Detecting Communities in Social Networks using Local Information

Jiyang Chen and Osmar R. Zaïane and Randy Goebel

Abstract Much structured data of scientific interest can be represented as networks, where sets of nodes or vertices are joined together in pairs by links or edges. Although these networks may belong to different research areas, there is one property that many of them do have in common: the network community structure. There has been much recent research on identifying communities in networks. However, most existing approaches require complete network information, which is impractical for some networks, e.g. the World Wide Web or the cell phone telecommunication network. Local community detection algorithms have been proposed to solve the problem 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 also propose a community discovery process for large networks that iteratively finds communities based on our measure. 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 [35], academic collaboration records [23, 34] and even political elections [1]. A community in the network can be seen as a subgraph such that the density of edges within the subgraph is

Jiyang Chen
Department of Computing Science, University of Alberta, e-mail: jiyang@cs.ualberta.ca

Osmar R. Zaïane
Department of Computing Science, University of Alberta, e-mail: zaiane@cs.ualberta.ca

Randy Goebel
Department of Computing Science, University of Alberta, e-mail: goebel@cs.ualberta.ca

greater than the density of edges between inside and outside nodes [15]. 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, which could enable search engines to increase the precision and recall of search results by focusing on narrow but topically-related subsets of the web [11]; groups within social networks might correspond to communities, which can be used to understand organization structures. Moreover, the influence of the community structure may reach further than these: a number of recent results suggest that networks can have properties at the community level that are quite different from their properties at the level of the entire network, so that analysis that focus on whole networks and ignore community structure may miss many interesting features [26]. For example, we may find that people in different community groups have different mean numbers of contacts in some social networks, i.e, individuals in one group might have many neighbours while members of another group are more reticent. Such social networks are reported in [2] and [13] for the study of HIV in sexual contact networks. Therefore, characterizing such networks by only quoting a single figure for the average number of contacts an individual has, and without considering the community structure, will definitely miss important features of the network, which is relevant to questions of scientific interest such as epidemiological dynamics [17].

The problem of finding communities in social networks has been studied for decades. Recently, several quality metrics for community structure have been proposed [25, 28, 37]. Among them, modularity Q is proved to be the most accurate [7] and has been pursued by many researchers [6, 10, 16, 26, 36]. However, most of those approaches require knowledge of the entire graph structure to identify communities, which we call *global communities*. A global community is a community defined based on global information about the entire network. That is, one needs to access and see the whole network information. This constraint is problematic for networks which are too large to know completely, e.g., the WWW. In spite of these limitations, finding communities, which we call *local communities*, would still be useful, albeit constrained by the small volume of accessible information about the network in question. A local community is a community defined based on local information without having access to the entire network. For example, we might like to quantify local communities of either a particular webpage given its link structure in the WWW, or a person given his social network in Facebook. Existing approaches [28, 6] also assume that each entity belongs to only one community, however in the real world one entity usually belongs to multiple communities, e.g., one researcher could publish in both the data mining community and the visualization community. We refer to these as overlapping communities.

Several techniques [3, 4, 5, 22] have been proposed to identify local community structure given limited information about network. However, parameters that are hard to obtain are usually required, such as the community size or density. Moreover, communities discovered by these algorithms include many outliers, which are nodes that are weakly connected to the community, 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. Moreover, we propose a community discovery process to discover overlapping communities in a large network where global information is not available. Given one or a set of start nodes, our algorithm starts from a local community, then iteratively identifies communities while expanding to the whole graph. We 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. Our iterative community discovery process is able to discover overlapping communities with only local information. Additionally it does not require any arbitrary thresholds or other parameters.

The rest of the paper is organized as follows. We discuss related work in Section 2. Section 3 defines the problem and reviews existing solutions. We describe our approach in Section 4 and report experimental results in Section 5, followed by conclusions in Section 6.

2 Related Work

Traditional data mining algorithms, such as association rule mining, supervised classification and clustering analysis, commonly attempt to find patterns in a data set characterized by a collection of independent instances of a single relation. However, for social networks, where entities are related to each other in various ways, naively applying traditional statistical inference procedures, which assume that instances are independent, can lead to inappropriate conclusions about the data [18]. For example, for a search engine, indexing and clustering web pages based on the text content without considering their linking structure would definitely lead to bad results for queries. The relations between objects should be taken into consideration and can be important for understanding community structure and knowledge patterns.

