HiggsTweet: Analyzing Influence Propagation During a Viral Event on Twitter

Kristian Flatheim Jensen
Norwegian University of Science and Technology

Gudbrand Tandberg
The University of British Colombia

In this project we analyze the influence dynamics within a subset of physics-interested Twitter users before, during and after the 4th July 2012 announcement of the discovery of the Higgs boson, made by researchers at CERN in Switzerland. We study a dataset consisting of a 450k-user Twitter follower network together with a 563k line event log, collected between the 1st and the 7th of July 2012. We use the event log to calculate influence probabilities between pairs of users, these weights are then used to run simulations of Independent Cascade (IC) diffusion processes and to compute near-optimal seed sets using a greedy algorithm. We experiment with several different preprocessing steps and heuristics for reducing the size and runtime of the simulation and optimization steps, and compare seed-sets and expected user influence across our experiments.

1 INTRODUCTION

Analyzing the spread of information through a large and complex social network is a difficult and interesting problem. In the last 10 years, the Big Data revolution has been driven, to a large extent, by innovations in understanding how humans interact and live in a mobile world. At least some of this progress has been made thanks to inter-disciplinary efforts by researchers in sociology, psychographics, epidemiology, and computer science into understanding the dynamics of social interactions. Already it has become clear the immense opportunities and dramatic shifts this revolution has brought about for marketers, news agencies, individuals and more. We will see in the near dystopian future how the study of social networks will in fact be key to developing political theory (and practice) into the 21st century.

The main goal of the broader research agenda we are following is the open-ended and general question of analyzing the power and influence dynamics of a web-community before, during, and after some major event. This study of course has a much narrower scope. Our lower level goals for this project are twofold, first, we wanted to get hands-on experience with some of the material we covered in class, notably Influence Maximization (IM), second, we wanted to perform an as-complete-as-possible scientific exposition of a new and interesting dataset. We will see how successful we were at the end!

Some questions that guided our efforts were

- *Which users in the network should we influence, and when should we influence, if we want to spread a rumor during a viral event?*
- *How do the power dynamics, as computed from an event log change over time during a viral event?*
- *What does the typical and the atypical user look like, in terms of event history, during a viral event?*
- *Do the seed-sets (as computed using IM) differ significantly from time to time, or is there overlap?*
- *To what extent can the selection of seed sets be simplified, or substituted for other statistics- or feature-based heuristics?*
- *Are users who are considered influential in the pre-announcement rumor phase still considered influential in the after phase?*

2 PRELIMINARIES & RELATED WORK

Network diffusion processes have a long history of study in the social sciences. Some of the early applications include modeling the adoption of new products and technologies, modeling disease outbreaks and word-of-mouth events, and understanding human social dynamics. With the relatively recent advent of global social network platforms such as Facebook and Twitter, many new lines of research in computer science have emerged. In particular, the process of influence propagation in social networks has received a lot of attention.

In the age of Youtube, Facebook and Twitter, the appeal of viral marketing is to many the ultimate free lunch: pick some small number of people to "seed" your idea, get it to "go viral", and watch while it relentlessly spreads to reach millions, all on a shoestring budget. In [23], the authors propose a new model called "Big Seed Marketing" that combines the power of traditional advertising and the extra punch provided by viral propagation. This follows from years on marketing research, see for example [8] for an early study in computing the "value" of a user in a social network.

The problem of selecting a set of seed-users that will trigger a large cascade of activity was first introduced in [12]. In this seminal work the computational problem of maximizing the expected spread of a seed set is formulated as a discrete optimization problem, proven to be NP-hard, and a greedy algorithm with provable approximation guarantees is presented for a class of diffusion models. Importantly, they find that they are able to attain significantly higher values of expected spread by solving the Influence Maximization (IM) problem, as opposed to both random and centrality-based seed-selection methods. Since the greedy algorithm can be relatively slow to run on large networks, several methods have been presented to deal more efficiently with IM. These include the optimized algorithms CELF, MIA, TIM and IMM, [6], [5], [20], [19].

Algorithms for solving the the IM problem all depend on a directed, weighted social network as input. The weight of a directed edge between two users is supposed to represent the degree of influence

Course Project, December 2017, Vancouver
.

the one has over the other. Estimating this number is an interesting and general question in itself, and there is still plenty of room for more research here. The problem of estimating influence probabilities was first presented in [11], where the authors present static and time-dependent models for learning influence weights from a log of events.

