Community and Influencer Detection by Affiliated Graph Model and TextM

Ajinkya Wasnik
Indian Institute of Technology, Delhi
eet182569@iitd.ac.in

Jeeban Sethi
Indian Institute of Technology, Delhi
eet182570@iitd.ac.in

1 Abstract

Network communities represent basic structures for understanding the organization of real-world networks. A community (also re- ferred to as a module or a cluster) is typically thought of as a group of nodes with more connections amongst its members than between its members and the remainder of the network. Communities in networks also overlap as nodes belong to multiple clusters at once. Due to the difficulties in evaluating the detected communities and the lack of scalable algorithms, the task of overlapping community detection in large networks largely remains an open problem. In this paper we've used AGMfit , an overlapping community detection method that scales to large networks of millions of nodes and edges. We took on a novel observation that overlaps between communities are densely connected and took that forward. This is in sharp contrast with past community detection methods which implicitly assume that overlaps between communities are sparsely connected and thus cannot properly extract overlapping communities in networks. In this paper, we develop a influencing nodes detection algorithm TextM that can detect densely overlapping, hierarchically nested as well as non- overlapping communities in massive networks and as well then visualise the graph to get the densely overlap argument clear from the visualisation.

2 Introduction

Communities are often interpreted as organizational units in social networks, functional units in biochemical networks, ecological niches in food web networks, or scientific disciplines in citation and collaboration networks. Even though methods for identifying overlapping as well as hierarchically-nested communities in networks have been considered in the past, identifying meaningful communities in large networks has proven to be a challenging task. Most methods have trouble scaling to large networks, and the lack of reliable ground truth makes evaluation of detected communities surprisingly difficult. Thus, while networks have been extensively- studied, and the existence and properties of communities in small networks is by now well-understood, it is still not clear how to identify realistic overlapping communities in very large networks that are increasingly common. Nodes in networks organize into densely linked groups that are commonly referred to as network communities, clusters or modules. There are many reasons why networks organize into communities. For example, in social networks communities emerge since society organizes into groups, families, friendship circles, villages and associations. In the graph of the World Wide Web topically related pages link more densely among themselves and communities naturally emerge. And in biological networks communities emerge since proteins belonging to a common functional module are more likely to interact with each other. Here we explore the community structure of a number of networks from many domains. We distinguish between structural and functional definitions of communities. Communities are often structurally defined as sets of nodes with many connections among the members of the set and few connections to the rest of the network. Communities can also be defined functionally based on the function or role of its members. For example, functional communities may correspond to social groups in social networks, scientific disciplines or research groups in scientific collaboration networks, and biological modules in protein-protein interaction networks. The premise of community detection is that these functional communities share some degree share some common structural signature, which allows us to extract them from the network structure.

3 Related work

The observation that community overlaps are more densely connected than the non-overlapping parts. Community-Affiliation Graph Model that explains the emergence of dense community overlaps and accurately models network community structure. Model-based community detection method that detects overlapping, non-overlapping, as well as nested communities in networks. Detecting communities of densely connected sets of nodes is an extensively researched area with a plethora of different algorithms and heuristics. For example, separate methods have been proposed for detecting communities in undirected networks that are disjoint, overlapping, or hierarchically nested. On the other hand, detection of 2-mode communities has been much less researched. An excep-

tion here is Trawling, which is a method for extracting 2-mode communities in large directed networks. The critical difference with our work here is that Trawling only identifies complete bipartite subgraphs of a given directed network. In contrast, our method is able to identify cohesive as well as bipartite communities in directed as well as undirected networks. Moreover community detection along with overlapping has been done by AGM algo whereas some other ways of looking at overlapping and knowing the influencing nodes is quite unexplored which we've thought of by other way.

4 Model

AGMfit detects overlapping communities (dense groups of nodes) in networks. AGMfit provides a fast and efficient algorithm to find communities by fitting the Affilated Graph Model to a large network. A community is a set of nodes that are densely connected each other. In many realworld networks, communities tend to overlap as nodes can belong to many communities or groups.The Affiliation Graph Model (AGM) is a generative model that produces a network from community affilation. AGMfit is a fast and scalable algorithm to detect overlapping communities from a given graph by fitting the AGM to the graph.
When a network is given, AGM can measure the likelihood of a community affiliation graph, and we can find the most likely community affiliation by fitting the AGM to the given network. If a user specifies the number of communities that the user want to detect, AGMfit finds the corresponding number of communities. If a user does not know how many communities would exist, which is more realistic, AGMfit automatically estimates the number of communities in the graph. User also can control the probability of edges between the nodes that do not share any communities.

TextM considers the no. of occurences of the node in the file and accordingly appends count with the node name in a new csv file.

5 Data Usage

Communities in National Collegiate Athletic Association(NCAA) football teams network are used.The edgelist(fig1) and nodelist(fig2) is shown in the figures section as an input into the AGMfit and TextM. Any graph can be given ass an input to the AGMfit and later its output to TextM.

