Probabilistic Influence Model on Social Network

Julian Gao, Wei-Ting Hsu, Qiwen Fu
Stanford University
julianyg, hsuwt, qiwenfu@stanford.edu

Abstract

Social networks are the one of the most common and complicated type of networks in the real world. One important research focus on social networks is how users influence each other. The message passing between individuals and information viral spreading inside network are interesting topics that draw attention from advertisement entities. In this work, we propose a probabilistic method to estimate the influential model for the network from a following relation graph and a set of timestamped retweet message ids. We construct a model that learns the distribution of probabilities of information transmission between nodes, that generates accurate approximation and requires only minimal information. To the best of our knowledge, this is the first work that is able to generate distributional influence model on social networks.

1 Background

Massive Social Network Service (SNS) is a rich resource for network research, and should be utilized to understand human social interactions. By studying and analyzing the Sina Weibo network, we can dig in deeper and analyze the influence set within the model, especially of certain group amongst the general public such as the affect of celebrity and marketing group on normal users. This information can bring insights to certain enterprise or individuals running on-line marketing campaign or propagation.

1.1 Sina Weibo

Sina Weibo is the largest micro-blogging platform in Asia, specifically in China. The number of active users have recently surpassed that of Twitter in early 2017 with over 360 millions active users monthly as of today. While the number of twitter users have plateaued in the past few years, the popularity and trend of growing user size of Sina Weibo have not yet seemed to stop. Despite the similarity between Weibo and Twitter, there are a few key difference amongst the two: Weibo is more media-rich, provides a better platform for celebrities and verified users, and display social trends in a more user friendly way. As a result, Weibo is deemed to be the best viral marketing tool in China. In this project, we aim to analyze the influencing behaviors of Weibo network by proposing a probabilistic graph model based on users preferences.

There is social significance behind this problem that motivates us. Due to language constraint, there have been much lesser research conducted on Weibo data than on Twitter, and we intend to utilize our knowledge in language and technology to come up with a influencing model in Asian network. With our probabilistic influential model, one can determine the optimal influence set based on our estimated probabilities, and identify the most influential account in the network. Weibo is more monetized than Twitter by having many marketing accounts and closely related to other financial services. The results of our research would not only benefit from a social research standpoint, but also bring advantages to enterprises or anyone who wants to build online business through micro-blogging platform.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

2 Related Work

2.1 Analysis on Sina Weibo Network

An Analysis of Microblogging Behavior on Sina Weibo: Personality, Network Size and Demographics [1] (Wang et al., 2013) This paper analyzes the micro-blogging behavior of Sina Weibo users from two perspectives, demographics and personality. It uses huge amount of user data, with abnormals, special users screened out, to perform a supervised correlation measurement. It quantized the blogging behavior by the number of followers, number of voting, and number of @. The personality and demographics information are gathered by online surveys.

This paper leverages the idea of supervision with graph analysis. The data collection process is closely related with course contents. The graph is generated using BFS to expand over the network. The relation of blogger follower and followee forms a directed graph, where average daily number of microbloggings and updates are attributes of the nodes. The abnormal behaviors are filtered out in the pre-processing, so the analysis assumes ordinary behaviors only.

To the best of our knowledge, this paper is the closest work from our idea of analyzing Weibo user behaviors. However, we are going to perform the full-range user behavior in an unsupervised way, analyze base on information extracted from graph data only.

Despite of the great amount of work of collecting data, and idea of user behaviors, this paper is done in bad quality and does not have a handful of either academic or application value. The analysis requires user surveys, which is a very unreliable approach per se; by correlating the behavior, personality, and demographics for users with a valid survey reported, it does not make use of the rest of collected Weibo data at all. The analyses are also completely depending on static data that only report average indicators, making no use of temporal information. The authors are especially missing the real world implications of conducted study. This work could have been more useful if the correlation between personality and behavior is further studied, to produce some insights into cyber-security and user study.

