Social Computing: Final Report R3

Niki Gitinabard & Farzaneh Khoshnevisan

Fall 2015

1 Data Preprocessing

Our data contains detailed information about each tweet including: text, hashtags, links, mentions, user and retweeted user information (if this tweet is a retweet). Another dataset we have is the mutual-follow graph of users in this network.
Each of these datasets have their own limitations. The most important one is that the graph does not contain all users that we have tweet data of them. So we had to eliminate some users data that the graph did not contain them. Another shortage is the structure of graph which is undirected, and it would be better if we had the directed graph of follower and followings.
First, we extracted tweet features: number of hashtags, number of mentions and whether the tweet has links or not. Then we defined a feature grade to assign to each person. The number of retweets a person can get, is related to the number of followers the person has, the degree of the followers interest in retweeting the tweets and recursively the follower of the followers. In this experiment, we assigned grade to each person in two levels. First we assigned each one a grade based on the following formula:

$$initialgrade = \frac{followers + \frac{retweets}{totaltweets} * Avg(followersoffollowers)}{Max(followers)}$$

Then we update the grades another time for each person by assigning a weight to their own grade and average grade of its followers by the following formula:
updated grade = 0.7 * initialGrade + 0.3 * Mean(initialGrade(followers))

With this assumption, for each tweet data if we consider the grade of user and the grade of the user this is retweeted from, as input features, then we know how much this person is ought to retweet and how influential this tweet is.

For traversing the mutual-follow graph and finding the grade for each user, we first used R-igraph library, but it was too slow and we switched to the python-snap. Then we built the graph and traversed all nodes twice for assigning their initial grade and then the updated one.

2 Learning Data and Evaluation

We divided our data into 80% training data and 20% testing data, with 5 features and one data label as retweet count. We normalized all features to the [0,1] interval using min-max method. Then we used neural network implemented by R to learn data.

We tried different structures of neural network and comparing the results together, a neural network with one hidden layer and two neurons was the best match to our data. We repeated the process of learning data for 10 times and each time collected the train and test data randomly. After each training process, we used our network to test data and predict number of retweets and compare it to the real value. We used Mean Square Error as evaluation measure.

3 Results

As it is mentioned we defined our error measure as the mean of squares of the difference of the guessed answer for number of retweets from the real value for test data, which will be the formula below:

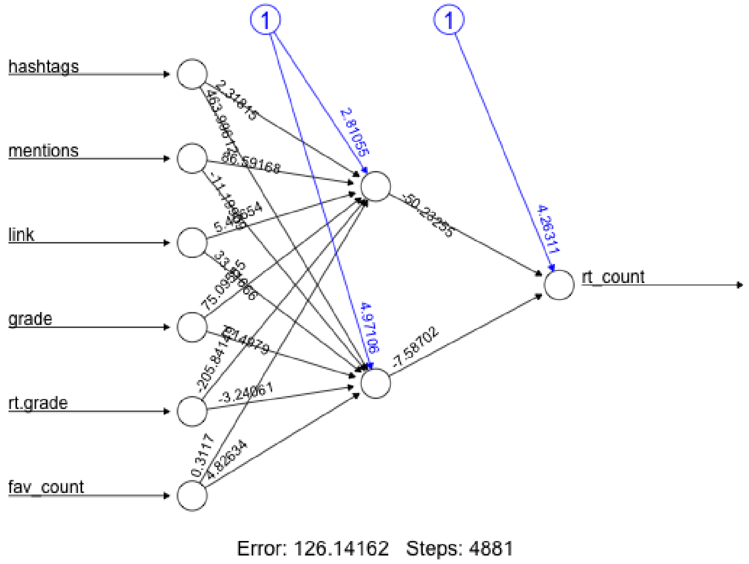
$$Error = \frac{\sum (guessed\_val - real\_val)^2}{count}$$

Then we partitioned our data to test and train randomly(with a constant seed to be able to compare results) 10 times, and checked the square root of the mean to be able to compare it to the range of results.
Our baseline method was to use the average retweet count of the author as the prediction. We used our created maps containing each persons total number of tweets(total_tweets) and total number of being retweeted(retweeted_by) and calculated this baseline as a map for each persons $\frac{retweeted\_by}{total\_tweets}$. Then we randomly selected 20% of the tweet files as test and the other 80% as train data. The error for that method was 1252.38, which compared to the range of the total retweets(0, 62000), is 2.02%.

First, not including our computed grade and defining our features as "hashtags, mentions, link, grade", the final error was 808.60, 1.30% of the result range which was better than the baseline.

Then, we added our computed grade and our error was 836.617, 1.349% of the range, which is not much different from the one before. We guess the network must have over-fit to the training data, so, we tried adding more features. Then we tried adding number of favorites and observed that the error changed to 796.032115, 1.28% of the range, which was less than the one before, so we decided that this feature has an effect on our result and we should keep it. Later, we discussed that the grade of the main owner of a retweeted tweet should have an effect on the results too. So we added another feature which was the grade of the main tweeter of a retweeted tweet, or 0 for the non-retweeted ones. Running this analysis took more time and gave us a final error of 756.29, 1.21% of the range, so the first author of the retweeted tweet has an effect on the final count

of retweets, too.
The final Neural Network was as below:



4 Findings

Our hypothesis was that the final number of retweets is related to factors "hashtag, link, mention, grade". But after more observation, we found out that it also depends on the number of total favorites of the user(perhaps because it shows the user's activity on twitter), and the grade of the main author if any.
Using our final neural network, if a certain user tweets something with a certain value for hashtags, mentions, links and according to users previous records which we have saved in 3 maps called total tweets, total favorites, and total being retweeted tweets, we can predict how many final retweets it might get which can help in commercial use and news sharing. We can know which user is best to share it at the first place among certain users.
Adding features one by one, and watching the effect on final error, we could find out that the combination of grades(for both original author and the current author) and the total amount of favorites have an eye-catching impact on the final retweet counts, changing the error from 1.34% to 1.21%, because it has the data from analysis of the social graph, also the grade of the original author is important because it can show the real value in the tweet text. A possible problem of this analysis is that the social graph didn't contain all the users and all the follower-following relationships, only the mutual ones were listed. We predict the results could be much better if we had the complete social graph.