Using Repeated Data to Predict Influential Users

Ann Kilzer* Aron Yu† C. Vic Hu‡

1 Introduction

We've all heard about viral marketing. Even Forbes.com compared the social graph to "crude oil," describing it as an "unrefined and complex natural resource containing many riches" [13]. Maciej Ceglowski, the developer behind Pinboard.in quipped "Social networks exist to sell you crap. The icky feeling you get when your friend starts to talk to you about Amway, or when you spot someone passing out business cards at a birthday party, is the entire driving force behind a site like Facebook" [1].

Advertisers want to run the most effective campaign using a limited budget. The whole idea behind viral marketing is to target users who will spread the word to their friends. But how do we choose which users to target? Our project is to find a scalable method for finding the most influential users in a social graph. We hypothesize that we can measure an individual's influence by the number of *repeaters* they connect to. By repeaters, we mean those users who redistribute content (Think Twitter's "retweets," Google+ and Facebook's "Share" features).

2 Setup

We will be running our tests on a Twitter dataset. Twitter is an online social network that specializes in "microblogging," sharing text snippets, or "tweets," of no more than 140 characters. It is a directed graph: you may follow others, subscribing to their tweets, but they may not necessarily follow you. Features of the service include the ability to "retweet," or repeat, verbatim, another's post. Additionally, users label their tweets via hashtags, short strings beginning with #, which provide some context to the tweet. For instance, a current popular hashtag is "#OcuppyOakland," which quickly provides context to tweets about the current political protests. Another feature is to directly reference users by putting an @ before their screen name.

Twitter distinguishes itself from other popular social networks in that it is easy to repeat content, either via retweets or repeating a hashtag. Posts are by default public, and twitter is a popular platform for celebrities, brands, and organizations to connect with their followers.

3 Project Status

3.1 Accomplishments

Over the past weeks, we have accomplished the following:

- acquired a dataset of tweets from the past year.
- wrote and tested python code to gather data on the twitter graph.
- wrote code to export python data to MATLAB

*Department of Computer Science, University of Texas at Austin, USA (akilzer@gmail.com)

†Department of Electrical & Computer Engineering, University of Texas at Austin, USA (aron.yu@utexas.edu)

‡Department of Electrical & Computer Engineering, University of Texas at Austin, USA (cvhu@utexas.edu)

- learned to use condor cluster to speed up our code
- investigated related work

3.2 Obstacles

Our twitter dataset is extremely large. For instance, 20 days worth of data from October was about 20G. We need to cut the problem down to an appropriate size so we can compute it in time. Additionally, we are having difficulty opening the large data files for processing. We are now using pypy and condor to speed things up.

3.3 Goals

Our plan is to crawl a subset of the twitter graph corresponding to the users in our tweet data. We will build a directed adjacency matrix solely based on the "follower" definition in twitter and export it into MATLAB. The *influence score* for each user will be calculated by our algorithm in MATLAB.

We split our data into two time periods. The first is used for training, the second is used for testing. We are assuming here that the users' behaviors remain relatively constant over the training and testing time periods.

During the training period, our algorithm first counts the number of tweets originating from each user and assigns an inidividual *resource score*. Then it counts the number of times each user retweets information or repeats a hashtag, and assigns an individual *repeat score*. Using these two scores and the adjacency matrix, we can compute the *influence score*. It is reasonable to assume that the most influential users will be the nodes with the highest resource score and the highest average repeat score among the followers. The resource score and the repeat score need to be weighed appropriately.

The simplest way of counting the influence score would be to count the number of followers with a repeat score over threshold N. More elaborate methods might sum the repeater score of all followers.

We plan to evaluate our algorithm by comparing it to other methods for computing influential users, such as degree count.

4 Related Work

To find the influential users in a social network, we need to understand and quantify some underlying properties such as centrality, communicability, and betweenness measures on a general matrix, and the framework that Estrada et al. put together meets this purpose perfectly [6]. By using the matrix exponentials of a network's adjacency, they introduced the *subgraph centrality* of node i $\left(I + A + \frac{A^2}{2!} + \cdots + \frac{A^k}{k!} + \cdots\right)_{ii} = (exp(A))_{ii}$[5] and *communicability* between node i and j $(exp(A))_{ij}$ [3]. Furthermore, to quantify the influence of a node,*betweenness* was introduced to describe the change of communicability when the node is removed, which is defined as

$$\frac{1}{(N-1)^2-(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{exp(A)_{ij} - exp(A - E(r))_{ij}}{exp(A)_{ij}}$$

where $i \neq j, i \neq r, j \neq r$, A-E(r) is the adjacency matrix when all edges connected to node r are removed[4].