The related question of identifying influential spreaders in complex networks was partly answered in [13]. In it, they find that there are circumstances where the best spreaders do not correspond to the best connected people or to the most central people (high betweenness centrality), rather, they are located within the core of the network as identified by the k-shell decomposition, and that when multiple spreaders are considered simultaneously, the distance between them becomes the crucial parameter that determines the extent of the spreading. More answers to the same question came in [1], where the authors investigate the attributes and relative influence of a large Twitter follower graph over a two month interval in 2009. They find that the largest cascades tend to be generated by users who have been influential in the past and who have a large number of followers. They also find that hashtags that were rated more interesting and/or elicited more positive feelings were more likely to spread. They also find that predictions of which particular user will generate large cascades are relatively unreliable. For a survey of computational models of influence propagation, see e.g. [3], or [9].

In [22], the authors present a community-based algorithm for mining top-k influential nodes in social networks. Their method is found to lead to a decrease in runtime, while not sacrificing much in terms of spread. This is perhaps not so surprising, given the above result that distance between seed-users becomes the crucial parameter that determines the extent of the spreading.

Combining the problems of influence maximization, influence estimation, and viral event dynamics, we are faced with the problem of real-time IM on dynamic social networks. This has been addressed in [16], and more recently in [21].

Some other relevant and recent research that we have taken inspiration from include studies of differences in mechanics of diffusion across topics [17], the dynamics of protest recruitment [10], sentiment reciprocality in reply networks [2] and prediction of social-link creation times [15].

The Higgs dataset was first presented in [7], where the authors present the dataset, explore the spatio-temporal properties of the data, and demonstrate a model for the information spreading in the social network during the event.

3 THE DATA SET

The dataset we are working with consists of a social network and an action log. The social network contains just over 450k unique Twitter users with just over 14.8M directed edges representing one user following another. The action log is a list of 563069 events of the form

```
user1, user2, event_type, time
```

where `user1` is the user who performs the action, `user2` is the user to whom the action is directed, `event_type` is one of either RT (retweet), MT (mention) or RE (reply), and `time` is the UNIX timestamp for the event. The dataset was scraped from the web using the Twitter API by the authors of [7]. The contents of the tweets and the identity of the tweeters is, unfortunately, not available. All we do know is that all of the tweets corresponding to events in the log contained one or more of the hashtags "LHC", "Higgs", "CERN", and "boson". It would have been very interesting to have access to not only the contents of the tweets, e.g. in order to perform some sort of semantic analysis, but also to have access to standalone tweets in the period, not just user to user interactions.

Of all the events in the log, roughly 62% of them are between two users where at least one of the users follows the other. This means that roughly 38% of the events are directed between "complete strangers". Out of all the events, 63% are retweets, 30% are mentions, and 7% are replies.

The authors of [7] identify four different periods into which the events can be divided:

Period I Before the 2nd July, there were some rumors about the discovery of a Higgs-like boson at the Tevatron accelerator

Period II On the 2nd July at 1PM GMT, scientists at the Tevatron accelerator in Fermilabs, Illinois, announced that they had discovered the Higgs boson with a 1 in 550 likelihood.

Period III After 2nd July and before 4th July there were many rumours about the Higgs boson discovery at the LHC.

Period IV The main event was the announcement on 4th July at 8AM GMT by the scientists at CERN. After 4th July, popular media covered the event.

The distribution of events in the four periods are described in Figure 1 and Table 2. As we can see from this distribution there is not much activity in the first periods, which is understandable considering the rumours were in an early phase. Therefore, in this paper we will divide the activity into two periods.

Before Rumour phase. This is the combination of period I, II and III above.

After Post-announcement. This is period IV above.

Table 1: Distribution of activities in the different periods defined in [16].

