COMMUNITY DETECTION OF FREE CODE CAMP GITTER CHAT USERS

Ajit Dhobale
School of Information Sciences
University of Illinois
Champaign, USA
dhobale2@illinois.edu

Kashish Kothari
School of Information Sciences
University of Illinois
Champaign, USA
kkotha4@illinois.edu

Saumil Shah
School of Information Sciences
University of Illinois
Champaign, USA
saumils2@illinois.edu

1) Keywords: Social Networks, Community detection, Centrality measures.

2) Abstract:

The main objective of this research is to detect different communities of users of Free Code Camp Gitter Chat. Community detection on this data will give us an insight into what kind of a specifically characterized group of users exist in Free Code Camp. In order to identify characteristics of these communities we initially started with performing centrality measure analysis on the data. Performing centrality measure analysis on the network of users helped us in finding the highly central nodes in the Free Code Camp network.

3) Introduction

The modern era of internet has led to an emergence of a large number of social networking websites. These social networking websites provide a platform for users to interact amongst themselves. The social network can be described as nodes which represent the users of the platform and edges which represent the interactions that take place between the users. The tendency of users of the network to interact with the other users on specific ideas and topics leads to the formation of specific different clusters in the network. These clusters in the network are also called as communities and each community represents a particular set or group of nodes (people/ users) discussing about one particular topic. Thus, we can say that social networks are actually formed as a result of combination of various different communities.

These communities which are present in a social network cannot be usually directly seen in the network. Community detection algorithms help us in identifying the different clusters or communities of the networks. Our research focuses on implementing these community detection algorithm on Free Code Camp Gitter Chat users' network. This will help us in identifying the different types of communities that are present and are being formed in the Free Code Camp network. It will also help us in identifying the basis on which these different communities are being formed. We would also be able to understand the topic of discussion amongst the users in each and every community of the network. Without implementing community detection a lot of this valuable information will get lost in the entire network. Community detection is a method that can help us in analyzing the social network of the Free Code Camp Gitter Chat users in detail.

4) Background

Community detection is a highly used machine learning method in analyzing different networks. It helps in detecting the underlying communities of the nodes of the networks and helps in analysis of the network. Our main research on implementing community detection on Free Code Camp network mainly started by reading and understand the prior work similar to this that has been done in the past. One of the first research work that we referred was **Community Detection in Political Twitter Network**. They developed three Nonnegative Matrix Factorization frameworks to investigate the contributions of different types of user communities and content information in community detection [1]. Another important work that we reviewed as reference for our research work was **Analyzing the Game of Thrones Network.** In this network, they have performed various different analysis methods to understand how various different nodes (people) in the Game of Thrones network interact with each other [3]. Performing different centrality measure analysis helped them in finding the most important nodes of the entire network. After that they performed community detection on the network to understand and find the different communities which form the entire network [3]. Another

work that helped us in understanding more about our research was **Community Detection and Link Prediction in Facebook** [2]. All these different research works done in the past helped us a lot in building the base of our research.

5) Data

The Gitter Chat data was released by Free Code Camp as a part of its open data initiative. This is a rich dataset with multiple features and huge number of records. There is a potential of finding many patterns among the git chatters, based on the text data and other features. Also, this kind of analysis has not been done by anyone and no one has even done a publication on this dataset, as it was released recently before 2 months.

The data collection was done by Free Code Camp for the time period 2015-2017. This data was originally published as 3 json files, which was then processed in Python Code and converted into a huge CSV file.

Data contains close to 5 million text posts by Gitter Chat users. Some of the important attributes in the data set are text messages, user_id, mentions of other user, user read count of text messages. Network will be formed from this data where nodes will represent gitter users and edges will represent user mentions in text messages. The node properties are user id, username and display name. The edge properties are read count of the message containing the mention and the sent timestamp. Text messages were in the following format "'your link works and I just owned @FreeCodeCamp Bonfire: Tweak HTML and CSS in CodePen'". We also plan to use text features for the analysis – like TFIDF, Part of Speech count, occurrence of specific words (for example software/tools). Approximate count of nodes is around 2 million and edges are going to depend on the number of user mentions in the text data.

