PLOS Complex Systems

RESEARCH ARTICLE

Fast algorithms to improve fair information access in networks

Dennis Robert Windham[1], Caroline J. Wendt[1], Alex Crane[2], Madelyn J. Warr[2], Freda Shi[2], Sorelle A. Friedler[3], Blair D. Sullivan[2], Aaron Clauset[1,4,5]*

1 Department of Computer Science, University of Colorado, Boulder, Colorado, United States of America, **2** School of Computing, University of Utah, Utah, United States of America, **3** Department of Computer Science, Haverford College, Pennsylvania, United States of America, **4** BioFrontiers Institute, University of Colorado, Boulder, Colorado, United States of America, **5** Santa Fe Institute, Santa Fe, New Mexico, United States of America

* aaron.clauset@colorado.edu

OPEN ACCESS

Citation: Windham DR, Wendt CJ, Crane A, Warr MJ, Shi F, Friedler SA, et al. (2026) Fast algorithms to improve fair information access in networks. PLOS Complex Syst 3(3): e0000094. https://doi.org/10.1371/journal.pcsy.0000094

Editor: Jae-Suk Yang, KAIST: Korea Advanced Institute of Science and Technology, KOREA, REPUBLIC OF

Received: July 2, 2025

Accepted: January 26, 2026

Published: March 4, 2026

Data availability statement: The data that support the findings of this study are publicly available at https://github.com/TheoryInPractice/FairInfoAccessHeuristics.

Abstract

We consider the problem of selecting k seed nodes in a network to maximize the minimum probability of activation under an independent cascade beginning at these seeds. The motivation is to promote fairness by ensuring that even the least advantaged members of the network have good access to information. Our problem can be viewed as a variant of the classic influence maximization objective, but it appears somewhat more difficult to solve: only heuristics are known. Moreover, the scalability of these methods is sharply constrained by the need to repeatedly estimate access probabilities. We design and evaluate a suite of 10 new scalable algorithms which crucially do not require probability estimation. To facilitate comparison with the state-of-the-art, we make three more contributions which may be of broader interest. We introduce a principled method of selecting a pairwise information transmission parameter used in experimental evaluations, as well as a new performance metric which allows for comparison of algorithms across a range of values for the parameter k. Finally, we provide a new benchmark corpus of 174 networks drawn from 6 domains. Our algorithms retain most of the performance of the state-of-the-art while reducing running time by orders of magnitude. Specifically, a meta-learner approach is on average only 20% less effective than the state-of-the-art on held-out data, but about 75-130 times faster. Further, the meta-learner's performance exceeds the state-of-the-art on about 20% of networks, and the magnitude of its running time advantage is maintained on much larger networks.

Author summary

A classic question in the study of how information flows over social networks is this: Where across a network should we seed information, such as news

Funding: This work was supported in part by the National Science Foundation (IIS 1956183 to AC; IIS 1956286 to BS, and IIS 1955321 to SF). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

about public health measures or employment opportunities, so that it spreads well? Using a simple gossip-style model of information spread on a network, we evaluate different algorithms for selecting a set of initial seeds and their subsequent performance on maximizing the minimum probability that anyone receives the spreading information. That is, we define "spreading well" as everyone in the network having a good chance of receiving the target information. We introduce a new, large-scale benchmark of 174 networks for evaluating information seeding algorithms, two new measures for quantifying and comparing algorithm performance on this task, and 10 new seeding algorithms. These new algorithms exploit theoretical insights about how network structure can influence information access, allowing them to avoid a significant computational bottleneck present in existing solutions and scale up to much larger networks. Finally, we show that a meta-learning algorithm that learns which particular algorithm to run on a network, based only on its structural signature, is more scalable than past solutions and remains highly accurate.

1 Introduction

Influence maximization [1,2] is one of the most intensively studied problems in data mining, machine learning, and social network analysis. Given some model of information diffusion, commonly the *independent cascade* of Kempe, Kleinberg, and Tardos [1], the task is to determine where we should seed information such that it spreads as widely as possible. Formally, given a graph G and $S \subseteq V(G)$, for each $i \in V(G)$ let $\pi_i(S)$ denote the probability with which i is activated by an independent cascade seeded at S; we will just write π_i when the set S is clear. Then for a given budget k, influence maximization is the problem of identifying a set $S \subseteq V(G)$ of cardinality k which maximizes $\sum_{i \in V(G)} \pi_i$. This formalization naturally adapts to other models of information diffusion, so long as node activation is a discrete, one-time event that depends in some way on the activation states of its neighbors.

Influence maximization is commonly motivated by commercial advertising, e.g., [2–4] but is also relevant in various other applications, including public health [5, 6], the distribution of economic opportunities [7], and the spread of scientific knowledge [8]. In these applications, the desiderata include not only widespread but also *fair* dissemination of information. The difficulty of the latter goal is a natural consequence of structural heterogeneity in networks, i.e., maximizing the number of nodes activated often means activating the most easily reached or best connected nodes.

Highly-connected or centrally located individuals have more opportunities to receive and spread information [9–11], while peripheral individuals with few connections participate less often in information exchanges [10,12]. For example, in a pandemic, access to crucial resources—such as money, food and healthcare—is more difficult for socially disadvantaged groups, in part due to their more limited connectedness in social networks [13]. Similarly, connectedness shapes employment opportunities in professional social networks such as LinkedIn: well-connected job seekers

are likely to fill lucrative openings sooner than others [14]. This situation has motivated the study of several variations of influence maximization, for example to ensure equitable information access with respect to demographic groups [15]. In this paper, we focus on the formulation of Fish et al. [12], which adopts a *Rawlsian* [16] notion of fairness in which the goal is to improve the information access of the worst-off individuals. We leave other notions of fairness or social equity, and other models of information diffusion for future work.

Formally, the problem we study here is as follows:

MAXIMIN INFLUENCE MAXIMIZATION

Input: A graph $G = (V, E)$ and an integer k.
Task: Select k vertices $S \subseteq V$ maximizing $\min_{i \in V} \pi_i$.

This maximin variant is NP-hard, and even a constant-factor approximation would imply P = NP [12], in stark contrast to the greedy $(1 - 1/e)$-approximation for the standard objective [1]. However, Fish et al. show that a heuristic approach (`Myopic`; see Sect 3.1) optimizes the objective well in practice. Indeed, this approach even performs well when evaluated via the classic influence maximization objective, which it does not directly optimize. Unfortunately, this heuristic relies upon repeated Monte Carlo simulations of the activation probabilities π_i, which are a severe computational bottleneck and thereby constrain the scalability of the method; see Fig 1. Additionally, the method has only been evaluated on a small corpus of six networks of small or modest size.

We are thus motivated to (a) introduce new algorithms which alleviate or completely avoid the bottleneck imposed by probability estimation; (b) compare the performance of our methods against that of `Myopic` in a principled manner,; and (c) expand the number and diversity of networks on which the MAXIMIN INFLUENCE MAXIMIZATION problem has been studied.

1.1 Related work and preliminaries

1.1.1 Fairness in ML and social networks. Fairness in machine learning is a well-studied area, and includes notions of fairness based both on individual characteristics and on group demographics [17–19] (for surveys, see [20, 21]). Recently, questions of fairness of information access of individuals and demographic groups in a network have come to the fore [12,15,22–26]. This interest is grounded in theoretical and empirical work in social network analysis that relates social connectivity and social capital [27,28], which builds on foundational work by Bourdieu, Loury, and Coleman,

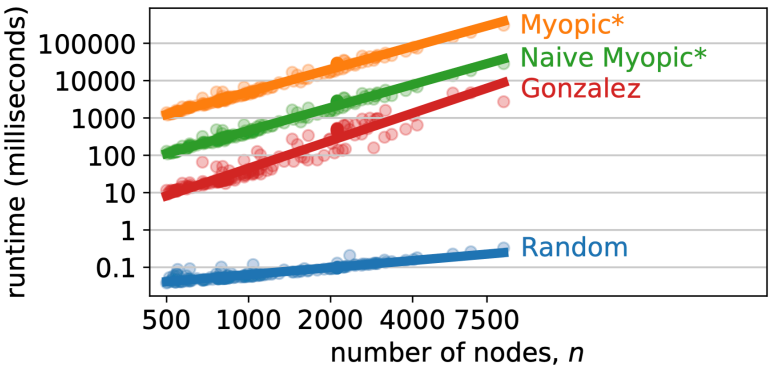

Fig 1. Algorithm runtime to select 10 new seeds vs. network size for algorithms in [12], averaged over 10 runs on an introduced large set of networks (see Sect 2.1). Algorithms requiring a Monte Carlo simulation (`ProbEst`) to select seeds are denoted by a *.

https://doi.org/10.1371/journal.pcsy.0000094.g001

among others. The key observation underlying this work is that in many social settings, individuals with more connections are substantially advantaged relative to those with fewer connections [29] (for many definitions of "advantage" and "connection").

