Local Community Identification in Social Networks

Jiyang Chen, Osmar R. Zaïane and Randy Goebel
Department of Computing Science
University of Alberta, Canada T6G 2E8
{jiyang, zaiane, goebel}@cs.ualberta.ca

Abstract

There has been much recent research on identifying global community structure in networks. However, most existing approaches require complete information of the graph in question, which is impractical for some networks, e.g. the World Wide Web (WWW). Algorithms for local community detection have been proposed but their results usually contain many outliers. In this paper, we propose a new measure of local community structure, coupled with a two-phase algorithm that extracts all possible candidates first, and then optimizes the community hierarchy. We compare our results with previous methods on real world networks such as the co-purchase network from Amazon. Experimental results verify the feasibility and effectiveness of our approach.

1. Introduction

Many datasets can be represented as networks composed of vertices and edges, including the World Wide Web (WWW), organization structures [1], academic collaboration records [2], [3] and even political elections [4]. A community in the network can be seen as a subgraph such that the density of edges within the subgraph is greater than the density of edges between its nodes and nodes outside it [5]. The ability to identify communities could be of significant practical importance. For example, groups of web pages that link to more web pages in the community than to pages outside might correspond to sets of web pages on related topics; this can enable search engines and portals to increase the precision and recall of search results by focusing on narrow but topically-related subsets of the web [6].

The problem of finding communities in social networks has been studied for decades. Recently, several quality metrics for community structure have been proposed [7], [8], [9]. However, most of those approaches require knowledge of the entire graph structure. This constraint is problematic for networks which are either too large or too dynamic to know completely, e.g., the WWW. In spite of these limitations, finding local community structure would still be useful, albeit constrained by the small volume of accessible information about the network in question. For example, we might like to quantify the local communities of either a particular webpage given its link structure in the WWW, or a person given his social network in Facebook.

Several techniques [10], [11], [12], [13] have been proposed to identify local community structure given limited information about network. However, parameters that are hard to obtain are usually required. Moreover, communities discovered by these algorithms include many outliers and thus suffer from low accuracy. In this paper, we propose a new metric, which we call L, to evaluate the local community structure for networks in which we lack global information. We then define a two-phase algorithm based on L to find the local community of given starting nodes, and compare our algorithm's performance with previous methods on several real world networks. In contrast to existing approaches, our metric L is robust against outliers. The proposed algorithm not only discovers local communities without an arbitrary threshold, but also determines whether a local community exists or not for certain nodes.

The rest of the paper is organized as follows. Section 2 defines the problem and reviews existing solutions. We describe our approach in Section 3 and report experimental results in Section 4, followed by conclusions in Section 5.

2. Preliminaries

Here we first define the problem of finding local communities in a network, then focus our efforts on reviewing existing algorithms.

2.1. Problem Definition

As mentioned in the introduction, local communities are densely-connected node sets that are discovered and evaluated based only on local information. Suppose that in an undirected network G (directed networks are typically first transformed to undirected ones), we start with perfect knowledge of the connectivity of some set of nodes, i.e., the known local portion of the graph, which we denote as D. This necessarily implies that we also have limited information for another shell node set S, which contains nodes that are adjacent to nodes in D but do not belong to D (note “limited” means that the complete connectivity information of any node in S is unknown). In such circumstances, the only way to gain additional information about

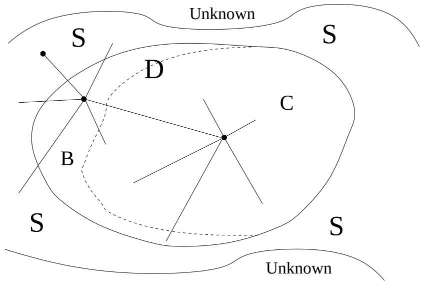

Figure 1. Local Community Definition

the network G is to visit some neighbour nodes s_i of D (where $s_i \in S$) and obtain a list of adjacencies of s_i. As a result, s_i is removed from S and becomes a member of D while additional nodes may be added to S as neighbours of s_i. This typical one-node-at-one-step discovery process for local community detection is analogous to the method that is used by web crawling systems to explore the WWW. Furthermore, we define two subsets of D: the core node set C, where any node $c_i \in C$ have no outward links, i.e., all neighbours of c_i belong to D; and the boundary node set B, where any node $b_i \in B$ has at least one neighbour in S. Figure 1 shows node sets D, S, C and B in a network. Similar problem settings can be found in [10], [11], [12], [13], however, the metrics used to discover and evaluate the local community are different, as explained in the next section.