NetworkX and NLTK library in Python is mainly focused in producing the network and extracting characteristic from text data that can be used in detecting communities.

As the data has been uploaded by Free Code Camp themselves as a part of their open data initiative, it is completely reliable and we can be sure about the authenticity of the data.

User characteristics or attributes like age, gender, and country are not a part of the dataset. Although we do not consider it to be a limitation because user privacy these days has become very important, it could have helped us in increasing the scope of our analysis. Also the data size is a bit large and that is why we could not work with the entire dataset.

We also kept in mind the ethical measures while using the data for our purpose. While working on this research we have not misused any user level features or characteristics to make biased decisions or judgements from our analysis. We are not even making any kind of conclusions that would violate user privacy and ethics.

The License for this database was Open Database License (ODbL) v1.0. Subject to the terms and conditions of this License, the Licensor grants to you a worldwide, royalty-free, non-exclusive, terminable (but only under Section 9) license to Use the Database for the duration of any applicable copyright and Database Rights. These rights explicitly include commercial use, and do not exclude any field of endeavor. To the extent possible in the relevant jurisdiction, these rights may be exercised in all media and formats whether now known or created in the future.

6) Method

Defining research questions: Before starting to implement our method and starting our analysis, we thought of first defining our research questions. After understanding the entire data properly and keeping in mind our final goal, we defined 5 important questions for our study. They are as follows: (1) To find out the most central and important users of the Free Code Camp network. (2) To find out and understand the underlying structure of the Free Code Camp network. (3) To find out the communities where specific web technologies exist. (4) To predict the link between the users of Free Code Camp network. (5) How responsive is the Free Code Camp network. Defining the important questions helped us to focus on the right things in our research.

Data Pre-Processing: The biggest challenge in front of us was the size of the data. It was really difficult for us to work on a data that was so huge and thus it was really important for us to reduce the size of the data. Also we realized that most of the users of the Free Code Camp network have sent a very few messages and considering users with low messages would have distorted the results of our analysis. Thus we decided to remove such

users before starting our analysis. This would also help us in reducing the size of the data. But the problem was to decide a cutoff value for the number of messages above which we can select the users. We call this cutoff frequency. Cutoff frequency is basically the frequency above which 2 users A and B both would have sent each other the number of messages given by the value of the frequency. To find out the right value of this cutoff frequency we plotted an inverse density vs cutoff frequency plot to find out the frequency value which corresponds to the maximum density. We observed that after a particular value of the frequency, the density remained constant or in other words we can say that the density got saturated (it was not increasing) and thus we selected that value of the cutoff frequency.

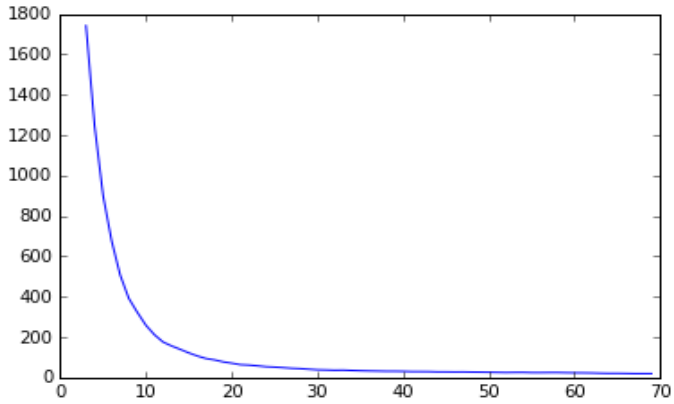

Figure 1: Inverse Density vs Cutoff Frequency

The above Figure 1 shows the plot of Inverse Density vs Cutoff Frequency. The Y axis shows the Inverse Density value and the X axis shows the cutoff frequency value. We can see from the plot that clearly after frequency = 32 the value of the density does not increase too much and it gets saturated. Thus, we selected 32 as the value of our cutoff frequency. We removed all the users who had sent messages less than the cutoff frequency. After removing the users below the cutoff frequency we observed that there were total 263 nodes and 942 edges in our network. Here nodes represents the users of the Free Code Camp and edges represent the messages sent from one user to other in Free Code Camp.