A significant body of work focuses on estimating the probability that an individual receives information that spreads over a network. This includes studies of seeding information at nodes to improve individual [12] or group [23,26] access; as well as interventions to add edges [30–33] under varying notions of fairness. Other work considers notions of group fairness based on the network structure [25,34,35]. Saxena, Fletcher, and Pechenizkiy provide a recent survey of such work [36]. Fair clustering of individuals has also received significant attention, e.g., [37].

1.1.2 Independent cascade. Considerations about the access of individuals to resources in a network build on structural concerns about social networks pioneered by Granovetter [38]. Necessary in any such study is a clear model for the dynamics of information propagation. Numerous models exist, notably including the independent cascade, generalized independent cascade, and linear threshold models [1,39,40]. A standard choice, also adopted in this paper, is the independent cascade, which can be defined via an iterative spreading process. At each step i, some subset S_i of nodes is *activated*, beginning with an initial seed set $S = S_0$. In round $i + 1$, each edge uv with $u \in S_i$, $v \notin \bigcup_{j \leq i} S_j$ activates v independently with probability α. The process stops after the first round i in which no nodes are activated, i.e., $S_i = \emptyset$, and a node v is said to be activated by the cascade if it is activated in any round, i.e., $v \in \bigcup S_i$.

1.1.3 Probability estimation. Important in any analysis of independent cascades is the ability to measure the probability π_i with which node i is activated. Unfortunately, exact computation of π_i is #P-hard [2]. The standard approach is to estimate these values via *reverse influence sampling* (RIS) [41,42], which may also be thought of as performing a series of Monte Carlo simulations of the cascade process. Theoretical bounds on the number of simulations needed to satisfy a given error tolerance ε have a quadratic dependence on ε in addition to quasilinear dependence on network size [42], and so in practice it is common to fix a reasonable number R of simulations, e.g., $R = 1000$. Algorithms for influence maximization and related problems then invoke a linear-time (regarding R as a constant) subroutine, referred to here as `ProbEst`, to provide estimated π_i values. In practice, however, algorithms requiring such a subroutine can be orders of magnitude slower than those avoiding probability estimation altogether; we again refer to Fig 1. In the influence maximization literature, much effort has been expended to lessen the complexity of `ProbEst` while retaining quality guarantees, e.g., [43–45], as well as to empirically compare various strategies [46]. In this work, our approach is to develop methods which avoid RIS entirely and evaluate the resulting solution quality via empirical comparison against the state-of-the-art. A purely heuristic approach, while difficult to accept for the classic influence maximization problem, is palatable in our setting because strong inapproximability bounds are known for Maximin Influence Maximization even in the presence of an oracle which perfectly computes the π_i values [12].

1.2 Summary of contributions

Network corpus. We introduce a large and structurally diverse corpus of 174 networks which serves as a novel benchmark for both our algorithms and the state-of-the-art.

Spreadability. We introduce a novel metric we call *spreadability*, which enforces mathematical rigor in the assessment of a particular choice of the parameter α as "low"- or "high"-spreading for a given network, under the independent cascade model of information transmission. We use this framework to select appropriate α values for each network in our corpus, ensuring that we can fairly measure algorithm performance under multiple distinct regimes of information spread.

Performance metric. We introduce a novel performance metric β that accounts for varying choices of budget k, random number generator seeding, and the impracticality of precisely measuring the problem objective. This metric allows us to fairly and systematically compare algorithm performance across many networks.

Algorithms. We introduce 10 new algorithms for the Maximin Influence Maximization problem, which we partition into three categories. The first two replace the Monte Carlo simulations used by the state-of-the-art to estimate activation

probabilities5cc8f;[dpspfpgc =v with heuristics based on breadth-first search and personalized page rank. The complexity of our subroutines is similar to the Monte Carlo approach, but the hidden constants are vastly improved, and this is reflected in improved running times. The third category of algorithms eliminates probability estimation entirely, instead using topological features of the network to inform seed selection.

Evaluation, meta-learning, and scalability. We conduct a comprehensive evaluation of our methods, together with a comparison against the state-of-the-art (`Myopic` [12]). We show that on most networks at least one of our new algorithms is nearly as effective as or even more effective than `Myopic`. Meanwhile, our algorithms are much faster, improving on the running time of `Myopic` by factors of 10–10,000. We then define a novel meta-learner, which uses only cheap-to-compute structural features of a network to predict which algorithm will be most effective on that network. This meta-learner recovers roughly 80% of the performance of `Myopic` while reducing running time by factors of about 75–130. These improvements in running time persist even on several much larger networks.

2 Network corpus and evaluation methods

2.1 Network corpus

In order to investigate the effects of network structure on algorithm performance, we construct a corpus of 174 networks from six domains: biological (34), social (44), economic (43), technological (32), transportation (17), and informational (4). We include non-social networks in our study in order to more fully characterize the behavior of algorithms on structurally diverse real-world networks. When relevant, we report our results by domain, so that social and non-social networks can be contrasted. An overview of the corpus is presented in Fig 2, with summary statistics in Table A in S1 File.

Networks were curated from the Index of Complex Networks [47], a large-scale index of research-quality networks spanning all domains of science, as well as the Netzschleuder network catalogue and repository [48] and the corpus of Ghasemian, Hosseinmardi, and Clauset [49]. All networks included in our corpus are simple graphs, meaning their edges are undirected, unweighted, and there are no self-loops. Further, they are unipartite, i.e., they only have one type of node.

Past work on such corpora indicates that domains (and even subdomains, e.g., online social networks vs. offline social networks) are highly distinguishable based on their structure alone [50]. Hence, a specific effort was made to (i) balance the classes, so that no domain was more than 25% of the corpus, (ii) avoid over-representing networks from particular sources (e.g., Twitter follower networks), and (iii) ensure that the minimum network size was large enough to provide good results for information spreading tasks (minimum number of nodes $n_{\min} = 500$). These choices improve the breadth and variety of network structure represented in the corpus, and its utility in the analyses of this study.

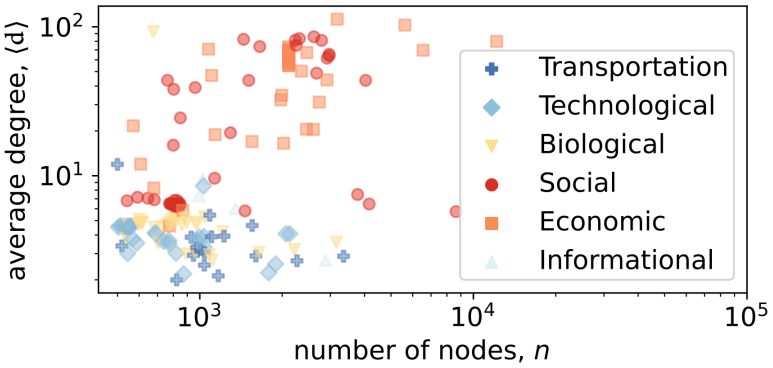

Fig 2. Average degree of a network as a function of network size (number of nodes) for the corpus of 174 networks from 6 distinct domains used in our study.

https://doi.org/10.1371/journal.pcsy.0000094.g002

2.2 Spreadability

Background. In our work, we leverage the independent cascade model [1] to study the spread of information in networks and estimate π_i. Given a network G and a set of activated nodes $Q \subseteq V$, we grow a forest on G under the independent cascade model by flipping a coin for each edge e_{ij}, where $i \in V \backslash Q$, $j \in Q$, exactly once, so that i is added to Q on a successful flip. Each edge e_{ij} is considered as a transmission path at most once, and we define the probability of successful transmission to be α. We note that mathematically, it is equivalent to conceptualize the independent cascade as follows: delete every edge in the graph independently with probability $(1 - \alpha)$; the set Q of activated nodes is exactly the set of nodes that remain reachable from the seed set S. As noted earlier, computing the exact probability of activation for a node i, denoted π_i, is #P-hard [2]. As such, we adopt the standard Monte Carlo simulation approach, but return later to consider practical consequences of this choice.