Social Network Analysis of Information Diffusion on Sina Weibo Micro-blog System [2] (Li et al., 2015) This paper focuses on two social networks retrieved from the most popular micro-blog networking service, Sina Weibo, which is known as the twitter in China. They study the "post-repost" relationship among bloggers. Instead of building the network directly from the "followee-follower" relationship, they first collect blogs of a specific topic in a specific range of time, represent the blogs instead of the bloggers as the nodes and represent "post-repost" relationship between two blogs. Then, they retrieve the network among the bloggers from the "post-repost" network by representing bloggers as nodes and adding links between two bloggers if one reposts the other's blog. They choose two topics of 2013 and built two networks, one is the iPhone 5 and the other is iPad Air, and they collect blogs posted in one week after the two products were officially launched respectively.

The paper analyzes the statistical characteristics of the two networks, including the average clustering coefficient, average path length and the average degree of the network. It also presents the log-log plot of the distribution of the degrees of the nodes. It compares the two networks and reach the conclusion that though the iPhone 5 network is larger, two networks are similar in terms of their structural characteristics and the online behavior patterns of the users.

The paper follows some of the basic techniques for network analysis introduced by CS224W. They tried to analyze the network from a different point of view by representing the network by the "post-repost" relationship among users instead of the direct "followee-follower" relationship and analyzing the network of a specific topic, which give us some insights on social network analysis.

However, the paper ignores several important problems. Firstly, when it builds the networks, it filters out the blogs posted by the governments, companies and other news organization accounts. Actually many users reposts blogs from these authoritative accounts or reposts after their followees instead of directly from other users. Filtering out these posts will throw away a large amount of network information. Secondly, the authors choose two similar topics, both of which are popular electronic products from Apple. It is intuitive that similar topics will lead to similar networks since the audiences who pay attention to these topics have a large overlap. Thirdly, they add a single link between two bloggers if one reposts the other's blog ignoring how many times of reposts. Using

multiple links between two bloggers might be a better representation. Besides, the paper does not elaborate on the different components of the networks.

2.2 Influence Model on Social Network

Sequential Influence Models in Social Networks [3] (Cosley et al., 2010) This paper attempts to model influence in social network with a probabilistic framework. The authors proposes that an un-adopted users is more likely to spread a particular behavior when more of its immediate neighbors have already adopted, and the probability is proportional to the fraction of users that are exposed to the same number of adopted neighbors in the entire network. Their investigation of such probability measurement is based on temporal data: as users adopt influence, the number of adopted neighbors changes for every user, and the fraction of users with particular number of adopted neighbors also changes. The author proposes a measurement for the probability of user to adopt new influences at every temporal step as the environment changes.

This paper is interesting because it attempts to decipher the mechanism of viral marketing at a temporal fashion. Since detailed temporal dynamics datasets are often hard to acquire, the paper also proposes model for "snapshot" observations which are more readily obtainable and demonstrate the relationship between the model for "snapshot" model to detailed temporal model.

There are several weaknesses in this paper where we think can be improved. First of all, the author uses Wikipedia data as network and define edge as interactions between editors on the user-talk page. However, each interaction are treated equally with the same weight of directed edge. In reality, some conversation should create stronger influences than others. In other words, some more influential users should produce stronger influence to others than ordinary users.

2.3 Discussion

The first two papers discussed above apply graph analysis theory to study the the social network of Sina Weibo and try to understand the behavior of the users of Weibo. They build the graph in different ways and focus on some specific aspects of the network. The first paper follows the "followee-follower" relationship to construct a graph and add some quantized variable as the weight of the edges. The second paper constructs the graph with the "post-repost" relationship. The third paper applies probabilistic models to study the sequential influence of social networks.

Though they provide some insights on network analysis, there are some flaws that cannot be neglected in their study. The data they collected are unreliable in different ways. The first paper collects data by survey, the second one uses an incomplete dataset and applies networks that are supposed to be similar to reach some obvious conclusion, and the third one uses Wikipedia and treats the interaction equally. They fail to explore the network comprehensively and to reach some general conclusions.

