Annals of Information Systems manuscript No.
(will be inserted by the editor)

ONDOCS: Ordering Nodes to Detect Overlapping Community Structure

Jiyang Chen · Osmar R. Zaïane · Jörg Sander · Randy Goebel

Received: 0-0-00 / Revised: 0-0-00
Accepted: 0-0-00 / Published online: 0-0-00

Abstract Finding communities is an important task for the discovery of underlying structures in social networks. While existing approaches give interesting results, they typically neglect the fact that communities may overlap, with some hub nodes participating in multiple communities. Similarly, most methods cannot deal with outliers, which are nodes that belong to no germane communities. The definition of community is still vague and the criterion to locate hubs or outliers vary. Existing approaches usually require guidance in this regard, specified as input parameters, e.g., the number of communities in the network, without much intuition. Here we present a general community definition and a list of requirements for a community mining metric. We review advantages and disadvantages of existing metrics and propose our new metric to quantify the relation between nodes in a social network. We then use the new metric to build a visual data mining system, which first helps the user to achieve appropriate parameter selection by observing initial data visualizations, then detects overlapping community structure from the network while also excluding outliers. Experiment results verify the scalability and accuracy of our approach on real data networks and show its advantages over existing methods that also consider overlaps. An empirical evaluation of our metric demonstrates superior performance over previous measures.

Jiyang Chen
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
Tel.: 1-780-492-4822
E-mail: jiyang@cs.ualberta.ca

Osmar R. Zaïane
Tel.: 1-780-492-2860
E-mail: zaiane@cs.ualberta.ca

Jörg Sander
Tel.: 1-780-492-5084
E-mail: joerg@cs.ualberta.ca

Randy Goebel
Tel.: 1-780-492-2683
E-mail: goebel@cs.ualberta.ca

1 Introduction

Many datasets of scientific interest can be modeled as networks, which consist of sets of nodes representing entities, connected by edges representing various relations between these entities. For example, the World Wide Web (WWW) can be viewed as a very large graph where nodes represent web pages and edges represent hyperlinks between pages. In social networks, nodes typically represent individuals and edges indicate relationships, e.g., in a tele-communication network, each node is a phone number and edges represent the fact that two nodes communicated. In such networks, the ability to detect closely-related entity groups, i.e., communities, can be of significant practical importance. For instance, the fact that web pages in the same community might focus on related topics can be used to help page ranking and recommendation. Social network communities can be used to understand implicit network structures, e.g., organization structures, academic collaborations or usage pattern in tele-communication networks.

In recent years, there has been a surge of research interests on finding communities in networks. A community (or *cluster*) 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 [1]. Existing community detection approaches, such as spectral clustering [2], modularity-based [3] and density-based methods [4] achieve good results for some datasets, and have proposed various metrics to measure the similarity between social entities. However, all of them implicitly define communities based on metrics which measure only partial aspects of the social network, thus existing community definitions can only identify specific types of communities. A new metric is needed to more thoroughly quantify the relation between two social entities.

Recent studies have also revealed that network models of many real world phenomena exhibit an overlapping community structure, i.e., a node can belong to more than one community, which is hard to take into account with classical graph clustering methods where every vertex of the graph belongs to exactly one community [5]. This is especially true for social networks, where individuals can connect to several groups in the network as *hubs*. Furthermore, in real networks we also have another node category, which belongs to no community, i.e., *outliers*. Therefore, a typical social network consists of communities, hubs and outliers. It is essential for community discovery methods to identify nodes in these three categories, since the isolation of hubs and outliers can be crucial for many applications. Unfortunately, a precise description of what a *community* really is has not yet been explicitly articulated. Moreover, the definition would be different across various domains, or even across different networks of the same domain. Therefore, most proposed approaches [1, 6, 7, 5, 4] for overlapping community detection require the user to describe the communities they are looking for by giving parameters, e.g., community size, density range, the number of communities, etc. However, appropriate parameters are usually extremely hard to determine without tedious and repeated testing. Moreover, arbitrary parameters may over-restrain the space in which communities are found and lead to inaccurate results. Overall, if the real value of community identification is to be achieved, we want tools that form the basis for community mining, so that useful and interesting structure emerges without too much parameter estimation required.

In this paper, we first define social network communities with a list of requirements for a community mining metric, based on observations of social network characteristics. After reviewing the advantages and disadvantages of existing metrics, we propose the *R (Relation)* metric to measure the similarity between any pair of entities in a so-

cial network, then show its advantages by comparison with existing metrics. We then propose our approach ONDOCS (Ordering Nodes to Detect Overlapping Community Structure). Our visual data mining approach first generates preliminary visualizations of the network in question by ordering nodes based on their reachability scores (RS) to help the user understand the network structure in order to choose appropriate parameters. Selected parameters are then used to extract communities, hubs and outliers from the network. We offer the following contributions in this paper:

- A new metric R to quantify the relation between entities.
- A visual data mining approach to assist the user in finding appropriate parameters to describe the communities they are looking for.
- A scalable and efficient method to discover communities, hubs, and outliers in social networks.

The rest of the paper is organized as follows. We discuss related work in Section 2. Section 3 introduces our community definition and reviews existing metrics. We present our R metric and the ONDOCS approach in Section 4 and report experimental results in Section 5, followed by conclusions in Section 6.

