

# Community Formation for Covid-19 Warriors using Android Application Covid Helper

Sunita Tripathi

Center of Computer Education, IPS  
University of Allahabad  
Prayagraj, India

Rajiv Ranjan Tewari

JK Institute of Applied Physics and  
Technology  
Uviversity of Allahabad  
Prayagraj, India

Rahul Kumar Mishra

Center of Computer Education, IPS  
University of Allahabad  
Prayagraj, India

**Abstract** - COVID-19 has affected the world in innumerable ways. And in such dark times hope has come to us in the form of corona warriors. And they are present not only in the form of identified medical/health workers. All the people who have volunteered to give assistance of any kind are as much of the warrior as any of the aforementioned people. The world is grateful to the corona warriors for battling it for us. We aim to give the volunteers a better chance at being able to dispense their services using our approach and providing them a chance to find people of the corona warrior community and also recommend them a community to be a part of and get a synergetic result for their efforts.

**Keywords** - community detection, humanity index, COVID-19, corona, corona warrior, android app, recommender system.

## I. INTRODUCTION

Though corona needs no introduction, we would proceed with it nonetheless. Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. You can be infected by breathing in the virus if you are within close proximity of someone who has COVID-19, or by touching a contaminated surface and then your eyes, nose or mouth. We are trying to bring some hope of help with technology playing a small role. Section II discusses what communities are, the need for this solution and implications, and we then propose our approach. Section III discusses the algorithm and functionalities of the Android app. Section IV discusses our algorithm of using the app, creating a data set, creating a new parameter and implementing it in community detection algorithms experimental analysis and then identifying communities for corona warriors. Section V discusses the conclusion wherein we discuss the areas it can be implemented.

## II. STUDY OF THE APPLICATION

### A. Motivation

Our application COVID HELPER application was initially designed for helping people in distress in corona times. Our app would help a person, a user of our application, who is in distress, to be able to send out help needed alert and another user, who has registered himself to be a corona warrior to be alerted for such requests. Now this setup was helpful by the users of both the type but by observation and general understanding we came to a conclusion that if all the warriors worked together the effect would be synergetic. So we aided the application with a community detection algorithm.

### B. Introduction to communities

Community is a graph with more than two nodes or a part of a graph and where every edge between any two nodes shows a relationship based on the properties of the nodes and if the relationship changes that affects node on other end or may start belonging to another community and this dynamic behavior represents flexibility of a community. The formation of a community is done by applying numerous methods and using different parameters. Each method has its own applicability hence different outcomes. So each requirement requires analysis for deciding the algorithm to be used. Also, the parameter or feature selection is also important as it will be responsible in the shaping up of all the communities being formed.

## III. WORKING OF APPLICATION

The application Covid Helper needs a user to register for availing the services of the application. A verified user can login. Other than the admin or super user there are mainly two types of users of the application. The first user is the seeker. Seeker is the one who needs help. The other user is the person who has registered himself as a helper or as we are referring them, corona warriors. As soon as a user logs in their location is saved. Location plays an important role as this would be a factor based on which alerts would be sent to the helpers on their mobile devices and emails. One complete interaction of a seeker and a helper is logged into the data base.

### A. The Dataset

In our approach we are also creating our own data set with our specified parameters. Different data set, according to the requirements can be created with Covid Helper app. After changing parameters like the geographical distance we can have a finer or coarser data set, as the need be. Adding a few parameters like time to our community detection algorithm can give us better communities. So we present this app and our algorithm as highly flexible in their build to be able to alter parameter values as well as add new parameters to get a community of desired specifications.

### B. Algorithm of the App

In this section we are discussing the algorithm of our application. The application of the result of this application to find out the warrior communities would be presented in the next section.