The weaknesses of the papers inspire us to study the network in a comprehensive way and understand the network behind Weibo better. We see the chance in mining Weibo network to discover patterns of the Weibo social influence. We use a representative dataset and try to construct the networks in different ways and combine the information retrieved from different networks. We aim to develop a probabilistic influential model that estimates the likelihood of information diffusion so that one can determine the optimal influence set and identify the most influential account in the network.

3 Method

3.1 Data Collection

3.1.1 Real Network Data

The real dataset is obtained from link `https://aminer.org/influencelocality`. The dataset involves 1,776,950 Sina Weibo users and the number of following relationship is 308,489,739. The number of original blogs is 300,000 and there are 23,755,810 reposts as well. The dataset contains several parts. The most important ones that we use are a following network which is collected at a certain time stamp and a repost network which reveals the repost behavior of users.

We parsed the original data and build networks with the basic graph type provided in Snap. To be specific, we built a directed network with PNEANet for the following network. For each user, the original data contains the number of his/her followees, the userIDs of those followees and whether their following relationship are reciprocal. In our graph, each node represents a particular user, and a directed link indicates the following relationship from one user to another. Due to the large size of the graph, loading and processing the network is not efficient enough to test our model. Hence we sampled the following graph uniformly and the resulting graph has 57832 nodes and 104772 edges, which represents 57832 users and 104772 following relationship. The other network we used is the network that represents the retweet relation. The original data contains the original tweet id and the ids of the user who posts originally and users who retweet. Each node in our graph represents a particular user and each link indicates one retweet behavior. We allow self-edges and multi-edges between nodes.

3.1.2 Synthetic Network Data

Since we are not able to obtain the true probability distribution underneath the network, we generate synthetic data to test our model so that we are able to evaluate the accuracy of our model. We use the sampled following graph that we mentioned in the previous section. We first generate random probability for each edge in our graph. Based on the assumption that a user has different probabilities to retweet tweets from people that he/she follows, we assign random probability to the in-links of each node and normalize them to sum to one. Then we generate retweet data based on the assigned probability. The number of path we generate shrinks exponentially with the length of the path. To generate a retweet path of a certain length, we first sample a random node from the whole graph as the original tweeter and the starting node of the path. Then we follow the out-links of the node based on the probability of the links which represents a retweet of a follower. We extend the path recursively until we reach the length we want.

3.2 Data Preprocessing

To avoid unnecessary repetitive computation and ensure the efficiency of the training process, we preprocessed the graph first to retrieve the information we need. From each original tweeter u_i, we perform a complete depth-first-search so that we can enumerate all possible path from u_i to u_j, for every u_j that retweets from u_i at least once. When we find a path τ ended with a node that once retweets from the original tweeter, we calculate the statistics of the path and store them with the path. The statistics are used to evaluate the likelihood of τ, calculated by checking the retweet timestamps. For each adjacent pair of nodes in τ, we examine their timestamps. If one or two nodes in the pair never retweet the original node, we call it a missing. If the latter timestamp is smaller than the former node, we call it a conflict. Otherwise, we call it correct. We count and store the numbers of missing m, conflict c, and correct o nodes in each path. The final results are saved in a path dictionary $\mathcal{D}$, whose keys are tuples such as (u_i, u_j), and values are sets of tuples τ, $\{(\tau_i^j, m_i^j, c_i^j, o_i^j)\}$ for each possible path τ_i^j from u_i to u_j.

