Maximizing Spread of Influence and Mitigating Overexposure in a Stochastic Social Network

Hangil Chung
hc685@cornell.edu

Soobin Han
sh767@cornell.edu

INTRODUCTION

Social networks serve a fundamental role in the spread of ideas throughout the world. With the increased connectivity of the world, social networks have become an ever increasing important aspect of how information spreads. For example, when a new phone app such as Instagram is introduced to the early adopters, these adopters continue to spread the app via the network. How these ideas and information is spread, how rapidly they are spread and to whom they are spread are vital questions surrounding the analysis of social networks.

Conventionally, the metric for evaluating a spread of an "idea" on the graph has been the total number of "activated" nodes, that receive these "ideas" from their neighbors. Examples of such cascading networks include spread of epidemics through a network of individuals, spread of product advertisement, etc. This type of evaluation assumes that each activated nodes only contribute monotonically to the overall influence of the "idea"; every infected person worsens the severity of an epidemic, and every person exposed to a new shoe commercial increases the performance of the marketing strategy.

However, we can easily imagine a scenario where a newly exposed person contributes negatively to the performance of the spread. Imagine a setting where the nodes in a social network are people to whom we want to introduce a new product. These people each have a liking/disliking to the product, and each have a set of neighbors to whom they deliver new ideas with some probability. Those who are satisfied with the product will contribute positively to the overall reputation of the product while those who are not will contribute negatively. Multiple studies have shown that a greedy approach of simply maximizing the spread can actually hurt the payoff of the strategy.

Under these assumptions, we want to mitigate for what [?] calls over-exposure, the spread of the "idea" to the nodes in the network who will be dissatisfied upon exposure and ultimately contribute negatively to the success of the spread. Therefore, the goal is to maximize the number of exposed and satisfied nodes, while minimizing the number of exposed but dissatisfied ones. A natural problem which rises from both these settings is how to choose the initial set of early adopters, to whom the product is introduced, such that the desired outcome is achieved.

[?] formulates this model and proposes a payoff metrics $\pi_i = p|C_i^o| - q|C_i^b|$, where C_i^o is the set of satisfied exposed nodes and C_i^b the set of dissatisfied exposed nodes. It considers two cases: 1) the unbudgeted problem in which the size of the seed set is not limited to a fixed budget, and 2) the budgeted problem in which the size of the seed set is limited to some budget. It shows that the payoff maximization problem under this model can be reduced to a simple max network flow problem and thus that there exists a polynomial time solution.

In this paper, we propose a new network model inspired by the formulation above. While [?]'s formulation assumes that any activated node deterministically activates all of its neighbors, we impose a stocasticity to all the edges. The motivation behind introducing stochasticity is to try to better reflect real-life social networks, where people do not always deterministically share a product they like to their friends and social circles, but rather share it with some probability. In other words, any activated node will only probabilistically activate each of its neighbors. This formulation now introduces an interesting change in that no one seed set will deterministically produce a maximum payoff. Instead, we can only compute the expected payoff for a seed set. If we use the payoff above, the expected payoff of a seed set S is:

$$E[\pi_i] = E[p|C_i^o| - q|C_i^b|] = \quad (1)$$

The equation above implies that we must account for every single path between a pair of nodes to compute the likelihood of a node being activated. [?] shows that counting all simple paths between a pair of nodes is #P-hard. While possible, it would be computationally infeasible for, say, a marketing agent to compute the expected payoff in a reasonably large network.