6 Working

Step 1: Both the Files namely .labels(fig 2) and .edgelist(fig1) files are required to be fed to the AGMfit with the help of following command
$agmfitmain - i : football.edgelist - l : football.labels - c : 12 - e : 0.1$
cmd: Detects 12 communities of universities (which correspond to NCAA (Refer fig3) conferences) from the network of NCAA football teams
here
Parameters:
-o: Output file name prefix (default:")
-i: Input edgelist file name. DEMO: AGM with 2 communities
-l: Input file name for node names (Node ID, Node label)
-s: Random seed for AGM
-c: Number of communities (0: determine it by AGM)
-e: Edge probability between the nodes that do not share any community: set it to be 1 / N2

NOTE: The best result with 5 trials with setting the edge probability of two nodes sharing no communities to be 0.1. Circular regions denote detected communities and node colors represent NCAA conferences.)

Step 2: It generates cmtyvv.txt and graph.gexf.(refer fig 4 & fig 5)
Here rows in the cmtyvv.txt file indicates number of communities and each column in a row indicates community name.

Step 3:We built a program TextM that scans through the file and extract the information about number of community count as described before and stores it in a simple csv file.(refer fig 6)

Step 4:Final visualisations by gephi.(remaining figures)

7 Result & Findings

We run that AGMfit on the graph over which we want to find the clusters and overlapping after which we get an output file which we use to do text based cluster count. We generate a new csv file where we have all nodes name and their count of clusters in which they are part of. We do counting by traversing the text with the help of our TextMine (code for that purpose).
•Now the file we have generated can be fed to any visualisation tool, we here preferably used Gephi for visualisation where we have ranked the nodes according to size and also shaded the nodes according to the count value we have attached to the graph file.
Now from visualisation we tend to find that:
•In the previous work done in this branch researchers tend to find the clusters and then the overlaps between those clusters whereas, we have tried to work upon different method wherein, we have the count of each of the nodes membership in cluster (using TextMine) and then we visualised that graph with the ranking on the basis of that count.
•When we look in the graph with all such properties then if we find that the nodes with more count are sharing more edges with those all node with similar high count nodes, they are those same ones who are in the overlapping.
•So if some churn has to be done on that we can work

from those specific nodes with high counts as theyll turn out as more connected.

•The results hence also show that the Nodes who belong to same cluster are usually sparsely connected rather those in the overlaps are more densely connected overturning normal point of view.

•Here above notion is inferred from the fact that the nodes with high counts share edges more with each other which also shows that nodes from different cluster are connected to each other in the overlap region.

•Overlaps in this novel method are quite virtual as we have approached things with different way rather than the way earlier all approaches were done.

8 Future Work

Usually we are working upon the AGMfit and extending the AGM model by working upon its output. But theres a scope of binding all the process we do to achieve the end result , which here we are doing in 3 steps could be done in single step as the result generation time would definitely come down and more observations henceforth can be achieved.

9 Conclusion

In this paper we developed a novel Influencers detection method (TextM) that discovers the overlapping community structure of real-world networks with use of AGMfit alongside for community detection. We observed that the overlaps of communities are more densely connected than the non-overlapping parts of communities, which is in sharp contrast to assumptions made by present community detection models and methods and is in line with the work of Jaewon Yang[1] and Jure Leskovec[1]. We note that the finding that community overlaps are denser than communities themselves nicely extends the notion of homophily in networks [3]. The strength of weak ties and small-world models[3] lead to the idea that homophily in networks operates in small pockets where inside the pocket nodes link strongly among themselves, and weakly to other pockets. Thus, network communities should not be thought of as a set of clusters but rather as a set of overlapping nodes where the density of the edges increases with the number of nodes in that overlap. Our work has several important implications: First, our analysis sheds light on the organization of complex networks and provides new directions for research on Influencer node detection. Second, overlapping is just not a single way to look at clusters rather the new way of finding influencing nodes which have rather greater range of reach in the network could be found. And last, the AGM alongwith TextM can be a realistic benchmark network on which new community detection algorithms can be developed and evaluated ahead by working along TextM and hence visualise to get insight into the data in a altogether new perspective.

10 Figures for representation

2	1
4	3
5	1
6	4
6	5
7	3
8	7
9	8
10	1
10	5
10	9
11	6
12	4
12	6
12	11
14	3
14	13
15	3
15	13
16	3
16	14
16	15
17	1
17	5
17	10
18	13
18	17
19	13
20	19
21	18
22	8
22	9
22	21
23	8
23	9
23	10
23	22
24	1

fig1: edgelist of the NCAA data we've used.