2 Related Work

Community Mining. The problem of finding communities in social networks has been studied for decades in many fields, including computer science, sociology, and physics. Originally, graph partitioning methods [8, 9, 2] were applied, but researchers soon realized that the condition for graph partitioning methods to be valid is that the number or the sizes of the communities into which the networks are divided should be fixed, which is not true for community mining. Various benefit functions have been proposed to solve the problem, such as *normalized cut* [8] and *min-max cut* [9], but they are still biased in favor of divisions into equal-sized parts and thus still suffer from the same drawbacks that make graph partitioning inappropriate for community detection. Recently, many quality metrics for community structure have been proposed [3, 10, 4]. Among them, modularity Q has been proved to be the most accurate [11] and has been pursued by many researchers [10, 12–16]. While all previous works focus on clique communities (defined in Section 3.1) and apply hierarchical methods, Xu et al. [4] propose the density-based SCAN algorithm to detect transitive communities (also defined in Section 3.1) and locate hubs and outliers in networks. However, all those metrics focus only on one type of community and do not consider a general community definition, not the whole picture of community mining in social networks, thus none of them satisfy all of the requirements listed in Section 3.

Overlapping Community Structure Detection. In general, there are two ways to detect overlapping community structure in a network. One natural idea is to first globally partition the network and then locally expand the discovered communities to locate overlapping components. Wei et al. [17] partition the network using the spectral clustering method and then locally expand to optimize a variation of the Modularity Q measure [3]. For overlapping community discovery in a name-entity network, Li et al. [18] generate community cores by merging triangles (3-clique) so that one vertex can be part of different communities if it belongs to several cliques. Similarly, Baumes et al. [19] initialize community cores using the Link Aggregate (LA) Algorithm and then refine the peripheries by an Iterative Scan (IS) procedure. Another mainstream research

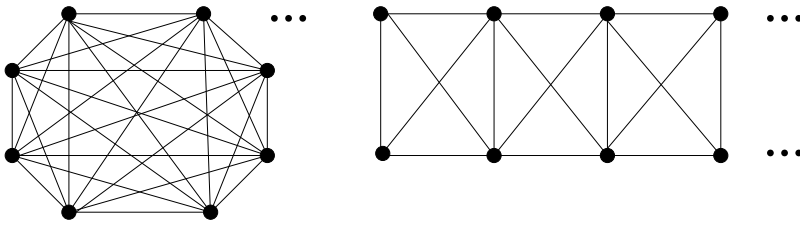
direction for this problem is based on fuzzy clustering. Zhang et al. [20] combine modularity and a fuzzy c-means clustering algorithm to identify overlapping communities. Nepusz et al. [7] propose a similarity function based on membership, and solve the fuzzy community detection problem as a constrained optimization problem. Recently, Palla et al. [5] propose the CFinder system to partition complex networks to k-clique communities, where k is a given parameter as clique size. Gregory proposes the CONGA algorithm [1] based on the betweenness score [3] and later extends it to the CONGO algorithm to improve the scalability [6]. He also shows that CONGO provides the same level of performance as CFinder, on synthetic networks. While all of the above methods successfully detect overlapping community structure, some major problems exist. Most methods do not consider outliers, which belong to no communities, thus many outliers would be classified as community members, i.e., they force outliers into existing clusters. Additionally, the fact that they intentionally focus on overlapping community structure makes them find or force overlap even for data without such structure. More importantly, many approaches not only require parameters that are difficult to determine but also their results are very sensitive to parameter settings, e.g., number of communities [1, 20], community density [5, 18], or size of a local community region [6].

Visual Data Mining.

Most community mining approaches apply data mining algorithms, e.g, agglomerative hierarchical clustering for a bottom-up merge, or partitional clustering for a top-down split. Having noted that community mining is also a data mining process, we believe that the idea of visual data mining could be helpful in the mining process, both to guide the mining towards goals, and to better understand the results, since visualization and interaction capabilities enable the user to incorporate domain knowledge to finding communities in social networks. Generally speaking, the areas of data mining and information visualization offer various techniques which effectively complement one another supporting the discovery of patterns in data. Whereas traditional (algorithmic) techniques are analyzing the data automatically, information visualization techniques can leverage the data mining process from an orthogonal direction, by providing a platform for understanding the data and generating hypotheses about the data based on human capabilities such as domain knowledge, perception, and creativity [21]. In the past few years, visualization techniques have been specifically designed to support human involvement in the data mining process. For example, Ankerst et al. [22] propose an interactive decision tree classifier based on a multidimensional visualization of the training data. They later extend the work [23] to include categorical attributes to interactively build decision trees and thus support a much broader range of applications. Similar visual data mining ideas are also applied in [24, 25] to help users determine parameters for decision tree construction and classification rule discovery.

3 Preliminaries

In this section, we propose a definition for network communities and provide a list of requirements for a good measure for community detection. We discuss two existing measures based on those requirements.