In this paper, we examine different computationally feasible *seed policies* to choose the initial set of seed adopters to both maximize the spread of influence throughout a social network while also mitigating for over-exposure. More precisely, given a set of positive targets we want to reach, and a set of negative targets we want to avoid, we examine how to choose the initial seed set to introduce a product to, such that the product is spread to positive targets and not spread to negative targets. Further, we consider a stochastic network where each edge has a propagation probability, in the sense that user A, who has liked the introduced product, has a certain probability of

CHI'16, May 07–12, 2016, San Jose, CA, USA

ISBN 123-4567-24-567/08/06...$15.00

DOI: http://dx.doi.org/10.475/123_4

introducing the product to its neighbor B. To extend the theme of computational feasibility, we discuss these policies with respect to their *efficiency* and *simplicity* and the trade-offs between the two. Below, we explain what we mean by these two terms:

Efficiency. The main goal of *seed policies* is to introduce the product to the positve target set and avoid introducing the product to the negative target set. We explain in subsequent sections a performance metric which measures how well a policy achieves this goal and thus how efficient the policy is.

Simplicity. While efficiency is a highly-desirable aspect of any policy, so is simplicity. If the calculations required to find the seed set is simple, it provides practical benefits for the user of the policy introducing the product. Particularly, if the network in consideration is extremely large, or if the given information of the network is not perfect, complex policies may no longer become feasible, emphasizing the desire for simple policies.

We propose and evaluate a number of policies that differ in the level of efficiency and simplicity in this regard and examine the inherent trade-offs between simplicity and efficiency.

DATA ANALYSIS AND MODEL DEFINITIONS

In this section, we describe the data-set used in our analysis, define the exact model we examine, and state the performance metric used to evaluate the various seeding-policies.

Data-Set

We used an email network of a large European research institution (found on the Snap database: 'email-Eu-core network') as the main social network to examine and test on. The network consists of 1005 nodes and 25571 edges. Each node in the network represents an email user, and an directed edge,$e_{i,j}$, exists if user i has sent an email to user j. The network was purposely chosen due to tis relatively small size such that the more calculation-intensive and complex seeding-policies could be run feasibly on the network.

Model Definitions

In our model, we consider the problem of introducing a product to a social network. The goal is to introduce the product to as many people who will like the product, and avoid introducing the product to people who will dislike it. Further, we assume people who like the product will introduce the product to their friends (neighbors), while people who dislike the product will not.

Formally, let us first define G to be the network described in the data-set, and V and E to be the set of nodes and edges in the graph respectively. Further, each node $v_i \in V$ has an "appeal threshold" parameter, t_i, associated with it. This parameter represents how appealing an product has to be for the node to like the product. The product's actual appeal parameter is defined as θ. Then for any node if $t_i \leq \theta$, the node will like the product and otherwise dislike the product. Let us define V^+ and V^- to be the set of positive and negative target nodes that like and dislike the product respectively. V^+ and V^- is defined as follows:

$$V^+ = \{v_i | \theta \geq t_i\} \tag{2}$$

$$V^- = \{v_i | \theta < t_i\} \tag{3}$$

In our analysis we consider nodes in V^+ to be propagating nodes and nodes in V^- to be blocking nodes. Propagating nodes, when introduced to the product, like the product and continue to propagate the node to any of it's neighbors who have yet to be introduced to the product. Blocking nodes, in contrast, dislike the product and do not propagate the product to any of it's neighbors. We can model this behavior, by removing any outgoing edges from nodes belonging to V^-.

Then we can partition V^+ into distinct clusters, by grouping by nodes that are reachable to each other. More formally, the cluster containing node $v_i \in V^+$ also contains all other nodes that are reachable from v_i when considering the propagation rules explained above. The process of identifying and defining these clusters is more formally defined in [2].

Now let us define G_θ to be the subgraph of G consisting of only the nodes in the largest cluster and the edges between these nodes. Then let us define V_θ, E_θ, V_θ^+, V_θ^- in the same manner, except limited to subgraph G_θ. In our analysis we will actually be focusing on these subgraphs G_θ which consist of only the largest cluster. The reason for this narrowing of scope is that we found G to be highly connected, meaning the smaller clusters are extremely small and provide little insight and value.Finally, to introduce stochasticity into the network, let us define every edge in E_θ to have a uniform propagation probability p, meaning the nodes in V_θ^+ propagate the product to each of it's neighbors who have not yet been introduced with probability p.

