Influence Maximization on Multi-Phased Multi-Layered Network

Yue Zhang yzhang91@stanford.edu
Department of Electrical Engineering

Shutong Zhang zhangst@stanford.edu
Department of Computer Science

Le Wang lewang2@stanford.edu
Department of Electrical Engineering, Stanford University

Abstract

With the development of social network, people start to pay attention to its internal information diffusion and propagation process. However, most state-of-the-art works mainly analyzed single-phased and single-layered network. In our project, we study the information spread on multi-layered network. We focus on one specific event - the information of the discovery of a new particle spreading in Twitter. The participated users and their corresponding actions have formed a multi-layered, multi-phased network. We explored different methods to aggregate multi-layered networks. Using different influence maximization algorithms and different methods to assign edge probability on the aggregated network in different phases, we are able to analyze the important nodes and layers in the given dataset. The experiment results show the effectiveness of our aggregation strategy on our multi-phased and multi-layered network.

1. Introduction

Information diffusion and propagation in social networks have been widely studied in various domains, including marketing strategies, promotion of new products, technology innovation diffusion and rumor spreading. Research has focused on two factors in the information diffusion process:

1. Models of the influence process
2. Algorithms finding the set of most influential users and their maximal influence

Finding the subset of nodes in such networks to trigger future cascade of information adoptions is posed as influence maximization problem. The optimization problem of selecting nodes with maximized influence is NP hard. To analyze and solve this problem, various prior work has conducted in mathematical modeling of the information propagation process, and practical algorithms for discovering such influential sets in efficient ways.

In our project, we target on analyzing this problem on a real-life multi-phased multi-layered network, with an emphasis on the multi-layered aspect overlooked by previous approaches. We evaluate the effectiveness of existing models and algorithms, and extend their approaches on real-world multi-layered network. Furthermore, we investigate the multi-layered network structure to provide insights on how different layers could contribute in the influence maximization process.

2. Literature Review

Influence maximization problem is originally proposed in [5] as finding the set of individuals who could trigger a large cascade of further adoptions of a product in the network. At early stage, nodes are selected based on their degree and distance centrality. Later, [11] modeled the information diffusion process using Linear Threshold (LT) model and Independent Cascade (IC) model. They also proposed the first provable approximation guarantee for efficient algorithms to solve the above problem, which is a NP-hard optimization problem. By introducing the concept of submodular function, the authors proved the performance-guaranteed algorithms. Their experiment on real-world citation dataset have shown that LT Model and IC Model outperformed other widely used node selection heuristics based on centrality.

Traditionally, based on LT and IC models, most influential nodes could be selected by Monte Carlo simulations, which is slow and computational expensive. Thus, various algorithms are proposed to perform node selection task in an efficient, effective, and scalable manner. [1] developed an efficient algorithm for influence maximization which improves the original greedy algorithm in [11] and reduces in running time. Their method also involves a new degree discount heuristics to improve influence spread. [13] demonstrated the property of 'submodularity' in outbreak detection objective. Based on this finding, they exploited submodularity to develop CELF algorithm that scales well and achieved near optimal selection with efficient execu-

tion time. [8] optimized CELF and empirically showed it is 35-55% faster than CELF. [2] proposed another scalable algorithm on directed acyclic graphs (DAGs) for LT model, based on their discovery that computing influence in DAGs can be done in time linear to the size of the graphs.[9] applied vertex cover optimization and look ahead optimization on greedy algorithm [11] of LT and developed SimPath algorithm. They claimed that this algorithm consistently outperforms the state of the art in running time, memory consumption and the quality of the chosen seed set.

Since LT and IC are two most widely used models for capturing the dynamics of information diffusion process [11], they often serve as an assumption for traditional influence maximization analysis. However, neighbors of influential nodes may be more engaged to the topic, and have the potential to influence the influential nodes. Also, the network structure and influence probability may not known. [15] applied adaptive seeding, a two-stage stochastic optimization model designed to leverage the potential in neighboring nodes. [10] proposed a scalable methods for adaptive seeding. [6] proposed methods to learn influence probabilities using expectation maximization methods in [6], and [7] utilized the learned probability to perform node selection for influence maximization based on Credit Distribution model. [12] proposes an online method of influence maximization with feedback to update the diffusion influence information.

