Evaluation of Fairness on Community Detection Strategies

Christos Gkartzios
chgartzios@cs.uoi.gr
University of Ioannina
Ioannina, Epirus, Greece

Manolis Konstantinos
kmanolis@cs.uoi.gr
University of Ioannina
Ioannina, Epirus, Greece

ABSTRACT

In this report, we study community detection algorithms in terms of fairness based on a protected attribute of the nodes of a network. We study these strategies on networks of different lengths and morphologies. We find that most of the community detection algorithms lead to less fair communities on average than the fairness of the original network. We also propose a post-processing algorithm that we believe can lead to more fair communities than the initial clustering does. We base our algorithm on the hypothesis that moving nodes with low in-community degree from its community, to one that has fewer nodes of the same protected attribute, can lead to more fair communities. We evaluate this procedure on five real-world datasets, and show that it raises the average fairness of the communities of the network.

CCS CONCEPTS

• **Information systems → Data management systems**.

KEYWORDS

datasets, community detection, fairness

ACM Reference Format:
Christos Gkartzios and Manolis Konstantinos. 2018. Evaluation of Fairness on Community Detection Strategies. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Community detection, also called graph partition, help us reveal hidden relations among the nodes of the network. Nodes of the network can easily grouped into sets of nodes such that each set of nodes is densely connected internally. In this way, the network is divided into groups of nodes with dense connections internally, but sparser connections between groups. Based on the attributes of a node (*degree, similarity, reachability*, there are many algorithms that achieve the community detection with different approaches adjacent to the node attribute.

Another important aspect of a social network studies, is the fairness inside a community. Every node in a social network represents a real-world entity with or without extra attributes. Some of that attributes might be sensitive and their study come to conclusions about the community they belong, but also for the network in general. Most community detection algorithms do not take into consideration those attributes, so the partitions they extract might be most of the cases unfair towards a protected attribute.

In our work, we do not only execute community detection algorithms to study the default fairness towards a protected attribute, but we also propose a post processing algorithm in order to correct the fairness in each community and consequently the network.

The rest of this report is structured as follows. In Section 2 we define the problem and present our algorithm, while in Section 3 we report experimental results. Section 4 concludes the report.

2 PROBLEM AND ALGORITHMS

We assume a graph where each node is assigned with a protected attribute. Concretely, a graph $G = (V, E)$ where V is the set of nodes and E is a set of edges where for each node a tuple *(k,v)* is assigned where k is the name of the protected attribute and *v*, where $v \in \{0, 1\}$.

For every graph G, we extract the largest connected component and create the induced subgraph *G'*, where $G' = (V', E')$. Then we perform three different community detection algorithms:

- *Label Propagation Communities*: Finds communities in G using a semi-synchronous label propagation method. Communities consists of vertices with identical labels, while vertices have more neighbors in their community than with other communities.
- *Louvain Communities*: A heuristic algorithm that extracts community structure of a network based on modularity optimization.
- *ComE*[1]: An algorithm that works on node embeddings and optimize the community detection procedure that performs the node embeddings, community embeddings and community detection procedures simultaneously.

We calculate the percentage of each protected attribute that is represented in each community. We introduce the definition of balance as proposed by [2]. Let C be a community set in a graph with subsets $C_1, C_2, ...C_N$ where N is the number of communities. We assume that each vertex V' in a community C is colored either red or blue: $V' \rightarrow \{RED, BLUE\}$ depending on the value of its protected attribute. A natural notion of balance.

Definition 2.1 (Balance). For a community $C_x \in C$ the balance of C_x *is defined as:*

$$balance(C_x) = min\left(\frac{\#RED(C_x)}{\#BLUE(C_x)}, \frac{\#BLUE(C_x)}{\#RED(C_x)}\right) \in [0, 1].$$

Authors' addresses: Christos Gkartzios, chgartzios@cs.uoi.gr, University of Ioannina, Ioannina, Epirus, Greece, 45110; Manolis Konstantinos, kmanolis@cs.uoi.gr, University of Ioannina, Ioannina, Epirus, Greece, 45110.

0730-0301/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

A community with an equal number of red and blue vertices has balance 1 (perfectly balance) while a monochromatic community has balance 0 (fully unbalanced). Balance encapsulates a specific notion of fairness, where a community which is monochromatic (i.e., fully unbalanced) is considered unfair. Given a community set C, we count for each community C_x the number of nodes with the red and blue protected attribute respectively. We also calculate the average fairness of the communities as it is used as metric in comparison with the fixed fairness later. Also we consider F_G, the network fairness of the graph as the $min\left(\frac{\#RED(C)}{\#BLUE(C)}, \frac{\#BLUE(C)}{\#RED(C)}\right)$.

Every node has an in-Community-Degree D as the number of the adjacent nodes from the same community. We assume that nodes with low in-Community-Degree are not strongly connected with the nodes of their community, so it is easier for them to leave the community. Knowing that property, we locate the communities with its fairness lower than the network fairness. For those communities, we specify and list those that are Red attributed dominant L_R and Blue attributed dominant L_B.

We propose a post-processing algorithm 1 that tries to fix the fairness on the communities with minimal changes on the initial community structure. As *input*, we are given the induced subgraph of largest connected components $G' = (V', E')$ where V' is the node set and E' the edge set, Network Fairness set F_N and the list of communities with dominant attribute red L_R & List of communities with dominant attribute blue L_B.