Given such a model, the problem examined is how to choose the initial set of seed nodes to introduce the product to. In our analysis we will examine the model under two regimes: budgeted and unbudgeted. The budgeted regime places the restriction that only a single seed node can be used and aligns with practical purposes, where companies have limited resources to introduce a new product. The second regime lifts the restriction and assumes an unlimited number of initial seed nodes can be chosen.

Performance Metric

The goal of seeding-policies is to introduce the product to nodes in the positive target set, V_θ^+, and not introduce the product to nodes in the negative target set, V_θ^-. For a seed set S, we can run a simulation to see which nodes the product is introduced to. Let us define this set to be T_S. Then let us define the score of the i^{th} simulation δ_i to be sum of the number of positive target nodes in T_S and the number of negative target nodes not in T_S. Formally, $\delta_i(S)$ is defined as follows:

$$\delta_i(S) = \sum_{v_i \in V_\theta^+} 1_{\{v_i \in T_S\}} + \sum_{v_i \in V_\theta^-} 1_{\{v_i \notin T_S\}} \tag{4}$$

where the notation $1_{\{v_i \in V_\theta^+\}}$ means the indicator function that is equal to 1 if $v_i \in V_\theta^+$, and is 0, otherwise.

To calculate the overall expected performance of a policy, we take the average performance over N simulations. Thus let us define $\Delta(S)$ to be the average score of $\delta(S)$ over N simulations. In our analysis N is set to be 1000, and $\Delta(S)$ is calculated as follows:

$$\Delta(S) = \frac{1}{N} \cdot \sum_{i=1}^{N} \delta_i(S) \tag{5}$$

Finally, to allow for comparison of scores across different θ values, we will present the results in a relative score. Theoretically, the best possible score achievable is equal to the sum of the cardinalities of the positive and negative target sets, and the relative score is calculated by dividing by this best possible score. Thus the overall relative score, $\phi(S)$ is calculated as follows:

$$\phi(S) = \frac{1}{|V_\theta^+| + |V_\theta^-|} \cdot \Delta(S) \tag{6}$$

POLICIES

In this section, we define a number of seeding policies. For each of the policies defined below, we define the policy under both the budgeted and unbudgeted regime.

Random

Budgeted: This policy simply choses at random a seed node from the set of positive target nodes, V_θ^+, and serves as the baseline policy with which to compare other heuristics.

Unbudgeted: This policy chooses at random $(1-p) \cdot |V_\theta^+|$ seed nodes from V_θ^+. The intuition behind choosing $(1-p) \cdot |V_\theta^+|$ seed nodes is that the propagation probability p captures how connected the network will be in hindsight of running a simulation. The more connected the network is, the fewer seed nodes we will need. Hence, we take $(1-p)$ fraction of the positive target set cardinality.

This is the simplest policy, and does not utilize any information of the social network.

Degree Centrality

Budgeted: This policy chooses from V_θ^+ the node which has the highest degree when considering positive nodes to be $+1$ degree and negative nodes to be -1. Formally, let us define $N(v_j)$ to be the neighbors of node v_j, then the seed node v^* is defined as follows:

$$v^* = \arg\max_{v_i} \sum_{v_j \in N(v_i)} 1_{\{v_j \in V_\theta^+\}} - 1_{\{v_j \in V_\theta^-\}} \tag{7}$$

where $1_{\{v_j \in V_\theta^+\}}$ means the indicator function that equals 1 if $v_j \in V_\theta^+$ and 0 otherwise. The intuition behind this policy is that this node will have the highest initial propagation rate to other positive target nodes and while avoiding negative nodes.

Unbudgeted: In the unbudgeted regime, the same algorithm is used, except the top $(1-p) \cdot |V_\theta^+|$ nodes with highest degree is used, using equation 5.

This policy serves as an example of a rather simple policy, as it only utilizes information about the direct neighbors of nodes and doesn't incorporate any distance measure in its calculations. This allows the policy to be scalable to larger networks.