Briefly, using `ProbEst`, given transmission probability α, number of simulation rounds R, and a seed set S, we estimate π_i for every $i \in V$ using R independent cascades originating from S [12]. The worst-case time complexity of `ProbEst` is $\mathcal{O}(R(|S| + 2m))$ [12], where m is the number of edges in G. In practice, the computational cost of `ProbEst` increases both as the seed set grows and as α increases, because both changes tend to increase the size of the induced information cascades. Past work used $R = 1000$ as a balance between statistical accuracy for π_i and computational cost, and we follow this precedent [12]. Here, `ProbEst` is used as a subroutine in some algorithms, as well as to evaluate the performance of algorithms by estimating the achieved minimum activation probability of a computed seed set.

Selecting α values. Prior work evaluated algorithm performance using transmission probabilities $\alpha \in \{0.3, 0.4, 0.5\}$ [12]. However, the resulting information cascades, and hence the associated access probabilities, are not a simple function of α; they instead also depend on the network's structure. For example, denser, more connected networks contain many more paths by which information can spread than do sparse networks. Hence the same α will tend to produce larger cascades on the former, and smaller cascades on the latter. To control for these structure-induced differences in information cascades, we introduce the concept of *spreadability*, which jointly accounts for the impact of network structure and transmission rate α on the sizes of information cascades.

We can construct a fine-grained spreadability function that relates a particular choice of α to the fraction of a network activated under the independent cascade model from a uniformly random initial seed. For a given α, the spreadability $f(\alpha)$ on a particular network the fraction of nodes that are activated from a uniformly random initial seed, averaged over R trials. This calculation produces a monotonically increasing curve, as seen in Fig 3. We find that computing spreadability for each $\alpha \in \{0.01, 0.02, ...0.99\}$ provides ample resolution to choose α close to target spreadabilities of 0.2,0.5,0.8 (which we

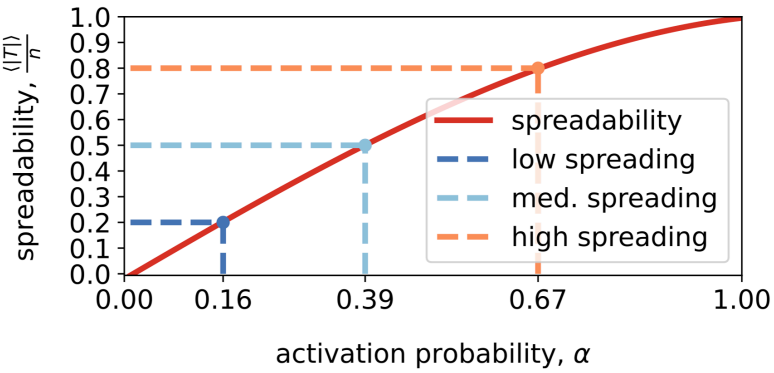

Fig 3. Spreadability on a network is quantified by the average fraction of a network's nodes $\langle |T| \rangle / n$ in a tree T grown through an independent cascade from a random initial seed for a given α. We define 'low', 'medium', and 'high' spreadability as the α that activates, on average, 20%, 50%, and 80% of the network, respectively.

https://doi.org/10.1371/journal.pcsy.0000094.g003

refer to as "low," "medium," and "high" spreadabilities respectively). We set $R = 1000$, as the spreadability curve tends to stabilize near this value and we get diminishing returns for larger R.

2.3 New metric for algorithm evaluation

There are several reasons it is not straightforward to compare the performance of algorithms, even with a fixed network and α value. First, we do not want to make strong assumptions about the "most useful" value for the parameter k (the number of seeds to be added), and it is possible that algorithm $\mathcal{A}$ obtains a better objective score (the minimum activation probability $\pi_{\min}$) than algorithm $\mathcal{A}'$ for one value of k, but not for another. Moreover, many algorithms (in particular those using `ProbEst`) have some randomness. Finally, even our analysis of the quality of a solution is inexact; recall that exact computation of the $\pi_{\min}$ achieved by a seed set S is #P-hard [2]. Thus, our ability to compare two solutions is limited by the precision of `ProbEst`.

We overcome these challenges by introducing a metric β, which intuitively measures the marginal gain in the problem objective (the minimum activation probability $\pi_{\min}$) which we can expect when asking an algorithm to add one additional seed. Formally, for a given algorithm $\mathcal{A}$, network G, independent cascade parameter α, and budget k, we run $\mathcal{A}$ on (G, α, k) and record a value $\pi_{\min}(k)$ indicating the achieved objective score. We repeat for each $k \in \{1, 2, \ldots, 10\}$, computing the points $(k, \pi_{\min}(k)) \in \mathbb{N} \times [0, 1]$. We then record the slope β_1 of the line of best fit for these points. We repeat the entire process multiple (generally 20) times, producing slopes $\beta_1, \beta_2, \ldots$. The metric β is the mean of these values, i.e., the average slope of the line of best fit; see Fig 4. Henceforth, when comparing the performance of algorithms, we do so via the metric β.

3 Algorithms for fair information access

3.1 Algorithms from prior work

Four previously introduced heuristics for choosing seed nodes to maximize $\pi_{\min}$ are `Greedy`, `Myopic`, `Naive Myopic` and `Gonzalez`, along with `Random` as a baseline for comparison [12]. `Random` selects k seeds by uniformly sampling nodes from the network. Given a partial seed set, `Gonzalez` selects as the next seed the node that is the furthest (in the shortest-path metric) from all nodes in the current set [12]. The `Greedy` algorithm iteratively selects a new seed by choosing the node with the highest marginal gain relative to the current seed set according to `ProbEst`. Due to its immensely high computational cost, `Greedy` is not practical for most networks [12], and is not used in our study.

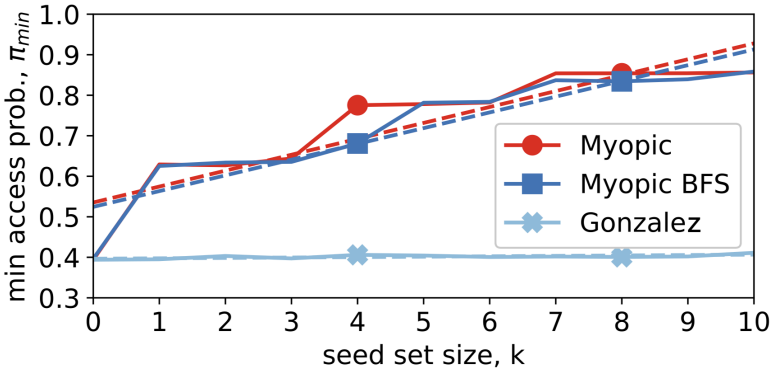

Fig 4. Minimum access probability $\pi_{\min}$ vs. seed set size k, with a best fit line (`Myopic` $\hat{\beta} = 0.039$), averaged over 20 runs, evaluated on a large economic network ($n = 2113$ nodes, $m = 57927$ edges), with $\alpha = 0.4$ and a budget of $k = 10$ seeds, plus one random initial seed.

https://doi.org/10.1371/journal.pcsy.0000094.g004

In contrast, `Myopic` uses `ProbEst` with the current seed set, and selects as the next seed the node with the lowest π_i. `Naive Myopic` is similar to `Myopic`, but only runs `ProbEst` once, at initialization, and then selects as the seed set the k nodes with the lowest π_i values. In past work, on a small set of networks, `Myopic` was found to perform best. However, because `Myopic` depends on `ProbEst`, it is computationally expensive, which limits its applicability to large networks. In practice, Myopic is always the second slowest algorithm, after `Greedy` [12].