2.2. Previous Approaches

Clauset has proposed the local modularity measure R [12] for the local community detection problem. R focuses on the boundary node set B to evaluate the quality of the discovered local community D.

$$R = \frac{B_{in\_edge}}{B_{out\_edge} + B_{in\_edge}} \tag{1}$$

where B_{in_edge} is the number of edges that connect boundary nodes and other nodes in D, while B_{out_edge} is the number of edges that connect boundary nodes and nodes in S. In other words, R measures the fraction of those “inside-community” edges in all edges with one or more endpoints in B. Therefore, the community D is measured by the ”sharpness” of the boundary given by B.

Similarly, Luo et al. later proposed the measure called modularity M [13] for local community evaluation. Instead of measuring the internal edge fraction of boundary nodes, they directly compare the ratio of internal and external edges.

$$M = \frac{number\ of\ internal\ edges}{number\ of\ external\ edges} \tag{2}$$

where “internal” means two endpoints are both in D and “external” means only one of them belongs to D. An arbitrary threshold is set for M so that only node sets that have $M \geq 1$ are considered to be qualified local communities. M is strongly related to R. Consider a candidate node set D where every node in D has external neighbours, thus we have $|C| = 0$ and $B = D$, which means $B_{in\_edge} = internal\ edges$ and $B_{out\_edge} = external\ edges$. The threshold $M \geq 1$ is equivalent to $R \geq 0.5$. It is straightforward to identify local communities with the R or M metric. Given a starting set D, in every step we merge the node into D from S which most increases the metric score, and then update D, B and S. This process is repeated until all nodes in S give negative value if merged in D, i.e., $\Delta R < 0$ or $\Delta M < 0$.

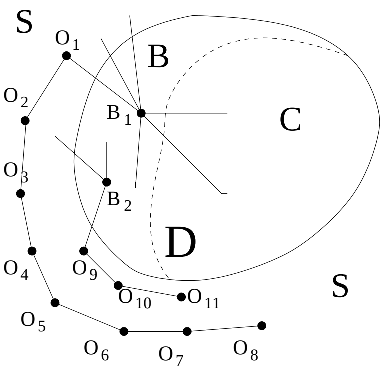

Figure 2. Problem of Previous Approaches

Indeed algorithms using these metrics are able to detect interesting communities in complex networks, however, their results usually include many outliers, i.e., the discovered communities have high recall but low accuracy, which reduces the overall community quality. Figure 2 illustrates the problem for R and M. In the figure, we have a local community D, its boundary B and nodes $O_1, ..., O_{11}$, which are outliers since they are barely related to nodes in D. Without loss of generality, let us assume that all nodes in S, except O_1 and O_9, will decrease the metric score if included in D. Now if we try to greedily maximize the metric R or M, all outliers (O_1 to O_8 and O_9 to O_{11}) will be merged into D, one by one. The reason is that every merge of node O_i does not affect the external edge number but will increase the internal edge number by one. Similarly, the algorithm would merge any node into D as long as it connects to the same number of nodes inside and outside the local community node set. Therefore, in addition to actual members, the resulting community would contain many weak-linked outliers, whose number can be huge for some networks, e.g., the WWW.

Bagrow et al. proposed an alternative method to detect local communities [11], which spreads a l-shell outward from the starting node n, where l is the distance from n to all shell nodes. The performance of their approach highly depends on the parameter l and the starting node because the result communities could be very different if the algorithm starts from border nodes instead of cores. The authors later proposed the "outwardness" metric Ω [10] to measure local structure, however, their method lacks an appropriate stopping criteria and thus still relies on arbitrary thresholds.