Farthest from Negative

Budgeted: This policy is based on the intuition that we want to avoid the "bad" negative nodes. Thus the policy first calculates the expected shortest-distance between all pairs of points within G_θ, and chooses the seed node to be the node which has the largest average expected distance to negative nodes. We use the term "expected distance" because we incorporate into our calculations the probability of reaching a node via the shortest path. Specifically, if the shortest-path to a negative node is 3 (3 edges away), then the expected distance is defined to be $\frac{1}{p^3}$ to reflect the probability of actually reaching the node. If we define $d_{i,j}$ to be the distance between v_i and v_j, then the seed node v^* is chosen as follows:

$$\arg\max_{v_i} \frac{1}{|V_\theta^-|} \sum_{v_j \in V_\theta^-} \frac{1}{p^{d_{i,j}}} \tag{8}$$

Unbudgeted: The same algorithm is used, except we choose the top $(1-p) \cdot |V_\theta^+|$ seed nodes using equation 6.

This policy proves to be a semi-simple policy. The policy is more complex than previous policies in the sense that it includes a distance measure in it's calculations. However, the policy is still relatively simple as it only accounts for the negative nodes and doesn't account for the positive nodes.

Near-Far

Budgeted: This policy combines our desire to be far from negative target nodes and near positive target nodes. Thus, using the same expected minimum-distance between all points, the policy calculates for each node the sum of the distances to the positive nodes minus the sum of the distances to the negative nodes, and chooses the node with the largest value to be the seed node. The seed node v^* is chosen as follows:

$$\arg\max_{v_i} \sum_{v_j \in V_\theta^+} \frac{1}{p^{d_{i,j}}} - \sum_{v-j \in V_\theta^-} \frac{1}{p^{d_{i,j}}} \tag{9}$$

Unbudgeted: In the unbudgeted regime, the same algorithm is used and we include all nodes which are more positive than negative. Intuitively, we can imagine the algorithm as considering all the positive and negative nodes and weighing each by the inverse of its distance and choosing nodes who weigh more positive. The set of nodes S is defined as follows:

$$S = \{v_i | \sum_{v_j \in V_\theta^+} \frac{1}{p^{d_{i,j}}} - \sum_{v_j \in V_\theta^-} \frac{1}{p^{d_{i,j}}} > 0\} \tag{10}$$

This policy proves to be a complex policy. It accounts for both the positive and negative nodes and also utilizes the distance metric.

Betweenness Centrality

Budgeted: This policy considers the betweenness centrality of the positive target nodes. More specifically, it calculates the betweenness measure of each node when considering only the positive target nodes in V_θ^+, and chooses the node with highest betweenness measure. The seed node v^* is calculated as follows:

$$\arg\max_{v_i} \sum_{v_j \neq v_k \neq v_i \in V_\theta^+} \frac{\sigma_{j,k}(i)}{\sigma_{j,k}} \tag{11}$$

where $\sigma_{j,k}$ is the number of shortest-paths between v_j and v_k, and $\sigma_{j,k}(i)$ is the number of those shortest-paths which pass through v_i.

Unbudgeted: In the unbudgeted regime, the same algorithm is used except similar to before, we choose the top $(1-p) \cdot |V_\theta^+|$ seed nodes using equation 9.

This policy also proves to be a semi-simple policy as it utilizes the distance metric but still fails to account for the negative nodes.

Katz Centrality

Katz centrality c_v for node v in a graph can be calculated as following:

$$c_v = \sum_{k=1}^{K} \sum_{u \neq v} \alpha^k (A_k)_{v,u} \tag{12}$$

, where (A_k) is the degree k adjacency matrix. In other words, $(A_k)_{v,u} = 1$ if nodes u and v are degree k neighbors, meaning that exists a length k path from node v to node u. α is an attenuation factor that discounts the effect of a degree k neighbor as k grows.

This can be considered an extended version of the degree centrality policy which only accounts for the immediate - i.e. degree 1 - neighbors for each node; instead, Katz centrality also accounts for the influence of all nodes within distance K. Furthermore, note that node u may be node v's degree k neighbor for multiple values of k. In this case, Katz centrality accounts for multiple paths between the two nodes.