In terms of influence maximization in multi-layer network, not much previous work was found. Rather, research has been conducted to understand the layered structure, and the diffusion process in such network. [14] investigates about the information cascades on multi-layered network. [3] proposed a contact-based information spreading model, and showed that the critical layer can be identified as the layer whose contact probability matrix has the largest eigenvalue.

Different from the previous work, our paper focuses on influence maximization of the real-life multi-layered network. The major contribution of our work has three folds:

1. Extending the existing influence maximization algorithm to multi-layered network.

2. Providing insights on information aggregation among layers.

3. Analyzing and comparing the importance of layers on twitter social network.

3. Problem Formulation

Given a multi-phased multi-layered network graph $G := (V, E, P, L)$ with $|V|$ vertices, $|E|$ edges, $|P| = 3$ phases and $|L| = 3$ layers. For certain phase p, we construct a aggregated multi-layered network $G_p := (V_p, E_p)$ with $|V_p|$ nodes and $|E_p|$ edges. V_p denotes the nodes that are involved in phase p and E_p denotes directed edges which represent the actions in phase p.

In each phase p, we have three seperate networks - the mention network G_{ME_p}, the retweet network G_{RT_p} and the reply network G_{RE_p}. These three networks are considered to be 3 different layers . Thus, we use the aggregation function $\mathcal{C}_p$ to construct the multi-layered network graph G_p. The aggregation function $\mathcal{C}_p$ can be written as:

$$\mathcal{C}_p(G_{ME_p}, G_{RT_p}, G_{RE_p}) = G_p$$

In the constructed multi-layered network graph G_p, our objective is to find a initial set S_p to maximization the influence in G_p. The initial set S_p for different phase p is:

$$S_p = arg\ max_{S_p \subseteq V_p}(\sigma(S_p))$$

where $\sigma(S)$ is the influence of S_p.

4. Data Description

4.1. Dataset

The dataset we will use is the same one used in [4]. This dataset was built by monitoring the spreading processes on Twitter before, during and after the announcement of the discovery of a new particle with the features of the elusive Higgs boson on 4th July 2012.

Only tweets about the dicovery of this new particle are considered in the dataset, which means all the actions (mention, retweet and reply) are relevant to the spreading of this event on twitter.

4.2. Data Preprocessing

The authors in [4] split the whole dataset into four different phases. In order to gain some insight on the information flow before, during and after the announcement of the discovery of a new particle, we on the other hand use three different phases in our model.

Phase I: On 2nd July at 1 PM GMT, scientists from CDF and D0 experiments presented results indicating expected mass of the Higgs particle;

Phase II: After 2nd July and before the announcement on 4th of July there were many rumors on twitter;

Phase III: The announcement of the new particle on 4th July at 8 AM GMT. After 4th July, popular media covered the event.

After splitting the dataset into different phases, we are able to analyze different characteristics of these three phases and find the optimal function $\mathcal{C}_p$ for each phase p.

4.3. Data Statistics

The whole dataset contains multiple layers of network:

1. retweet network
2. reply network (to existing tweets)
3. mention network

Table 1 includes some basic information about the dataset.

Table 1. Dataset statistics of multiple layers

	Social Networks	Retweet	Reply	Mention
Nodes	456626	256491	38918	116408
Edges	14855842	328132	32523	150818

The number of nodes and edges of Phase I, Phase II and Phase III are listed below.

Table 2. Data statistics for seperate phases

Phase	Number of nodes	Number of edges
Phase I	6061	7712
Phase II	87457	169526
Phase III	247440	378243

The log-scale of number and cummulative number of messages sent per hour is shown in **Figure 1**.