Algorithm initialization. `Myopic`, `Naive Myopic`, and `Gonzalez` all start with an initial seed. Past work chose this initial seed to be the highest degree node. Our initial experiments indicate that this choice confers a substantial advantage to these methods (Fig 5), and that some of the previous positive results are thus attributable to this initial seed choice rather than to the algorithms' subsequent choices. To mitigate this bias we instead initialize all heuristics with a seed set composed of a single uniformly randomly selected initial seed. Moreover, for each evaluation round on a particular network, we initialize all algorithms to use the same random seed, which further controls for the effects of different initial seed choices. This initial seed choice is not counted against the budget k (Fig 4).

3.2 New fast algorithms

In addition to the four heuristics from prior work, here we introduce 10 new heuristics to maximize the minimum π_i on a budget. These heuristics are designed to be computationally lightweight, scaling to far larger networks, while also matching or exceeding the performance of `Myopic` on the corpus. We group the new algorithms into three families: BFS-based, PPR-based and Topology-based.

BFS-based: Myopic BFS and Naive Myopic BFS. In the two BFS-based heuristics, `Myopic BFS` and `Naive Myopic BFS`, we swap the `ProbEst` component of `Myopic` and `Naive Myopic` with a simple breadth-first search to estimate π_i. The breadth-first search component is initialized with a random seed k_0, and transmission probability α. It proceeds to "peel" the network, starting at k_0, in breadth-first fashion, estimating π_i for each node as the probability that i receives a transmission from k_0 through any nodes it connects to in the previous BFS 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 does not exactly measure π_i, it is much faster than `ProbEst`, taking $\mathcal{O}(n \cdot \langle k \rangle)$ per iteration, because for most networks, including those in our corpus, the mean degree $\langle k \rangle \ll 1000$ (Fig 2). The key design principle of this algorithm is to capture complex network structures that `Gonzalez` could not account for. In Fig 4, we find that `Myopic BFS` almost matches `Myopic`'s performance in situations where `Gonzalez` lags behind.

PPR-based: Myopic PPR and Naive Myopic PPR. In the PPR-based heuristics, `Myopic PPR` and `Naive Myopic PPR`, we use Personalized PageRank (PPR) to estimate π_i instead of `ProbEst` or BFS. Personalized PageRank (implemented with networkx [51]) performs a random walk that probabilistically restarts from nodes in the seed set [52]. The

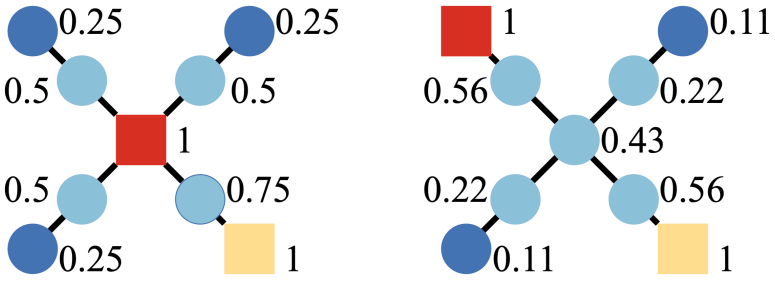

Fig 5. Illustrations of two runs of `Myopic` for different initial seeds (red), with new selected seeds (yellow), and fixed $\alpha = 0.5$. Numbers indicate π for each node after the new seed is selected. Initialization significantly affects the performance of `Myopic`.

https://doi.org/10.1371/journal.pcsy.0000094.g005

ranking produced by PPR is not a direct estimate of π_i; we treat the PPR values as being correlated, such that a lower PPR score is a proxy for a lower π_i value.

Topology-based: LeastCentral and MinDegree. These heuristics are based on the intuition that nodes with low π_i values have distinctive structural positions or patterns of connectivity. `LeastCentral` selects the non-seed node i with the lowest closeness centrality c_i as the next seed. Similarly, `LeastCentral_n` selects the lowest centrality node i's highest degree neighbor as the next seed. Here, the closeness centrality of a node i is the inverse of the average shortest path length from i to all other reachable nodes in the same connected component [53,54]. 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 heuristics exploit a network's degree structure to make decisions, based on the observation that in practice `Myopic` tends to select nodes with low degree as seeds. The first two heuristics take a non-seed node with the lowest degree, breaking ties by choosing the node with the lowest harmonic centrality [54,55] among same-degree nodes. In the case of `MinDegree_hc`, it chooses that node itself as the next seed, while `MinDegree_hcn` chooses the highest-degree neighbor of that node. The two remaining variations replace harmonic centrality in the aforementioned logic with the neighbor degree, i.e. breaking ties by choosing the node with the highest neighbor degree (sum of degrees across neighbors). These heuristics are called `MinDegree_nd` and `MinDegree_ndn`, respectively.

Recall from Sect 1 that strong approximation lower bounds are known. Indeed, most of the algorithms proposed here have worst-case approximation factors which are exponentially small as a function of n; Sect E in S1 File. The same is true of the current state-of-the-art [12]. Our approach will be to evaluate algorithms based on their empirical performance on real-world networks.

4 Experimental results

4.1 Performance of algorithms on the corpus

We evaluate and compare the performance of 14 algorithms in total (10 new algorithms and 4 from prior work), applied to all 174 networks in the corpus. Our code and data are available at https://github.com/TheoryInPractice/FairInfoAccessHeuristics. We are interested in algorithms that operate well on a tight budget and so let $k \in \{1, 2, \ldots, 10\}$ seed nodes. For each spreadability level (low, medium, high), we produce a $10 \times 14 \times 174$ matrix, where each entry is the β performance of an algorithm after adding k seeds in a network, averaged over 20 runs. We focus on the medium spreadability regime here, and include results for low and high spreadability Sect C in S1 File. Fig 6 displays the mean performance of each algorithm on networks by domain under medium spreadability (see Fig A in S1 File for low and high spreadability results). Across domains, `Myopic` produces the best average performance.

To further examine the results, we sort within each domain by network size in ascending order, and then score algorithms as better than, "equivalent" to (within one std. err.), within 80% of, or worse than `Myopic` (see Fig 7); low and high spreadability results in Fig B in S1 File. In this way, we can assess whether the better average performance of `Myopic` applies to individual networks compared to other algorithms. This experiment reveals that `Myopic` is not universally the best algorithm for all networks in any spreadability setting; on many individual networks other algorithms perform equivalently or better. In the medium spreadability setting, for only 24% of networks is there no algorithm that performs at least 80% as well as `Myopic` (Fig 7).

This variability in performance across networks suggests that network structure plays a critical role in governing the relative performance of different algorithms. Fig 8 plots the instances for which each algorithm was the best performing algorithm on a network against that network's mean degree $\langle k \rangle$. `Myopic` tends to perform better on networks with lower average degree, although it also performs well on many networks with larger mean degree. In practice, we find that the final seed set produced by `Myopic` is often composed primarily of low-degree nodes located in a network's periphery, and these nodes may be too far removed from other disadvantaged nodes to meaningfully improve their π_i values.

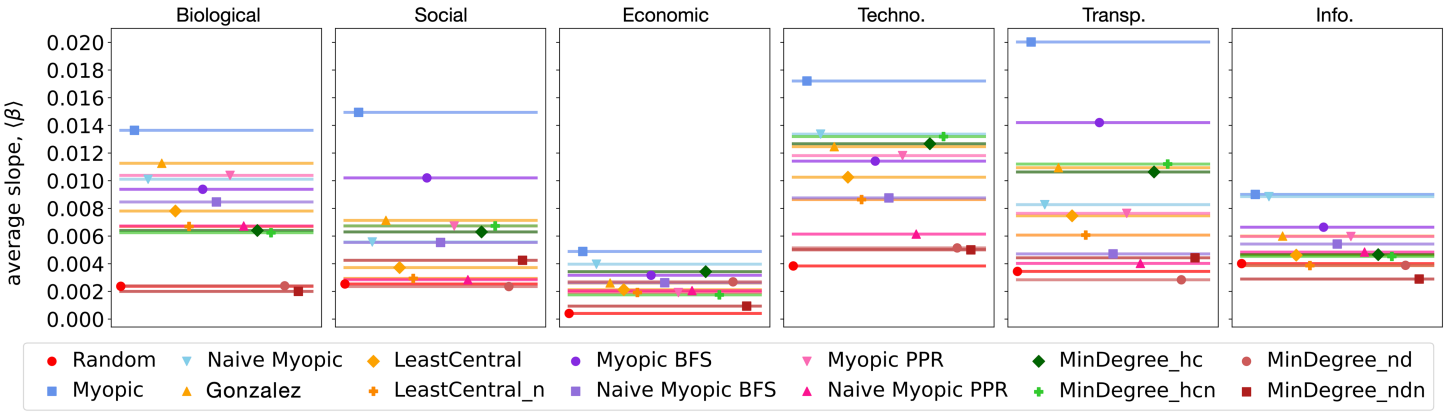

Fig 6. Mean performance of the intervention algorithms on each domain in the corpus under medium spreadability. Each algorithm's performance is averaged over a given domain, 20 runs per network.