To accommodate our model, however, we make one alteration to this measure. Instead of having a binary degree k adjacency matrix, where $(A_k)_{v,u}$ is assigned 1 if v and u are degree k neighbors and 0 otherwise, we differentiate when the node u is a positive or a negative target node. More specifically, our algorithm generates degree k adjacency matrices A_k^* such that:

$$(A_k^*)_{v,u} = \begin{cases} 1 & v \text{ and } u \text{ are degree } k \text{ neighbors and } v \in V_\theta^+ \\ -1 & v \text{ and } u \text{ are degree } k \text{ neighbors and } v \in V_\theta^- \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, we choose α to be p, the propagation probability that we impose on all the edges. For an immediately adjancent pair of nodes $u \in V_\theta^+$ and $v \in V_\theta^+$, the probability that v activates u is p. For a pair of degree k nodes u and v, both in V_θ^+, the probability that one activates the other via a particular length k path is p^k. By setting the attenuation factor as p, we attempt to capture node u's influences on node v via length k paths for multiple k's.

Finally, we consider the parameter K. K indicates the greatest degree neighbor we take into account when computing the centrality of a node. Greater values of K lets us take into account a greater number paths with different lengths between any pair of nodes. A short analysis of our sample graph revealed that most pairs of nodes have a shortest paths of length at most 5, meaning that setting $K = 5$ allows us to account for the influence of most nodes in the graph when computing the centrality of the node of interest. While greater values of K may provide us greater insight, the computational complexity of computing Katz centrality grows linearly with K; therefore we choose $K = 5$ for our algorithm.

Budgeted: For the budgeted case, this policy selects node v^* such that

$$v^* = \arg\max_v \sum_{k=1}^{K=5} \sum_{u \neq v} p^k (A_k^*)_{v,u} \tag{13}$$

Unbudgeted: For the budgeted case, we select $(1-p) \cdot |V_\theta^+|$ nodes with the greatest values of Katz centrality describe above in equation 12.

This policy is another example of a complex policy, that utilizes the distance metric and also accounts for both the positive and negative nodes.

RESULTS

In this section we present the results for the seeding-policies when run across various regimes with different θ and p values. We first focus on our results during the Budgeted regime, then show the results for the Unbudgeted regime. The scores for the Budgeted regime are shown in Figure 1, and the scores for the Unbudgeted regime are shown in Figure 2. Thereafter, we compare the policies across the two regimes in Figure 3. All scores, $\phi(S)$ are given as a fraction of the optimal possible score. We begin by giving a high-level summary of our most important findings before providing more details for each of them.

Key Observations of Budgeted Results

Considering the plots presented in Figure 1, it is noticeable that the Farthest policy (Farthest from Negative) performs well when $\theta < 0.5$ and performs very poorly when $\theta > 0.5$. In contrast, the Betweenness policy performs well when $\theta > 0.5$ and