Period	#activities
I	4170
II	3720
III	171196
IV	383983

For completeness, the distribution of in- and out-degrees of each user (number of followers and following, respectively) can be seen in Figure 2, and the average values of the same quantities in Table 3

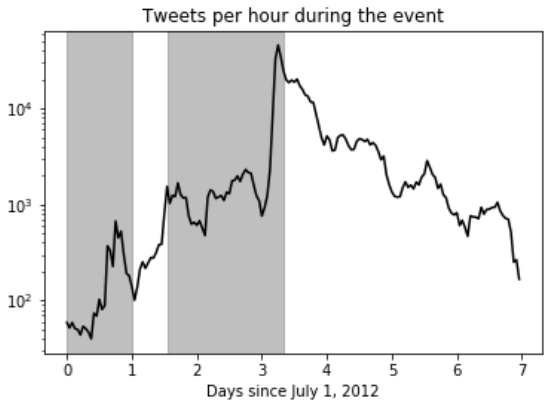

Figure 1: Activity levels through time during the event.

Table 2: Distribution of activities before and after the Higgs announcement.

Period	#activities
Before	179086
After	383983

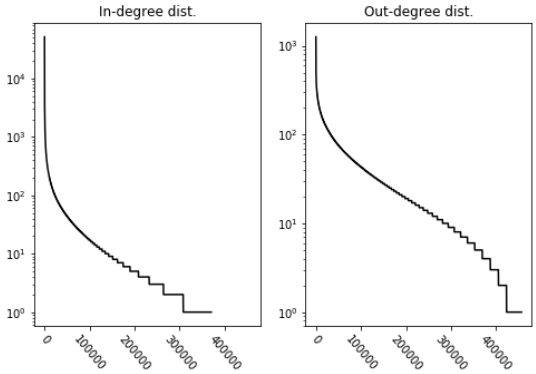

Figure 2: Distribution of in- and out- degree of users in the social network

Table 3: Average number of followers/following in the social network

	#followers	#following
mean	32.5	32.5
median	4.0	16.0

4 OUR APPROACH

4.1 Computing Influence Probabilities

In this paper, our goal is to experiment with IM on the Higgs dataset presented above in order to answer some of the questions stated in the introduction. In order to do this, we first combine the social network and the action log into a hybrid multi-digraph as follows: every node in the social network is first added to the (initially empty) hybrid network. Then, every link in the social network is added as a link in the hybrid network with the label "FR". Finally, all events in the action log are added as links in the hybrid network labelled as either "RT", "MT", or "RE". Later, we will also experiment with creating these graphs on a per-period level, so as to compare seeds over time.

The next step is to convert the hybrid multi-digraph into a weighted digraph, where, hopefully, the weights in the graph correspond roughly to the level of influence one user exerts over the other. This digraph can then be used to run IM experiments. Figure 3 shows the process schematically in a simple setting.

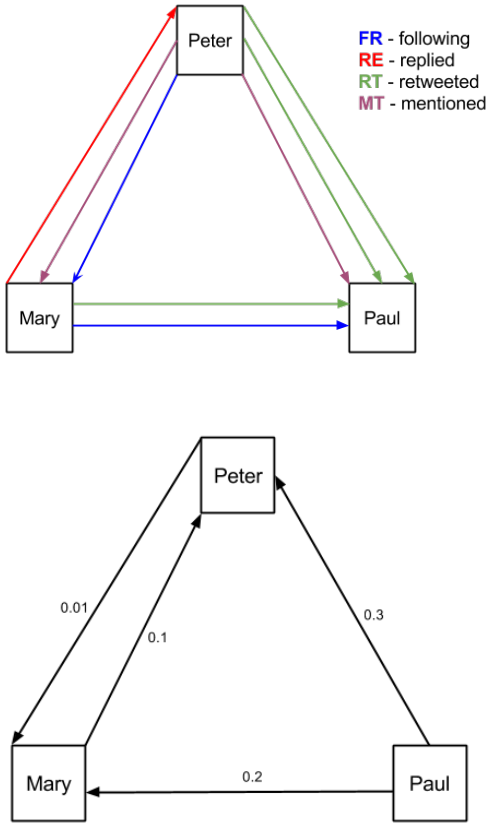

Figure 3: *Top:* original hybrid unweighted multi-digraph *Bottom:* influence-weighted digraph

As mentioned earlier, there are many viable ways of estimating the weights in the digraph. We propose a simple additive-weights method, where all parallel edges in the network are combined using predetermined weights per type, $\mathcal{W} = \{w_{\text{FR}}, w_{\text{RT}}, w_{\text{MT}}, w_{\text{RE}}\}$.