https://doi.org/10.1371/journal.pcsy.0000094.g006

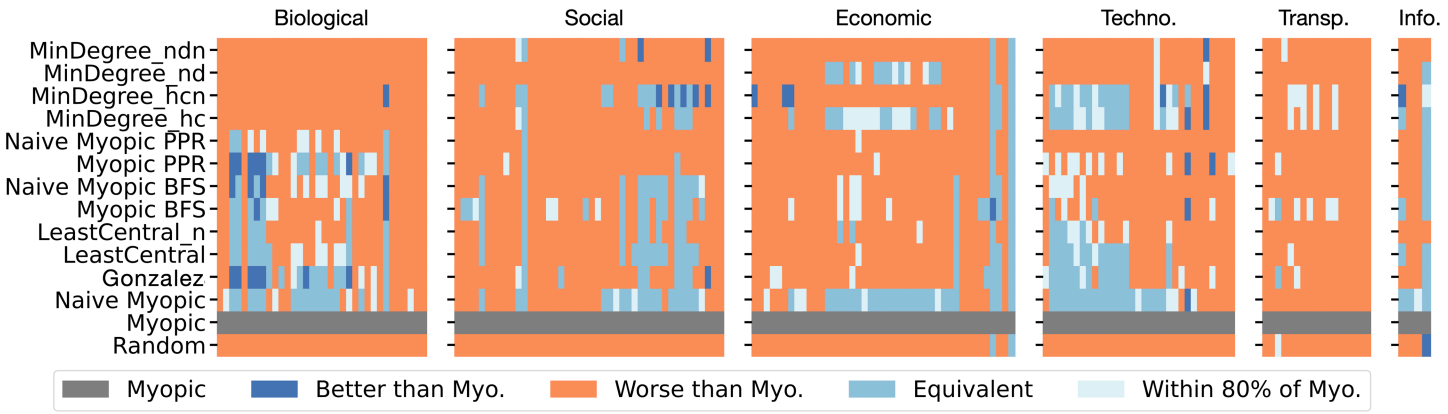

Fig 7. Performance of intervention algorithms on the network corpus, relative to `Myopic` and sorted in ascending order by network size within each domain. Under medium spreadability, 24% of networks have no algorithm better than or within 80% of `Myopic`'s performance. "Equivalent" is defined as within one standard error of β for `Myopic`; typically about 0.001.

https://doi.org/10.1371/journal.pcsy.0000094.g007

A second takeaway is that a few specific algorithms tend to perform better than `Myopic` in certain settings (Fig 8), specifically `MinDegree_hcn` and `Gonzalez`. The performance of `MinDegree_hcn` in particular tends to improve over `Myopic` with increasing average degree, while `Gonzalez` does best in networks with very low mean degrees. Furthermore, we note that in Fig A in S1 File, on average `MinDegree_hcn` outperforms `Myopic` in the economic domain, and from Table A in S1 File, we see that the economic domain has the highest mean degrees. The `MinDegree_hcn` algorithm selects as seeds the highest-degree neighbors of low-degree nodes. As result, new information cascades seeded at these nodes will tend to spread quickly to a number of disadvantaged nodes in the network.

4.2 Algorithm runtime

We evaluate all algorithm runtimes on the corpus, selecting $k=10$ new seeds, averaged over 10 runs. For fair comparison, we run all algorithms on a single core of an AMD Ryzen 5900X, overclocked to 5.00Ghz, with 32GB RAM, and measure the runtime in milliseconds. Two major performance bottlenecks are `ProbEst`, used by `Myopic` and `Naive Myopic`,

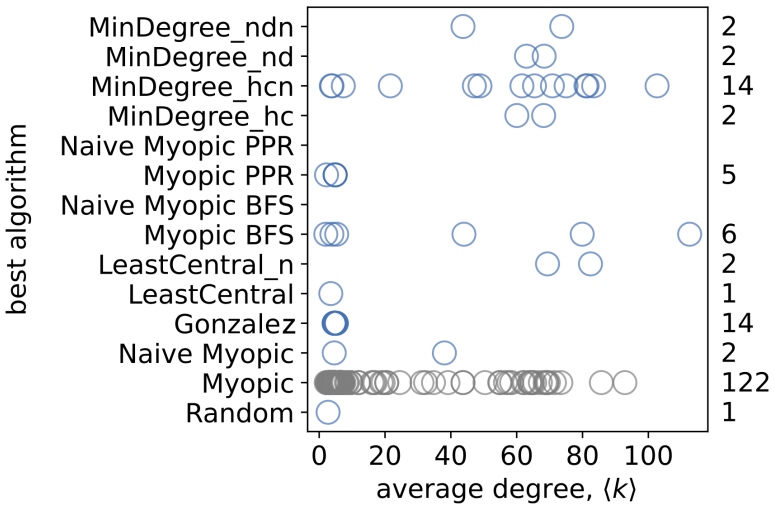

Fig 8. Best-performing algorithm vs. mean degree of the network (medium spreadability), for all networks. Counts on the right show total circles per line, i.e., the number of times an algorithm was the best over the whole corpus.

https://doi.org/10.1371/journal.pcsy.0000094.g008

and an All-Pairs-Shortest-Paths (APSP) computation, used by `Gonzalez`, `LeastCentral`, `LeastCentral_n`, `MinDegree_hc`, and `MinDegree_hcn`. Both `ProbEst` and `APSP` have efficient parallel implementations, but we restrict them to a single core to ensure fair comparison with other algorithms.

All of the new algorithms are substantially faster than `Myopic` and `Naive Myopic` (Fig 9). Because they only require sorting two lists, `MinDegree_ndn` and `MinDegree_nd` are the most efficient, improving running times over `ProbEst`-based algorithms by a factor of 1000-10000x, depending on the size of the network (Fig 9). `BFS`-based algorithms are marginally slower than these fastest algorithms, and algorithms that use an All-Pairs-Shortest-Paths (APSP) calculation fall between the `ProbEst` and `BFS` algorithm groups. As expected from the asymptotics, `BFS` algorithms tend to scale more slowly (in terms of runtime) with network size than do `ProbEst` algorithms, while APSP algorithms scale more quickly (Fig 9).

The low upfront cost of APSP-based algorithms makes them far faster than `ProbEst`-based algorithms in many practical settings, being 10-100x faster on networks with less than $n = 10^6$. However, the asymptotic cost of APSP is cubic in the network size, implying that for sufficiently large networks, `ProbEst` will be faster. For our corpus, we estimate the crossover point when Myopic becomes more efficient than `MinDegree_hc` to occur between 1,262,000 and 1,515,000 nodes (95% CI, from 1000 bootstraps). However, approximation algorithms for APSP could potentially extend their practical efficiency much further.

4.3 A fast meta-learning algorithm

We can exploit the variability in algorithm performance, and the fact that even among non-`Myopic` algorithms no alternative is superior on all networks, by introducing a meta-learner algorithm that combines multiple scalable heuristics to approximate the state-of-the-art performance of the `Myopic` algorithm. We compare this algorithm to a fast ensemble algorithm that uses an oracle to make perfect predictions about which scalable algorithm is best to apply on a particular test network, thus providing an upper bound on the meta-learner's possible performance.