Centrality Measure Analysis: After doing data pre-processing we decided to go ahead and find out the most important and central users of the Free Code Camp network. That would even help us in answering one of our important research questions and that is why we decided to go ahead with that before we actually implement community detection and link prediction. For finding out the most central and the important nodes of our network we performed centrality measure analysis and we found out the top 5 nodes with highest betweenness, closeness, clustering, degree and eigenvector.

The results and analysis of these centrality measures are described in the next section named “Results”.

Network Structure: Another important question that we wanted to answer was the underlying structure of the Free Code Camp network. We wanted to check whether the Free Code Camp network is a Small World Network, Preferential Attachment Network or Random Network. Clustering coefficient and Average Path length both are important parameters to find out whether a network is a Small World or not. We found out that the clustering coefficient of our network was 0.3825 but we had no way to calculate the Average Path Length of the network and that is why we were not able to find out whether the network is small world or not. So we tried to plot the log of the degree distribution and based on the plot we tried to understand how the network actually is.

The degree distribution plot and its interpretation both are described in the next section named “Results”.

Community detection: To perform community detection on our network we were confused with which algorithm to use. Based on our past research we found two methods which are important and that can be helpful for our application. These methods are “Louvain Method for Community Detection” and “Girvan-Newman Method for Community Detection”. Both have their individual pros and cons. Girvan-Newman method is a very popular method and has been used by a lot of researchers in the past while Louvain method is not that popular. But Louvain method is based on optimizing modularity. Basically the communities detected by Louvain method are well separated and better defined than Girvan Newman Method. Thus, Louvain method has a high modularity value than Girvan Newman Method and it also has higher speed and more efficiency. To make sure that these theories are correct we did community detection using both the methods and we found out that the Girvan Newman method gave us inly 6 communities and the Louvain method gave us 11 different communities and that is why we restricted our analysis based on only the Louvain method.

After applying the method, we had to detect the communities to which a particular web technology belongs. Because Free Code Camp is a web technology

platform we decided a few popular web technologies for which we wanted to detect specific communities. These web technologies were CSS, JAVA, JavaScript, Ruby, PHP, HTML, Python, AJAX, Laravel and XML. After selecting these web technologies and then used these as words in a word search algorithm to find out which is the community where these web technologies are discussed the most.

The results of this Community Detection method are described in the next section named "Results".

Link Prediction:

We wanted to predict possible future links between the users in the Free Code Camp network. The motivation for doing this was to understand that if a particular user puts a message on the Forum then which user can possibly answer his question which will form a link between the two users. We wished to use user attributes and characteristics as predictors for link prediction but keeping in mind the privacy measures the data did not have such user attributes and that is why we decided to do link prediction using each node's centrality values that we had calculated earlier. Thus, we used degree, closeness, betweenness, clustering and eigenvector centrality as predictors for link prediction. We also found out the individual importance of each of these predictors in deciding the link between the users. To perform link prediction we used Random Forest algorithm. The data was divided into train and test set using 10 fold cross validation.

The results of Link Prediction are described in the next section named "Results".