$$p_{uv} = \min\left\{1, h\left(\sum_{e=(v,u,t)} w_t\right)\right\}, \tag{1}$$

where the sum ranges over all edges e from v to u with type t. For our experiments, we "squeeze" the sums using a function h. The intuition behind this is that we want multiple actions between the same users to exhibit a diminishing returns property, i.e. the first time user A mentions user B counts more towards increasing the influence B has over A than the tenth time A mentions B. To accomplish this we use the simplest possible concave function that satisfies $h(0) = 0$ and $h(\infty) = 1$, which is $h(x) = 1 - e^{-x}$. It is also possible to both simplify (h is the identity function) or to complicate (say, by learning a parameterized function h_θ). We find our method to be simple, effective and intuitive, but it is hard to validate whether our formula actually captures the relative influence levels present in the user-set. We set the weights in $\mathcal{W}$ using a combination of trial and error and relative frequency of action-types in the event-log. An advantage of our weights-method is that we can simply choose to set $w_{\text{FR}} = 0$, resulting in a digraph which has roughly 15M fewer edges, drastically decreasing the runtime of the algorithms, as we will see later.

It is an interesting question whether the weights can be set in a more principled manner, ideally by learning from data, but this is really a whole new problem, which we avoid in this project and leave for later work. However the weights p_{uv} are computed, it is evident that they are paramount to the later success of our IM campaign. Given a set of weights, it is hard to validate their accuracy without expensive and intrusive user-surveys, or access to extensive event-logs. Furthermore, the weight's impact on both the diffusion process and the runtime of the algorithm is significant. For these reasons we stick to our simple formula for the remainder of this paper. The distribution of influence weights in the network when using type-weights $\{w_{\text{FR}} = 0.0, w_{\text{RT}} = 0.1, w_{\text{MT}} = 0.1, w_{\text{RE}} = 0.1\}$ can be seen in Figure 4.

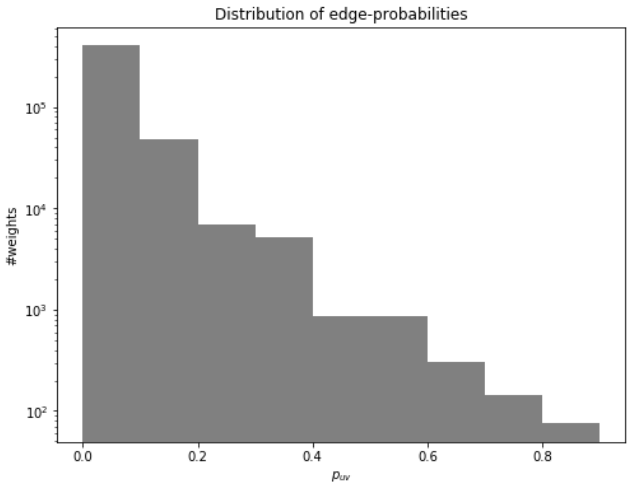

Figure 4: Distribution of edge-weights in the social/action hybrid network

4.2 Influence Maximization

Suppose we are an evil galactic empire, and due to universe-threatening circumstances we need to shut down research in particle physics on planet earth. But we are very far away from planet earth, and the only way to do this is to monitor the radio signals in space. Under these circumstances, one way to "shut down"[1]the particle physics community would be to infiltrate their twitter feed and manipulate them in some way.

We hope to make some progress towards an effective solution to this problem in this paper. However, we are not an evil galactic empire..

The aim of Influence Maximization (IM), is to "activate" a large number of users in a social network $G = (V, E, p_{uv})$, by activating a smaller number of seed-users, S. Given any discrete diffusion process on a social network and a seed set S denote the sets $S = A_0, A_1, \ldots, A_n, \ldots$ as the sets of active users at each timestep $0 \leq n \leq \infty$. Assuming it exists, denote the terminal state of the diffusion process by $\Phi(S) \subseteq V$. We define the *expected spread* of a seed set S as

$$\sigma(S) = \mathbb{E}\left[|\Phi(S)|\right], \tag{2}$$

and our goal is to maximize this function over all seed sets subject to a cost constraint. In our experiments, we employ the Independent Cascade (IC) diffusion model which works as follows: we start with an initial set of active nodes A_0, and the process unfolds in discrete steps according to the following randomized rule. When node u first becomes active in step t, it is given a single chance to activate each currently inactive neighbor v; it succeeds with a probability p_{uv} (if v has multiple newly activated neighbors, their attempts are sequenced in an arbitrary order.) If u succeeds, then v will become active in step $t + 1$. The process terminates when no more activations are possible.