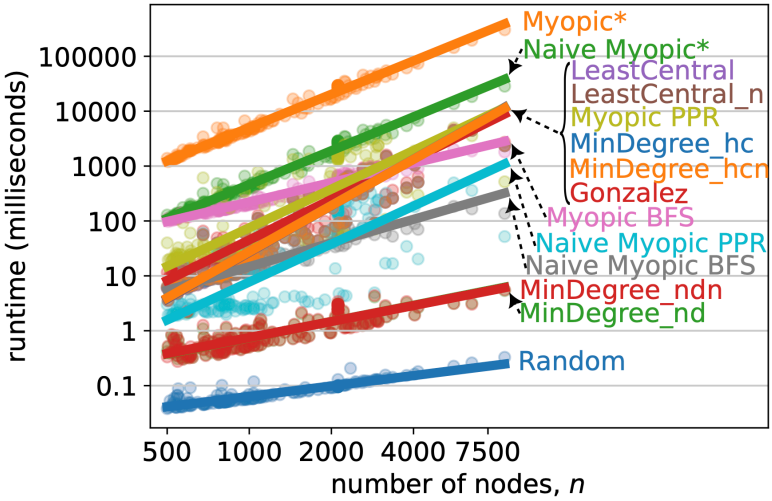

Fig 9. Runtime of old and new algorithms across the network corpus with $k = 10$ seeds averaged over 10 runs, showing a substantial advantage in running time for the new algorithms. Note: algorithms with * require `ProbEst`.

https://doi.org/10.1371/journal.pcsy.0000094.g009

The meta-learner algorithm leverages the scalability of non-`ProbEst` algorithms while retaining good overall performance under all spreadability regimes. The task is as follows: given a particular network G and knowledge of the information's spreadability (low, medium, or high), select the scalable algorithm with the best marginal benefit β for improving information access.

As shown previously, many of the heuristics do not perform well, and so we begin by narrowing the set of available algorithms. For each of the three spreadability settings, we select the set $\mathcal{X}$ of five algorithms (excluding `Myopic` and `Naive Myopic`) that maximize the number of networks for which at least one among the set performs at least 80% as well as `Myopic`. This produces sets

- $\mathcal{X}_{\text{high}}$={`Gonzales`, `LeastCentral`, `Myopic BFS`, `Naive Myopic BFS`, `MinDegree_hc`}
- $\mathcal{X}_{\text{medium}}$={`Gonzales`, `Myopic BFS`, `Myopic PPR`, `MinDegree_hc`, `MinDegree_hcn`}
- $\mathcal{X}_{\text{low}}$={`Gonzalez`, `Myopic BFS`, `Myopic PPR`, `MinDegree_hcn`, `MinDegree_ndn`}

For the meta-learner algorithm, we then learn a 5-way random forest classifier to predict the best algorithm in $\mathcal{X}$ to apply to a given network, using nine of the network's topological features as the feature set (Fig E in S1 File). We train and evaluate the meta-learner approach using an 80-20 train-test split among networks in the corpus, with meta-learner algorithm selection and model training both performed on the training set, and we report the mean performance over the test set. The meta-learner's runtime is the runtime of the trained model and the single algorithm it selects.

In the fast ensemble algorithm, for a given network in the corpus, the oracle runs all algorithms in $\mathcal{X}$ and evaluates the performance of each using `ProbEst` to calculate their respective βs, and then returns the single algorithm with the highest β for that network. In this way, the oracle acts like an optimal classifier over $\mathcal{X}$ (cf. the meta-learner algorithm). The runtime of the fast ensemble algorithm is simply that of the single algorithm it selects.

Compared to `Myopic`, the meta-learner is dramatically more efficient, with an average runtime that is 76.26 ± 64.07 times faster under medium spreadability (Fig 10) and 133.35 ± 79.32 times faster under high spreadability (Fig G in S1

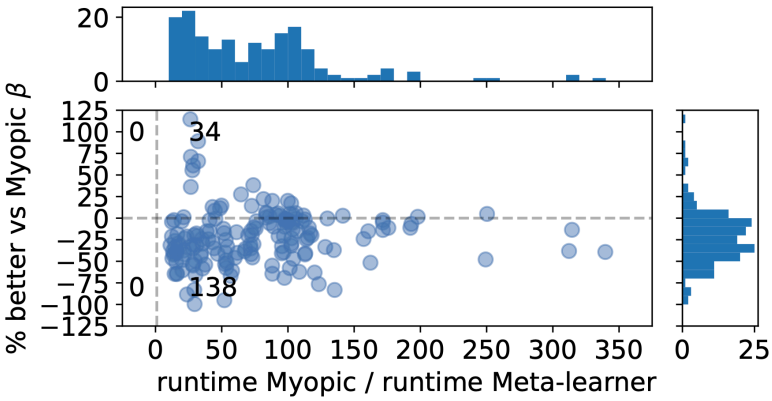

Fig 10. Performance difference vs. speedup for the meta-learner algorithm under medium spreadability, with marginal histograms, averaged over 1000 runs. Extreme outliers have been removed for visualization purposes. Average performance difference relative to `Myopic` is $-20.11\% \pm 29.34$ (mean $\pm$ stddev), for an average speedup factor of 76.28 ± 64.07. For 34 of the networks (19.8%) the meta-learner strictly outperforms `Myopic`.

https://doi.org/10.1371/journal.pcsy.0000094.g010

File). Improvement in scalability comes with a modest cost to performance, such that the meta-learner produces β values that are, on average, $20.11\% \pm 29.34$ lower than those of `Myopic` under medium spreadability (with similar results for high); we note that the wide variance in these numbers reflects the broad range of difficulty across networks in the corpus. In contrast, the fast ensemble algorithm's performance is only $9.34\% \pm 28.34$ lower for medium spreadability (similar results for high), indicating both room for improvement by the meta-learner with a better feature set as well as an upper limit to that improvement with the current scalable algorithms. We note, however, that lower performance is not universal: for 34 and 22 networks (20% and 12.8%), the meta-learner outperforms `Myopic` on medium and high spreadability, respectively (Fig 10, and Fig G in S1 File). Under low spreadability, the fast meta-learner's average performance generally exceeds `Myopic` because of the inherent precision limitation of `ProbEst` in this setting.

4.4 Scaling to larger networks

In this section, we verify that our methods remain much faster than `Myopic` even on much larger networks. We consider two networks, Email Network (EU Research Institution) (which we shorten to Email (EU)) and Google+ (2013). These have ~ 34 thousand and ~ 87 thousand nodes, respectively; other summary statistics are reported in Table B in S1 File. On these networks, we evaluate `Myopic` against the five algorithms in $\mathcal{X}_{\text{high}}$, as defined in Sect 4.3: `Gonzales`, `Least Central`, `Myopic BFS`, `Naive Myopic`, `BFS`, and `MinDegree_hc`. We use high-spreadability α values. We ran these experiments on identical hardware equipped with 40 physical cores (Intel(R) Xeon(R) Gold 6230 CPU @ 2.10 GHz) and 19100 MB of RAM. As with the smaller networks, the budgets tested are $k \in \{1, 2, \ldots, 10\}$.

We change several parameters from our previous runs so that our experiments are better suited to the size of these networks. After initial experiments with the number of Monte Carlo simulations in `ProbEst` set to $R = 1000$, it became clear that 1000 simulation rounds was not enough for networks with tens of thousands of nodes, as some of our evaluations produced a negative β. With exact probability computations, π_{min} would monotonically increase with each seed added, so $\beta < 0$ indicates highly inaccurate estimations. Thus, we increase the number of Monte Carlo simulations to $R = 10000$ in order to better estimate π_i for large networks. However, due to the increase in simulation rounds and network size, running `ProbEst` to evaluate the algorithms' performances is extremely expensive. To balance between computation time and a thorough analysis, we run only 3 trials of our algorithms with high-spreadability α values. We also use only 1000 simulations when executing the spreadability computations (recall Sect 2.2) to select the appropriate α. We note that calculating the target α would have required approximately 30 days of compute time for Google+ (2013) with $R = 10000$.

After using 1000 simulations to complete the spreadability computations, we tested the selected α values with $R = 10000$, verifying that they activate on average 78.9% and 79.4% of Email (EU) and Google+ (2013), respectively, which is quite close to the target activation percentage of 80%.