Generally speaking, we can divide previous research of finding communities in networks into two main principle lines of research: *graph partitioning* and *hierarchical clustering*. These two lines of research are really addressing the same question, albeit by somewhat different means. There are, however, important differences between the goals of the two camps that make quite different technical approaches desirable [27]. For example, *graph partitioning* approaches usually know in advance the number and size of the groups into which the network is to be split, while *hierarchical clustering* methods normally assume that the network of interests divide naturally into some subgroups, determined by the network itself and not by the user.

Graph Partitioning. There is a long tradition of research by computer scientists on graph partitioning [31]. Generally, finding an exact solution to a partitioning task is believed to be an NP-complete problem, making it prohibitively difficult to

solve for large graphs. However, a wide variety of heuristic algorithms have been developed and give good solutions in many cases [12], e.g., multilevel partitioning [19], k-partite graph partitioning [20], relational clustering [21], flow-based methods [11], information-theoretic methods [8] and spectral clustering [30]. The main problem for these methods is that input parameters such as the number of the partitions and their sizes are usually required, but we do not typically know how many communities there are, and there is no reason that they should be roughly the same size. Various benefit functions have been proposed to avoid the problem, such as the *normalized cut* [33] and the *min-max cut* [9]. However, these approaches are biased in favour of divisions into equal-sized parts and thus still suffer from the same drawbacks that make graph partitioning inappropriate for community mining.

Hierarchical Clustering. The approaches developed by sociologists in their study of social networks for finding communities are perhaps better suited for our current purpose than the aforementioned clustering methods. The principle popular technique in use is *hierarchical clustering* [32]. The main idea of this technique is to discover natural divisions of social networks into groups, based on various metrics of similarity (usually represented as similarity x_{ij} between pairs (i, j) of vertices). The hierarchical clustering method has the advantage that it does not require the size or number of groups we want to find beforehand, therefore, it has been applied to various social networks with natural or predefined similarity metrics, such as the modularity and betweenness measure [6, 14, 25, 28]. However, they are usually slow and the performance depends highly on the corresponding metrics.

Recently, real world networks have been shown to have an overlapping community structure, which is hard to grasp with classical clustering methods where every vertex of the graph belongs to only one community. Based on these observations, fuzzy methods [15, 24, 29, 38] have been proposed for overlapping structure. Recent work by Xu et al. [37] proposed a fast SCAN algorithm to detect not only clusters, but also hubs and outliers in networks. However, the performance of these approaches depends on input parameters, which are very sensitive.

While all these methods successfully find communities, they implicitly assume that global information is always available. However, that is usually not the case for large networks in the real world. Clauset [5] and Luo et al. [22] proposed similar metrics for community detection with local information, which are presented in detail in Section 3. Bagrow et al. proposed an alternative method to detect local communities [4], which spreads an l-shell outward from the starting node n, where l is the distance from n to all shell nodes. The performance of their approach 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 [3] to measure local structure, however, their method lacks an appropriate stopping criteria and thus still relies on arbitrary thresholds.

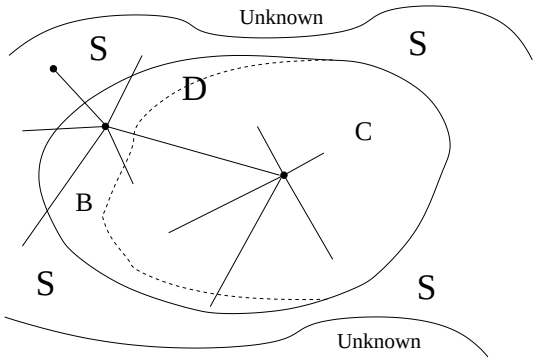

Fig. 1 Local Community Definition

3 Preliminaries

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

3.1 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. (Note that D may start with one node, but can later contain a set of nodes and connections between them as a local community.) 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 the network G is to visit some neighbour nodes s_i of D (where $s_i \in S$) and obtain a list of adjacent nodes 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 [3, 4, 5, 22], however, the metrics used to discover and evaluate the local community are different, as explained in the next section.

