Predicting the Future of Communities using Time Series Data

11th June 2021

Kayle Ransby[1], Shuzhen Heng[2], Yun Chen[3]
Zhihao Song[4], Jooyoung Kim[5]
Giulio Dalla Riva[6]
Department of Mathematics
and Statistics
University of Canterbury
Christchurch, New Zealand
krr39@uclive.ac.nz[1], she125@uclive.ac.nz[2], ych215@uclive.ac.nz[3]
zso22@uclive.ac.nz[4], jki95@uclive.ac.nz[5]
giulio.dallariva@canterbury.ac.nz[6]

***Abstract*—This paper proposes a method to use a community's structure to predict its growth and future. Data was collected from Twitter and wrangled into an edge list containing user-to-user connections. Communities were defined using the Louvian method and strung together based on their node similarity. Graph measures of these communities then allowed for different prediction models to be trained in order to predict if a given community is born or dies. In addition, if a community survives into the next week, we can predict whether it has grown.**

***Index Terms*—Community detection, Network analysis, Social Network, Graph theory**

I. Introduction

This paper is a partner explanation to the methodology developed for using a community's structural elements to predict its growth. All source code implementing the proposed methodology explored in this paper can be found at [1].

Can a community's structural elements be used to predict its growth and future? In order to realise this, we need to think about what the *structural elements* of a community actually are, and whether it is feasible for such a question to be answered. In terms of graph theory, we could define *structural elements* as transitivity, modularity, motifs, degree, etc. The list goes on, but a problem arises when thinking from this direction; Communities, more specifically online communities behave sporadically based on external factors that would require an unreasonable amount of research to uncover depending on the subject these communities are based on.

This leads on to the option of a natural language processing or a corpus linguistics approach to the prediction. This has its own set of problems, namely, generating a network based purely on text can prove troublesome when compared to simply analysing interactions themselves. On the other hand, detecting habitus could allow for a more realistic model of community structure within the network. Similarly, topic modelling could allow for the detection of a tone shift within community communications, allowing for another element to use for prediction

Despite these differences, this paper focuses on an approach surrounding graph theory. This proved to be the most realistic option for us, and the proposed method seems appropriate given the data at our disposal.

II. Background

A. Data source

The number of potential social networks to perform analysis on are vast, but not all are created equal. Initially we considered using either Facebook or Reddit for this type of analysis, as both have easily distinguishable community structures within their platforms (Facebook groups, subreddits). Unfortunately, Both of these options do not have easily-accessible or open-source solutions for scraping their data, or their solutions do not offer "time series" data, which we require to answer our thesis.

Given the restrictions of the data we require, we turned our attention to Twitter, which as multiple open-source solutions for scraping data. The method we ended up using allowed for essentially infinite data on any timescale we want, which is perfect for us. A limitation of Twitter is that it lacks pre-defined community or group structures like Facebook or Reddit, meaning we'll have to detect communities using other means.

B. Defining Communities

An important part of this project was to determine where communities are in a network, but before we could do this, we must first define what a community is. Unfortunately, this is not a straightforward task. A community can be defined by many different standards: habitus, essentialism or sectarianism,

mist communities have high elasticity as well. So, it is hard to define a community in a definitive way.

For our research, we had to make a compromise and say a connection/cluster in a network was a community. The cluster of nodes can be formed according to a defined relationship, like a mutual hobby, same religion, or participating in the same sports club. A connection in our case is defined by a tweet in a Twitter conversation. We are forming a community that shares the same tweet conversation, aiming to find the trend of community change in a particular tweet topic.

C. Defining Growth

Since we take the community as a cluster from a network, the growth of a community can by defined by the growth of a cluster. We are using two different measures to define growth: the change of number of nodes or the change of number of edges.