- 1. Register user - Seeker
  - 1.1 Verify email
  - 1.2 Login user
    - Seeker logs in
  - 1.3 Capture current location and update in database
  - 1.4 Want help
    - Create help request, specify title and description and submit.
  - 1.5 According to location find all users in a radius of 20 km (our specification for local area – can be increased or decreased to suit our requirements).
  - 1.6 Assign this request for each user and send an email to notify.
- 2. Register user - Helper
  - 2.1 Verify email
  - 2.2 Login user
    - Helper log in
  - 2.3 Go to requests tab
  - 2.4 View all requests with title and description and select that you want to help
  - 2.5 To help access seeker's contact details
  - 2.6 Contact the seeker and help.
- 3. Help seeker mark request as complete or delete it.
  - 3.1 Ask for registered email id of helper
  - 3.2 Key in details and mark as complete
  - 3.3 Save data to network table

The following screenshots of the application can be seen for reference:

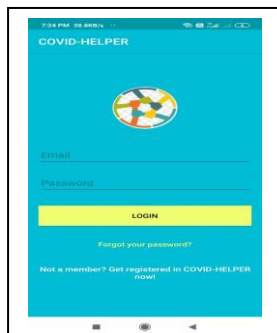


Fig. 1. The login screen

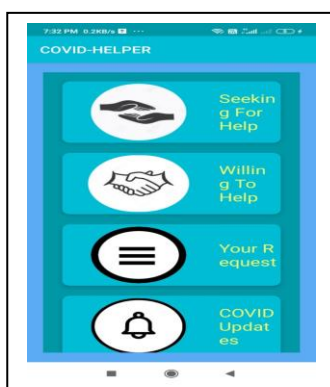


Fig. 2. The panel to raise or serve a request

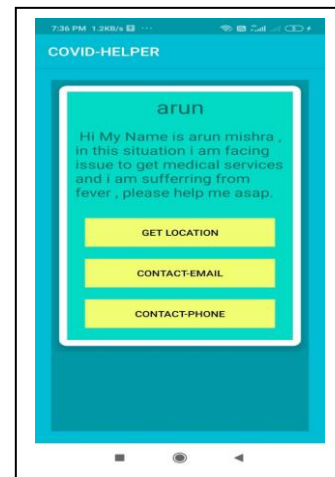


Fig. 3. The helper panel displaying help request

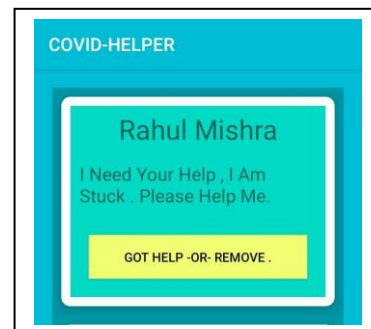


Fig. 4. The seeker panel reporting a complete request or canceling it

### C. New parameter for finding and assigning communities

The task of finding communities has been a tedious one considering the plethora of parameters, thresholds, and constraint which come into picture while doing so. Thus finding a new community detection parameter was benefitted using our application Covid Helper as it could also be later modified as and when needed.

For creating communities we take a snapshot instance of the collected data set. This approach would be our static community detection method.

Our algorithm introduces a new parameter which is relevant in this scenario. The parameter is "humanity index". This is a parameter assigned to each corona helper. This index is calculated for each corona helper by finding the ratio of the number of requests served to the total number of requests raised altogether.

In the next section we would be discussing the algorithm of calculating the "humanity index".

### D. Formula to calculate Humanity Index

Each corona helper registered in the system is referred to as  $CH_i$  where  $i = 1, \dots, n$  and  $n$  is the total number of

helpers registered till that time. We must keep this in mind that this approach has been designed for static community detection where the number of people involved in forming a community is fixed and their number and links are not changing over the time.

For each  $CH_i$ , the “humanity index”  $HI_i$  is calculated as

$$HI_i = \frac{\text{Total requests served}}{\text{Total requests recieved in the application}}$$

Algorithm to detect communities

To implement our algorithm the pre requisites have been discussed and they would be as follows:

#### E. Running the application

To be able to create a data set of a good size, we firstly need to run the application for a certain amount of time. This would ensure that we have enough helper and seeker registrations and also a workable number of requests have been raised and fulfilled.