3.2 Previous Approaches

Clauset has proposed the local modularity measure R [5] 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 [22] 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 straight-forward 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 $\Delta M < 0$.

Indeed algorithms using these metrics are able to detect 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.

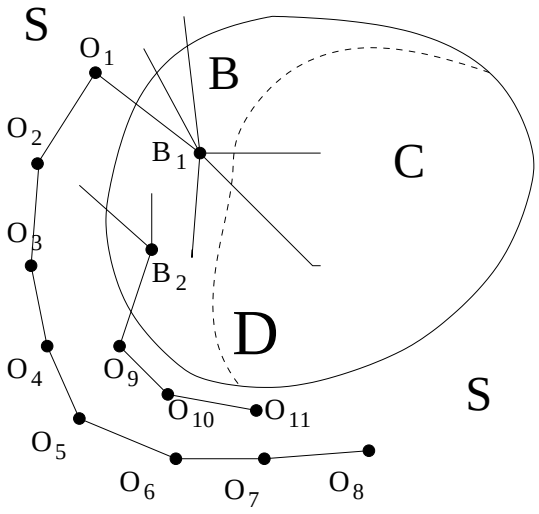

Fig. 2 Problem of Previous Approaches

4 Our Approach

Existing approaches discussed in Section 3 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.

4.1 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 3.2 and Section 5.1.1 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.

4.2 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 (See Algorithm 1). Finally, the starting node should still be found in D, otherwise, we believe a local community does not exist if we start from n_0. (See Algorithm 2.)

Algorithm 1 General Local Community Identification

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

Algorithm 2 Single Local Community Identification

Input: A social network G and a start node n_0.
Output: A local community D for node n_0.
1. Apply algorithm 1 to find a local community D for n_0.
2. 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.

$$L_i' = \frac{\frac{Ind+2*Ind_i}{|D|+1}}{\frac{Outd-Ind_i+Outd_i}{|B'|}} \tag{6}$$

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.

4.3 Iterative Local Expansion

Algorithm 1 is for identifying one local community for a specific set of starting nodes. However, we could apply this algorithm iteratively to cover the whole graph or a large section of the graph if the iterative process is terminated. In other words, instead of one-node-at-one-step, we expand as one-community-at-one-step to discover the community structure in the network. See Algorithm 3.

Algorithm 3 Iterative Expansion Algorithm

Input: A social network G, a start node n_0 and the community number m (optional).
Output: A list of local communities.
1. Apply algorithm 1 to find a local community l_0 for n_0.
2. Insert neighbours of l_0 into the shell node set S
3. **While** ($|S|! = 0$ && ($i \leq m$))
   Randomly pick one node $n_i \in S$.
   Apply algorithm 1 to find a local community l_i
   for n_i.
   Remove n_i and nodes in S that are covered by l_i.
   Update S by neighbours of l_i that are not covered
   yet.
4. Output m local communities $l_0, l_1, l_2 \ldots, l_m$, m could be given as a stop parameter if necessary.

In algorithm 3, we recursively apply the local community identification algorithm to expand the community structure. Every time we find a local community, we update the shell node set, which is actually a set of nodes whose community information is still unclear. Note that here we accept identified local communities even if the starting node is not included. The shown algorithm stops when we have learned the whole structure of the network; however, we could also give parameters as stopping criteria if exploring the whole network is unnecessary or impractical, such as the number of discovered communities (m), or the number of nodes that has been visited (k). The algorithm could also be parallel and have multiple starting nodes, where several local community identification procedures start simultaneously from different locations of the network. Obviously, the complexity of the Algorithm 3 is still $O(kd|S|)$.

As previously discussed, in real world networks, one entity usually belongs to multiple communities. However, most of the existing approaches cannot identify such overlapping communities. Fortunately, even though we do not specifically focus on finding the overlapping property, our approach is able to discover overlapping communities, since in our algorithm nodes could be included in multiple local communities based on their connection structure.

5 Experiment Results

In this section we conducted several experiments to validate the effectiveness of the proposed approach.

5.1 Comparing Metric Accuracy

Since the ground truth of local communities in a large 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 [4, 5, 22]. 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 [5] (we refer to it as algorithm R) outperforms all other known methods for local community detection. Furthermore, similar to our approach, R does not require any initial parameters while other methods [3, 4, 22] 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.