The IC model is one of the conceptually simplest diffusion models, based on work in interacting particle systems. A particular useful property of the model is that the corresponding expected spread function σ is monotone and submodular. This means we can use a greedy algorithm to achieve a $(1-1/e)$-approximation to the optimal solution. However, since the greedy algorithm needs to compute the marginal gain of adding every node in the network to the seed set at each time step, the runtime of the basic greedy algorithm is simply prohibitive for networks of the size we are working with. For this reason we chose to use the Cost Effective Lazy Forwarding (CELF) algorithm, which has a significant speedup in runtime at no sacrifice in approximation guarantee. Even so, the runtime of running a complete IM trial is quite high on our dataset. Most of the time is spent in a subprocedure that runs Monte Carlo simulations to approximate the expected spread of a seed set.

[1]Stop, hinder, end, obliterate, assassinate, etc.

4.3 Heuristics

It has been the main focus of our project to experiment with heuristics, alternatives, and optimizations for running a full greedy optimization on a large network, without resorting to algorithms that sacrifice another ϵ on the approximation guarantee, such as MIA and TIM. In addition, we have tried to use our prior knowledge of the exact nature of the Higgs dataset to guide our experimentation. As simpler alternatives to CELF, we propose three different centrality-based heuristics for extracting top-k user-sets. We chose these particular heuristics partly based on earlier work (notably the importance of centrality as presented in [11]), and partly on our own intuition.

(1) **infl:** The *influence* of a user is simply the sum of the weights of all outgoing edges in the processed digraph. A user who has many followers, or is mentioned, replied to, or retweeted many times is considered influential.

(2) **outdeg:** Is the out-degree of a user in the processed digraph. A high value of outdeg means this user should have influence over a large number of users.

(3) **indeg:** Is the in-degree of a user in the processed digraph. A high value of indeg means this user has been very active, or follows a large number of users.

(4) **random:** This is the absolute minimum benchmark to compare all other methods with; k nodes are selected at random from the entire user-set.

Our next step is to use our coarse before/after separation to further reduce experimental cost. Suppose we have some method for detecting the presence of a rumor within a certain user community. Alternatively, we might *know* that a rumor is currently being disseminated across a network and we also know that that whatever is being rumored will be publicly announced at a certain time. In this case, we might consider performing some kind of costly preprocessing or computation steps early on in the rumor phase, in order to be ready to deploy a viral campaign as soon as the announcement is made. We use this idea in our experiments: we compute all the top-k sets in the before phase, and use these to compute spread in the after phase.

Many variations on this theme are possible. In addition to k-core and other centrality-based heuristics that have been considered earlier, we also considered the option of transforming every user into a feature-vector and simply scoring each user on a scale from 0 to 1 on how likely this user is to generate large cascades. Such feature-vectors could be constructed using simple graph- and log-based statistics such as in-degree, times mentioned, etc., but could also be combined with more personal or high-level features such as age, nationality, customer history, etc., all depending on what data is available. This approach could be realized using a supervised learning approach (say, a logistic regression model trained with labels obtained from actually running some IM-solution method).

Other heuristic methods for finding good seed-user sets might include outlier detection techniques or clustering. We followed up on this idea in our experiments, by turning the action log into a training set by simply counting the number of times every user was active in the four periods under study and running k-means on the PCA-reduced data matrix. We ran this algorithm in a two-level hierarchy, this gave us 6 clusters of users with different activity "fingerprints". The centroids of the clusters are plotted in Figure 5.

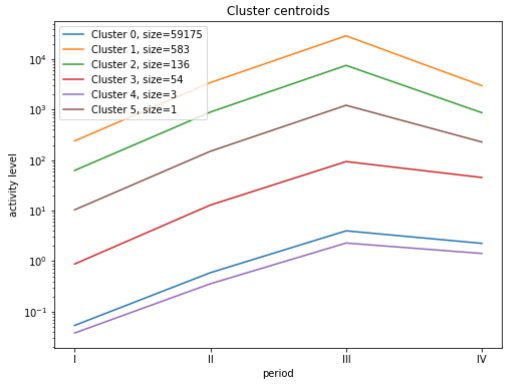

Figure 5: Centroids obtained after two levels of hierarchical k-means on the activity levels of users in each period.

