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

Hangil Chung
hc685@cornell.edu

Soobin Han
sh767@cornell.edu

ABSTRACT

Write Abstract

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. Particularly, imagine the 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 whom they interact with, and may or may not spread the product to. In some settings one might want to maximize the spread of the product throughout the network. In other cases; however, one might want to mitigate for over-exposure and minimize the number of people who are introduced to the product and dislike it. 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.

In this paper, we examine different *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 introducing the product to its neighbor B. The design of these policies involve trade-offs between *efficiency* and *simplicity*. 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 information of the network is not perfect, complex policies may no longer become feasible, emphasizing the desire for simple policies.

We show that... WHAT DO WE SHOW!!

Related Work

Include related work: what motivated this.

DATA ANALYSIS AND MODEL DEFINITIONS

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

Data-Set

The data set we used comprises of an email network from a large European research institution (found on the Snap database: 'email-Eu-core network'). The network consists of 1005 nodes and 25571 edges. Each node in the network represents an email user, and an undirected edge exists between two users if they have exchanged emails.

Model Definitions

First, let us 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, let us define that the product we aim to introduce has a parameter θ which represents how appealing the product is. Each node $v_i \in V$ in the network also has a threshold level t_i, such that if θ is greater than t_i, the node v_i likes the product and belongs in the set of positive nodes, V^+. If θ is less than the threshold level, the node dislikes the product and belongs in the set of negative nodes V^-. Formally, V^+ and V^- is defined as follows:

$$V^+ = \{v_i | \theta \geq t_i\} \tag{1}$$

$$V^- = \{v_i | \theta < t_i\} \tag{2}$$

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

Further, in our analysis we consider nodes in V^+ to be propagating nodes and nodes in V^- to be blocking nodes. More specifically, propogating 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 and blocking nodes dislike the product and do not propagate the product to any of it's neighbors. In this sense, we can partition G into clusters consisting of nodes belonging to V^+ and V^-. More formally, for any node v_i in V^+, the cluster of nodes it belongs to is the set of nodes which are reachable from v_i. The process of identifying and defining these clusters is more formally defined in [2].

Then 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 as above, 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 that G to be highly connected, meaning that the smaller clusters are extremely small and provide little insight.

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. Thus, given an initial seed set of nodes S to which the product is introduced, each node in S starts to propagate the product to each of its neighbors who have not yet been introduced, with probability p if the node belongs to V^+, and stops propagation otherwise.

Performance Metric

The goal of the policy 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^-. Thus, for a particular simulation in which we choose the seed set to be S, let us define T_S to be set of nodes which are introduced to the product. Then let us define the score of the $i^t h$ simulation δ_i to be sum of the number of positive target nodes in T_S and the number of the negative target nodes not in T_S. Formally, $\delta_i(S)$ is defined as follows:

$$\delta_i(S) = \sum_{v_i \in V_\theta^+} 1_{\{v_i \in T_S\}} + \sum_{v_i \in V_\theta^-} 1_{\{v_i \notin T_S\}} \tag{3}$$

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 and calculated as follows:

$$\Delta(S) = \frac{1}{N} \cdot \sum_{i=1}^{N} \delta(S) \tag{4}$$

POLICIES

In this section, we define a number of seeding policies. In our analysis, we examine the model under two regimes: budgeted and unbudgeted. In the budgeted regime, we assume we can only pick one starting seed. In the unbudgeted regime we assume we can choose as many starting seed nodes as needed. Note that because we consider G_θ, which consists of the single largest cluster, that a single seed can theoretically reach all nodes within the cluster. 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: The 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 when imagining that we flip a coin for each edge to determine if the edge propagates or not in the simulation. Thus, 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.

Degree Centrality

Budgeted: This policy chooses from V_θ^+ the node which has the highest degree when considering only other positive target nodes. More specifically, the policy chooses the node within V_θ^+ which has the most number of neighbors also belonging to V_θ^+. Formally, the seed node v^* chosen can be defined as follows:

$$v^* = \arg\max_{v_i} \sum_{v_j \in N(v_i)} 1_{v_j \in V_\theta^+} \tag{5}$$

The intuition behind this policy is that this node will have the highest initial propagation rate to other positive target nodes.