1	BrighamYoung	0	7	
2	FloridaState	8	0	
3	Iowa	5	2	
4	KansasState	7	3	
5	NewMexico	0	7	
6	TexasTech	7	3	
7	PennState	5	2	
8	SouthernCalifornia	0		8
9	ArizonaState	0	8	
10	SanDiegoState	0	7	
11	Baylor	7	3	
12	NorthTexas	0	10	
13	NorthernIllinois		2	6
14	Northwestern	5	2	
15	WesternMichigan	2	6	
16	Wisconsin	5	2	
17	Wyoming	0	7	
18	Auburn	4	9	
19	Akron	2	6	
20	VirginiaTech	6	1	
21	Alabama	4	9	
22	UCLA	0	8	
23	Arizona	0	8	
24	Utah	0	7	
25	ArkansasState	0	10	
26	NorthCarolinaState	8		0
27	BallState	2	6	
28	Florida	4	9	
29	BoiseState	0	11	
30	BostonCollege	6	1	
31	WestVirginia	6	1	
32	BowlingGreenState	2		6
33	Michigan	5	2	
34	Virginia	8	0	
35	Buffalo	2	6	
36	Syracuse	6	1	
37	CentralFlorida	2	5	
38	GeorgiaTech	8	0	
39	CentralMichigan	2	6	

fig2: nodelist of the NCAA data we've used.

```
ajinkya@ajinkya-HP-Pavilion-g6-Notebook-PC:~/Desktop/agm-package/agmfit$
 ./agmfitmain -i:football.edgelist -l:football.labels -e:0.1
cpm. build: 23:12:50, Nov 13 2018. Time: 12:56:52 [Oct 31 2018]
============================================================
Output file name prefix (-o:)=
Input edgelist file name. DEMO: AGM with 2 communities (-i:)=football.ed
gelist
Input file name for node names (Node ID, Node label)  (-l:)=football.lab
els
Random seed for AGM (-s:)=0
Edge probability between the nodes that do not share any community (defa
ult (0.0): set it to be 1 / N^2) (-e:)=0.1
Number of communities (0: determine it by AGM) (-c:)=0
Graph: 115 Nodes 613 Edges
conductance computation completed [0.00s]
6 communities needed to fill randomly
initial likelihood = -970.463906
299999 iterations completed [0.44]
```

fig3: *giving input to agmfit to get no. of clusters from terminal*

```
WakeForest      FloridaState    GeorgiaTech     Virginia
SouthernMethodist       TexasElPaso     SanJoseState    Nevada
TexasElPaso     TexasChristian  Rice    FresnoState     Nevada
```

fig4: *output text file generated by command*

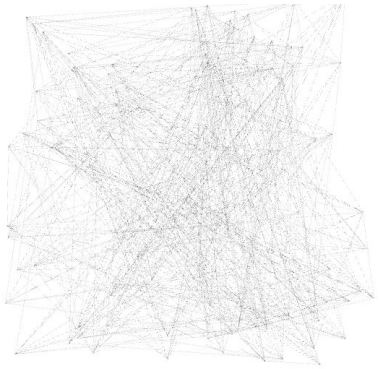

fig5: *output .gexf file*

id	count
0	0
1	1
2	0
3	0
4	1
5	1
6	1
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	1
21	1
22	1
23	0
24	0
25	1
26	0
27	1
28	0
29	0
30	0
31	0

fig6: *output generated by TextM*

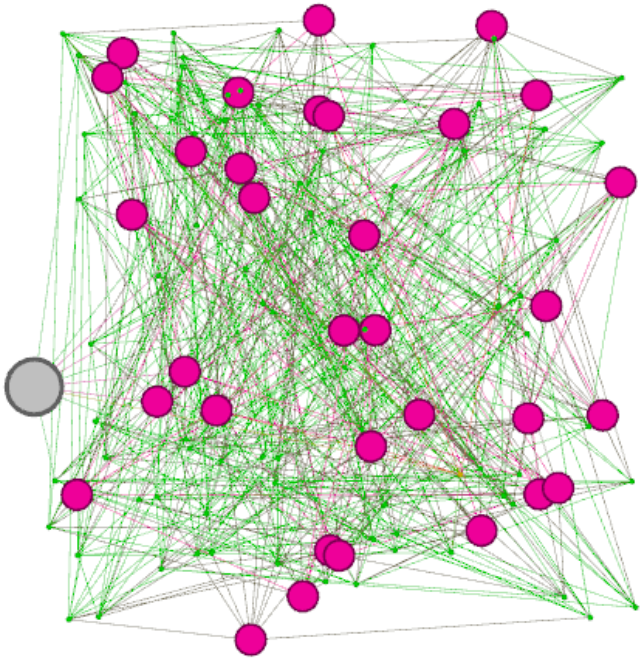

fig7: *output generated by TextM run on gephi* both edges and nodes are here weighted according to the count of their cluster membership.

References

[1] Jaewon Yang, Jure Leskovec. 2012. *Community-Affiliation Graph Model for Overlapping Network Community Detection*

A[2] Jaewon Yang Jure Leskovec 2012. *Overlapping Community Detection at Scale: A Non-negative Matrix Factorization Approach.*

[3] M. S. Granovetter. 1973. *The strength of weak ties. American Journal of Sociology*.

[4] Y Halberstam, Brian Knight. 2014. Alternation. *Homophily, group size, and the diffusion of political information in social networks:evidence from twitter*