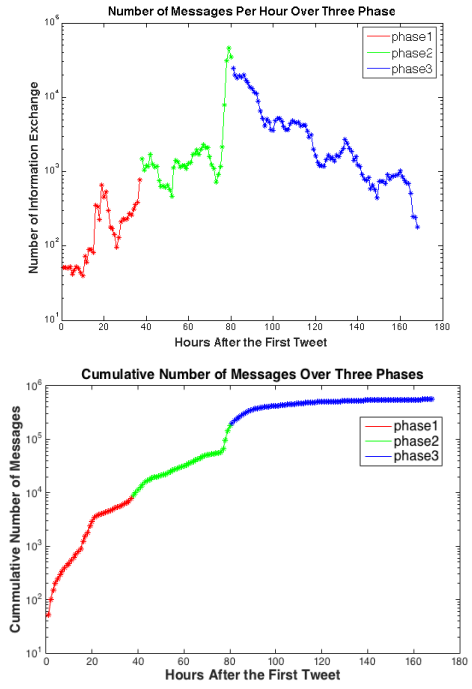


Figure 1. number and cummulative number of messages sent per hour

From **Figure 1**, we can see several peaks of information exchange. Although the official announcement happened on 4th July at 8AM GMT, 4th July 3AM GMT is actually the moment that the highest information exchanges happened. Moreover, from the graph, we can clearly find the differences between the trend in different phases.

In the following part, we will use Phase I as an example to show the properties of each layer.

The data statistics of different layers in Phase I are listed in **Table 3**.

Table 3. Data statistics for different layer

Layer	Nodes	Edges	Diameter	Average Path length
Retweet	4537	4040	5	1.477
Mention	3134	2966	6	1.666
Reply	896	563	4	1.118

The reply network is very sparse and less informative with less than 1000 edges. The network only consists of small discrete components. Most of these discrete components only have 2 or 3 nodes. On the other hand, in the mention and retweet network, several influential nodes exist in both the mention network and the retweet network, which means that most of the users in the network knows the information from these influential nodes. With a proper aggregation function $\mathcal{C}_p$, we should be able to find the nodes that have the maximized influence in the aggregated graph.

5. Methods

5.1. Diffusion Models

In this work, we use the traditional LT and IC models to simulate the information propagation process in the social network. Their definition are described below:

1. **Linear Threshold Model**
   For a node v, we have an monotone function f based on v's neighbors that are already active.
   Thus,
   $$\begin{cases} f(v) > \theta_v & v \ is \ active \\ f(v) \leq \theta_v & v \ is \ inactive \end{cases}$$
   Start from a set S, using the above inequalities, we can determine whether a node v is active or not, thus compute $\sigma(S)$ - how many nodes can be activated from the initial S.

2. **Independent Cascade Model**
   Each node v has a probability $p_v(u, T)$ to activate its neighbor u, where T is u's neighbors who has already tried to activate u.
   Note that for each node pair $(v, u) \in G$, there is only one chance for v to activate u. Based on the previous process, we can compute $\sigma(S)$.

5.2. Information Maximization Algorithms

We apply SimPath [9] and CELF++ [8] for initial set selection to achieve maximized influence. We define the influence as the cumulative number of activated nodes in the information process starting with the initial set.

1. **SimPath**
   SimPath algorithm is based on LT model. The spread of a set S is the sum of the spread of each node $u \in S$ on subgraphs induced by $V - S + u$. That is,

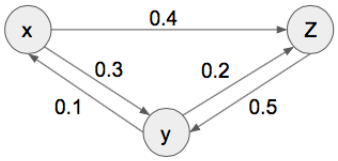
$$\sigma(S) = \sum_{u \in S} \sigma^{V-S+u}(u)$$

   Different from the general LT model, the SimPath model compute the spread of a set S by the sum of the spread of each node in S and the spread of each node can be computed by summing the weights (i.e., probabilities) of all simple paths originating from it. That is,

$$r_u = \sum_{v \in G} r_{u,v}$$

   And

$$r_{u,v} = \sum_{P \in \mathcal{P}(u,v)} Pr[P]$$



   Where $Pr[P]$ is the probability of a path P being live and $\mathcal{P}(u, v)$ is the set of all paths from node u to v. Take the above network as an example, if the initial set is $S = \{x, y\}$, we have,

$$\sigma(S) = r_x + r_y$$

   Where

$$r_x = r_{x,x} + r_{x,y} + r_{x,z}$$

$$r_{x,z} = 0.3 \cdot 0.2 + 0.4 = 0.46$$

2. **CELF++**
   CELF used the greedy algorithm with the heuristic that each node has unit cost. Start with empty placement $\mathcal{A}_0 = \varnothing$, and iteratively, in step k, adds the location s_k which maximizes the marginal gain

$$s_k = \text{argmax}_{s \in \mathcal{V} \setminus \mathcal{A}_{k-1}} R(\mathcal{A}_{k-1} \cup \{s\}) - R(\mathcal{A}_{k-1})$$

   The algorithm stops, once it has selected B elements.