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

Farthest from Exterior

Budgeted: This policy is based on the intuition that we want to avoid the "bad" negative nodes. Thus the policy first calculates the expected minimum-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 probabilities of each edge. Formally 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:

$$\arg\max_{v_i} \frac{1}{|V_\theta^-|} \sum_{v_j \in V_\theta^-} \frac{1}{p^{d_{i,j}}} \tag{6}$$

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

Near-Far

Budgeted: This policy combines our desire to be far from negative target nodes and desire to be 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 good nodes minus the sum of the distances to the bad nodes, and chooses the node with the largest value to be the seed node. The seed node v^* is chosen as follows:

$$\arg\max_{v_i} \sum_{v_j \in V_\theta^+} \frac{1}{p^{d_{i,j}}} - \sum_{v-j \in V_\theta^-} \frac{1}{p^{d_{i,j}}} \tag{7}$$

Unbudgeted: In the unbudgeted regime, the same algorithm is used except 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. Thus the set of nodes S can be defined as follows:

$$S = \{v_i| \sum_{v_j \in V_\theta^+} \frac{1}{p^{d_{i,j}}} - \sum_{v-j \in V_\theta^-} \frac{1}{p^{d_{i,j}}} > 0\} \tag{8}$$

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 centrality. Thus the seed node v^* is calculated as folllows:

$$\arg\max_{v_i} \sum_{v_j \neq v_k \neq v_i \in V_\theta^+} \frac{\sigma_{j,k}(i)}{\sigma_{j,k}} \tag{9}$$

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 the equation described in the budgeted regime.

RESULTS

In this section we present the results for the various policies when run across various regimes with different θ and p values. Particularly, we present results when $\theta \in \{0.2, 0.4, 0.6, 0.8\}$, and for $p \in [0,1]$. To maintain the same scale across varying θ values, we present the results in relative terms of proportion of optimal score achieved. More specifically, for a G_θ the theoretical optimal score is when all positive target nodes are reached, and all negative nodes are avoided. Obviously, this is not always actually possible. However, it does provide an upper bound on the possible score and lets us present the results in relative scores.

CONCLUSION

We have proposed a number of data-driven policies to guide incentives for rebalancing in bike-sharing systems via crowdsourcing. While our analysis clearly displays the performance differences between these policies, the superior performance of the more dynamic policies comes with the cost of greater complexity for users and operators alike.

There are other important considerations beyond their simplicity and performance. For example, when comparing the performance of the Static and Cluster Hindsight policies, it seems unclear at first glance what additional value the Cluster Hindsight policy provides, given that it relies on heavier machinery – after all, they perform very similarly. However, the Static Hindsight policy can only be defined for stations for which the Static policy had been in place, whereas the Cluster Hindsight policy can be defined for other stations as well. Thus, in a way, each of the policies presented has its own advantage.

Most importantly, our analysis shows that slightly limiting the online fashion of decision-making only causes limited decreases in performance. On the academic side, this adds a data-driven analysis to a recent stream of literature in operations management that compares dynamic and static decision-making in similar applications. On the practical side, our analysis led to Citi Bike adopting a version of the Dynamic CC (30) policy in 2017.

REFERENCES

1. Siddhartha Banerjee, Daniel Freund, and Thodoris Lykouris. 2017. Pricing and Optimization in Shared Vehicle Systems: An Approximation Framework. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 517–517.

2. Siddhartha Banerjee, Ramesh Johari, and Carlos Riquelme. 2015. Pricing in ride-sharing platforms: A queueing-theoretic approach. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM, 639–639.

3. Yiwei Chen and Ming Hu. 2017. Pricing and matching with forward-looking buyers and sellers. (2017).

4. Iris A Forma, Tal Raviv, and Michal Tzur. 2015. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation Research Part B: Methodological* 71 (2015), 230–247.

