Georgia Institute of Technology
CSE 6730

Simulation of rumor spread in social network

Checkpoint Report

Haomin Lin
(hlin374@gatech.edu)

Yuntian Zhang
(yzhang3469@gatech.edu)

Zichao Feng
(zfeng78@gatech.edu)

February 2020

1 Problem Description

Rumor spreading is an important factor in social communication, which has triggered vast research interest. It is much more complex than epidemic spreading or news spreading since there are uncertainty during the process. Sometimes a person doesn't believe the rumor at all, sometimes he buys it and becomes a rumor spreader. The probability will affect the result of the rumor spread. It is fairly useful in many areas, like market branding, entertainment business, communication, etc. Our project aims to simulate the process of rumor spread in social network using Discrete Event Simulation. Currently, there are many researches focusing on social network and rumor spreading. SIR models[1] are most famous by adopting the idea mentioned above. Also, there are some simulations focusing on large-scale social network, like SUPE-Net model[2]. However, there are several limits among current study :

(1) With social network being dynamic, it usually has a lot of interaction. The complexity of the model is much higher than a static model. So the original model can be too simple to be real. We need a parallel model to simulate this properties[3].

(2) Social network usually involves higher participation, which makes the model harder to simulate due to its large scale. Some large-scale models even need large scale parallel computers to simulate.

(3) In a social network system, the outcome of rumor spreading can be largely different if some uncertainties happens. Most of the models are built with a lot of assumptions. Therefore, the prediction of a model will only be correct if there are no uncertainties at all, which seldom happens.

All these properties bring more difficulties to building an accurate social network simulation, which will also be the biggest challenges for our project.

2 Literature review

While most of the SIR model only involves the interaction between spreader,ignorant and stifler, our model adds one more role in the social network which is called truther. It has the same function as the spreader. In order to simplify our calculation modeling process, we set the probability that an ignorant can be influenced by a truth that has the same as the one of a spreader. Similarly, In the model mentioned in Laijun Zhao's paper about SIHR rumor spreading model in social networks[4], it introduces a new group called hibernators. The concept of introducing this new role is the same as our incentive of introducing the truther function. While the difference is that the hibernators mentioned in the paper above won't actually influence the ratio of different S-I-R people in the social network but our truther will actually have.

While another major problem of SIR model is the consideration of trust mechanism. The SIR model was first introduced by Daley and Kendall[5−7] who proposed a standard rumor spreading model. The major shortcomings of the DK model is that it doesn't consider the effect of network typologies on the dynamic behaviors of rumor spreading. A lot of model proposed like Borge-Holthoefer model[8,9] and LiveJournal. Zhang model focusing on an online social blogging platform still have similar question.

However, in real rumor spreading system, the trust degree differs a lot between different relationships. We know that, in real world, the greater the degree of intimacy between two individuals, the greater the trust degree between them. Therefore, in order to imitate the social network more vividly and thus get the result more realistic, a trust mechanism is necessary to be introduced. Therefore, In paper related to SIR Rumor model in recent days, trust mechanism is usually introduced. However, the model is usually hard to imitate the true one 100%, many research use some alternative way to make it happen. For example, the paper *Rumor Spreading Mode with Trust*

Mechanism in Complex Social Networks[10] mentioned a way that propose some rate to show the proportion of the trusted neighbors.

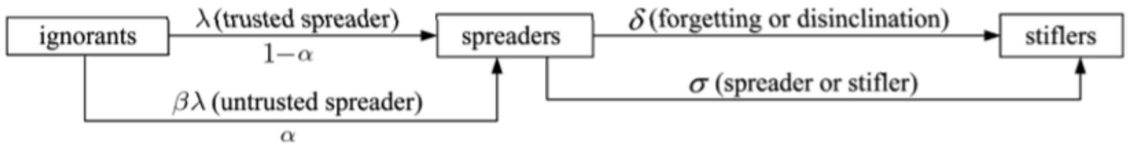

Figure 1: Structure of SIR rumor spreading process.

In our project, at this point, we haven't really introduce the trust mechanism since some technical problems. But we really definitely considering introducing this mechanism to our model in future process in order for refining our model.