III. Method

A. Data Collection

As stated in Section II-A, the data source is Twitter. There are a few notable approaches for extracting data from Twitter, but in the end, we settled with the Twitter Intelligence Tool (TWINT) [2]. TWINT is a simple open-source script and Python library that bypasses the limitations of the Twitter Application Programming Interface (API). It was used in its Python library form to create a script for retrieving tweet data relevant for network creation.

Given the source and means for retrieving any tweet data we need, we had to define the topic to search for through TWINT, as well as the desired time frame. For search topics, we decided to use search terms related to the five different political parties present in the 53rd New Zealand Parliament throughout the 2020 elections. The search terms included the names of the parties and their respective Members of Parliament (MPs), as well as the twitter usernames of the aforementioned two.

The time frame to retrieve the data from is restricted to a time series based on this paper's research question, otherwise we wouldn't be unable to predict anything. Given this restriction, the data was scraped from Twitter in 7 day (1 week) chunks over a period of 52 weeks (1 year). The python script works by scraping from the end of the period back to the start. We defined the end of the search period as October 17th, 2020, which was the end of the 2020 elections in New Zealand.

For faster development and testing of our methodology, the TWINT search area was restricted to only New Zealand. This was achieved by placing a bounding circle with an 800km (kilometer) radius at the geographic center of New Zealand. There is a monument in Nelson that signifies this position, Figure 1 shows the restricted search area.

We also restricted the type of data that TWINT would retrieve, as some of the fields available are irrelevant for our purposes. The fields we ended up using for network creation with their respective descriptions are shown in Table I.

Fig. 1. The search area that TWINT is limited to.

TABLE I
Data Structure

Tweet Data	Data Type			
	conversation_id	*user_id*	*date*	*week*
Description	Conversation this tweet is part of	User who posted this tweet	Date the tweet was posted	Week the tweet was posted

This data was exported as a comma-separated values (.CSV) file to be imported into RStudio for analysis.

B. Network Creation

After loading the .CSV file from the previous section into RStudio, we are able to transform the data into a bipartite network edge list.

This is achieved by transforming the `user_id` field into a new field group `to` and the `conversation_id` into a new field group `from` and only allowing connections between nodes from different groups. Note that we also include the `week` field here so that we can extract a network from each week of the edge list.

It is also important to note that we group the entries in the edge list together and take only a single occurrence of each entry. This is because we are only interested in whether a user has participated in a conversation, not how many times they have participated in said conversation. The code that performs this sequence is shown in Listing 1

```
# Retrieve bipartite links from tweet dataframe
bipartite_links <- tweet_dataframe %>%
  rename(to = user_id,from = conversation_id) %>%
  group_by(to, from, week) %>%
  slice(1) %>%
  ungroup()
```

Listing 1. Code for generating the bipartite edge list

Given the type of analysis we would like to perform, the bipartite network is inconvenient. We can simply transform this into a unipartite network by joining two users together

if they are part of the same conversation. This eliminates the `conversation_id` (`from`) field, and allows for direct user-to-user links. Listing 2 shows the process for this.

```
# Generate unipartite edge list from bipartite edge
    list:
user_links <- bipartite_links %>%
  full_join(bipartite_links, by = c("from", "week")) %>%
  filter(to.x != to.y) %>%
  select(-from) %>%
  rename(from = to.x, to = to.y) %>%
  group_by(to, from, week) %>%
  slice(1) %>%
  ungroup()
```

Listing 2. Code for generating the unipartite edge list

We now have a means for creating a network object from the scraped tweet data using the igraph [3] and tidygraph [4] R libraries.

C. Community Detection

Community detection was performed on tidygraph networks generated using the unipartite edge list based on each week. As we mentioned in the background, the detection of community is transformed to the detection of the cluster in the network.

There are a few methods from tidygraph library for groups detecting, computing different parameters of a network. Group nodes via short random walks, group edges by biconnected components, group nodes by density, etc. In our research, the Louvain Method (`group_louvain()`) within the tidygraph library was used for identifying communities.