   CELF++ optimizes CELF by exploiting submodularity. The key improvement in CELF++ is that if the node $u.prev$ best is picked as a seed in the current iteration, we don't need to recompute the marginal gain of u w.r.t $(S \cup \{prev\_best\})$ in the next iteration. Thus, CELF++ is more efficient compared to CELF.

5.3. Edge Probability Assignment

In LT and IC model, we have to assign each edge from node i to node j a influential score p_{ij}, which measures how likely information can flow from node i to node j. In our project, we use two different methods to assign edge probability.

1. **Beta distribution**
   We use Beta distribution $B(\alpha_{ij}, \beta_{ij})$ to generate influence score p_{ij} for each edge from node i to node j. The hyperparameter α_{ij} is set to be a fixed value and β is set to be propotional to the in-degree of node j.

$$\begin{aligned} \alpha_{ij} &= 19 \\ \beta_{ij} &\propto InDegree(j) \end{aligned}$$

2. **Online Probability Learning**

Algorithm 1 Online Probability Learning

procedure COMPUTE PROBABILITY
  $z \leftarrow$ Bernoulli*(e)*
  if $z = 0$ **then**
    $p_{ij} = \frac{1}{\alpha_{ij}+\beta_{ij}}(\alpha_{ij} + \sqrt{\frac{\alpha_{ij}+\beta_{ij}}{\alpha_{ij}+\beta_{ij}+1}})$
    $S \leftarrow IM(G, k, n)$
  else
    $p_{ij} = \frac{\alpha_{ij}}{\alpha_{ij}+\beta_{ij}}$
    $S \leftarrow IM(G, k)$
  $S \leftarrow IM(G, k)$
procedure UPDATE HYPERPARAMETERS α, β
  $\alpha \leftarrow 1$
  Using binary search to compute β for $f(\beta) = 0$, where
  $f(\beta) = \sum_{m_{ij} \neq 0} \frac{1}{\beta + m_{ij}} - \sum_{h_{ij} \neq 0} \frac{1}{\alpha + h_{ij}}$
  $p_{ij} \sim B(\alpha + h_{ij}, \beta + m_{ij})$

Using the idea in [1], the given network can be seen as a uncertain influence graph with uncertain influence score p_{ij} between each pair of nodes. Then, we adopt the Explore-Exploit strategy to update the influence score p_{ij} in the the given network, which can select seed nodes using either the current influence probability estimation or the confidence bound on the estimation. Then any existing Influence Maximization Algorithms(IM) can be used to determine the initial set S and we use the simulation process to update the

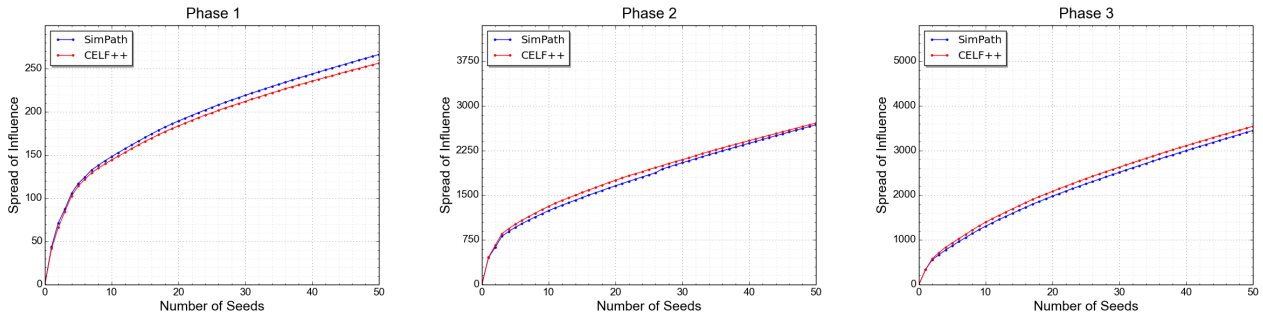


Figure 2. Influence spread vs number of seeds by random probabilities. (From left to right: phase I, II, III)