(A) Clique Community (B) Transitive Community

Fig. 1 Examples for Clique Community and Transitive Community

3.1 Community Definition

Recent research has proposed community detection methods in two different ways based on various motivations and similarity measures. First, hierarchical methods [3, 10] tend to find communities *globally* so that nodes, which are more densely connected to nodes in the same community than outside nodes, are grouped together; second, density-based approaches [4] classify nodes into communities based on their *local* structure, i.e., nodes are in the same community if they share many neighbours. In experiments, these two approaches typically yield noticeably different results on the same datasets. They actually target two different kinds of communities. On one hand, hierarchical methods partition networks by greedily maximizing an objective function, which increases for pairs of connected nodes that are in the same community and decrease for pairs of disconnected nodes also in the same community. Their methods favor communities where every node connects to everyone else in the same community, which we call *Clique Communities* (Fig 1 A). On the other hand, density-based approaches expand communities from nodes that are structurally dense, i.e., have enough neighbours, judged by appropriate parameters. Therefore, these approaches do not consider global properties but only the local network structure. They find communities where nodes may not directly connect to many others in the same community but are indirectly connected to every other node via some connections, which we call *Transitive Communities* (Fig 1 B). The difference between these two strategies is analogous to hierarchical-based and density-based methods in the data clustering field [26].

While existing methods implicitly describe specific types of communities based on their metrics and algorithms without clearly defining them, we give a general definition for social network communities based on the observations highlighted above: ***A community is a network partition such that entities within the same community share some common trait or proximity, judged by some defined entity similarity or relationship metric***.

No matter how communities are defined, there are two major issues for community mining that remain to be addressed. First, each pair of nodes should be measured by their similarity or relationship; second, pairs with high similarity or strong relationship should be put in the same community. Although it is the algorithm (hierarchical or density-based) that decides the community type to be found (clique or transitive), a good similarity metric is vital for both clique and transitive community structure detection. We present the requirements of a good metric in the following section.

3.2 Requirements for A Good Community Mining Metric

It is easy to confuse graph partitioning with community mining since these two lines of research are really addressing the same question, which can be described as dividing vertices of a network into some number of groups. There are, however, important differences between network characteristics of the two camps that make quite distinct approaches and metrics desirable. For instance, in social network community mining, the relation between two nodes is asymmetric. (Take MySpace.com as an example: user A might list user B as one of his best friends while he is not even in the friend list of user B.) Thus, existing measures and approaches that are shown to be effective for some graph partitioning may not fit for community mining, since they do not take these differences into consideration. In the following, we propose a list of requirements, which we believe should be satisfied by a good metric for community mining.

1. ***A metric should measure the similarity between every pair of nodes.***
   A similarity score between two nodes is required for all algorithms to decide whether to put these two nodes into one community or not. The metric should be able to measure all pairs, connected or disconnected. Metrics, which do not consider disconnected pairs of nodes, may be able to find some community structure, but they naively assume that disconnected pairs should not be in the same community.
2. ***A metric should reflect not only similarity but also dissimilarity.***
   In other words, the metric not only measures whether two nodes should be in the same community but also measures whether they should not be in the same community. For instance, the metric should provide a means to solve a disagreement while merging a node n in a community when some existing nodes relate to n and others do not.
3. ***A metric should consider the asymmetric nature between pairs.***
   The pair asymmetry in social networks means that $Relation(i \rightarrow j) \neq Relation(j \rightarrow i)$, e.g., consider people pair (i, j) where i has many friends and is j's only friend, i is much more important to j than j is to i. For undirected graphs, where the similarity measure are usually required to be symmetric, the asymmetric nature between the node pairs should still be considered.
4. ***An overlapping community metric should handle both hubs and outliers.***
   We think there are three kinds of nodes in a social network: hubs (nodes that have many connections and can be seen as community overlaps), outliers (nodes that have very few connections and do not belong to any community) and normal nodes (nodes that have some connections and belong to a community). The influences of hubs and outliers to community discovery have to be minimized by the metric.

3.3 Existing Metrics for Community Detection

Newman et al. proposed the modularity Q as a quality measure of a particular division of a network [3]. For a social network with k communities, the modularity is defined as $Q = \sum_{c=1}^{k} [\frac{e_c}{m} - (\frac{d_c}{2m})^2]$ where m is the number of edges in the network, e_c is the number of edges between nodes within community c, and d_c is the sum of the degrees of the nodes in community c. The modularity Q measures the fraction of the edges in the network that connect vertices of the same community, i.e., within-community edges, minus the expected value of the same quantity in a network with the same community

division but with random connections between the vertices. Q can be transformed as a sum of similarity scores for all node pairs [12,13]:

$$Q = \sum Q_{ij} = \sum_{i,j} (\frac{A_{ij}}{2m} - \frac{d_i}{2m} * \frac{d_j}{2m}) \tag{1}$$

where $A_{ij} = 1$ if nodes i and j are connected, 0 otherwise, d_i, d_j are the degree of node i and j, m is the edge number. Note that $Q_{ij} = 2 * (\frac{A_{ij}}{2m} - \frac{d_i}{2m} * \frac{d_j}{2m})$ since each pair (i, j) is calculated twice in the sum as (i, j) and (j, i). Also note that, Q_{ij} represents the difference between the probability of the event $i \leftrightarrow j$ *(node i and j are connected)* in the given graph structure ($P(i \leftrightarrow j) = \frac{A_{ij}}{m}$) and that in a random model with the same number of vertices, edges and degrees ($P(i \leftrightarrow j) = 2 * \frac{d_i}{2m} * \frac{d_j}{2m}$). (See [3,13] for detail.)