3. Our Approach

Existing approaches discussed in Section 2 are relatively simple: an effective local community detection method should be simple, not only because the accessible information of the network is restricted to merely a small portion of the whole graph, but also because the only means to incorporate more information about the structure is by expanding the community, by one node at one step. With these limitations in mind, we present our L metric and the local community discovery algorithm.

3.1. The Local Community Metric L

Intuitively, there are two factors one may consider to determine whether a node set in the network is a community or not: 1) high value node relations within the set, and 2) low value relations between inside nodes and the rest of the graph. Therefore, almost all existing metrics directly use the internal and external degrees to represent these two significant factors, and identify local communities by maximizing the former while minimizing the latter. However, their community results might include many outliers and the overall community quality is questionable (See Section 2.2 and Section 4.1 for examples). The important missing aspect in these metrics is the *connection density*, because is not the absolute number of connections that matters in community structure evaluation. For instance, even if there are one million edges within one node set N and no outward links at all, it is not sensible to identify N as a strong community if every node in N connects only one or two neighbours. We therefore propose to measure the community internal relation L_{in} by the average internal degree of nodes in D:

$$L_{in} = \frac{\sum_{i \in D} IK_i}{|D|} \tag{3}$$

where IK_i is the number of edges between node i and nodes in D. Similarly, we measure the community external relation L_{ex} by the average external degree of nodes in B:

$$L_{ex} = \frac{\sum_{j \in B} EK_j}{|B|} \tag{4}$$

where EK_j is the number of connections between node j and nodes in S. Note that L_{ex} only considers boundary nodes instead of the whole community D, i.e., the core nodes are not included since they do not contribute any outward connections. Now we want to maximize L_{in} and minimize L_{ex} at the same time. Fortunately, this can be achieved by maximizing the following ratio:

$$L = \frac{L_{in}}{L_{ex}} \tag{5}$$

Note that it is possible to quantify the density L_{ex} by other means, e.g., by using the average number of connections from the shell nodes to community nodes to measure L_{ex}. However, this method fails for the local community identification problem because the shell set is usually incomplete. For example, while the friend list of user A is available in Facebook, the list of the users that choose A as a friend is hard to obtain.

3.2. Local Community Structure Discovery

Using L to evaluate the community structure, one can identify a local community by greedily maximizing L and stopping when there are no remaining nodes in S that increases L if merged in D. However, this straight-forward method is not robust enough against outliers. Take Figure 2 as an example. Although L_{in} for O_1 would decrease because O_1 only connects to one node in D, the overall L might increase because the denominator L_{ex} decreases as well (O_1 only connects to one node outside D). Therefore, it is still possible to include outlier O_1 in the community. To deal with this problem, we look further into the metric instead of simply maximizing the score in a greedy manner. We note there are three situations in which we have an increasing L score. Assume i is the node in question and L'_{in}, L'_{ex} and L' are corresponding scores if we merge i into D, the three cases that will probably result in $L' > L$ are:

1) $L'_{in} > L_{in}$ and $L'_{ex} < L_{ex}$
2) $L'_{in} < L_{in}$ and $L'_{ex} < L_{ex}$
3) $L'_{in} > L_{in}$ and $L'_{ex} > L_{ex}$

Obviously nodes in the first case belong to the community since they strengthen the internal relation and weaken the external relation. Nodes in the second case, e.g., O_1 in Figure 2, are outliers. They are weakly connected to the community as well as the rest of the graph. Finally, the role of nodes in the third case cannot be decided yet, since they are strongly connected to both the community and the network outside the community. More specifically, when we meet a node i, which falls into this case during the local community discovery process, there are two possibilities. First, node i can be the first node of an enclosing community group that is going to be merged one by one; Second, i connects to many nodes, inside or outside the community, and can be referred to as a "hub." We do not want hubs in the local community. However, it is too early to judge whether the incoming node is a hub or not. Therefore, we temporarily merge nodes in the first and third cases into the community. After all qualified nodes are included, we re-examine each node by removing it from D and check the metric value change of its merge again. Now we only keep nodes in the first case. If node i is a member of an enclosing group, L'_{ex} should decrease because all its neighbours are now in the community as well, while hub nodes would still belong to the third case. Finally, the starting node should still be found in D, otherwise, we believe a local community does not exist. (See Algorithm 1.)