We chose not to follow up on this idea in our main experiments, although there is potential for a useful method here, since we did in fact notice a correlation between users chosen by CELF and the other heuristics and users not present in cluster 0. However we suspect simpler more traditional methods of outlier detection might be more natural.

5 RESULTS

We now present the results of the experiments we have run. In all of the experiments, the main obstacle has been tractability. We made the (in hindsight) somewhat unpractical choice of using Python. The reasons for this include ease-of-implementation, earlier experience, efficient multi-processing support, and the significant practicality of the Python Notebook format. If we were to revisit this project, we would definitely migrate the most critical computations to C. That being said, we did our best at speeding up our experiments as much as possible given our means. This involved parallelizing the CELF/expected spread computations and executing the programs using the PyPy interpreter. All experiments were run on the UBC thetis server which has a Intel Xeon E5-2698 v3 with 16 cores (32 threads) @ 2.3GHz (boosted to 2.8GHz). Our graph needs about 7 GB of RAM to be stored and the 192GB on thetis makes it possible efficiently parallelize without trashing.

The first set of results are for the weighted digraph built using the method described in the beginning of section 4.1, and using $\{w_{\mathrm{FR}} = 0.0, w_{\mathrm{RT}} = 0.1, w_{\mathrm{MT}} = 0.1, w_{\mathrm{RE}} = 0.1\}$. Unfortunately, this choice is almost a necessary evil if we want to run all the desired experiments, as the number of following-edges (15M) dwarfs the number of action edges (563k). We also compute all spreads for both the before and the after phase. In this case the network reduces to

$300k$ nodes and $563k$ edges, rendering the problem barely tractable, in that it takes roughly 17 hours to compute. The results can be seen in Figure 6.

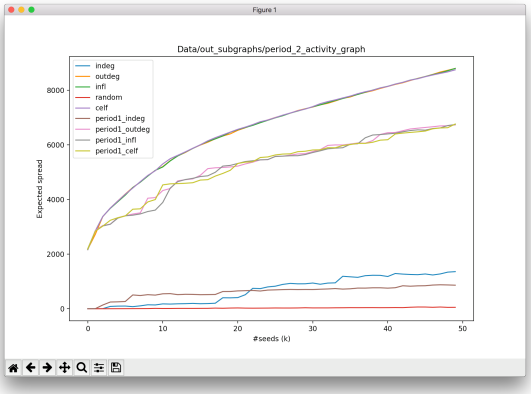

Figure 6: Placeholder

We see two main results in Figure 6. First, we see that the outdeg and infl heuristics perform just as well as CELF. This is somewhat surprising, and perhaps too good to be true. In future work, it is definitely worth looking into the possible reasons for this performance. In any case, from a viral marketing campaign's HQ, this is good news. Second, we find that the seeds selected by all methods in the before period give quite good spread in the after period. This is also good news, as this can be used by our campaign to get a head start in the post announcement phase.

We find that for both the outdeg and infl heuristics, there is a roughly 40% overlap in the selected seed users in the before phase and the after phase. Furthermore, within a given period, we find that there is almost 100% overlap between the infl and outdeg heuristics. It would be interesting to see how this number changes as the weights vary, and across other heuristics, as well as comparing with the seeds selected by CELF.

In the next set of results, we used weights $\{w_{\text{FR}} = 0.01, w_{\text{RT}} = 0.1, w_{\text{MT}} = 0.1, w_{\text{RE}} = 0.1\}$. This results in a much larger network with $450k$ nodes and $15M$ edges. For this network, running the CELF algorithm is tremendously intractable, so we only compute the spreads using our heuristics. Assuming the results above are somewhat generalizable, we expect the realized spreads using the outdeg and infl heuristics to be close to what the greedy solution might obtain. Also, we only compute the spreads in the after period, again relying on the above result that the good seed users in the before period also perform well in the after period. These results can be seen in Figure 7.