The modularity Q provides a similarity score for all pairs of nodes. Whether the score is positive or negative depends on whether two nodes are connected or not, which reflects both similarity and dissimilarity. By taking the global information (the total edge number m) into consideration in the score calculation such that the higher degree the nodes have the lower score the pair gets, modularity handles the influence from hub nodes. However, the measure neglects the asymmetric nature between pairs in social networks by assuming $P(i \rightarrow j) = P(j \rightarrow i)$. Moreover, the method fails to handle outliers. Since outliers have small degrees and can achieve high scores given the formula, they are usually inaccurately merged first into a community by hierarchical algorithms.

Recently, Xu et al. [4] proposed another similarity measure S:

$$S_{ij} = \frac{|N_i \cap N_j|}{\sqrt{|N_i| * |N_j|}} \tag{2}$$

where N_i is the neighbourhood of node i, including i itself and all nodes connecting to i. This metric normalizes the number of common neighbours by the geometric mean of the two neighbourhoods' sizes in order to compare the neighbourhood structure of the two vertices in question.

The S metric considers the local structure of compared nodes (the common neighbour number) as well as their local attributes (the sizes of both neighbourhoods), thus it minimizes the score for both hubs and outliers. However, this metric does not measure dissimilarity, e.g., the score will be zero if two nodes share no neighbours, disregarding the network structure, and it fails to include pair asymmetry as well. Although this metric is easy to be extended for all pairs of nodes, it was originally proposed for connected pairs only. Additionally, even though the S metric considers the neighbourhood size of the two nodes in question, it neglects the degrees of other nodes in the neighbourhood, i.e., every node in the neighbourhood is weighted equally as 1 disregarding whether it is a hub, an outlier or a normal node.

We have summarized two state-of-the-art similarity metrics for community mining and analyze their advantages and disadvantages (See Table 1). While they successfully find communities for some datasets, they do not satisfy all given requirements and thus need to be improved.

4 Our ONDOCS Approach

In this section, we first present our characterization of the relation between nodes, then introduce the algorithm to generate network visualizations, and then show how to detect overlapping community structure based on observed parameters.

4.1 Relationship Definition

Originally, ONDOCS is inspired by the OPTICS algorithm proposed by Ankerst et al. [27], where points are ordered for data clustering. However, unlike their clustering approach, we do not have a distance measure between nodes, so we need to define a new node relationship. The existing community metrics reviewed in Section 3 are designed to find optimal communities of a specific type, i.e., Q for clique communities and S for transitive communities, which means they focus only on partial aspects of network structure. We think that comparing the community structure to a random model, in which nodes are randomly connected in a network, is a practicable way to quantify node relations. The intuition is that community structure can be identified as that which is non-random; so developing a measure with a notion of random connections should help identify non-random structure. The neighborhood around any two nodes in question is also important in assessing their relationship. Therefore we proposed a new measure R to combine these two aspects, defined as follows:

$$R(i,j) = \frac{R(i \to j) + R(j \to i)}{2} = \frac{\sum_{x \in N_j} r(i,x) + \sum_{x \in N_i} r(x,j)}{2} \tag{3}$$

where N_i is the neighbourhood of node i, including i itself and all nodes that connect to i. The similarity between node i and j is defined as the average of $R(i \to j)$, representing the relationship from i to j's neighbourhood, and $R(j \to i)$, representing relationship from j to i's neighbourhood. $R(i \to j)$ is defined as the sum of relation scores r between i and all nodes in j's neighbourhood, similarly for $R(j \to i)$ with respect to j and i's neighbourhood. Next, in order to quantify the relation $r(i,j)$ between node i and j, we compare the probability of the event that *i and j are connected* in the original graph G to a random model, where we only keep the same node number n and node degrees $k_1, ..., k_n$ and leave the rest random. In such a random model, it is obvious that the probability of a node i having a connection to any other node is $P(i) = \frac{k_i}{n-1}$. Here we assume G is undirected so that the events of i connecting to j and j connecting to i are equivalent, thus the probability of i and j being connected is the maximum of $P(i)$ and $P(j)$:

$$P(i \leftrightarrow j) = max(P(i \to j), P(j \to i)) = max(P(i), P(j)) = \frac{max(k_i, k_j)}{n-1} \tag{4}$$

Now we define the relation score $r(i,j)$ between node i and j:

$$r(i,j) = A_{ij} - \frac{max(k_i, k_j)}{n-1} \tag{5}$$

where $A_{ij} = 1$ if nodes i and j are connected in G, 0 otherwise. Here we omit directed graphs since that is a straightforward extension. The proposed metric R, r and the random model are justified in the next section.