As with the broader corpus, when evaluated on the two larger networks, the five representative algorithms from $\mathcal{X}_{\text{high}}$ are orders of magnitude faster than `Myopic`, with some of the algorithms maintaining seed choice quality comparable to `Myopic`. Indeed, the *slowest* of the algorithms from $\mathcal{X}_{\text{high}}$ is 135 times faster than `Myopic` on Email (EU), and 96 times faster on Google+ (2013). These results (see Fig 11) indicate that the running time advantage of our methods over `Myopic` is not diminished on networks much larger than those in the corpus of Sect 2.1. Moreover, the performance loss of our methods (relative to `Myopic`) is similar (though slightly greater) to that observed for the larger corpus; see Fig J in S1 File.

5 Discussion and conclusion

We evaluate a set of new algorithms on a large and structurally diverse network benchmark, and introduce a meta-learning method for choosing k seed nodes to maximize the minimum access probability π_i of a node in a network. The meta-learner achieves a large (75x) speedup over the existing state-of-the-art, with only a modest decrease in performance. The Monte Carlo method for estimating a node's information access, which is the basis of the existing state-of-the-art, has high computational costs that limit the applicability of algorithms that rely on it. In contrast, the algorithms we introduced here scale-up to much larger networks by leveraging insights from network theory and meta-learning.

These findings highlight the utility of taking a pluralistic approach to algorithm design for maximizing the minimum access probability on a network: our systematic investigation revealed that no algorithm was the best, or the worst, on every network in our benchmark. Although the meta-learner is highly accurate at selecting the best heuristic given a particular network's structural features, future work could improve this performance by developing and incorporating new lightweight heuristics that exploit other relationships between access and network structure. The two metrics we

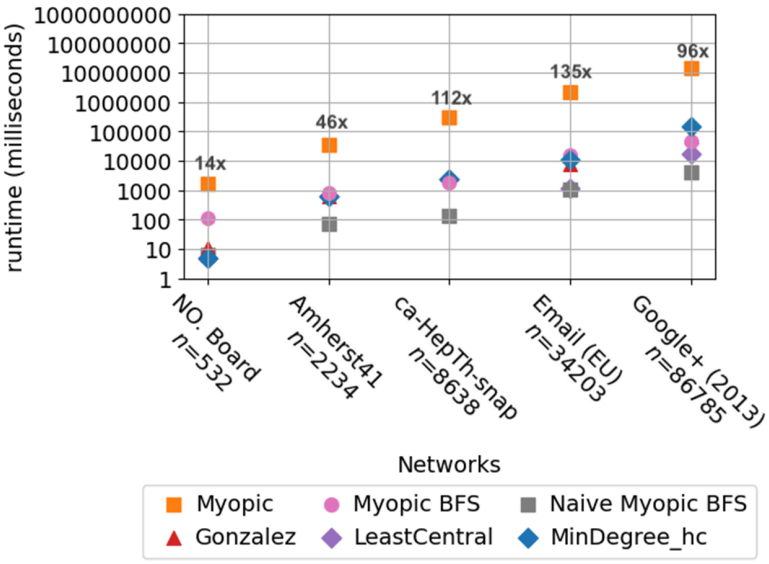

Fig 11. Runtime of `Myopic` and the algorithms of $\mathcal{X}_{\text{high}}$ on five social networks of various sizes under high spreadability: Norwegian Board of Directors (2006), Amherst41, ca-HepTh-snap, Email (EU), and Google + (2013). Numbers in the plot indicate how much slower `Myopic` is than the second-slowest algorithm for that network.

https://doi.org/10.1371/journal.pcsy.0000094.g011

introduce—spreadability, which standardizes performance comparisons across networks, and β, which standardizes performance comparisons across algorithms—should facilitate such future work.

A key conclusion from our investigation is that the structure of the network plays a significant role in the performance of the algorithms, with average degree being a particularly important factor, along with degree assortativity, mean path length, and the degree distribution's variance. These findings may have implications for the design of intervention strategies that aim to improve information access of the disadvantaged individuals in real networks.

Finally, future work may also consider different models of how information spreads across a network, such as the linear threshold model for "complex contagions," the invitation-aware diffusion (IAD) model [56], and the independent cascade with invitation (ICI) model [56], as well as alternative definitions of access, e.g., group-level access, or other models of social advantage or disadvantage. Work in these directions would have many benefits. It would help shed new light on the more general questions of fairness and access in networks. It would clarify the degree to which meta-learning approaches can adapt to different models of information spread or alternative objective functions. And, it would better connect algorithmic work with real-world systems and potential interventions.

Supporting information

S1 File. Algorithm definitions and supplementary results for individual algorithms and meta-learner, including on large networks.
(PDF)

Acknowledgments

The authors thank Daniel B. Larremore and Zachary Kilpatrick for helpful feedback, and they acknowledge the BioFrontiers IT group at the University of Colorado Boulder for their support with data storage infrastructure, data management services, and High Performance Computing resources.

Author contributions

Conceptualization: Dennis Robert Windham, Caroline J. Wendt, Sorelle A. Friedler, Blair D. Sullivan, Aaron Clauset.

Data curation: Dennis Robert Windham, Caroline J. Wendt, Aaron Clauset.

Formal analysis: Dennis Robert Windham, Caroline J. Wendt, Alex Crane, Madelyn J. Warr, Freda Shi.

Funding acquisition: Sorelle A. Friedler, Aaron Clauset.

Investigation: Dennis Robert Windham, Caroline J. Wendt, Alex Crane, Aaron Clauset.

Methodology: Dennis Robert Windham, Caroline J. Wendt, Sorelle A. Friedler, Blair D. Sullivan, Aaron Clauset.

Project administration: Sorelle A. Friedler, Blair D. Sullivan, Aaron Clauset.

Resources: Dennis Robert Windham, Caroline J. Wendt, Madelyn J. Warr, Freda Shi, Blair D. Sullivan, Aaron Clauset.

Software: Dennis Robert Windham, Alex Crane.

Supervision: Sorelle A. Friedler, Blair D. Sullivan, Aaron Clauset.

Validation: Dennis Robert Windham, Alex Crane.

Visualization: Dennis Robert Windham, Alex Crane, Aaron Clauset.

Writing – original draft: Dennis Robert Windham, Alex Crane, Sorelle A. Friedler, Blair D. Sullivan, Aaron Clauset.

Writing – review & editing: Sorelle A. Friedler, Blair D. Sullivan, Aaron Clauset.