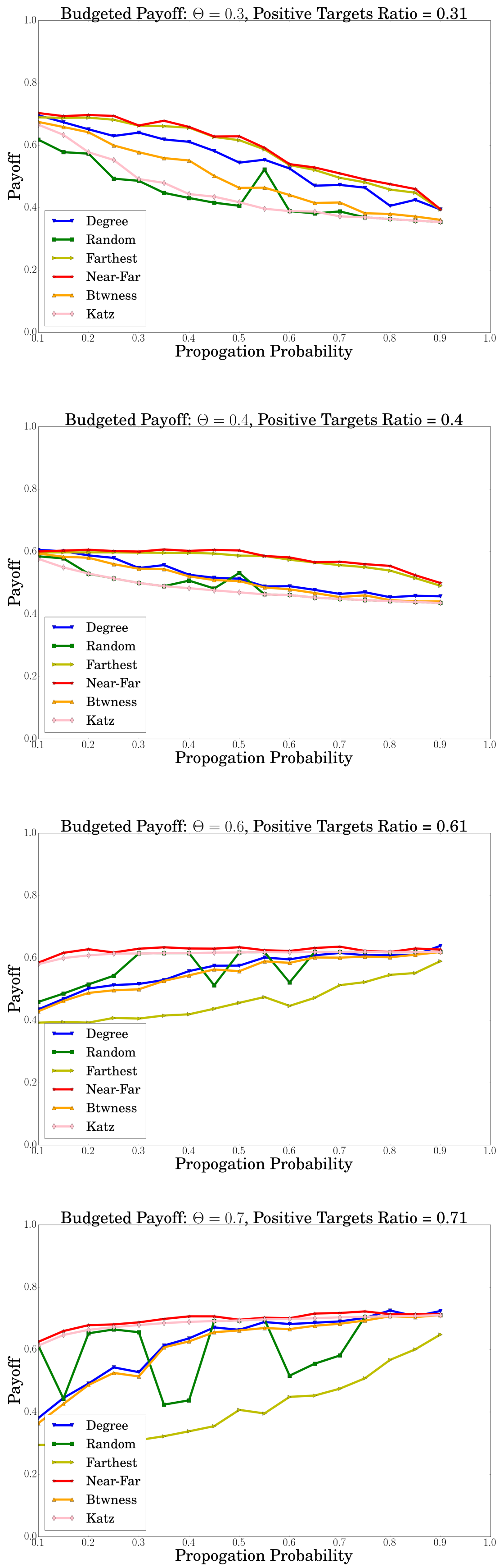

Figure 1: Policy Budgeted Results

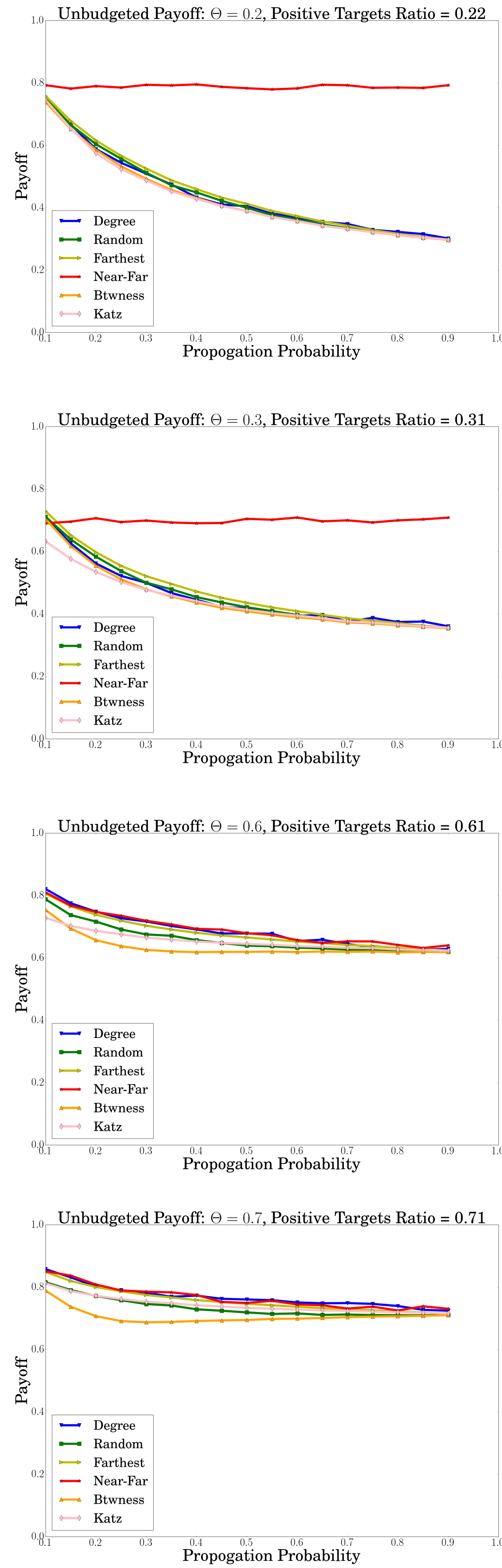

Figure 2: Policy Unbudgeted Results

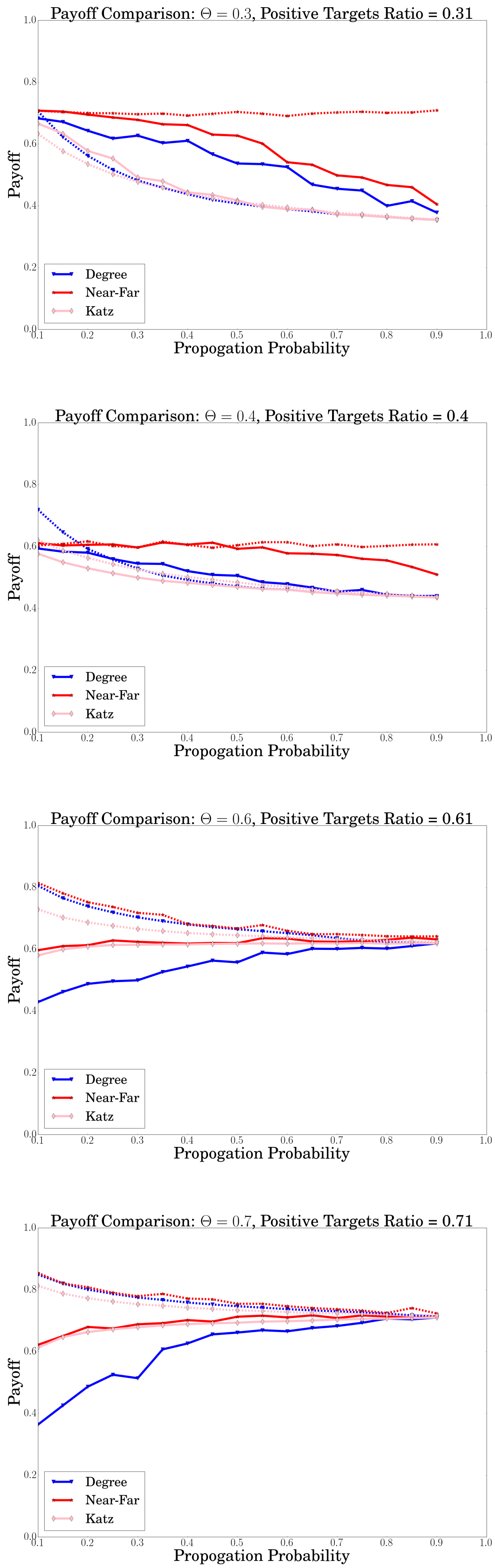

Figure 3: Comparison of Budgeted and Unbudgeted Results. Dashed lines are Unbudgeted scores.

performs very poorly when $\theta < 0.5$. These two policies show the limitations of too simple policies. Next we observe that the Near-Far policy outperforms all other policies regardless of the θ value and is less sensitive to it, highlighting the efficiency of the policy at the cost of simplicity. Finally, we see that there does seem to exist an inherent trade-off between simplicity and efficiency.

Farthest and Betweenness Performance

From Figure 1, we see an interesting trend that the Farthest policy performs well when $\theta < 0.5$ and performs very poorly when $\theta > 0.5$ and the Betweenness policy is the exact opposite. Intuitively, this makes sense. The Farthest policy takes into account only the negative nodes, and tries to pick nodes that are as far away as possible, making it well suited for regimes when $\theta < 0.5$ as the majority of the nodes in the graph are negative target nodes. In contrast, the Betweenness policy takes into account only the positive nodes, and chooses the node with highest betweenness measure amongst the positive nodes, making it well suited for regimes when $\theta > 0.5$ as the majority of the nodes in the graph are positive target nodes. The performance of these two policies show the limitations of only considering either the positive or negative target nodes and exemplifies the cost in efficiency for simpler policies.