Metric	Metric Requirements			
	All Pairs	Similarity & Dissimilarity	Asymmetry	Hub & Outlier
Q	All	Yes	No	Only Hub
S	Connected	No	No	Both
R	All	Yes	Yes	Both

Table 1 Comparing Community Mining Metrics

4.1.1 Analyzing the R measure

We evaluate our R metric using the requirements listed in Section 3. First, R assesses similarity for both connected and disconnected pairs of nodes. Two nodes are measured by the relation between them and their neighbourhoods. Second, while the relation score r between each pair will be positive for connected pairs and negative for disconnected ones, R in Equation 3 considers all pairs within the local neighbourhood so that the R score represents an overall similarity, therefore $R(i,j)$ can be positive even if $r(i,j)$ is not. Similarly, $R(i,j)$ can be negative even if $r(i,j)$ is not. Third, the R metric is divided into two parts: $R(i \rightarrow j)$ and $R(j \rightarrow i)$, each of which represents the similarity between one node and the other's neighbourhood. The asymmetric characteristic of social networks is thus considered. Finally, the influence from hubs or outliers to other nodes are minimized. Hubs have big degrees which lead to large $\frac{max(k_i,k_j)}{n-1}$ and small r scores. Outliers have small neighbourhoods so R is small since there are few pairs to contribute in the sum. Therefore, as shown in Table 1, the R metric satisfies all requirements for a good community mining measure.

We now justify the formula for the relation score r and the random model presented in Section 4.1. Recall that the intuition behind the r score is to compare the probability of the event E, *that two nodes i and j are connected*, in the original graph structure with the probability of the same event in a random model, which has the same node number and degrees. Only if the probability of having these two nodes connected in the random model is low, does the fact that they are indeed connected show us strong relationship. Since the probability of E in the original graph is simply 1 or 0 given the network structure, we only need to answer the following question: *In an undirected graph G with n nodes, degrees $k_1, ..., k_n$ and the rest random, what is the probability of event E?* In this model, it is obvious that the probability of the event A: i connecting to j, equals to $\frac{k_i}{n-1}$ and the probability of the event B, j connecting to i, equals to $\frac{k_j}{n-1}$. However, either A or B confirms E, therefore we set $P(E) = max(P(A), P(B))$. In other words, with respect to i, the probability of selecting j as one of i's neighbours is $\frac{k_i}{n-1}$. We cannot achieve a higher score unless $k_j > k_i$, thus the probability of the fact that two nodes are connected is decided by the node with the higher degree. Note that $P(E) \neq P(A) * P(B)$ since the two events A and B are dependent on each other.

4.2 Ordering Nodes to Visualize Networks

Now we can generate network visualizations by ordering nodes based on their relation scores. Given the relationship function we defined above, for a node n_i, we create a list of nodes l_i ordered by their relation to n_i from high to low. (Note that we can limit candidate nodes to those which have $R > 0$, i.e., they are connected to or share at least one neighbour with n_i.) We define the k^{th} value in this list to be l_{ik}. Here, our

approach takes one input parameter s. However, as we will show in Section 5, s does not strongly affect the output. In practice, we usually generate several visualizations with s ranging from 2 to 8 and let the user make a choice based on their observations. For a node n_i, we define its community score C_s to be the s^{th} value in its node list l_i, i.e., $C_s(n_i) = l_{is}$, and $C_s(n_i) = 0$ if there are less than s nodes in the list. Then we define the reachability of node j with respect to i as

$$reach_s(i,j) = \begin{cases} R(i,j) \ if \ C_s(n_i) > R(i,j) \\ C_s(n_i) \ otherwise \end{cases}$$

Intuitively, the parameter s represents the expected number of nodes that one node is similar with in order to be a member of any community. C_s is the lowest relation score between node i and its similar neighbours in one community. Then the reachability score from node i to j ($reach_s(i,j)$) is the relation score between node i and j if j is not among the top s nodes of l_i and is the community score of i otherwise. Thus, $reach_s(i,j)$ measures the community relationship between i and j. It is their direct distance score if i and j are far away from each other, and equals to the community radius of i if j is close enough. Therefore, a decreasing order of the reachability scores (RS) indicates a node list for i, starting from i's most related neighbours to the least ones.

Algorithm 1 The ONDOCS Algorithm: Network Visualization

Input: A social network G with n nodes and m edges, a start node n_{start} and possible s values s_0, s_1, s_2....
Output: A list of nodes L with their Reachability Scores RS for each s.
1. Sort a node list l_i for each node n_i, ordered by their relation score to n_i, from high to low.
2. For each s :
  Initialize a max-heap h, insert n_{start} in h with $RS = 0$.
  Select the s^{th} largest element in l_i for each node n_i as its community score $C_s(n_i)$.
  While (there is still nodes in heap h) :
    Pop the node α in h with largest value ϵ.
    Store α in L_s with $RS_\alpha = \epsilon$.
    For all nodes x in l_α:
      If $x \notin h$, insert x into h with $reach_s(\alpha, x)$.
      If $x \in h$, update its value if $reach_s(\alpha, x)$ is larger.
    Update max-heap h.
3. Return list L_s with RS values for each s value.