3.3 Algorithm

3.3.1 Formulation

Let G_f, R denote the directed user-following relation graph and the time-stamped retweet information, correspondingly. G_f is a directed graph where edge direction $u_i \rightarrow u_j$ represents user j following user i. R contains information of all retweet messages, where each element r_i is a set of $\{u_j, \mathcal{V}, \mathcal{T}\}$. $\mathcal{V}$ is the set of users v_k that retweet the message from u_j at time t_{jk}, $t_{jk} \in T$. Here we make a strong assumption on the graph: the user can only retweet messages from users that he/she is following. The extreme cases of popular tweets where users may retweet from non-following users are rare and ignored. The problem is defined as: given G_f, R, find the set of retweet probabilities P between any two users u_i and u_j, that best fit the ground truth provided in R: $P' = \text{argmax}_P f(G_f, R, P)$. Note that the model learns a Gaussian distribution for P, $P \sim \mathcal{N}(M, \Sigma)$. Denote the probability of user i retweeting message from user j as p_i^j. To solve the problem, we take two approaches with two different models, and compare results in the the results section.

3.3.2 Node Model

The node model takes the assumption that each user has a value of popularity q. For user u_i that follows multiple users, the probability of p_i^j is proportional to the popularity score q_j of user j. We assign μ, σ for each node, and the value q is sampled from $\mathcal{N}(\mu, \sigma^2)$. The probability for each pair of users is computed as

$$p_i^j = \frac{q_j}{\sum_{u_k \in \text{In}(u_i)} q_k} \tag{1}$$

We initialize μ for each node by the assumption that nodes with higher out-degree are more popular. We have an initial preference bias value μ_b assigned to each node.

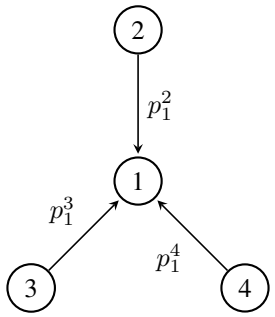
$$\mu_i = \frac{\text{outdeg}(\mu_i)}{\max_{j \in G_f} \text{outdeg}(u_j)} + \mu_b \tag{2}$$

We initialize σ for each node proportionally to its out-degree. This is to prevent the absolute effect of exploration on nodes with different out-degree: for nodes with higher out-degree, the popularity value tends to explore more than those with smaller out-degree.

3.3.3 Edge Model

For the edge model, we assign μ and σ for each edge (in-link), instead of nodes.

We initialize μ and σ for each edge according the the in-degree of the node its going to:



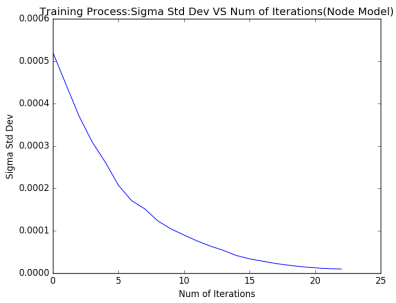
μ_1^2 is assigned according to the probabilistic model $\mathcal{N}(\frac{1}{n} + \mu, \sigma^2)$. In the above example, $\mu = \frac{1}{3} + \mu_b$, and $p_1^2 \sim \mathcal{N}(\frac{1}{3} + \mu_2, \sigma_2^2)$, stands for probability of user 1 retweeting tweets from user 2. For user i following user j, μ_i^j is the bias of u_i's preference to u_j, and σ_i^j is the fluctuation factor that measures the possibility of user j generating topics attractive to user i. The probabilities are sampled from these normal distributions and normalized to make sure $\sum_{i \in \{2,3,4\}} p_1^i = 1$. The ground truth probability is defined as $p_a^b = \frac{r_a^b}{\sum_{i \in \mathbb{N}_{\text{followed}}(a)} r_a^i}$.

3.3.4 Cross-Entropy Method

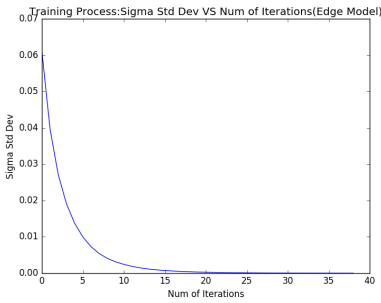
We adopt the Cross-Entropy Method (CEM) to solve the problem. We will first introduce the algorithm, then explain our choice of h and g.