5. Daniel Freund, Shane G. Henderson, and David B. Shmoys. 2017. Minimizing multimodular functions and

allocating capacity in bike-sharing systems. In *Integer Programming and Combinatorial Optimization Proceedings (Lecture Notes in Computer Science)*, F. Eisenbrand and J. Koenemann (Eds.), Vol. 10328. Springer, 186–198. arXiv preprint arXiv:1611.09304.

6. Daniel Freund, Ashkan Norouzi-Fard, Alice Paul, Shane G. Henderson, and David B. Shmoys. 2016. Data-Driven Rebalancing Methods for Bike-Share Systems. (2016). Working paper.
7. Christine Fricker and Nicolas Gast. 2012. Incentives and regulations in bike-sharing systems with stations of finite capacity. *arXiv preprint arXiv:1201.1178* (2012), 2.
8. Christine Fricker and Nicolas Gast. 2016. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *Euro journal on transportation and logistics* 5, 3 (2016), 261–291.
9. David K George. 2012. *Stochastic Modeling and Decentralized Control Policies for Large-Scale Vehicle Sharing Systems via Closed Queueing Networks*. Ph.D. Dissertation. The Ohio State University.
10. Sin C Ho and WY Szeto. 2014. Solving a static repositioning problem in bike-sharing systems using iterated tabu search. *Transportation Research Part E: Logistics and Transportation Review* 69 (2014), 180–198.
11. Nanjing Jian, Daniel Freund, Holly M Wiberg, and Shane G Henderson. 2016. Simulation optimization for a large-scale bike-sharing system. In *Proceedings of the 2016 Winter Simulation Conference*. IEEE Press, 602–613.
12. Deepak Merugu, Balaji S Prabhakar, and N Rama. 2009. An incentive mechanism for decongesting the roads: A pilot program in bangalore. In *Proc. of ACM NetEcon Workshop*.
13. NYCBS. 2016. Private Communication. (2016).
14. NYCBS. 2017a. Bike Angel Program. (2017). `https://bikeangels.citibikenyc.com`
15. NYCBS. 2017b. Citi Bike JSON Feed. (2017). `https://gbfs.citibikenyc.com/gbfs/en/station_status.json`
16. NYCBS. 2017c. Citi Bike System Data. (2017). `https://www.citibikenyc.com/system-data`
17. Eoin O'Mahony, Shane G. Henderson, and David B. Shmoys. 2016. (Citi)Bike sharing. (2016). working paper.
18. Eoin O'Mahony and David B Shmoys. 2015. Data Analysis and Optimization for (Citi) Bike Sharing. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 687–694.
19. Pulkit Parikh and Satish Ukkusuri. 2014. Estimation of Optimal Inventory Levels at Stations of a Bicycle Sharing System. *TRB 94th Annual Meeting* (2014).
20. Tal Raviv and Ofer Kolka. 2013. Optimal inventory management of a bike-sharing station. *IIE Transactions* 45, 10 (2013), 1077–1093.
21. Tal Raviv, Michal Tzur, and Iris A Forma. 2013. Static repositioning in a bike-sharing system: models and solution approaches. *EURO Journal on Transportation and Logistics* 2, 3 (2013), 187–229.
22. J. Schuijbroek, R.C. Hampshire, and W.-J. van Hoeve. 2017. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research* 257, 3 (2017), 992 – 1004. DOI: `http://dx.doi.org/https://doi.org/10.1016/j.ejor.2016.08.029`
23. Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. 2015. Incentivizing Users for Balancing Bike Sharing Systems.. In *AAAI*. 723–729.
24. WY Szeto, Ying Liu, and Sin C Ho. 2016. Chemical reaction optimization for solving a static bike repositioning problem. *Transportation research part D: transport and environment* 47 (2016), 104–135.
25. Ariel Waserhole and Vincent Jost. 2014. Pricing in vehicle sharing systems: optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics* (2014), 1–28.