We present our algorithm to generate node lists ordered by their reachability scores in Algorithm 1. More specifically, our algorithm creates an ordering of network nodes, additionally storing a reachability score $RS(i)$ for each node i. It starts at a given node n_{start} and inserts n_{start} into a max-heap structure h, which is maintained to store the reachability of candidate nodes. At each step, the node j, which has the highest reachability score in h, is chosen to be the next node in order and the popped score is stored as $RS(j)$. All nodes that are in j's neighbourhood are then inserted into h with their reachability according to j if they are not yet in h. The value in h is updated if the node is already in h and its new score is higher. Then h is updated to maintain its max-heap property. Therefore, the top node of heap h has the highest RS value to one of the nodes that have already been included in the list L, i.e., the RS score for each

node in the list represents its highest reachability from any of the prior nodes in the sequence. The algorithm stops after all nodes in the network are visited.

The computational complexity of ONDOCS is $O(n \log n)$ for dense graphs and $O(n)$ for sparse ones. The list generation and sort step takes $O(c \log cn)$ where constant c is the average number of similar nodes for each node. Note that based on our relationship function, one node can only be similar to another if they are connected or share one or more neighbours. In step 2, there are n insertions to the heap h and updating h for each insertion takes $O(\log n)$ time for dense graphs and $O(1)$ for sparse networks. Thus, the actual running time of our algorithm on experimental networks is $O(n)$ as shown in Section 5.

In summary, given a network with a list of s values, Algorithm 1 produces a sequence of nodes with their reachability scores for each s value, which can be visualized as a 2-D graph by tools such as GNUPlot [28]. The visualizations show interesting community information such that nodes in the same communities are consecutive in the list with high RS scores while the RS score apparently drops between two groups of community nodes (See Figure 3). The goal of visual data mining is to help user acquire accurate parameters by observing this phenomenon, which is presented in the next section. (A detailed example of how to choose the parameters is given in Section 5.2 and Figure 3 after explaining the experiments.)

4.3 Detecting Overlapping Community Structure: Communities, Hubs and Outliers

We have generated lists of nodes given specific s values, where we found that the ordering of the corresponding RS values has interesting community properties. For example, if we start from one node i, we will first visit other nodes in i's community in sequence. This is because the reachability score from i to these nodes are higher than nodes outside i's community. Therefore, each community can be seen as a group of consecutive nodes with high RS scores. In a 2D visualization, these groups are represented as curves in a "mountain" shape or peak. A noticeable drop of subsequent RS scores after a "mountain" indicates that this community has ended, which is represented as a curve in a "valley" shape or trough. The "valley" between two "mountains" represents a set of hubs, which belong to several communities. For instance, if we start from nodes in community α, the fact that hubs have neighbours from different communities makes RS scores of hubs lower than that of those single-community nodes in α but still higher than nodes in communities other than α. Therefore, after all single-community nodes in α are visited, hubs are next to follow before nodes in other communities, which form the "valley" between "mountains."

As we have discussed in the introduction, there is no global community definition, thus communities in specific networks need to be defined by parameters given by the user. For this purpose, our visual data mining approach generates visualizations with different s values first. After the user selects the suitable one based on their observation, they need to further provide two parameters to define the communities in this network, *Community Threshold (CT)* and *Outlier Threshold (OT)*. While such parameters are usually hard to obtain for previous methods, parameter selection for our approach becomes easy since we provides a visualization of the network structure with "mountains" representing strongly related communities, and "valleys" representing hub nodes that connect to both communities. Outliers are usually found at the end of the list, since their RS scores to any other nodes in the network are low. Examples of choosing pa-

rameters for real networks are presented in Section 5. Note that we do not require k, the number of communities to discover, as a parameter. The number of communities is a byproduct of the mining process given the parameters OT and CT which are determined by the user after exploiting our visualization output. The visualization of the network helps the user understand the structure first and then decide about reasonable thresholds for communities and outliers, i.e not the numbers per se but has a similar effect.

Given the two parameters CT and OT, our algorithm works as the following: from the first node in the sequence as the starting community, we scan all nodes along the list. One node n_i is merged into the current community if $RS(n_i) \geq CT$. If $CT > RS(n_i) > OT$, n_i is classified as a hub. If $OT \geq RS(n_i)$, it is an outlier. Since the first node of a community in the list has a low RS score, e.g., the starting node has $RS = 0$, we refine the outlier and hub nodes by moving any node n_i into corresponding communities if we have $RS(n_{i+1}) \geq CT$. (Also see Algorithm 2) The complexity of Algorithm 2 is $\theta(n)$.

Algorithm 2 The ONDOCS Algorithm: Overlapping Community Structure Detection

Input: A list L of nodes $n_0, n_1, ...$ and their RS scores, the Community Threshold CT and the Outlier Threshold OT.
Output: A list of communities $c_0, c_1, ...$, hubs $h_0, h_1, ...$ and outliers $o_0, o_1, ...$
1. Create a community c, set $k = 0$.
2. for each $n_i \in L$ **do**
  If $RS(n_i) \geq CT$, classify n_i as a community node.
    else if $CT > RS(n_i) > OT$, classify n_i as a hub.
    else classify n_i as an outlier.
  end if
  If i is not classified as a community node but $RS(n_{i+1}) \geq CT$
    classify i as a community node.
  end if
  If n_i is a community node, insert n_i into c.
    else (n_i is a hub or an outlier)
      If $|c| != 0$, save c as a community c_k for output
          reset c for the next community, increase index k by 1
      end if
  end if
end for
3. Return communities $c_0, c_1, ...$, hubs $h_0, h_1, ...$ and outliers $o_0, o_1, ...$