Another big problem concerning the SIR model is the conversion probability between each state of the people in the network, aka, if a ignorant is influenced by a spreader, what possibility will it still ignorant, become stifler, or even become a new spreader? The answer for this question vary in different models. In the paper *Theory of rumour spreading in complex social networks*[11], the author introduces a dynamic probability calculation from Markov chain mean-field equations, that is, the probability of a node (a person) becomes from an ignorant to a spreader is decided by the state of his n-nearest node. This concept is reasonable since it reflects the fact that every one tends to believe what their most close friends believe about a rumor or other news. The author also gives us an equation based on time interval t. However, some other paper just simplify this question by giving each one in the network a equal number. The author in *Why Rumors Spread Fast in Social Networks*[12] believes that since the huge sufficiency of the network, the average speed can be used to replace each individual nodes. This largely reduce the complexity of the model. In [4], the author also mentioned this question by saying that the rate of if an ignorant buys the rumor largely depends on its background. He used the Chinese salt-buying frenzy case to strengthen his point of view. Therefore, this could be an very tricky question since the probability can be vary a lot under a lot of conditions. In our model, we imitate the model first mentioned in the paragraph.

3 Simulation Design

3.1 SUI description

We simulate a rumor spreading model. There are different types of people in this system. Spreader, Truther, Ignorant, Stifler. They are assigned into two distinct lists: active user and all user . When the simulation starts, we pick up one people from the active user list and make he/she interact with some close people in the all user list except himself. We pick up people from all user list according to the closest relation – the graph distance. Make these two interact and use stochastic value to decide what type they will turn into. After such process, we will bring new member into the whole system and continue this process. I will discuss more about this in the next conceptual model section.

3.2 Data ultilization

Our research is related to the society and human's interaction. As a result, it is very important for us to look up some data from daily life. There are many aspects that need data support in our model, such as the time interval of each conversation. We decide to use the data originated from people's phone talking length. According to the paper we chose, the time interval of mobile calls will vary because of different situations. We use these different data to come up with a average conversation time to represent our conversation time interval[13]. We also use some data to decide how the network will enlarge[14]. There is a report about the number of friends that Americans feel satisfied with. We utilize that data to decide how fast will this network grow.

3.3 Conceptual model

There are many components in our model that have lots of dynamic behaviors. The following two parts made up our conceptual model.

3.3.1 Structure

There are two classes in our system. One is user and the other is event. We use user class to represent the persons in this system. They have id and status. They will be involved into the interaction of humors. The event class is defined to represent the interaction event. We will record the talker ID and receiver ID and finish/begin to define a discrete event. In our model, the events will be put into a queue, which is called the future event list. The order of assignment of this queue depends on the interaction of users. The interaction between uses are in two people's group, which means that there is only one people will interact with the talker at one time. The following people want to interact with that talker have to wait. So each people in our model will be considered as a distinct station. They all have a priority queue to arrange the next interactive people. In our model, the relation of different people is compared to a graph. We will use the numerical distance value to represent the people's relationship, which will decide which two people will start interaction.

3.3.2 Behavior

The behavior of user class is related to the interaction. The people who are spreader or truther will be considered as active people and they will go to interact with ignorant people. The result of interaction will be decided by random number and this interaction will be packaged into an event. After such interaction, more people will be brought into the "all user" list. We implement such behavior because we want this model can be enlarged. In real society, people will enlarge their social network as well. The interaction event will just keep in line in the FEL. When one event has been solved, two people will be assigned to other people to get new interaction.

4 Simulation software

4.1 Softeware Diagram

After simulation design, we decide to deploy a diagram of our model according to what I have said above:

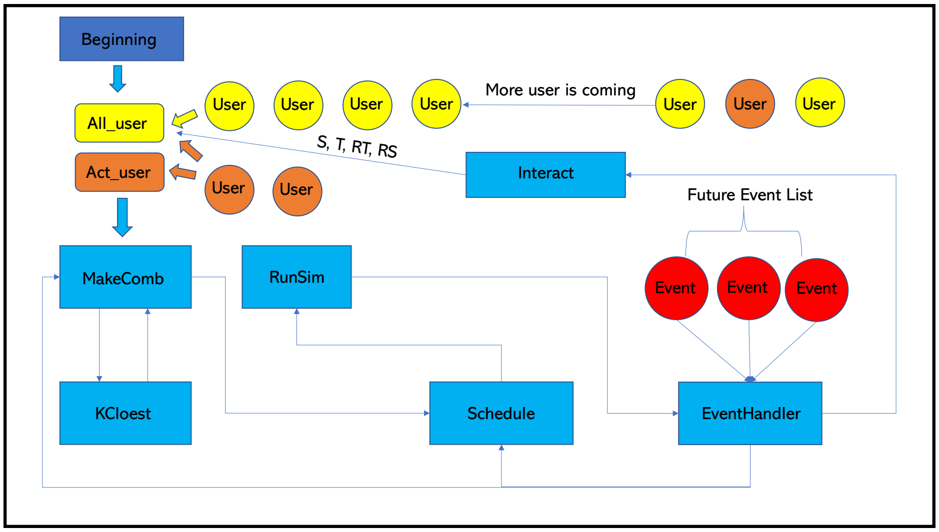

Figure 2: Software Diagram

4.2 Softeware verificaiton

In our simulation program, we aim to simulate the rumor spreading in a social network. We firstly generate several spreaders and truthers, who are labeled as "active users" in the network. With their generation, several new ignorants are also introduced into the network, the number is reflected as "numFriend" in the "User" structure.

```
User #0 introduces 2 friends
This is User #0:
His Type is S with value: 1
wait_num: 0, freetime: 0, numFriend: 2
Coordinates: (0.924305616786,0.381653149833)
```

Figure 3: User generation

After initialization, the simulation begins to run by making combination with spreaders or truthers and ignorants. Once the pair is made, the interaction between "talker" and "receiver" will be scheduled in the Future Event List (FEL). After all the active users have been allocated with their conversation partner, the simulation begins.

```
The person chosen for User #0 is #9
Combination made, talker:0, receiver: 9
The person chosen for User #3 is #0
Combination made, talker:3, receiver: 0
The person chosen for User #8 is #11
Combination made, talker:8, receiver: 11
The person chosen for User #14 is #19
Combination made, talker:14, receiver: 19
The person chosen for User #18 is #31
Combination made, talker:18, receiver: 31
Initial event list:
```

Figure 4: User combination for interaction

In the scheduling process, once an active user encounters a user who he wants to talk with is already occupied, he will have to wait until the current conversation ends, queuing after someone who is currently in the talk.

```
Now processing
S (ID:145) interaction beginning with S (ID:207)
@ 412.032415075
#145 User has been occupied, scheduled after 428.406112633
```

Figure 5: Queuing action

In the combination making process, we implement a graph made of 2D coordinates for combination making, storing the coordinates in a list "dists", corresponding to the users. To realize this, when making combinations, we calculate several coordinates closest to the current talker, and choose one randomly from them, find the user and then make the combination. If the number of interactions for someone has exceed his "numFriend", then he will not be assigned new partners anymore.

```
Now 3/3
#257 has interacted with all the friends, now stifled
```

Figure 6: Stifled user

When handling an event, if both sides in the conversation are not in a talk at the moment, the simulation will schedule a "finished" event for them and decide the length of conversation according to function "interval". However, a "begin" event will be scheduled with one of the users in a conversation, indicating that this pair has to wait until the currently talking person finishes. The waiting and occupying state are reflected in the "wait num" and "occupy" attributes in the "User" structure. And the new waiting ones will have to wait after those who come earlier.

```
T (ID:25) interaction finished with I (ID:65)
@ 546.715239909

T (ID:114) interaction beginning with T (ID:65)
@ 546.715239909

T (ID:67) interaction beginning with T (ID:65)
@ 546.715239909
```

Figure 7: Multiple queueing entities

In the "interaction", we introduce some preliminary possibility distribution to test the executability of our simulation. The conversion will take place according to the types of those who participate in the conversation and the possibility distribution for this case. And the result will reflect in the user's "Type" attribute. Type "I" can be converted to "RS"/"RT"/"S"/"T", and "T" can be converted to "RT", "S" can be converted to "RS". "I", "RS" and "RT" will not trigger new conversations.

```
Now is 53.0160431252
Now processing
T (ID:8) interaction finished with I (ID:11)
@ 53.0160431252, result:
T (ID:8) , and T (ID:11)
```

Figure 8: Interaction result

5 Future plans

In the future, we plan to implement a more sophisticated formula to decide the type conversion in the interaction, by looking into the distribution of people around a specific receiver and the time of the interaction. Also, we will introduce varying time interval between a certain person's interactions. After that, we plan to use this simulation to execute several simulations on the problems of interest, which include:

1. Study the distribution of the interacted people in our network with different cases of initial allocation of spreaders and truthers;
2. Study the power of spreading rumor overall with different network structure;
3. Study the outcome of rumor spreading with different simulation time in the same network;
4. Study the outcome of rumor spreading with same simulation time in the same network while with different credibility of the rumor;
5. Study the dynamic of network spreading by checking the network status in different time steps.