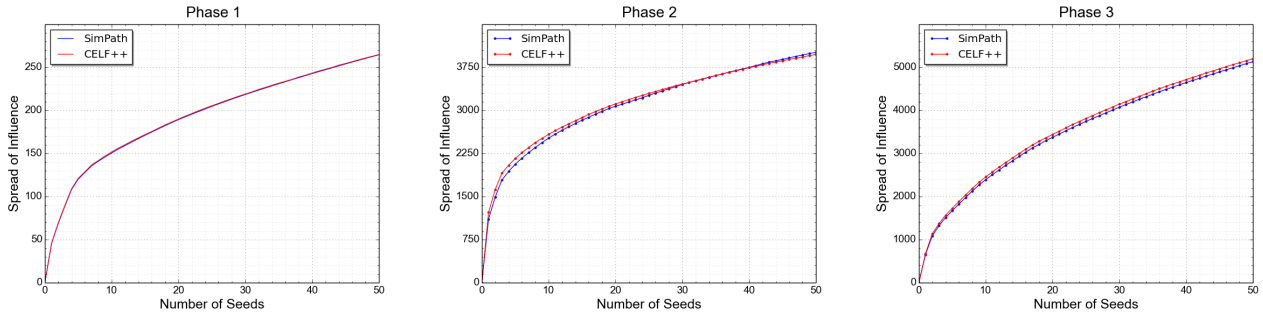


Figure 3. Influence spread vs number of seeds by heuristic probabilities. (From left to right: phase I, II, III)

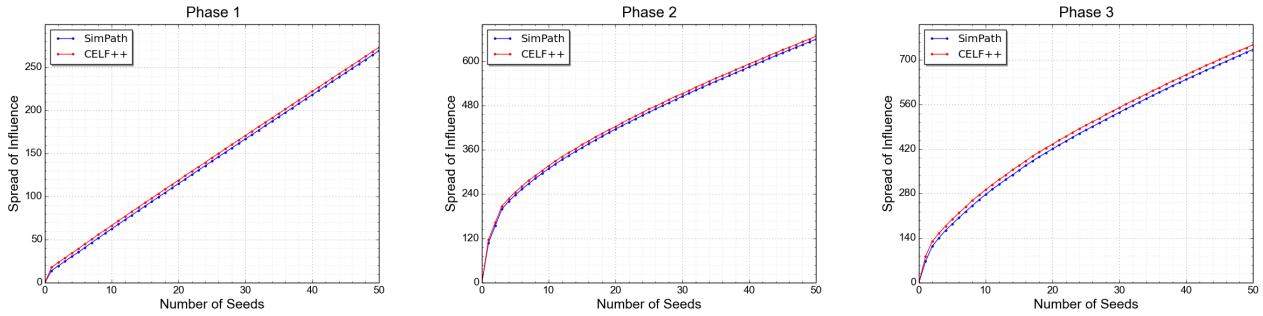


Figure 4. Influence spread vs number of seeds by Online Probability Learning. (From left to right: phase I, II, III)

hyperparameters α, β in beta distribution. The pseudo-code are listed at **Algorithm 1**.

5.4. Multi-layered Network Aggregation

We focus on experimenting with different layer aggregation methods. Given the twitter dataset with retweet, mention and reply layers of information spreading trace, we experimented with various ways of integrating those methods. So far, we mainly explored possible ways to assign edge probabilities in the aggregated network.

The baseline is to assign random edge probability to all layers, and integrate all layers into a single network by overlapping them. In other words, we use this as a unified network where there is no difference between different layers.

For the second probability assigning strategy, we first assign the probability for each edge by applying Beta distribution discussed above. Then we scale the edge probability based on the weight of three types of communications in the network. For example, the ratio of total number of edges in mention, retweet and reply is 6:3:1, then we scale them to make the mean of mention edge probability, the mean of retweet edge probability and the mean of reply edge probability also as the ratio of 6:3:1. To fit in the LT model, we also assign the probability to make the largest in-weight in the network smaller than 1. This assignment is based on the assumption that the larger number of communication

will result in larger influence in the information flow in the network. To better simulate the uncertainty in the network, among the same type of edge, we assign their probability based on a normal distribution with expectation calculated as mentioned above.