Each week's network data was then converted to a data frame and stored in one large list containing information for each node including its unique name, the week it inhabits, and the community it is part of within that week. This allows us to compare communities with each other and identify whether that community survives to the next week, dies in the current week, or was born in the current week.

D. Community Growth

Naturally, the life-cycle of a community includes birth, growth, contraction, and death. Multiple communities can merge into one community, and reversely, one community can split into multiple communities. If we call the current stage of the community "father" and call the community at next stage that related to the current community (either by growth, contraction, or split) "son", the "father and son" relationship will be a many-to-many relationship. "Community Growth" stands for all the state changes between father and son.

We are using the logic as following to define the growth of community. Look at the communities from week(n) and week(n+1), for each community from week(n), repeat the comparison with all the communities from week(n+1):

- If none of them have an intersection, we call the community dead;
- If some of them have a certain size of intersection with the chosen community from week(n)
  - Calculate the proportion of the intersection against the chosen community from week(n).
  - If the proportion bigger than the threshold (0.3 for example), we call the community from week(n) "Father", and the community from week(n+1) "Son".

The intersection and threshold are mathematical indicators to quantify the community change in our research. Intersection stands for the heritage nodes from the father community, the threshold stands for the percentage of the heritage nodes number against the father community nodes number. We only define the communities that inherit more than the threshold of the father community as the son generation.

After the detection of the community growth week by week, we can generate a father and son community data frame as the baseline for the community prediction. Figure 2 shows this time series data for the communities for more context.

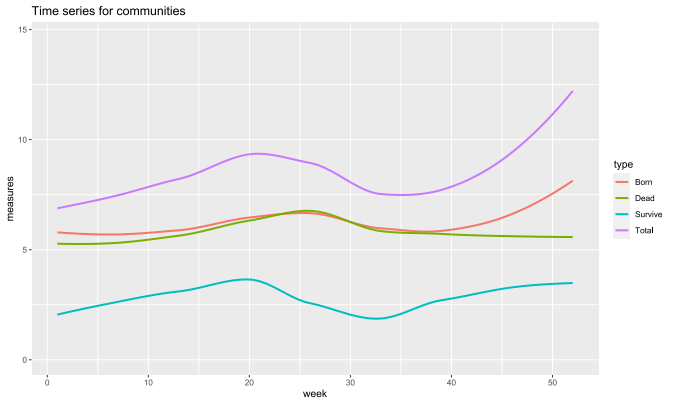

Fig. 2. Graph showing the number of communities in total, that have died, been born or continue to live for each week.

E. Predictor Extraction

In order to predict the future of a community, we need to define a set of predictors X, where X can be used to predict Y (with number of edges, number of nodes $\in Y$). Here, we use 4 different graph measures as the predictors in X. Every community within a given week has it's on unique set of predictors. They are:

Average degree: Average degree is simply the average number of edges per node in the community. It is easy to calculate using the following equation:

$$\overline{d} = \frac{|E|}{|N|}$$

Where $\overline{d}$ = average degree, $|E|$ = the total number of edges and $|N|$ = the total number of nodes. Average degree provides a strong tool to analyse the social network.

Transitivity: Transitivity of the community is a measure of the tendency of nodes to cluster together and the overall probability for the network to have adjacent nodes interconnected, thus revealing the existence of tightly connected communities (or clusters, subgroups, cliques). High transitivity means that the network contains communities or groups of nodes that

are closely connected internally. Following a social science analogy, "a friend of a friend is a friend of mine." It is calculated by the ratio between the observed number of closed triplets and the maximum possible number of closed triplets in the community.

Density: The density of the network is the number of connections divided by the number of possible connections. Fully linked networks have a density of 1, while the density of the other networks is a decimal value that represents the percentage of possible links that actually exist.

Motif Distribution: Network motifs are sub-graphs that repeat themselves in a specific network or even among various networks. Each of these sub-graphs, defined by a particular pattern of interactions between vertices, may reflect a framework in which particular functions are achieved efficiently [5].