5.1.1 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) [37]. 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 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 3.2, 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

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

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 [22] has a similar "deletion step", however, that approach depends on arbitrarily selected thresholds.

2006 NCAA League		Algorithm Results						
		Algorithm using metric R			Algorithm using metric L			
Conference	Size	P	R	F	No Community	P	R	F
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 (P), Recall (R) and F-measure (F) score are all average values for all nodes in the community).

5.1.2 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 [22]. 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. Note that in this dataset we are looking for communities of "items" instead of communities of "people". There are 585,283 nodes and 3,448,754 undirected edges in

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.

this network with a mean degree of 5.89. Similar datasets have been used for testing in previous works [5, 22].

In table 2, we 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 node. While both algorithms find 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 3.2.

5.2 Iteratively Finding Overlapping Communities

After evaluating the accuracy of the L metric and our algorithm for single community identification, here we apply Algorithm 3 on the Amazon network to find overlapping communities iteratively. Table 3 shows several local community examples of our result. Note that start nodes of some communities may be removed by

	Items (Books) in the Local Communities
1	Mozart: A Cultural Biography
2	The Cambridge Companion to Mozart (Cambridge Companions to Music)
3	The Mozart Compendium: A Guide to Mozart's Life and Music
4	Mozart: The Golden Years
...	...
19	The Complete Mozart: A Guide to the Musical Works ...
1	Chopin In Paris: The Life And Times Of The Romantic Composer
2	The Cambridge Companion to Chopin (Cambridge Companions to Music)
3	Chopin (Master Musicians Series)
4	Chopin: The Man and His Music
5	Chopin's Letters
...	...
15	The Parisian Worlds of Frederic Chopin
1	The Cambridge Companion to Schubert (Cambridge Companions to Music)
2	The Cambridge Companion to Mozart (Cambridge Companions to Music)
3	The Cambridge Companion to Chopin (Cambridge Companions to Music)
4	The Cambridge Companion to Stravinsky (Cambridge Companions to Music)
5	The Cambridge Companion to Ravel (Cambridge Companions to Music)
...	...
9	The Cambridge Companion to Beethoven (Cambridge Companions to Music)
1	The New Webster's Grammar Guide
2	Hardcover, Longman Grammar of Spoken and Written English
3	Editorial Freelancing: A Practical Guide
4	The Oxford Dictionary for Writers and Editors
...	...
52	Modern American Usage: A Guide
1	Shakespeare's Language
2	Imagining Shakespeare
3	Hamlet: Poem Unlimited
4	... A Complete Pronunciation Dictionary for the Plays of William Shakespeare
...	...
66	William Shakespeare: A Compact Documentary Life

Table 3 Overlapping Local Community Examples for the Amazon Network

our algorithm. Such communities are not included using Algorithm 2 for single local community identification in earlier experiments.

The first community has 19 nodes, originated at the book *Mozart: A Cultural Biography*. It naturally includes other books about the life and music of the legendary musician. Similarly, we have another 15-node-community about the famous Polish pianist Chopin. The third community is a book series, which is the *Cambridge Companions to Music*. Finally, the fourth community and fifth community contain books about English grammar and William Shakespeare. Note that many other global community detection algorithms, e.g., FastModularity [6], become slow for such huge networks. Moreover, they may not apply if the global network information is unavailable.

Aside from local communities of books in Amazon, our approach also finds overlaps between communities. For example, the two books *The Cambridge Companion to Mozart (Cambridge Companions to Music)* and *The Cambridge Companion to Chopin (Cambridge Companions to Music)* are found both in the community of the book series and the community of the subject. One could easily justify there is indeed some overlap.

6 Conclusion and Future Work

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. Moreover, we extend the algorithm to an iterative local expansion approach to detect communities to cover large networks. 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.

In this work, we assume the social network to be "static". It would be interesting to investigate the possibility of extending the proposed metric and algorithms to discover communities in a dynamic social network. Our future work also includes the investigation of a means to validate the effectiveness of overlapping community detection in a large network without ground truth.