Kommentarer

Finally, we note that all our experiments agree on at least one thing: user 88 is by far the single most important user in the data set. This user appears in the top 2 of every set of seeds and it appears in a singleton cluster in the clustering experiments (cluster 5 in Figure 5). It would be interesting to see who this user ID corresponds to—Neil

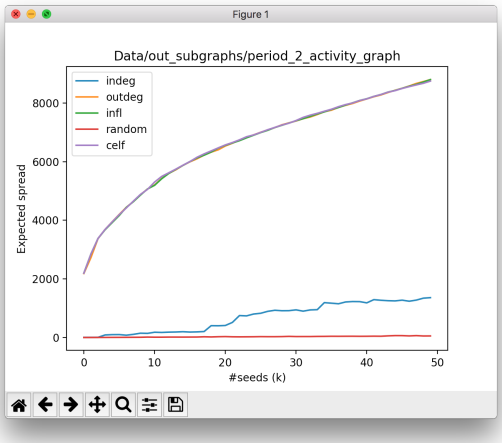

Figure 7: Placeholder

Tyson? the president of CERN? Brian Cox? Unfortunately we will never know.

6 CONCLUSION

In this paper we investigated the spread of a scientific rumor from a Influence Maximization standpoint. We experimented with different ways of preprocessing the data; building graphs, computing influence probabilities and reporting summary statistics. We implemented the CELF algorithm for seed selection and proposed a few different heuristics for faster seed selection. Our main results were that the heuristics performed comparably to CELF, and that influential users in the rumor phase tend to remain influential in the post-announcement phase.

6.1 Future Work

In this paper we have only just started learning about influence dynamics on social networks. We have touched upon many interesting, difficult, and open questions in this line of research.

Continuous time models,

$$x$$

Content of tweets - sentiment analysis.

Compute p_{uv} using unsupervised learning approach. Perform evaluative analysis.

REFERENCES

[1] Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM, 2011.

[2] Catherine A Bliss, Isabel M Kloumann, Kameron Decker Harris, Christopher M Danforth, and Peter Sheridan Dodds. Twitter reciprocal reply networks exhibit assortativity with respect to happiness. *Journal of Computational Science*, 3(5):388–397, 2012.

[3] Francesco Bonchi. Influence propagation in social networks: A data mining perspective. *IEEE Intelligent Informatics Bulletin*, 12(1):8–16, 2011.

[4] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[5] Wei Chen, Chi Wang, and Yajun Wang. 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*, pages 1029–1038. ACM, 2010.

[6] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[7] Manlio De Domenico, Antonio Lima, Paul Mougel, and Mirco Musolesi. The anatomy of a scientific rumor. *Scientific reports*, 3:2980, 2013.

[8] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[9] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):21, 2012.

[10] Sandra González-Bailón, Javier Borge-Holthoefer, Alejandro Rivero, and Yamir Moreno. The dynamics of protest recruitment through an online network. *Scientific reports*, 1:197, 2011.

[11] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[12] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[13] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888–893, 2010.

[14] Ling-ling Ma, Chuang Ma, Hai-Feng Zhang, and Bing-Hong Wang. Identifying influential spreaders in complex networks based on gravity formula. *Physica A: Statistical Mechanics and its Applications*, 451:205–212, 2016.

[15] Brendan Meeder, Brian Karrer, Amin Sayedi, R Ravi, Christian Borgs, and Jennifer Chayes. We know who you followed last summer: inferring social link creation times in twitter. In *Proceedings of the 20th international conference on World wide web*, pages 517–526. ACM, 2011.

[16] Manuel Gomez Rodriguez and Bernhard Schölkopf. Influence maximization in continuous time diffusion networks. *arXiv preprint arXiv:1205.1682*, 2012.

[17] Daniel M Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 695–704. ACM, 2011.

[18] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-based intelligent information and engineering systems*, pages 67–75. Springer, 2008.

[19] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554. ACM, 2015.

[20] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 75–86. ACM, 2014.

[21] Yanhao Wang, Qi Fan, Yuchen Li, and Kian-Lee Tan. Real-time influence maximization on dynamic social streams. *Proceedings of the VLDB Endowment*, 10(7):805–816, 2017.

[22] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1039–1048. ACM, 2010.

[23] Duncan J Watts, Jonah Peretti, and Michael Frumin. *Viral marketing for the real world*. Harvard Business School Pub., 2007.