This results in 11 different predictors in X for each community. Note that the graph is undirected, so there are two possible 3-motifs and six possible 4-motifs that we can use.

F. Prediction

The two-stage model was used in the prediction section; In the first stage, a binary model was built to predict the mortality of the communities. After the samples predicted to be predicted as a survival community, then the model in the second stage will be used to produce a specific number of growth rate.

To measure the performance for the first model, a baseline performance has been set. According to the data, the ratio of dead communities and survival communities is 302:148. Hence, the baseline accuracy rate is 67%. Adaptive Boosting method generated the best result for the first model. The data has been randomly split in to train and test with a ratio of 9:1, after that, cross-validation was applied to choose the best model. As a result, the accuracy rate is 69% and has been optimized to 73% where data had applied Principal component analysis and scaling.

In the second stage, the baseline model will be mean square error between $\hat{y}$ and the value of $\hat{y}$ in the previous week. The observations with survival communities have been used to train the model, hence, the baseline root mean square error is 39.92. After filter out the dead communities, there are 148 observations left. To maximise the train data, leave one cross-validation was used in the stage two model, and the cross-validation RMSE were used to against the baseline RMSE. The best model with cubist method produced the best of 10.14, which has largely improved from the baseline performance.

IV. Results and Discussion

In the first model, the result has slightly better performance than the baseline, it is insufficient to guarantee a reliable model. Therefore, the variables used in this model cannot make good prediction of weather a community survival or not. However, the second stage model can make relatively precise prediction for the growth of a survived community. Furthermore, the prediction performance of the second stage model is based on the performance of the first stage model. To produce a decent prediction for the growth of a community, more information is required at least for the survival model.

In summary, the limitations of our first stage of prediction meant we were not able to find sufficient results to let us confidently predict whether a community lives or dies. However, the second stage of prediction using the cubist method allowed us to predict the growth of surviving communities with a RMSE of 10.14. The accuracy of the second stage of prediction relies on the accuracy of the first stage, and given the best accuracy we were able to achieve in the first stage was 73%, our overall accuracy was hindered. However this could be due to the dataset we picked and the way we went about using it. Perhaps in a different context other than the New Zealand Elections, our prediction model will have more meaningful results.

A. Limitations

Our code's time complexity resulted in futile attempts to perform analysis on networks with a large size. We attempted to analyse a network with an edge list with ≈ 32 Million edge links with no success.

The definition of community in our research is a comprise of the real word community. The complexity and unlisted outside factors of the impact for the network are not considered in the research. As the result, the outcome of the research cannot reflect the true reality of community growth prediction. However, the research method is based on the network cluster, the result can still be applied to further network analysis-related research.

Given our relatively small data set, the community similarity proportion (percentage similarity between two communities), had to be quite low. To get viable relations between weeks, the proportion had to be 30% in our testing, meaning there are most likely more relations than necessary. This would not be a problem with a larger dataset however.

B. Future Research

Incorporating natural language processing of the raw tweet and # (hashtag) data may allow for better prediction accuracy by including context and potential external discourses.

Optimising the code for this method and allowing for the analysis of larger networks would allow for more precise outputs and more realistic community traces.

Applying the proposed methodology to different contexts and datasets would allow for more concrete results and validation of the thesis.

References

[1] K. Ransby, S. Heng, Z. Song, Y. Chen, and G. D. Riva. Group project for data419. [Online]. Available: https://github.com/krransby/DATA419_project

[2] F. Poldi and C. Zacharias. Twint - twitter intelligence tool. [Online]. Available: https://github.com/twintproject/twint

[3] The igraph core team. igraph r package. [Online]. Available: https://igraph.org/r/

[4] T. L. Pedersen. A tidy api for network manipulation. [Online]. Available: https://tidygraph.data-imaginist.com/

[5] Wikipedia contributors. (2021, May) Network motif. [Accessed: 09-06-2021]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Network_motif&oldid=1023735468