7 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. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 u.s. election: divided they blog. In: LinkKDD '05, pp. 36–43 (2005)
2. Aral, S.O., Hughes, J.P., Stoner, B., Whittington, W., Handsfield, H.H., Anderson, R.M., Holmes, K.K.: Sexual mixing patterns in the spread of gonococcal and chlamydial infections. American Journal of Public Health **89**, 825–833 (1999)
3. Bagrow, J.P.: Evaluating local community methods in networks. J.STAT.MECH. p. P05001 (2008)
4. Bagrow, J.P., Bollt, E.M.: Local method for detecting communities. Physical Review E **72**(4) (2005)
5. Clauset, A.: Finding local community structure in networks. Physical Review E **72**, 026,132 (2005)
6. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**, 066,111 (2004)
7. Danon, L., Duch, J., Diaz-Guilera, A., Arenas, A.: Comparing community structure identification. J. Stat. Mech p. P09008 (2005)
8. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: KDD, pp. 89–98 (2003)
9. Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: ICDM, pp. 107–114 (2001)
10. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Phys. Rev. E **72**, 027,104 (2005)
11. Flake, G.W., Lawrence, S., Giles, C.L.: Efficient identification of web communities. In: KDD, pp. 150–160 (2000)
12. Fortunato, S.: Community detection in graphs. Physics Reports **486**, 75–174 (2010)
13. Garnett, G.P., Hughes, J.P., Anderson, R.M., Stoner, B.P., Aral, S.O., Whittington, W.L., Handsfield, H.H., Holmes, K.K.: Sexual mixing patterns of patients attending sexually transmitted diseases clinics. Sexually Transmitted Diseases **23**, 248–257 (1996)
14. Girvan, M., Newman, M.: Community structure in social and biological networks. In: PNAS USA, 99:8271-8276 (2002)
15. Gregory, S.: An algorithm to find overlapping community structure in networks. In: PKDD, pp. 91–102 (2007)
16. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature **433**, 895–900 (2005)
17. Gupta, S., Anderson, R.M., May, R.M.: Networks of sexual contacts: Implications for the pattern of spread of hiv. AIDS **3**, 807–817 (1989)
18. Jensen, D.: Statistical challenges to inductive inference in linked data (1999)
19. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. Journal of Parallel and Distriuted Computing **48**(1), 96–129 (1998)
20. Long, B., Wu, X., Zhang, Z.M., Yu, P.S.: Unsupervised learning on k-partite graphs. In: KDD, pp. 317–326 (2006)
21. Long, B., Zhang, Z.M., Yu, P.S.: A probabilistic framework for relational clustering. In: KDD, pp. 470–479 (2007)
22. Luo, F., Wang, J.Z., Promislow, E.: Exploring local community structures in large networks. In: WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 233–239 (2006)
23. Nascimento, M.A., Jörg Sander, Pound, J.: Analysis of sigmod's co-authorship graph. SIGMOD Record **32**(2), 57–58 (2003)

24. Nepusz, T., Petroczi, A., Negyessy, L., Bazso, F.: Fuzzy communities and the concept of bridgeness in complex networks. Physical Review E **77** (2008)
25. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E **69** (2004)
26. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E **74** (2006)
27. Newman, M.E.J.: Modularity and community structure in networks. PNAS USA **103** (2006)
28. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004)
29. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**, 814 (2005)
30. Ruan, J., Zhang, W.: An efficient spectral algorithm for network community discovery and its applications to biological and social networks. In: ICDM, pp. 643–648 (2007)
31. Schaeffer, S.E.: Graph clustering. Computer Science Review **1**, 27–64 (2007)
32. Scott, J.: Social network analysis: A handbook (Sage, London 2nd edition(2000))
33. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE. Trans. on Pattern Analysis and Machine Intelligence (2000)
34. Smeaton, A.F., Keogh, G., Gurrin, C., McDonald, K., Sodring, T.: Analysis of papers from twenty-five years of sigir conferences: What have we been doing for the last quarter of a century. SIGIR Forum **36**(2), 39–43 (2002)
35. Tyler, J.R., Wilkinson, D.M., Huberman, B.A.: Email as spectroscopy: automated discovery of community structure within organizations. Communities and technologies pp. 81–96 (2003)
36. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: SIAM (2005)
37. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: Scan: a structural clustering algorithm for networks. In: KDD, pp. 824–833 (2007)
38. Zhang, S., Wang, R., Zhang, X.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A **374**, 483–490 (2007)