Algorithm 1 CEM for Influence Set

procedure CEM ITERATION
    Initialize M, Σ
loop:
    Generate n sets of P using $g(\mathcal{N}(M, \Sigma))$
    Calculate n scores using $h(P, G_t)$
    Select m top scores P'
    $M, \Sigma \leftarrow g^{-1}(P')$



(a) Node Model



(b) Edge Model

Figure 1: Training Process of Synthetic: Avg Standard Deviation of Σ vs. Number of Iterations

Now define $M_f = \cup_{i \in G_f} \mu_i$, $\Sigma_f = \cup_{i \in G_f} \sigma_i$, and $P \sim g(\mathcal{N}(M_f, \Sigma_f))$. We rely on two functions: 1. $g(\mathcal{N})$ that maps $\mathcal{N}$ to a set of edge probabilities, and 2. a heuristic $h(P, G_t)$ that evaluates the feasibility score s of G_f having these probabilities, by comparing with G_t. The problem is defined as: find M and Σ, that generates a P that best conforms to G_t.

$$M, \Sigma = \text{argmax}_{M_f, \Sigma_f} h(g(\mathcal{N}(M_f, \Sigma_f)), G_t) \tag{3}$$

With M and Σ, we can generate probabilities for the entire graph and find an accurate influence set model.

The algorithm is defined in 3.3.4. We first assign initial values to M, Σ, and then iterate until some convergence criteria is reached. Now we take a closer look to the core functions h and g.

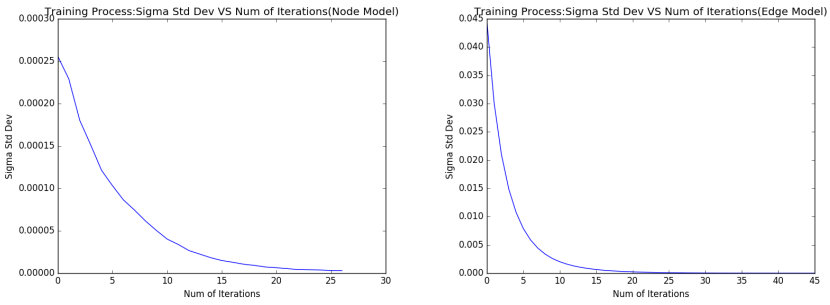
g: given M, Σ, we assign probabilities p_i^j sampled from the Gaussian distribution as defined above to each edge in G_f. $P = \cup_{(i,j) \in E_f} p_i^j$. h: Given P, we can compute a cascading probability for each retweet path from user i to user j, $\prod p_l^k$, where l, k represents the users that are in the path from i to j. The paths are loaded from preprocess defined in 3.2. With multiple paths existing between a post user and a retweet user, we take a soft learning approach towards path score evaluation. Instead of picking the one with the maximum probability p_i^j among all paths, we take account of all paths for evaluation, but weight each path by their likelihood. For a given path τ_i^j, the likelihood weight is defined as $L = \prod_{k=i}^{j} \text{indeg}(u_{k+1}) p_k^{k+1}$. For computing score, we need to make use of $\{(\tau_i^j, m_i^j, c_i^j, o_i^j)\}$ from 3.2. For each path τ_i^j, we define the score $s_{ij} = m_i^j a + c_i^j b + o_i^j c$, where a, b, and c are tunable penalize factors for the case of missing, conflicting, and correct node pairs. The final score for a tweet-retweet message is the average sum of these path scores.

4 Results

4.1 Training

To compare the performance of our models, we first run them on graph with synthetic probabilities which act as ground truth. The convergence criteria of both of our Node Entity and Edge Entity model is that the average sigma across all nodes or edges have to be below 3×10^{-6}. This critera means the probabilities we are seeking have stabalized and are not changing much across all entities. The number of example we generate is 50, and we select the top 10 to fit the new mu and sigma. It takes around 25 iterations for Node Entity model and around 50 iterations for Edge Entity model to converge. The process takes about 4-5 hours for each of them. The convergence plot is as 1a and 1b.

