Project Final Report:

Information Flows on Twitter

005670213 Yu-Wei Lin

*cooperate with Huang-Wei Chang, Te-Yuan Huang as their cs224w project (constructing epidemic model)

1 Introduction

Twitter has become a very popular microblog website and had attracted millions of users up to 2009. It is generally considered as a social networking website but gradually also used as a media for people or companies to spread out news or marketing information. One reason for considering Twitter as a media that broadcasts information instead of a social network is that on Twitter the friendship between two users are asymmetric and according to[1] only 22.1% of the relationships are reciprocal. This is very different from the online messenging services such as MSN or Google Chat. However, different from the traditional media like TV or newspapers, a user can easily propagate information she saw from other users to her followers by retweet, which is an automatic way of duplicating others' posts on the reader's Twitter board. The retweet function is critical for making the information available to a large number of users. However, in the online environment, I know sometimes people see but not really read the content.

Therefore, the goal of the project is to predict the "infection rate" between information provider and receiver, and study information flow on Twitter network. I construct epidemic models according to the design of Twitter. Please see Figure 1 for a summary of our epidemic model. With epidemic model, our prediction function of infection rate can better capture the nature of information cascade on the Twitter network.

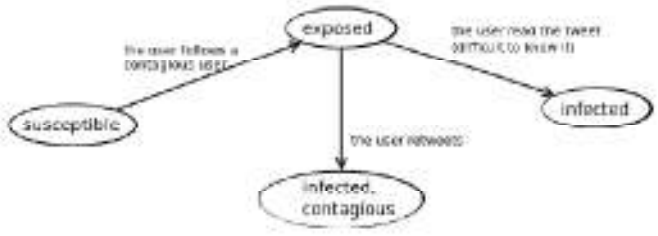

Figure 1: The Epidemic Model

In Section 3, I'll talk about the brief analysis of the profile data and, based on this, how I normalize them. Also, the process to get the reply and retweet rate. Then the learning model for predicting the rate in Section 4, and how and why I evaluate the result of the

learning function in Section 5. Finally in Section 6, I will summarize our project and describe a few take way ideas.

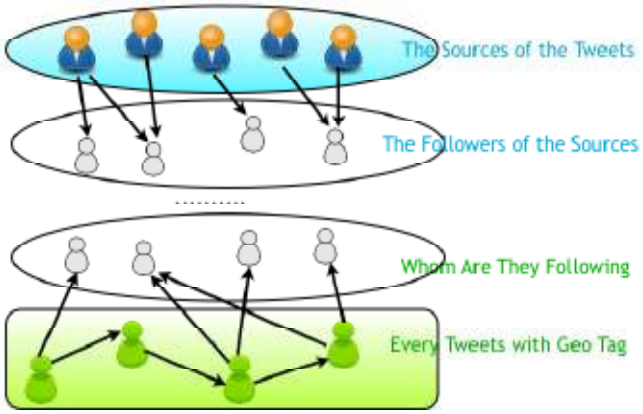

Figure 2: Data Collection for Building NYC Network

2 Our Dataset

In this project, I collected the dataset ourselves in New York City, Bay Area, and Chicago at summer quaretr. For our dataset, I are focused on general mobile users who use twitter on-the-go; the users involve in the data are common people like you and me. In this section, I will detail on how I collect the NYC dataset from twitter.

In order to identify mobile users in New York City, I use Twitter's APIs to collect all the tweets generated by mobile devices and with GPS location within the NYC. Their geo-location APIs allows us to collect tweets that are posted within a pre-configured geographical range, and I further use the name of the twitter client to select the tweets generated by the twitter clients on mobile phones, such as "twitter for iPhone" and "twitter for android". I collect all these tweets from the 10 most populated cities in the United States, including New York City, Chicago, BayArea, Los Angeles, Houston, Dallas, San Antonio, Phoenix, Philadelphia and San Diego. However, among all the cities I collect data from, New Yorkers seems to use twitter the most and the most frequent. I can collect around 40-50MB of tweet from NYC each day. For the rest of the cities, it's around 10-20MB of tweet per day. Therefore, in the rest of the project, I will use the data from NYC to analyze how information flows on twitter's network.

Since I are interested in knowing how the messages are propagated between people, I are also collecting the followee/follower relationship of the users in our trace. For each retweet, I would like to know how the tweet is propagated to the user. First of all, I collect whom the user's following, i.e., the user's friends in Twitter's terminology. Secondly, since retweets follows the format: "RT @Source: content", I parse the content of the retweet to

retrieve the source of the tweet. After I learn the sources, I collect the sources' follower. In Figure 3, I plot out the data I collected and their relationship between each other. In our dataset, there are 440,135 tweets and 15,919 of them are retweets. The retweets are generated by 1407 unique users, whom I called seeds; while the sources of the retweets are 8517 unique users, whom I called sources. Among the 1407 seeds, 1244 of them allowing us to collect their friend list, and among the 8517 sources, 6167 of them allowing us to collect their follower list. After collecting the friends of the seeds and the followers of the sources, I have roughly 250,000 users (or nodes) and 4,000,000 edges in our network. In other words, this network is very well connected and most of the retweets can travel from the sources to the seeds within two steps.

3 Data Analysis

The features I used in the prediction function includes the following information of both sources(information providers) and seeds(information receiver):

status count The number of tweets the user has.

followers count How many people follow this user.

friend count How many people the user follows.

status count How many retweets the use made before.

time zone The time zone of the user.