7) Results

Centrality Measure Analysis: Based on the results of centrality measure analysis we found out the top 5 important nodes for each centrality measure. These nodes with their respective centrality measure value can be seen in the tables below

Degree centrality:

Nodes	Degree
Rphares	0.362595
Abhishekp	0.358779
Anthonygallina1	0.274809
Apottr	0.206107
CEREBR4L	0.179389

Betweenness centrality:

Nodes	Betweenness
Abhishekp	0.240410
Rphares	0.221833
Anthonygallina1	0.118653
Terakilobyte	0.111836
apottr	0.095906

Closeness centrality:

Nodes	Closeness
Abhishekp	0.555339
Rphares	0.541581
anthonygallina1	0.526175
Iheartkode	0.482855
CEREBR4L	0.482855

Eigenvector centrality:

Nodes	Eigenvector
Rphares	0.267230
Abhishekp	0.265093
Anthonygallina1	0.255497
CEREBR4L	0.214146
revisualize	0.194881

Clustering centrality:

Nodes	Clustering
Rameshsyn	1
Kaelandekker	1
P1xt	1
Karagulamos	1
briangunsel	1

After finding these important nodes we started manually looking at the messages of these users from our dataset. We found out that most of these top nodes like "Abhishekp" are Free Code Camp Forum Moderators who are present to promote online courses based on users' interest.

Network Structure: To understand the structure of the network we plotted the node degree distribution against the frequency of messages. We first took the log to the base 10 of all the nodes and then plotted the graph. Based on the graph we could see that most of the nodes in the network had a very low node degree and only a few nodes had a high degree. By understanding this from the plot we were able to confirm that the Free Code

Camp Network is a Preferential attachment network. The plot of node degree distribution can be seen below

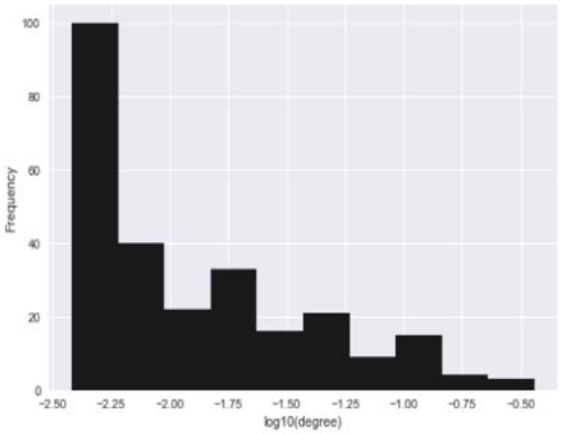

Figure 2: Node Degree Distribution

Community Detection: Like discussed above we performed community detection using 2 different methods but the Girvan Newman Method gave us very few communities which were also not well defined. Thus, we decided to stick to Louvain method for analyzing the results of community detection.

Louvain method gave us total 11 different communities and it was time for us to use word search algorithm to detect the web technologies that we had decided into each of these communities.

After using the word search algorithm we got specific communities for each web technology and these results can be seen in the table below.

Web technologies	Communities they belong to
CSS	Community 6
JAVA	Community 2
JavaScript	Community 5
Ruby	Community 3
PHP	Community 9
HTML	Community 8
Python	Community 1
AJAX	Community 7
Laravel	Community 2
XML	Community 6

We also visualized the communities that we found using Louvain Method which can be seen below in Figure 3