However, it can be notice that the average sigma of the Node Entity model starts a lot lower, which plays a role in its faster convergence. This is because with Node Entity, we have significantly lower parameters to train, which is the number of nodes, versus the number of edges in Edge Entity model, thus is a simplier model and is easier to minimize the average sigma across all trainable parameters. Another thing to notice is the convergence capability of both models. From the convergence plot of Edge Entity model, we can see that average sigma can hardly be minimize even if we run more



(a) Node Model (b) Edge Model

Figure 2: Training Process of Real Data: Sigma Avg Dev VS Num of Iterations

iterations, while Node Entity model seems to be able to minimize further. However, we want to keep a fair comparison between the two and thus we keep the convergence criteria the same.

We also train our models on the real retweet data with the same parameters, the convergence plot is 2a and 2b and they exhibit the same behavior as with synthetic data.

4.2 Experiments

4.2.1 Synthetic Data With Ground Truth

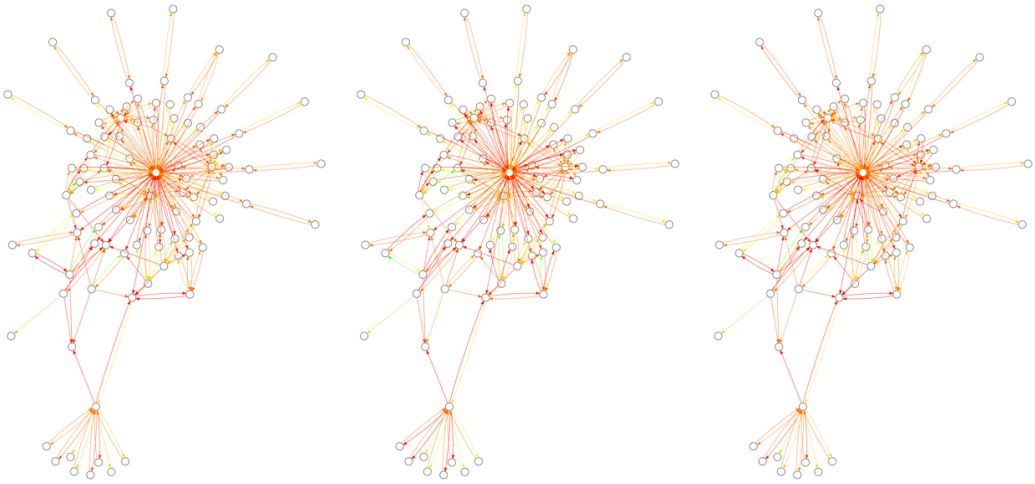
To evaluate the performance of our models, we compute the L2 loss between the obtained probabilities from both models and ground truth probabilities, and compare them with the L2 loss that a random model would produce. We find that Node Entity model yields a L2 loss of 265.8528 ; Edge Entity model yields 133.3816; and a random model yields 520.1048. This comparison suggests that both models produces meaningful results by producing significantly lower L2 loss than a random model, and that Edge Entity model performs better than Node Entity model. The second implication mentioned above should also be expected, since Edge Entity model imposes less assumption of how retweet should behave amongst the followers who see the original tweet. In our ground truth synthesis, we also made no assumption on the probabilities of how a tweet propagates through a user's followers, thus only Edge Entity model can learn the different edge probabilities coming out of a particular node, yielding a much lower L2 loss.

To visualize the probabilities that we obtained through both models and compare it with ground truth. We sample a small portion of the follower graph and use color to denotes probabilities. The greener the edge, means that the probability is higher than its average value of $\frac{1}{deg}$. On the other hand, the redder the edge, means that the probability is lower than its average value of $\frac{1}{deg}$. By placing the ground truth, and the probabilities we obtained from Node Entity and Edge Entity model side by side 3a 3b and 3c, we can see that both model is able to obtain similar probabilities compared to the ground truth. While taking a closer look in the color shades, one can actually see that the edge model is closer to the ground truth, which corroborate with the numeric loss results we mentioned above.