References

1. Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. 2003. p. 137–46. https://doi.org/10.1145/956750.956769
2. Chen W, Wang C, Wang Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 2010. p. 1029–38. https://doi.org/10.1145/1835804.1835934
3. Domingos P, Richardson M. Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. 2001. p. 57–66. https://doi.org/10.1145/502512.502525
4. Richardson M, Domingos P. Mining knowledge-sharing sites for viral marketing. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. 2002. p. 61–70. https://doi.org/10.1145/775047.775057
5. Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007. p. 420–9. https://doi.org/10.1145/1281192.1281239
6. Yadav A, Chan H, Jiang AX, Xu H, Rice E, Tambe M. Using social networks to aid homeless shelters: dynamic influence maximization under uncertainty. In: AAMAS. vol. 16; 2016. p. 740–8.
7. Banerjee A, Chandrasekhar AG, Duflo E, Jackson MO. The diffusion of microfinance. Science. 2013;341(6144):1236498. https://doi.org/10.1126/science.1236498 PMID: 23888042
8. Rogers EM, Singhal A, Quinlan MM. Diffusion of innovations. An integrated approach to communication theory and research. Routledge; 2014. p. 432–48.
9. Cheng C, Wang H-Y, Sigerson L, Chau C-L. Do the socially rich get richer? A nuanced perspective on social network site use and online social capital accrual. Psychol Bull. 2019;145(7):734–64. https://doi.org/10.1037/bul0000198 PMID: 31094537
10. Picard PM, Zenou Y. Urban spatial structure, employment and social ties. Journal of Urban Economics. 2018;104:77–93. https://doi.org/10.1016/j.jue.2018.01.004
11. Tóth G, Wachs J, Di Clemente R, Jakobi Á, Ságvári B, Kertész J, et al. Inequality is rising where social network segregation interacts with urban topology. Nat Commun. 2021;12(1):1143. https://doi.org/10.1038/s41467-021-21465-0 PMID: 33602929
12. Fish B, Bashardoust A, Boyd D, Friedler S, Scheidegger C, Venkatasubramanian S. Gaps in Information Access in Social Networks?. In: The World Wide Web Conference. 2019. p. 480–90. https://doi.org/10.1145/3308558.3313680
13. Garoon JP, Duggan PS. Discourses of disease, discourses of disadvantage: a critical analysis of National Pandemic Influenza Preparedness Plans. Soc Sci Med. 2008;67(7):1133–42. https://doi.org/10.1016/j.socscimed.2008.06.020 PMID: 18656294
14. Wang D, Uzzi B. Weak ties, failed tries, and success. Science. 2022;377(6612):1256–8. https://doi.org/10.1126/science.add0692 PMID: 36108030
15. Stoica AA, Riederer C, Chaintreau A. Algorithmic glass ceiling in social networks: the effects of social recommendations on network diversity. In: Proceedings of the 2018 World Wide Web Conference; 2018. p. 923–32.
16. Rawls J. A theory of justice. Harvard University Press; 2009.
17. Feldman M, Friedler SA, Moeller J, Scheidegger C, Venkatasubramanian S. Certifying and removing disparate impact. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining; 2015. p. 259–68.
18. Hardt M, Price E, Srebro N. Equality of opportunity in supervised learning. Advances in Neural Information Processing Systems. 2016;30:3323–31.
19. Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. 2012. p. 214–26. https://doi.org/10.1145/2090236.2090255
20. Caton S, Haas C. Fairness in machine learning: a survey. ACM Comput Surv. 2024;56(7):1–38. https://doi.org/10.1145/3616865
21. Pessach D, Shmueli E. A review on fairness in machine learning. ACM Comput Surv. 2022;55(3):1–44. https://doi.org/10.1145/3494672
22. Stoica A-A, Chaintreau A. Fairness in social influence maximization. In: Companion Proceedings of The 2019 World Wide Web Conference. 2019. p. 569–74. https://doi.org/10.1145/3308560.3317588
23. Stoica AA, Han JX, Chaintreau A. Seeding network influence in biased networks and the benefits of diversity. In: Proceedings of The Web Conference. 2020. p. 2089–98.
24. Ali J, Babaei M, Chakraborty A, Mirzasoleiman B, Gummadi K, Singla A. On the fairness of time-critical influence maximization in social networks. IEEE Trans Knowl Data Eng. 2021. p. 1. https://doi.org/10.1109/tkde.2021.3120561
25. Mehrotra A, Sachs J, Celis LE. Revisiting group fairness metrics: the effect of networks. Proceedings of the ACM on Human-Computer Interaction. 2022;6(CSCW2):1–29.
26. Tsang A, Wilder B, Rice E, Tambe M, Zick Y. Group-fairness in influence maximization. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019. p. 5997–6005. https://doi.org/10.24963/ijcai.2019/831
27. Wasserman S, Faust K. Social network analysis: methods and applications. Cambridge University Press; 1994.

28. Portes A. Social capital: its origins and applications in modern sociology. New critical writings in political sociology. 1st ed. Routledge; 2009. p. 24.
29. Blumenstock J, Cadamuro G, On R. Predicting poverty and wealth from mobile phone metadata. Science. 2015;350(6264):1073–6. https://doi.org/10.1126/science.aac4420 PMID: 26612950
30. Bashardoust A, Friedler S, Scheidegger C, Sullivan BD, Venkatasubramanian S. Reducing access disparities in networks using edge augmentation. In: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency. 2023. p. 1635–51.
31. Becker R, D'Angelo G, Ghobadi S. Improving fairness in information exposure by adding links. AAAI. 2023;37(12):14119–26. https://doi.org/10.1609/aaai.v37i12.26652
32. Bhaskara A, Crane A, Mazumder MMHU, Sullivan BD, Yalamanchili P. Optimizing information access in networks via edge augmentation. arXiv preprint 2024. http://arxiv.org/abs/2407.02624
33. Barnes K, Ellis-Einhorn M, Chavez-Ruelas C, Hasan N, Fanous M, Sullivan BD. Edge interventions can mitigate demographic and prestige disparities in the Computer Science coauthorship network. arXiv preprint 2025. https://doi.org/arXiv:250604435
34. Jalali ZS, Chen Q, Srikanta SM, Wang W, Kim M, Raghavan H, et al. Fairness of information flow in social networks. ACM Trans Knowl Discov Data. 2023;17(6):1–26. https://doi.org/10.1145/3578268
35. Bashardoust A, Beilinson HC, Friedler SA, Ma J, Rousseau J, Scheidegger CE. Information access representations and social capital in networks. arXiv preprint 2020. https://arxiv.org/abs/2010.12611
36. Saxena A, Fletcher G, Pechenizkiy M. FairSNA: algorithmic fairness in social network analysis. ACM Comput Surv. 2024;56(8):1–45. https://doi.org/10.1145/3653711
37. Chierichetti F, Kumar R, Lattanzi S, Vassilvitskii S. Fair clustering through fairlets. In: Advances in Neural Information Processing Systems, Long Beach, CA, USA. 2017. p. 5029–37.
38. Granovetter MS. The strength of weak ties. American Journal of Sociology. 1973;78(6):1360–80. https://doi.org/10.1086/225469
39. Granovetter M. Threshold models of collective behavior. American Journal of Sociology. 1978;83(6):1420–43. https://doi.org/10.1086/226707
40. Goldenberg J, Libai B, Muller E. Talk of the network: a complex systems look at the underlying process of word-of-mouth. Marketing Letters. 2001;12(3):211–23. https://doi.org/10.1023/a:1011122126881
41. Borgs C, Brautbar M, Chayes J, Lucier B. Maximizing social influence in nearly optimal time. In: Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms. SIAM; 2014. p. 946–57.
42. Tang Y, Xiao X, Shi Y. Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data; 2014. p. 75–86.
43. Tang Y, Shi Y, Xiao X. Influence maximization in near-linear time: A martingale approach. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data; 2015. p. 1539–54.
44. Nguyen HT, Thai MT, Dinh TN. Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In: Proceedings of the 2016 international conference on management of data. 2016. p. 695–710.
45. Huang K, Wang S, Bevilacqua G, Xiao X, Lakshmanan LVS. Revisiting the stop-and-stare algorithms for influence maximization. Proc VLDB Endow. 2017;10(9):913–24. https://doi.org/10.14778/3099622.3099623
46. Ohsaka N. The solution distribution of influence maximization: a high-level experimental study on three algorithmic approaches. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. 2020. p. 2151–66. https://doi.org/10.1145/3318464.3380564
47. Clauset A, Tucker E, Sainz M. The Colorado index of complex networks. 2016. https://icon.colorado.edu/
48. Peixoto TP. The Netzschleuder network catalogue and repository; 2020. https://doi.org/10.5281/zenodo.7839981
49. Ghasemian A, Hosseinmardi H, Clauset A. Evaluating overfit and underfit in models of network community structure. IEEE Trans Knowl Data Eng. 2020;32(9):1722–35. https://doi.org/10.1109/tkde.2019.2911585
50. Ikehara K, Clauset A. Characterizing the structural diversity of complex networks across domains. arXiv preprint 2017. http://arxiv.org/abs/1710.11304
51. Hagberg AA, Schult DA, Swart PJ. Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux G, Vaught T, Millman J, editors. Proceedings of the 7th Python in Science Conference. Pasadena, CA USA; 2008. p. 11–5.
52. Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Computer Networks. 1998;30:107–17.
53. Freeman LC. Centrality in social networks conceptual clarification. Social Networks. 1978;1(3):215–39. https://doi.org/10.1016/0378-8733(78)90021-7
54. Newman MEJ. Networks. 2nd ed. Oxford, UK: Oxford University Press; 2018.
55. Boldi P, Vigna S. Axioms for centrality. arXiv preprint 2013. http://arxiv.org/abs/1308.2140
56. Zhang S, Sun J, Lin W, Xiao X, Huang Y, Tang B. Information diffusion meets invitation mechanism. In: Companion Proceedings of the ACM Web Conference 2024. 2024. p. 383–92. https://doi.org/10.1145/3589335.3648337