Near-Far Outperforms

The Near-Far policy outperforms all other policies regardless of the θ and p values, and can be seen in Figure 1. Further, it seems to be the only policy that is less sensitive to the θ value and performs well regardless.

Simplicity and Efficiency Trade-off

We saw earlier that the Farthest and Betweenness policies, while they utilize the computationally heavy distance metric, are still rather simplistic policies in the sense that they only consider either the positive or negative nodes. As a result, these policies can only handle one side of the spectrum of θ values and performs poorly on the other. The Degree policy, in contrast, considers both the positive and negative nodes, but does not perform any distance metric calculations, once again making it simplistic. As a result, we see that the Degree policy performs mediocrely across all θ values. Finally, we see that with the Near-Far policy, which considers both the positive and negative nodes, and also utilizes the distance metric is a more complex policy and proves to outperform under all regimes. These results highlight the inherent trade-off between simplicity and efficiency within the policies. As we sacrifice information and calculations, we have a more simplistic policy but at the cost of efficiency.

Key Observations of Unbudgeted Comparison

Considering the plots presented in Figure 2, we see that when $\theta < 0.5$, all policies but the Near-Far performs very poorly in the unbudgeted regime, exemplifying the Near-Far's superior efficiency. When $\theta > 0.5$, we see the scores are more similar, but the trend of Near-Far outperforming all other policies still remains. Finally, looking at Figure 3 that compares the scores in the budgeted and unbudgeted regime, we see that the Near-Far policy performs better when unbudgeted. The other policies; however, perform better in the budgeted regime when $\theta < 0.5$.

Near-Far Outperforms

The similar trend that Near-Far outperforms all other policies can be seen in the unbudgeted regime. This is especially obvious when $\theta < 0.5$ and still prevalent when $\theta > 0.5$. At first glance, one might think that this is because many of the policies besides Near-Far pick $(1-p) * |V_\theta^+|$ seed nodes. Intuitively, it is natural that when $\theta < 0.5$, there are more negative nodes, and thus enforcing one to pick many seed nodes forces the policies to be bad, while when $\theta > 0.5$ this is not a problem. However, the Katz policy, like Near-Far, does not have this restriction and still performs poorly. This shows that the Near-Far policy is actually good at handling the unbudgeted regime and choosing only necessary seed nodes.

Unbudgeted vs. Budgeted

From Figure 3, we see that the Near-Far policy is able to take advantage of the unbudgeted regime and always performs better than when it is budgeted, regardless of the θ value. Besides the Near-Far policy, we see the general trend that the policies score higher in the unbudgeted regime only when $\theta > 0.5$ and worse when $\theta < 0.5$. This makes sense, as the other policies are not good at choosing only necessary seeds. Thus, when $\theta > 0.5$ and there are more positive nodes, the unbudgeted regime is able to perform better, while it underperforms in the $\theta < 0.5$ setting.

CONCLUSION

We have shown several heuristics that vary in the level of knowledge and computing resources available and their effectiveness in maximizing positively activated nodes while minimizing the negatively activated nodes. We see a clear tradeoff between the simplicity and the efficiency of the algorithms. Simpler algorithms that require less information about the network or are less computationally complex, such as Degree policy are easier to compute but result in poor payoff, whereas more complex algorithms, such as Near-Far, which requires us to compute the pairwise distances for all pairs of nodes, perform well across all conditions. However, it is worth noticing that a simpler Degree algorithm only performs slightly worse in most cases and returns a reasonable seed set even when compared to Near-Far. While having sacrificed performance slightly, Degree algorithm may be an attractive choice especially when the size of the network calls for a heavy computation.

Another interesting observation that we can make is that the in unbudgeted case in which we can choose any number of seed agents, we do not see a significant increase in the eventual payoff. One would expect that increasing the size of the seed set would enable us to select a seed set that results in a much higher payoff closer to the optimal payoff of 1. However, we observe that the policy selection contributes more to the performance, as it can be see with the superior performance of Near-Far.