To represent that hubs can belong to k communities, for each hub node i, we use a vector of "belonging factors" $v = (f_{(i,1)},\ f_{(i,2)}\ ...\ f_{(i,k)})$ where each coefficient $f_{(i,k)}$ measures the strength of the relationship between node i and community k. For every community C_k, we can quantify the Overall Relationship between i and C_k as

$$OR_{(i,k)} = \begin{cases} \sum_{x \in C_k} R(i,x) & if \ \sum_{x \in C_k} R(i,x) > 0 \\ 0 & otherwise \end{cases}$$

We then normalize the vector to get the coefficients so that we have $\sum_{x=1}^{k} f_{(i,x)} = 1$. Therefore, one node can belong to many communities at the same time, weighted by the relationship value in the range $[0, 1]$ and the sum of belonging coefficients to communities is the same for all nodes in the network, except outliers.

In summary, the community mining process is aided by visual data mining in our approach. Instead of asking the user to arbitrarily provide vital parameters, we

Datasets	Vertices	Edges	Runtime / s			
			CONGO [6]		CF [5]	ONDOCS
			h = 3	h= 2		
football [4]	180	787	8	2	1	< 1
protein_protein [5]	2640	6600	114	11	3	11
blogs [6]	3982	6803	41	8	4	12
PGP [30]	10680	24316	772	104	>20000	62
word_association [5]	7207	31784	15922	230	102	161
blogs2 [6]	30557	82301	15148	380	319	269
cond-mat [31]	27519	116181	> 20000	1486	490	544

Table 2 Comparing Running Time of CONGO, CF and ONDOCS on Real World Networks

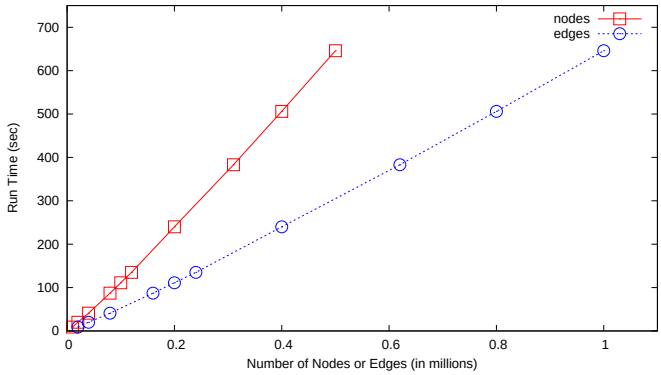


Fig. 2 ONDOCS Algorithm Running Time

generate visualizations of the network in question so that the user is able to observe the structure and relations between communities before they give parameters. After appropriate parameters are determined, hubs and outliers are extracted together with communities. Note that another advantage of our approach is that while parameters are easy to be altered, the impact on the change of discovered communities can be clearly perceived by observing the visualization.

5 Experiment Results

Here we evaluate the ONDOCS approach using both synthetic and real world datasets. The performance of ONDOCS is compared with CFinder [5] and CONGO [6], which are shown to be two of the most efficient algorithms for finding overlapping community structure [6]. The comparison is measured by the well known F-measure score and Adjusted Rand Index (ARI) [29]. All experiments were conducted on a PC with a 3.0 GHz Xeon processor and 4GB of RAM.

5.1 ONDOCS Scalability

To evaluate the scalability of our algorithm, we generated ten random graphs of vertices ranging from 10,000 to 500,000 and the number of edges ranging from 20,000 to 1,000,000. The edges are randomly distributed in the network. Figure 2 shows the

performance of our algorithm on those networks. It clearly shows that, although the running time of ONDOCS is $O(n \log n)$ in the worst case, our approach actually runs very close to linear time with respect to the number of vertices and edges.

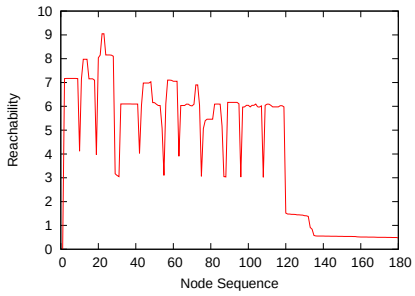
To further evaluate the efficiency of the algorithm, we apply three algorithms on several real-world networks. Table 2 shows the source of each network, its statistics, and the execution times for CONGO to compute the entire dendrogram, CFinder (v1.21) to generate solutions for $3 \leq k \leq 8$ and ONDOCS to create dataset visualizations for $2 \leq s \leq 8$. From the table, we can see that ONDOCS works well overall, while CONGO's running time increases dramatically with respect to h and CF's clique detection becomes slow on some particular networks. (Note that it may seem to be unfair to compare since ONDOCS merely generates visualizations but not communities yet. However, the intent of run time comparison is to demonstrate that our approach is no more time consuming than previous methods but on the contrary in most cases faster. Additionally, the complexity of extracting communities after parameter setting, i.e, selecting CT and OT, is negligible compared to the visualization generation.) Unfortunately, we do not have ground truth to validate the accuracy of our results for these datasets, thus we turn to several real world datasets with ground truth to evaluate the accuracy of our approach.