Algorithm 1: Post processing algorithm for fair communities

Input: Network Fairness F_N,List of communities with dominant attribute red L_R, List of communities with dominant attribute blue L_B.
Output: List of updated Communities with better fairness L'_C.

for *Community* C_i *in* L_R **do**
  Compute *inCommunityDegree* of the nodes v_i that belong in C_i
  Create list of candidate to leave nodes with the lowest *inCommunityDegree* L_v
  for *Community* C_j *in* L_B **do**
    for *node* u *in* C_i **do**
      add u to C_j
      remove u from C_i
      compute fairness F_i, F_j of C_i and C_j
      if F_i >= F_N *and* F_j >= F_N **then**
        add C_i, C_j to L'_C
        remove C_i from L_R
        remove C_j from L_B
        break
      end
      if F_i >= F_N **then**
        add C_i to L'_C
        remove C_i from L_R
        break
      end
      if F_j >= F_N **then**
        add C_j to L'_C
        remove C_j from L_B
        break
      end
    end
  end
end
if *len*(L_R) >0 **then**
  for C_i *in* L_R **do**
    add C_i to L'_C
  end
end
if *len*(L_B) >0 **then**
  for C_j *in* L_B **do**
    add C_j to L'_C
  end
end
return L'_C

The idea is to transfer Red nodes from a community with dominance of Red nodes to a community with dominance of Blue nodes and vice versa. For a community $C_x \in L_R$ we try to send a node $v \rightarrow \{Red\}$ to a community $C_y \in L_B$ and vice versa. This procedure will continue until all the communities have $F \geq F_G$ or there are no longer available nodes to be sent to other communities. In the occasion of the either $L_R = \emptyset \parallel L_B = \emptyset$ we assume that there is no available community to receive nodes from the other group so the fairness cannot be fixed. The output of the algorithm is the updated Communities with its new fixed fairness.

3 EVALUATION

The evaluation of the post processing algorithm we propose is based on five different real world networks:

deezer[1] Nodes are Deezer users from European countries and edges are mutual follower relationships between them. The attribute we study is the gender of the users.

facebook[2] Nodes are Facebook users and edges are friendship relationships between them. The attribute we study is the gender of the users.

gplus[3] Nodes are Google+ users and edges are friendship relationships between them. The attribute we study is the gender of the users.

pokec[4] Nodes are Pokec users and edges are friendship relationships between them. The attribute we study is the gender of the users.

twitch[5] Nodes are Twitch users and edges are mutual follower relationships between them. The attribute we study is mature content.

We provide the code used during the experiments at the following link[6]

Table 1: Characteristics of each Network

Network	Nodes	Edges	Directed	Protected Attribute
deezer	28,281	92,752	No	Gender
facebook	4,039	88,234	No	Gender
gplus	107,614	13,673,453	No	Gender
pokec	1632803	30,622,564	No	Gender
twitch	168,114	6,797,557	No	Mature Content

The statistics of the Networks are shown in Table 1.

We used three different community detection strategies. Label Propagation, the Louvain method and ComE, which uses community embeddings. The communities detected with each method for the Networks we used are presented in Table 2.

Table 2: Communities detected with each method

Network	Label Propagation	Louvain	ComE
deezer	1984	84	5
facebook	44	16	5
gplus	165	32	5
pokec	14851	38	
twitch	79	19	5

We compute the fairness, based on the protected attribute,that is observed in the whole Network, presented in Table 3. We find that *gplus* and *pokec* are almost perfectly balanced.

[1]http://snap.stanford.edu/data/feather-deezer-social.html
[2]http://snap.stanford.edu/data/ego-Facebook.html
[3]http://snap.stanford.edu/data/ego-Gplus.html
[4]https://snap.stanford.edu/data/soc-Pokec.html
[5]https://snap.stanford.edu/data/twitch_gamers.html
[6]https://github.com/kostaspm/Fair-Community-Detection

Table 3: Network Fairness

Network	Network Fairness
deezer	0.79
facebook	0.61
gplus	0.92
pokec	0.97
twitch	0.88

In Table 4 we present the average fairness that we observed in the resulting communities. We find that for every Network the most balanced communities are created with *ComE*, while *Label Propagation* leads to the most unbalanced communities.

Table 4: Average Community Fairness with each method

Network	Label Propagation	Louvain	ComE
deezer	0.47	0.62	0.77
facebook	0.44	0.61	0.63
gplus	0.36	0.49	0.69
pokec	0.52	0.67	
twitch	0.28	0.54	0.65

In Table 5 we present the average community fairness that is observed in the networks, after the proposed post processing strategy is used. We find that, with the exception of the communities created with *ComE*, our strategy leads to more balanced communities.

We believe that due to the fact that *ComE* uses community and node embeddings to optimize community detection, the resulting communities are more similar to the original network than the other methods. This can be observed in Table 5

Table 5: Average Community Fairness with each method after post processing

Network	Label Propagation	Louvain	ComE
deezer	0.68	0.70	0.77
facebook	0.53	0.61	0.63
gplus	0.36	0.49	0.69
pokec	0.73	0.85	
twitch	0.45	0.74	0.81

4 CONCLUSIONS

In this report, we studied the problem of fairness on community detection strategies. Furthermore we proposed a post processing algorithm that can lead to more fair communities. We tested our algorithm on communities, created with three different community detection strategies, on five different networks.

We found that *ComE*, the community detection strategy that uses community embeddings leads to communities that have average fairness near to the global fairness of the network. On the contrary the other two strategies lead to communities that have worse average fairness than the corresponding global. Our post processing algorithm leads to fairer communities when used on communities created with the *Label Propagation* and *Louvain* strategies while for *ComE* the average fairness stays mostly the same.

REFERENCES

[1] Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. 2017. Learning Community Embedding with Community Detection and Node Embedding on Graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 377–386. https://doi.org/10.1145/3132847.3132925

[2] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair Clustering Through Fairlets. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/978fce5bcc4eccc88ad48ce3914124a2-Paper.pdf