Algorithm 1 Local Community Identification Algorithm

```
Input: A social network $G$ and a start node $n_0$.
Output: A local community with its quality score $L$.
1. Discovery Phase:
  Add $n_0$ to $D$ and $B$, add all $n_0$'s neighbours to $S$.
  do
    for each $n_i \in S$ do
      compute $L'_i$
    end for
    Find $n_i$ with the maximum $L'_i$, breaking ties randomly
    Add $n_i$ to $D$ if it belongs to the first or third case
    Otherwise remove $n_i$ from $S$.
    Update $B$, $S$, $C$, $L$.
  While ($L' > L$)
2. Examination Phase:
  for each $n_i \in D$ do
    Compute $L'_i$, keep $n_i$ only when it is the first case
  end for
3. If $n_0 \in D$, return $D$, otherwise there is no local
community for $n_0$.
```

The computation of each L'_i can be done quickly using the following expression.

$$L'_i = \frac{\frac{Ind+2*Ind_i}{|D|+1}}{\frac{Outd-Ind_i+Outd_i}{|B'|}} \qquad (6)$$

where Ind and $Outd$ are the number of within and outward edges of D before merging i, and should be updated after each merge; Ind_i and $Outd_i$ are the number of edges from node i to the community and the rest of network; B' is the new boundary set after examining all i's neighbour in D. In the discovery phase, L'_i need to be recomputed for every node in S to determine the one with the maximum ΔL, thus the complexity of the algorithm is $O(kd|S|)$, where k is the number of nodes in the D, and d is the mean degree of the graph. However, in networks for which local community algorithms are applied, e.g., the WWW, and where adding a new node to D requires the algorithm to obtain the link structure, the running time will be dominated by this time-consuming network information retrieval. Therefore, for real world problems the running time of our algorithm is linear in the size of the local community, i.e., O(k). Note that in Algorithm 1 we begin with only one node n_0, but the same process could apply for multiple nodes to allow a larger starting D, C, B and S.

4. Experiment Results

Since the ground truth of local communities in a large and dynamic network is hard to define, previous research usually apply their algorithms on real networks and analyze the results based on common sense, e.g., visualizing the community structure or manually evaluating the relationship between disclosed entities [11], [12], [13]. Here we adapt a different method to evaluate the discovered local communities. We provide a social network with absolute community ground truth to the algorithm, but limit its access to network information to local nodes only. The only way for the algorithm to obtain more network knowledge is to expand the community, one node at a time. Therefore, we can evaluate the result by its accuracy, while satisfying limitations for local community identification. Based on our observations, the greedy algorithm based on metric R [12] (we refer to it as algorithm R) outperforms all other methods for local community detection. Furthermore, similar to our approach, R does not require any initial parameters while other methods [10], [11], [13] rely on parameter selection. Therefore, in this section we compare the results of our algorithm and algorithm R on different real world networks to show that our metric L is an improvement for local community detection.

4.1. The NCAA Football Network

The first dataset we examine is the schedule for 787 games of the 2006 National Collegiate Athletic Association (NCAA) Football Bowl Subdivision (also known as Division 1-A) [9]. In the NCAA network, there are 115 universities divided into 11 conferences[1]. In addition, there are four independent schools, namely Navy, Army, Notre Dame and Temple, as well as 61 schools from lower divisions. Each school in a conference plays more often with schools in the same conference than schools outside. Independent schools do not belong to any conference and play with teams in all conferences, while lower division teams play only few games. In our network vocabulary, this network contains 180 vertices (115 nodes as 11 communities, 4 hubs and 61 outliers), connected by 787 edges.