As the logarithm histogram of each feature in Figure 4, I can find out that count of followers, friends, and status are in normal distribution after taking logarithm, so I normalize these three by *normalized = (log(#) – mean(log))/ std(log)*.

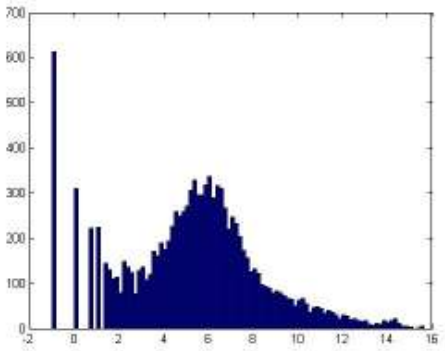

Fig 4(a) distribution of logarithm of followers count

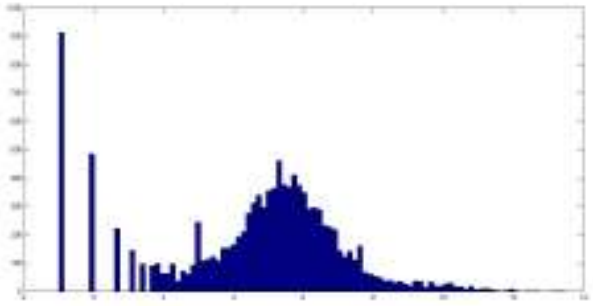

Fig 4(b) distribution of logarithm of friends count

4 Learning for Rate Prediction Function

There are two main parameters in our epidemic model: the infection rate, which is the probability for the user to read a post from whom the user follows, and the contagion rate (or retweet rate), which is the probability for the user to retweet a post after reading it. As I have shown in previous sections, the information cascade on the Twitter network is decided by the two parameters.

In stead of fitting the parameters of the network for each edge in the Twitter network, I want to learn functions to predict the rates for a given pair of Twitter following information. Using the Twitter API, I can obtain the posts by a given user. Furthermore, if a post is a retweet instead of an original post by this user, there will be a mark ”RT“ before it. Therefore, I can estimate the overall retweet rate. Even more, the retweet posts also record who is the source. Therefore, fix a pair of users (u; v), in which u follows v, I can compute the retweet rate between them.

To be specific, I select a set of (directed) edges E0 in the NYC network, in which for each edge (u; v) in E0 there exists at least one retweet 2. I use the profile of u and v as the input (features) and compute the retweet rate of (u; v) as the output of a linear function f. Denote the features of the i-th edge in our dataset as $x^{(i)}$ and the retweet rate to be $r^{(i)}$.

I want to learn the coefficients α of $f(x) = \alpha^T x$ so that $f(x^{(i)})$ is close to $r^{(i)}$. I For the closeness I can use the least square distance. Besides, since the rate is the parameter of a Bernuli distribution, I can compute the KullbackLeibler divergence (KL-divergense) or the Bhattacharyya distance between the distributions as the distance. To sum up, I use the following three different cost functions for $f(x^{(i)})$ and $r^{(i)}$:

Least square distance :

$$\|f(x) - r\|^2$$

KullbackLeibler divergence :

$$D_{KL} = f(x)\log\frac{f(x)}{r} + (1-f(x))\log\frac{(1-f(x))}{1-r}$$

Bhattacharyya distance :

$$\sqrt{f(x)r} + \sqrt{(1-f(x))(1-r)}$$

For the case of using least-square distance, I solved it as a linear regression problem using the package in Matlab. For the other two cost functions, I do the optimization using gradient descent.

5 Evaluation of the Result

To evaluate the learned prediction function, I compute the log-likelihood of our data. From the histogram of post#, I have the weight function of loglikelihood be logarithm of status instead directly using number of status; that is,

$$\sum_{i=1}^{|E'|}\left[\#reweet(u^{(i)},v^{(i)})\times\log f(x^{(i)}) + (\#post(v^{(i)}) - \#reweet(u^{(i)},v^{(i)}))\times\log(1-f(x^{(i)}))\right].$$

Besides the aforementioned three models using different cost functions, I also compute the loglikelihood using the total average retweet rate for comparison. The table below is the result:

Retweet	**NYC**	**Bay Area**	**Chicago**
Normal	-68.7986	-22.3473	-32.3284
KL	-70.2389	-24.3290	-31.8923
Bhattacharyya	-70.0540	-22.8463	-33.4383
Logistic	-69.1939	-23.3248	-32.7436
Average	-93.2763	-33.3463	-45.3673

Reply	**NYC**	**Bay Area**	**Chicago**
Normal	-70.3467	-23.1389	-34.1238
KL	-70.3123	-25.2349	-36.9230
Bhattacharyya	-71.1238	-24.8123	-34.3290
Logistic	-71.9123	-25.0139	-36.0239
Average	-95.1823	-36.0349	-49.4098

6 Summary

To sum up, during the past several weeks I did the data preparation and preprocessing. Besides, I also investigated the data to figure out what kind of learning model and features I can use to predict the model parameters. The model parameters and the coefficients of the prediction functions should be able to reflect some interesting properties of information cascade on Twitter network.

References

[1] Kwak H., Lee C., Park H., Moon S. (2010), Proceedings of the 19th International World Wide Web (WWW) Conference