6. Experiments and Evaluation

6.1. Experiments

We first conducted experiments to compare, for each phase, the performance of SimPath and CELF++ on both networks with edge probabilities assigned randomly or by our heuristics, in terms of influence spread.

Second, while running CELF++, we removed one type of edges (i.e. one type of MT: mention, RE: reply and RT: retweet) each at a time, and compare the resulting influence spread during each phase. This experiment could illustrate to what extent the removal of each type of edges would deteriorate the influence spread.

We also targeted the top influencer in each phase and analyzed their actions in that phase. More specifically, we analyzed the type and time of their actions.

6.2. Experiment Results

Table 4. Top Influencers Selected by SimPath and CELF++ using Random Probabilities

	Phase I	Phase II			Phase III		
SIMPATH	250519	88	677	1988	88	14454	677
CELF++	250519	88	1988	677	88	14454	677

Table 5. Top Influencers Selected by SimPath and CELF++ using Heuristic Probabilities

	Phase I	Phase II			Phase III		
SIMPATH	250519	88	1988	677	88	14454	677
CELF++	250519	88	1988	677	88	14454	677

Table 6. Top Influencers Selected by SimPath and CELF++ using Online Probability Learning

	Phase I	Phase II			Phase III		
SIMPATH	250519	88	1988	677	88	14454	677
CELF++	250519	88	1988	677	88	14454	677

Figure 2, **Figure 3** and **Figure 4** show the performance of SimPath and CELF++ on different networks with edge probabilities assigned randomly, by our beta distribution heuristics and by online probability learning, during each phase I, II and III. For all networks CELF++ outperformed SimPath, during all phases. Comparing the influence spread in different network during the same phase (e.g. the right plot in Figure 2, Figure 3 and Figure 4), we could conclude our heuristics to assign edge probabilities are more reasonable than just randomly assign probabilities, since the influence spread faster. The spread of influence scores are in Figure 4 is not as good as those in Figure 2 and 3. However,

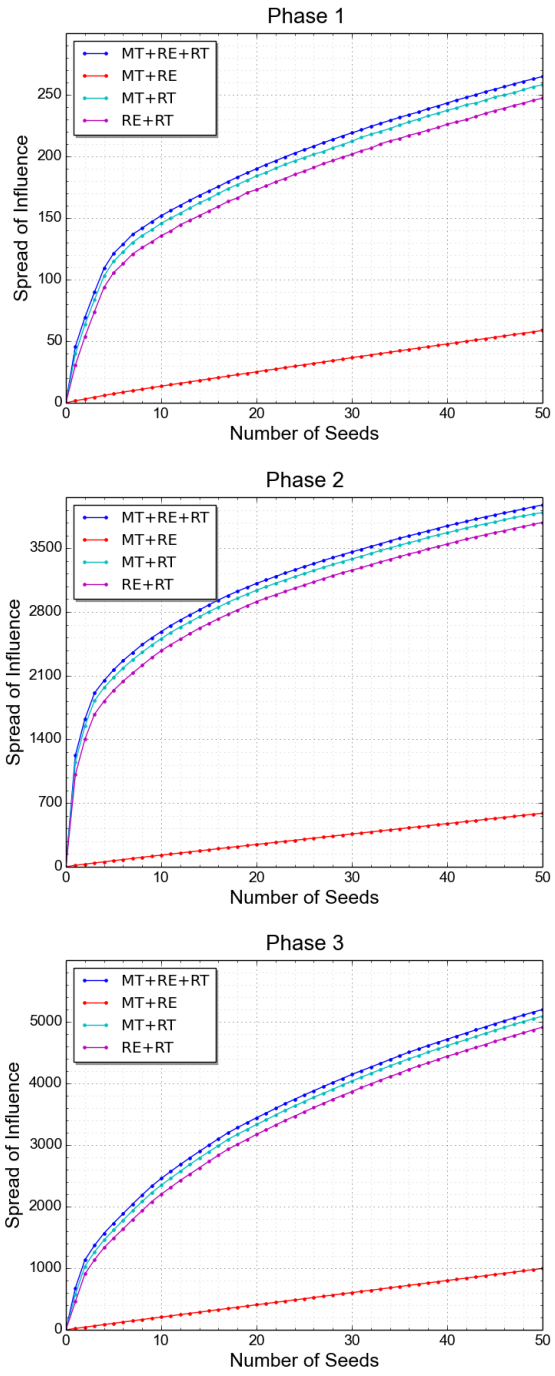


Figure 5. Influence spread vs number of seeds by using different combination of layers. (Where MT: Mention, RE: Reply, RT: Retweet. From top to bottom: phase I, II, III)