Our research plans are not limited to the above, we will update all the issues we study in the final report. What's more, we will also implement a visualization to the network to visualize the activity of the rumor spreading and try to implement some data structure like the Ladder Queue[15] to construct the priority queue so that we can get our simulation run faster. in the future.

6 Contribution

During this phase, every one in the group contribute equally greatly to this checkpoint product in different aspects:

Haomin Lin is responsible for revising the report, composing the codes to simulate combination making using graph, user administration and event scheduling as well as debugging the software.

Yuntian Zhang is responsible for composing the literature review part of the proposal, reorganizing the report and composing the user generation part of the program.

Zichao Feng is responsible for composing the simulation design and problem description part of the proposal, revising the report, as well as composing the engine part of the program.

All three of us dedicate equally to the design of the simulation.

References

[1] Laijun Zhao, Hongxin Cui, Xiaoyan Qiu, Xiaoli Wang, Jiajia Wang. "SIR rumor spreading model in the new media age," *Physica A: Statistical Mechanics and its Applications*, 1998, 392(4): 995-1003

[2] B. Hou, Y. Yao, B. Wang and D. Liao. "SUPE-Net: An Efficient Parallel Simulation Environment for Large-Scale Networked Social Dynamics," *2010 IEEE/ACM Int'l Conference on Green Computing and Communications* & *Int'l Conference on Cyber, Physical and Social Computing*, Hangzhou, 2010, pp. 628-635.

[3] Hou B, Yao Y, Wang B, et al. "Modeling and simulation of large-scale social networks using parallel discrete event simulation," *Simulation: Transactions of the Society for Modeling and Simulation International*, 2013, 89(10): 1173-1183.

[4] Zhao Laijun, Wang Jiajia, Chen Yucheng, Wang Qin, Cheng Jingjing and Cui Hongxin "SIHR rumor spreading model in social networks" *Physica A: Statistical Mechanics and its Applications*, 2012, 391(7):2444-2453

[5] Daley, D., Kedall, D, "Epidemics and Rumours" *Nature*, 1964, 1118(204)

[6] Newman, M. E. J. and Forrest, Stephanie and Balthrop, Justin, "Email networks and the spread of computer viruses" *Phys. Rev. E*, 2002, 66(3):035101

[7] Wang, Fang and Moreno, Yamir and Sun, Yaoru, "Structure of peer-to-peer social networks" *Phys. Rev. E*, 2006, 73(3):036123

[8] J. Gu, W. Li, X. Cai, "The Effect of the Forget-remember Mechanism on Spreading" *Eur. Phys. J. B*, 2008, 62(2):247-255

[9] De Lellis, Pietro, Jinxian Li, Yanping Hu, Zhen Jin, "Rumor Spreading of an SIHR Model in Heterogeneous Networks Based on Probability Generating Function" *Complexity*, 2019:1-15

[10] Ya-Qi Wang and Xiao-Yuan Yang and Yi-Liang Han and Xu-An Wang, "Rumor Spreading Model with Trust Mechanism in Complex Social Networks" *Communications in Theoretical Physics*, 2013, 59(4):510-516

[11] M. Nekovee, Y. Moreno, G. Bianconi, M. Marsili, "Theory of rumour spreading in complex social networks," *Physica A: Statistical Mechanics and its Applications*, 2007 ,374(1):457-470.

[12] Doerr, Benjamin, Fouz, Mahmoud, Friedrich, Tobias, "Why Rumors Spread So Quickly in Social Networks," *Communications of The ACM*, 2012, 55:70-75. 10.1145/2184319.2184338.

[13] Peilin Luo, Leopoldina Fortunati, Shanhua Yang, "New technologies in global societies," *World Scientific*, 2006:168.

[14] Joseph Carroll, "Americans Satisfied With Number of Friends, Closeness of Friendships" *Gallup News Service*, 2004.

[15] Tang W T, Goh R S M, Thng I L J, "Ladder queue: An O (1) priority queue structure for large-scale discrete event simulation," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2005, 15(3):175-204.