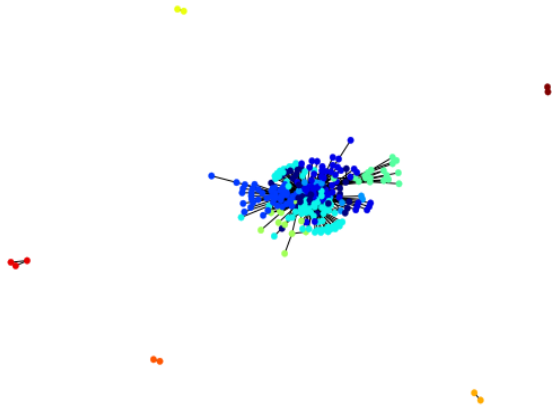

Figure 3: Community detection using Louvain Method

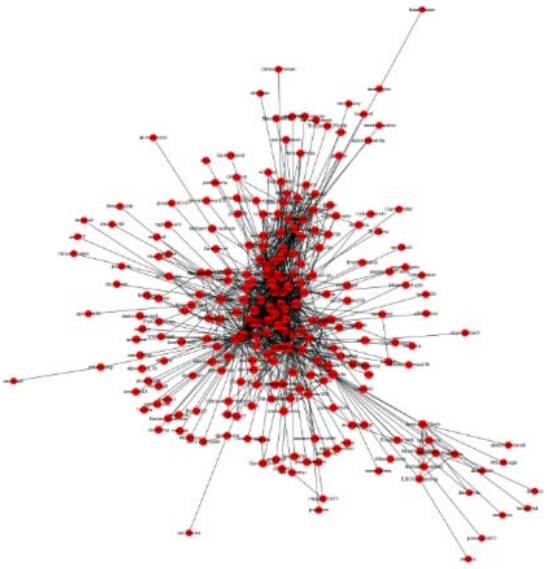

Figure 4: Free Code Camp Network

Link Prediction: Applying 10 fold cross validation to Random Forest Algorithm we were able to find out that the total number of existing instances of links are 1884 and the total number of absent instances of links are 67022. Although it is a bit hard to believe, we got an accuracy of 99% using 10 fold cross validation method. We got a recall of 99% and precision of 80%. We also found out the importance of each of the centrality measure predictors in predicting the link between the users and the importance can be seen in the table below.

Centrality Measures (Predictors)	Importance
Closeness	0.257368
Betweenness	0.235284
Eigenvector	0.232123
Clustering	0.148698
Degree	0.126527

From the importance values we found that while all predictors are important Closeness is the most important of them.

8) Conclusions:

From our analysis and the results that we obtained we were able to answer all our research questions:

Based on our centrality measure analysis we were able to find out the most important and central users of the Free Code camp network. After analyzing the messages of these users in detail we found out that most of these users are Free Code Camp Forum moderators who are there in the network to promote online courses based on people's interests.

From the node degree distribution plot we understood that most of the users have a very less node degree value and only a few users have a high node degree value. Such a distribution of node degree can only be of a Preferential attachment network and thus we were able to conclude that Free Code Camp is a Preferential attachment network.

Based on the results on community detection, we found that although there are a few communities which overlap for more than one web technology, most of the web technologies have a specific community and users of that community only discuss about that particular web technology.

We also found out that closeness centrality was the most important of all the centrality measures to predict the links between two users of the Free Code Camp network.

We also observed that the Free Code Camp network is not very responsive which means that when people ask questions on the forum, not a lot of people answer to their questions.

We learnt a lot of new things from this research. Till now we had just heard about specific communities being present in different social networking sites. Through this research we were actually able to validate these theories on our own.

These findings can actually be used by Free Code Camo to improve their Marketing Strategy and to also give Product recommendations:

Marketing Strategy: Free Code camp is a non-profit organization and their courses are for free. They do not earn any money from users for their courses. But using the results and insights of community detection Free Code Camp can actually find an external source of income. They can put up specific advertisements in different communities and earn money based on that.

Product Recommendations: Free Code Camp has a lot of courses on different Web Technologies and thus they can use these results of Web Technologies to promote their online courses because now they have a specific target of users who would be interested in that particular we technology and promoting online courses to them would increase their chances of getting more users.

9) Limitations

User privacy is a very important topic these days. And thus we understand why user attributes are not a part of our dataset. But having these attributes would have increased the scope of analysis for our research.

Also the data size is very large and thus working with such a huge dataset and building a network from it is more of a challenge for us.

10) References

[1].Bedi, P., & Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(3), 115-135. https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1178

[2] Amtsha09, https://github.com/amtsha09/Facebook-Community-Detection-and-Link-Prediction

[3] William Lyon, (26 June, 2016), https://www.lyonwj.com/2016/06/26/graph-of-thrones-neo4j-social-network-analysis/