this score is not comparable between algorithms, as we used different methods to normalize them.

Figure 5 shows the effects of removing one type of action to the influence spread. In all three phases, the removal

of RT: retweet edges will cause the most significant effects on the influence spread, Which is consistent to our intuitive mentioned in section 5.3. **Figure 5** also illustrates the extent of this effects, the removal of RT would dramatically deteriorate the influence spread, compared with the removal of the other two types of actions, especially during phase II and III.

Table 7. Top users' actions in different phases

	Phase I	Phase II			Phase III		
ID	250519	88	1988	677	88	14454	677
Total	540	21059	6005	6005	14496	5493	5493
Mention	29	8841	1132	1132	6443	306	306
Reply	3	658	85	85	655	36	36
Retweet	505	11549	4784	4784	7393	5151	5151

Table 4, **Table 5** and **Table 6** show the top influencers selected by SimPath and CELF++ in different networks during each phase (only one top influencer is included in phase I). **Table 7** shows the actions the top influencers took in each phases. In each phase, the top influencers not just participate a lot but more importantly, their tweets are retweeted thousands of times and they are frequently mentioned by other users. This is consistent with our analysis above that the retweets play a more significant role in the spreading of influence, the users with top influence are usually those whose tweets are retweeted more.

Through dissecting the actions of those users, we also got some interesting results: most influential user '88' in phase II and III took his/her action very early in that phase. In both phase, '88' tweeted almost at the beginning of that phase and those tweets are repeatedly retweeted during that phase. However, the most influential user '250519' in phase I did NOT take his/her actions very soon, his/her first tweet appeared almost in the middle of phase I. But that retweet still got massively retweeted through phase I. So we could conclude the most influential nodes are not necessarily those taking their actions very early.

7. Conclusion and Future Work

In our project, we focuse on a specific dataset - the spread of discovery of a particle on Twitter, which is a multi-phased, multi-layered network. We have explored some aggregation model for multi-layered network, under different assumptions. Using Information Maximization algorithms we are able to determine the most influential nodes in multi-layered network. The exprimental results shows that our strategy of assigning edge probabilities outperforms state-of-arts methods.

In the future, we would like to adopt online edge probability learning strategies to more general network, and explore the change of edge probabilities to measure the effectiveness of the online learning methods.

References

[1] W. Chen, Y. Wang, and S. 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.

[2] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 88–97. IEEE, 2010.

[3] E. Cozzo, R. A. Banos, S. Meloni, and Y. Moreno. Contact-based social contagion in multiplex networks. *Physical Review E*, 88(5):050801, 2013.

[4] M. De Domenico, A. Lima, P. Mougel, and M. Musolesi. The anatomy of a scientific rumor. *Scientific reports*, 3, 2013.

[5] P. Domingos and M. 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.

[6] A. Goyal, F. Bonchi, and L. V. 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.

[7] A. Goyal, F. Bonchi, and L. V. Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011.

[8] A. Goyal, W. Lu, and L. V. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.

[9] A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 211–220. IEEE, 2011.

[10] T. Horel and Y. Singer. Scalable methods for adaptively seeding a social network. In *Proceedings of the 24th International Conference on World Wide Web*, pages 441–451. International World Wide Web Conferences Steering Committee, 2015.

[11] D. Kempe, J. Kleinberg, and É. 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.

[12] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart. Online influence maximization (extended version). *arXiv preprint arXiv:1506.01188*, 2015.

[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[14] Z. Li and Y. Jiang. Cross-layers cascade in multiplex networks. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 269–276. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[15] L. Seeman and Y. Singer. Adaptive seeding in social networks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 459–468. IEEE, 2013.