We provide this network as input to our algorithm and algorithm R. Every node in a community, which represents one of the 115 schools in an official conference, has been taken as the start node for both algorithms. Based on the ground truth posted online, the *precision*, *recall* and *f-measure* score, which is defined as the harmonic mean of

[1] The ground truth of communities (conferences) can be found at http://sports.espn.go.com/ncf/standings?stat=index&year=2006

2006 NCAA League		Algorithm Results						
		Greedy Algorithm R using metric R			Our Algorithm using metric L			
Conference	Size	Precision	Recall	F-measure	No Community	Precision	Recall	F-measure
Mountain West	9	0.505	0.728	0.588	0 node	0.944	1	0.963
Mid-American	12	0.392	0.570	0.463	1 nodes	0.923	1	0.96
Southeastern	12	0.331	0.541	0.410	3 nodes	1	1	1
Sun Belt	8	0.544	0.891	0.675	3 nodes	1	1	1
Western Athletic	9	0.421	0.716	0.510	4 nodes	0.6	1	0.733
Pacific-10	10	0.714	1	0.833	0 nodes	1	1	1
Big Ten	11	0.55	1	0.710	9 nodes	0.729	1	0.814
Big East	8	0.414	0.781	0.534	5 nodes	1	1	1
Atlantic Coast	12	0.524	0.924	0.668	3 nodes	1	1	1
Conference USA	12	0.661	1	0.796	1 nodes	1	1	1
Big 12	12	0.317	0.465	0.355	5 nodes	1	1	1
Total	115	0.488	0.783	0.595	34 nodes (29.6%)	0.927	1	0.952

Table 1. Algorithm Accuracy Comparison for the NCAA Network (Precision, Recall and F-measure score are all average values for all nodes in the community).

precision and recall, of all the discovered local communities are calculated. We average the score for all schools in one conference to evaluate the accuracy of the algorithm to detect that particular community. Finally, an overall average score of the precision, recall and f-measure score of all communities is calculated for comparison.

The experiment results are shown in Table 1. We first note the disadvantage of metric R we reviewed theoretically in Section 2.2, which is vulnerability against outliers, has been confirmed by the results: for all communities, Algorithm R gets a higher recall but a much lower precision, which eventually leads to an unsatisfactory f-measure score. On the other hand, the accuracy of our algorithm is almost perfect, with a 0.952 f-measure score on average. Second, we see that our algorithm does not return local communities if starting with certain nodes in the network, namely 34 of the 115 schools representing 29.6%. (Note that in these cases the local community is considered not existent and is not included in the average accuracy calculation even though the starting nodes are not outliers.) However, this result actually shows merit of our approach instead of weak points. Generally speaking, in one local community, nodes can be classified into cores and peripheries. It would be easier for an algorithm to identify the local community if it began from cores rather than peripheries. For example, if the algorithm starts from a periphery node i in community c, the expansion step might fall into a different neighbour community d, which has some members connecting to i, due to lack of local information. It would be more and more difficult to return to c as the algorithm proceeds, because members of d are usually taken in one after another and finally, the discovered local community would be d plus node i, instead of c. Fortunately, our algorithm detects such phenomena in the examination phase since i will be found as an outlier to d. Therefore we do not return the result as a local community for i since we realize that it is misdirected in the beginning. As a possible solution for this problem, we can always start with multiple nodes, by which we provide more local information to avoid the possible misdirection. Note that while our algorithm handles such situations, algorithm R returns communities for every node without considering this problem, which is one reason for its low accuracy. Also note that another approach [13] has a similar “deletion step”, however, that approach depends on arbitrarily selected thresholds.

4.2. The Amazon Co-purchase Network

While mid-size networks with ground truth provide a well-controlled testbed for evaluation, it is also desirable to test the performance of our algorithm on large real world networks. However, since ground truth of such large networks is elusive, we have to justify the results by common sense. We applied our algorithm and algorithm R to the recommendation network of Amazon.com, collected in January 2006 [13]. The nodes in the network are items such as books, CDs and DVDs sold on the website. Edges connect items that are frequently purchased together, as indicated by the “customers who bought this book also bought these items” feature on Amazon. There are 585,283 nodes and 3,448,754 undirected edges in this network with a mean degree of 5.89. Similar datasets have been used for testing in previous works [14], [13].