#### F. Data Processing

The following steps are involved in processing the data:

- Accessing data stored in the network table and getting its CSV File.
- Cleaning the data and passing to the model

#### G. Community Detection

For each corona helper when we have calculated the humanity index, using any existing community formation algorithms like *k-means* or *Louvain*. We are using k-means for our purpose as we have the distance between a helper and seeker also as a key factor. As well as, two helpers would be able to form a community to work together better if they are at physically closer locations as compared to the ones far away. Using  $HI$  for each  $i$  as the parameter for being in community would be justified by the facts that people who are able to provide their services more number of times would be benefitted by staying close to people.

We also use a threshold of proximity  $P_{ij}$  between  $CH_i$  and  $CH_j$ . Two helpers can be considered for being a part of the same community only if their proximity threshold is less than a specified limit. In our case we have kept it to be 20. Physically it would represent the distance in kilometers between two helpers. The idea of two helpers being in a community would be defeated if two helpers lie too far away physically. Proximity threshold would lose its relevance in a few scenarios when distance would not be a botheration, for example, when a request could also be served online.

#### CONCLUSION

The application can be used for creating data sets and suggesting community options to all the new corona warriors using our app. We have worked with static community structure till now. Our application Covid Helper and our approach can be extended to work for larger geo-spatial data and for dynamic communities. This is the theme for our next work which would be an extension of the presented work.

#### REFERENCES

- [1] Chakraborty, T., Dalmia, A., Mukherjee, A., & Ganguly, N. (2017). Metrics for community analysis: A survey. *ACM Computing Surveys (CSUR)*, 50(4), 1-37.
- [2] Newman, M. E., & Park, J. (2003). Why social networks are different from other types of networks. *Physical review E*, 68(3), 036122.
- [3] Schaeffer, S. E. (2007). Graph clustering. *Computer science review*, 1(1), 27-64.
- [4] Fortunato, S. (2009). Community detection in graphs/Santo Fortunato. *Physics Reports*
- [5] Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Physical review E*, 80(5), 056117..
- [6] Todd Waskiewicz (Air Force Research Laboratory, AFRL/RIEA 525 Brooks Road, Rome, NY 13441-4505), “Friend of a Friend Influence in Terrorist Social Networks”
- [7] Yang, J., McAuley, J., & Leskovec, J. (2013, December). Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining* (pp. 1151-1156). IEEE.
- [8] Nguyen, N. P., Dinh, T. N., Nguyen, D. T., & Thai, M. T. (2011, October). Overlapping community structures and their detection on social networks. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing* (pp. 35-40). IEEE.
- [9] Bedi, P., & Sharma, C. (2016). Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3), 115-135.
- [10] Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043), 814-818.
- [11] Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4), 046110.
- [12] Guimerà, R., Sales-Pardo, M., & Amaral, L. A. N. (2007). Module identification in bipartite and directed networks. *Physical Review E*, 76(3), 036102.
- [13] Sawardecker, E. N., Sales-Pardo, M., & Amaral, L. A. N. (2009). Detection of node group membership in networks with group overlap. *The European Physical Journal B*, 67(3), 277-284.
- [14] Fortunato, S. (2009). Community detection in graphs/Santo Fortunato. *Physics Reports*.
- [15] de Solla Price, D. J., Solomonoff, R., & .., and A. Rapoport (1965), *Science* 169, 510. *Bull. Math. Biophys.* 13, 107
- [16] Duan, D., Li, Y., Li, R., & Lu, Z. (2012). Incremental K-clique clustering in dynamic social networks. *Artificial Intelligence Review*, 38(2), 129-147.
- [17] Everett, M. G., & Borgatti, S. P. (1998). Analyzing clique overlap. *Connections*, 21(1), 49-61.
- [18] Lancichinetti, A., Radicchi, F., Ramasco, J. J., & Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one*, 6(4).
- [19] Bansal, S., Bhowmick, S., & Paymal, P. (2011). Fast community detection for dynamic complex networks. In *Complex Networks* (pp. 196-207). Springer, Berlin, Heidelberg.
- [20] Xing, Y., Meng, F., Zhou, Y., Zhou, R., & Wang, Z. (2015). Overlapping Community Detection by Local Community Expansion. *J. Inf. Sci. Eng.*, 31(4), 1213-1232.
- Yang, J., McAuley, J., & Leskovec, J. (2013, December). Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining* (pp. 1151-1156). IEEE.