As pointed out in a research paper by Kempe et al. at Cornell University [11], selecting the most influential users in a social network is a NP-hard optimization problem, and even a provable approximation for efficient algorithms can be challenging to solve. Inspired by two diffusion models of interacting particles, *Linear Threshold*[9] and *Independent Cascade Models*[8], where each node can be turned on as *active* from *inactive* based on its neighbor weights and activating threshold, the team used a natural greedy hill-climbing approach related to [2] to show that the performance is guaranteed to be at least 63% (1-1/e) of optimal, which significantly out-performs other algorithms targeting high-degree or "central" nodes. The group further generalized the proved strategies to more realistic marketing scenario, where marketing actions are no longer simplified as making a single

node active, but rather have stochastic effects to increase a subset of nodes' probability of becoming active (and each individual has different response to each actions.) To maximize the expected size of the final active set, they applied the hill-climbing algorithm on the expected revenue with respect to a vector of investment actions.

When given a social graph and a list of actions propagating through it, Mathioudakis et al. designed the SPINE algorithm to find the 'backbone' of the network through the use of the independent-cascade model [12]. The algorithm proved to be an effective pre-processing step for solving influence-maximization problems. The effectiveness of SPINE came from its ability to reduce computation speed significantly while giving up little accuracy. Its main applications included propagation characterization, feed ranking, and viral marketing.

To approximate a massive graph of networks data, Savas and Dhillon[14] provided a faster and better performance framework using clustered low rank matrices. While preserving essential information structure of a massive graph, their algorithm partitions nodes into clusters, computes low rank approximations independently, and combines each low rank approximations to obtain the approximation of the entire graph. The approximation errors were derived to have deterministic bounds using a stochastic algorithm introduced by Halko et al.[10] The clustering approach was experimented and shown to outperform traditional approximation methods in both speed efficiency and memory usage significantly.

Wang et al. took a community-based approach to finding the influential nodes by observing information diffusion [15]. Their research targeted the mobile social network as it modeled the spread of information on a large scale. Communities within a given network were identified through a modified version of the current community detection algorithms. The changes included the addition of information diffusion as a main factor, the handling of weighted directed graphs, and the ability to modify the threshold of community decision boundaries. Instead of mining every single communities in the network, they allowed the mining decision to be dynamically chosen by the algorithm. The selection of algorithm for finding top-K influential user is trivial. This new algorithm provided provable performance guarantee just like previous algorithms. When tested on a network with 15 times more user nodes than previous testing, the new algorithm proved to be orders of magnitude faster than the leading greedy algorithm, while maintaining a small error margin.

The dynamics of friendship and information sharing in a social network fit into Galeotti et al.s model of Network Games [7]. Here, players are connected in a graph. Players can either act or not act. In a game of *strategic substitutes*, a neighbor's action replaces the need for the player to act. In the *strategic complements* game, a neighbor's action gives the player greater incentive to act, while a neighbor's inaction gives the player less incentive to act. Viral marking in a social network displays features of both strategic complements and substitutes. For instance, if Alice shares content, her friend Bob may feel less need to duplicate content, since typically some of their friends (including Charlier) overlap. Perhaps Charlie only needs to hear the campaign once to be influenced. However, we might also imagine that Charlie would be more interested in the campaign if many of his friends were sharing information about it.

5 Acknowledgements

Thanks to Joseph Reisinger for providing us data from twitter crawls.

References

[1] Maciej Ceglowski. The social graph is neither, November 2011.

[2] P. Domingos and M. Richardson. Mining the network value of customers. *Seventh In-*

ternational Conference on Knowledge Discovery and Data Mining, 2001.

[3] E. Estrada and N. Hatano. Communicability in complex networks. *Phys. Rev. E*, 77(2007)(036111), 2007.

[4] E. Estrada, D. J. Higham, and N. Hatano. Communicability betweenness in complex networks. *Phys. A*, 388(2009):764–774, 2009.

[5] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Phys. Rev. E*, 71(2005)(056103), 2005.

[6] Ernesto Estrada and Desmond J. Higham. Network properties revealed through matrix functions. *SIAM Rev*, page 52:696714, 2010.

[7] Andrea Galeotti, Sanjeev Goyal, Matthew O. Jackson, Fernando Vega-Redondo, and Leeat Yariv. Network games. *Review of Economic Studies*, page 77, 2010.

[8] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development. *Academy of Marketing Science Review*, 2011.

[9] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, pages 85(6):1420–1443, 1978.

[10] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *Tech. re*, 2009.

[11] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of inuence through a social network. *ACM SIGKDD international conference on Knowledge discovery and data mining*, page 137146, 2003.

[12] Michael Mathioudakis, Francesco Bonchi, Carlos Castillo, Aristides Gionis, and Antti Ukkonen. Sparsication of inuence networks. *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011.

[13] Venkatesh Rao. The social graph as crude oil (go ahead, build that yasn!), October 2011.

[14] Berkant Savas and Inderjit S Dhillon. Clustered low rank approximation of graphs in information science applications. *SIAM International Conference on Data Mining*, 201.

[15] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k inuential nodes in mobile social networks. *16th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 10391048, 2010.