Due to lack of space, here we only present discovered local communities for one popular book (*The Lord of the Rings (LOR)* by J.R.R. Tolkien), which is used as the starting

Alg.	Items (Books) in the Local Community
Both	Smith of Wootton Major*
	LoR: A Reader's Companion$^{\#}$
	LoR: 50th Anniversary, One Vol. Edition*
	(The starting node) LoR [BOX SET]*
L	On Tolkien: Interviews, ... and Other Essays$^{\#}$
	Tolkien Studies: ... Scholarly Review, Vol. 2$^{\#}$
	Tolkien Studies: ... Scholarly Review, Vol. 1$^{\#}$
	... Grammar of an Elvish Language from LoR$^{\#}$
	J.R.R. Tolkien Companion and Guide$^{\#}$
	The Rise of Tolkienian Fantasy$^{\#}$
	... Celtic And Norse in Tolkien's Middle-Earth$^{\#}$
R	Farmer Giles of Ham & Other Stories*
	... Farmer Giles of Ham*
	Roverandom*
	Letters from Father Christmas, Revised Edition*
	Bilbo's Last Song*
	... Wonderful Adventures of Farmer Giles*
	Poems from The Hobbit*
	Father Christmas Letters Mini-Book*
	Tolkien: The Hobbit Calendar 2006*

Table 2. Algorithm Comparison for the Amazon Network. $*$ indicates the author is J.R.R. Tolkien while $\#$ is not.

node. The results are shown in Table 2. While both algorithms find interesting communities, our algorithm detects books by authors other than Tolkien but are strongly related to the topic. On the other hand, more than 90% of the books in R's community are written by Tolkien. Moreover, after reading the reviews and descriptions on Amazon, we found that many of the books are for children, e.g, *Letters from Father Christmas*. These books are not related to dragons and magic, but are included in the community because they weakly connect to the starting node since they share the same author, as we discussed in Section 2.2.

5. Conclusions

We have reviewed problems of existing methods for constructing local communities, and propose a new metric L to evaluate local community structure when the global information of the network is unavailable. Based on the metric, we develop a two-phase algorithm to identify the local community of a set of given starting nodes. Our method does not require arbitrary initial parameters, and it can detect whether a local community exists or not for a particular node. We have tested our algorithm on real world networks and compared its performance with previous approaches. Experimental results confirm the accuracy and the effectiveness of our metric and algorithm.

6. Acknowledgments

Our work is supported by the Canadian Natural Sciences and Engineering Research Council (NSERC), by the Alberta Ingenuity Centre for Machine Learning (AICML), and by the Alberta Informatics Circle of Research Excellence (iCORE). We wish to thank Eric Promislow for providing the Amazon data and Xiaowei Xu for the NCAA data.

References

[1] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman, "Email as spectroscopy: automated discovery of community structure within organizations," *Communities and technologies*, pp. 81–96, 2003.

[2] M. A. Nascimento, Jörg Sander, and J. Pound, "Analysis of sigmod's co-authorship graph," *SIGMOD Record*, vol. 32, no. 2, pp. 57–58, 2003.

[3] A. F. Smeaton, G. Keogh, C. Gurrin, K. McDonald, and T. Sodring, "Analysis of papers from twenty-five years of sigir conferences: What have we been doing for the last quarter of a century," *SIGIR Forum*, vol. 36, no. 2, pp. 39–43, 2002.

[4] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 u.s. election: divided they blog," in *LinkKDD '05*, 2005, pp. 36–43.

[5] S. Gregory, "An algorithm to find overlapping community structure in networks," in *PKDD*, 2007, pp. 91–102.

[6] G. W. Flake, S. Lawrence, and C. L. Giles, "Efficient identification of web communities," in *KDD*, 2000, pp. 150–160.

[7] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, 2004.

[8] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, 2004.

[9] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger, "Scan: a structural clustering algorithm for networks," in *KDD*, 2007, pp. 824–833.

[10] J. P. Bagrow, "Evaluating local community methods in networks," *J.STAT.MECH.*, p. P05001, 2008.

[11] J. P. Bagrow and E. M. Bollt, "Local method for detecting communities," *Physical Review E*, vol. 72, no. 4, 2005.

[12] A. Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, p. 026132, 2005.

[13] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," in *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, 2006, pp. 233–239.

[14] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very lage networks," *Phys. Rev. E*, vol. 70, p. 066111, 2004.