4.2.2 Evaluation of Real Data

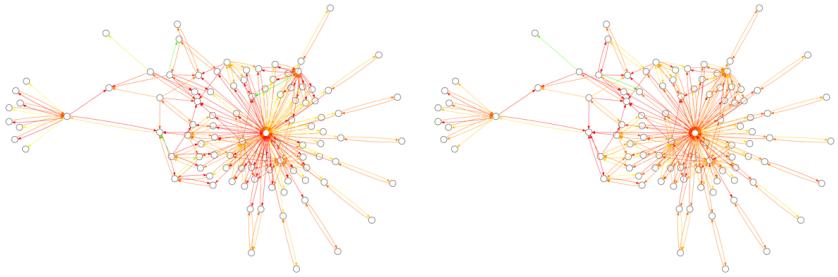
Given our models are able to approximate ground truth as evaluated above with the synthetic data, the then investigate the results of running on real retweet data, the visualization graph is shown 4a and 4b.

As it can be seen from the graph, there are minor discrepancies between the two models, but in general both models converges to roughly the same value for most of the edges. It can also be notice that most of the edges are more red and only a few are very green. This suggest that there are a few individuals that would retweet a post from a particular persons that he/she follows with high probability, which is a likely behavior for someone like a fan of a celebrity.



(a) Ground Truth (b) Result of Node Model (c) Result of Edge Model

Figure 3: Visualization of Synthetic Data



(a) Node Model (b) Edge Model

Figure 4: Visualization of Real Data

4.3 Future Work

There are certain aspect of our research that can be further extended or improved. First of which is to continue explore a bigger set of nodes and edges in our data. Throughout our investigation we were only evaluating a relatively small subset of the entire dataset we have. Even though for the main purpose of this paper our goal is to validate the proposed method and we believe the small subset is representative of the social network, it would still be a good interest to explore the entire dataset using our models. The limitation we face is that given a large followers graph, searching all possible path between a user who retweets and the original author requires tremendous amount of memory: a retweet on average traverse 14.4 users and each users have couple of hundreds followers on average. Part of the future work is to compute such paths efficiently through better algorithms or perhaps use heuristics to approximate and ignore similar paths.

Another area we can improve on is the optimization of our tunable hyperparameters, and our implementation of evaluation function. One can potentially design different evaluation strategies and hyperparameters in the evaluation functions, and compare the L2 loss amongst them. In designing the evaluation functions, one can also incorporate other properties of the graph other than the timestamps of retweet that our current evaluation function is based on. For example, we can favor path that go through user that have more frequent retweet behavior, etc. By further investigating the hyperparameters and evaluation functions, we can get an even more accurate model.

Our implementation of cross-entropy method can also use some improvement. Once again, we could explore the effects of some hyperparameters such as convergence criteria, number of examples to generate, and number of example to select with top score, and potentially improve our model. Also,

we could improve our convergence criteria to include simulated annealing of sigma. This will prevent our training process to converge too quickly without fully explore a particular value of sigma. This can increase the likelihood of finding global optima while training.

With the probabilities obtained from our models, we can solve more interesting problems such as maximal influence sets, predicting information cascade behavior, target marketing, etc. One can further investigate these problems and further validate the feasibility of our models.

References

[1] Wang, Lingyu & Qu, Weina & Sun, Xianghong (2013) An Analysis of Microblogging Behavior on Sina Weibo: Personality, Network Size and Demographics. In Rau, P. L. Patrick, *Cross-Cultural Design. Methods, Practice, and Case Studies*, pp. 486–492. 5th International Conference, CCD 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I

[2] Li, Feng & Lin, Ning (2015) Social network analysis of information diffusion on Sina Weibo micro-blog system *IEEE International Conference on Software Engineering and Service Science.*, pp. 233–236.

[3] Cosley, Dan & Lan, Xiangyang (2010) Sequential influence models in social networks *In ICWSM*