5.2 ONDOCS Accuracy

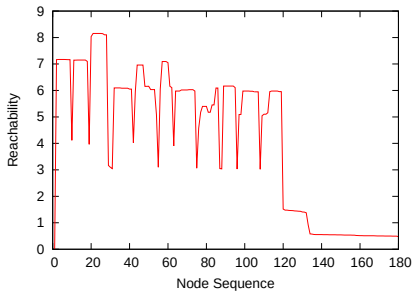
The first dataset we examine is the schedule for 787 games of 2006 National Collegiate Athletic Association (NCAA) Football Bowl Subdivision (also known as Division 1-A) [4]. In the NCAA network, there are 115 universities divided into 11 conferences. In addition, there are four independent schools at this level, namely Navy, Army, Notre Dame and Temple, as well as 61 schools from lower divisions. Each school in the division 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 different conferences, while lower division teams play only very 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.

First, the ONDOCS approach generates several visualizations with different s values for the user to choose. We show all visualizations for $2 \leq s \leq 8$ in Figure 3. As we can see, most images are very similar to each other. The only one that shows a different structure is the visualization for $s = 8$. Recall that the parameter s represents the expected number of nodes that one node is similar with in order to be considered as a community member. When s is raised to a large value, some communities might disappear if their size is smaller than s. In this case, ONDOCS visualizations only show the structure of communities whose size is greater or equal to s. The larger the s value is, the smoother the curves are and the fewer "spikes" we have. Nevertheless, we have seven visualizations that clearly represent the network structure, where there are 11 communities, a few hubs and a set of outliers.

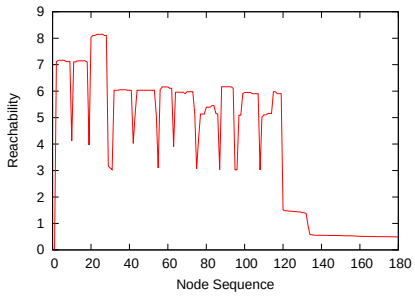
The parameter selection is solely based on users' visual interpretation of the visualized network. First we choose the visualization with $s = 2$, where the community structure is shown in most detail since pair relations are mostly measured as direct distance. In Figure 4, we note that nodes in sequence from 120 to 180 are barely related to the rest and can be considered as outliers, therefore we set $OT = 2$. Note that OT can also be set as 2.5, or any other close number. Different OT value will not give



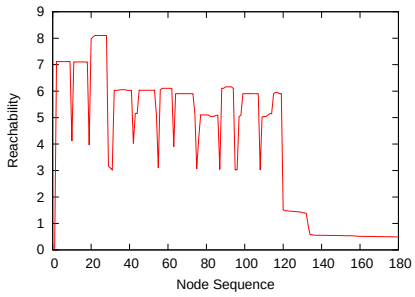
(a) S=3



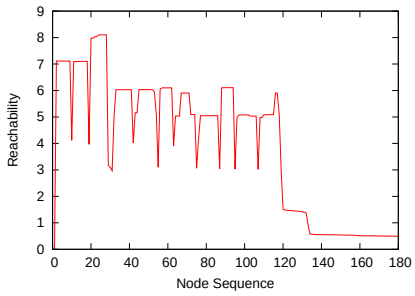
(b) S=4



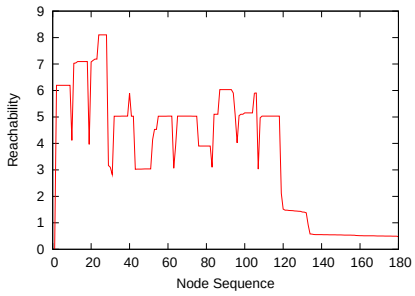
(c) S=5



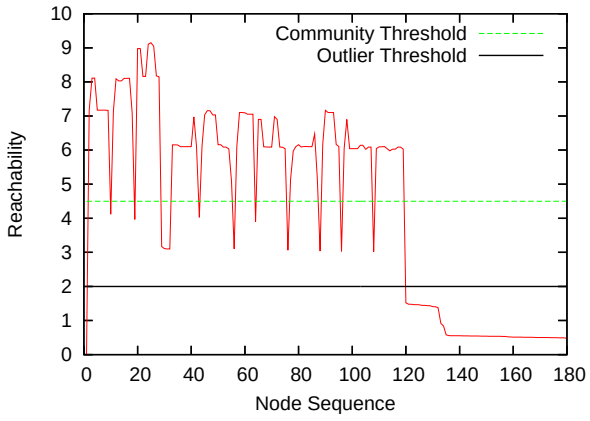
(d) S=6



(e) S=7



(f) S=8



(g) S=2

Fig. 3 Community Visualizations of the football network with different S value

Data Setting		Algorithms		
		CONGO (h=2)	CF (k=4)	ONDOCS (s=2) (CT = 4.5, OT = 2)
115 Nodes in 11 Clusters	Cluster	11*	11	11
	Hub	92	6	0
	ARI	0.047	0.945	1.00
Plus 4 Hubs	Cluster	11*	12	11
	Hub	100	8	3
	Hub F-measure	0.038	0.167	0.857
Plus 4 Hubs and 61 Outliers	Cluster	11*	12	11
	Hub	96	8	3
	Hub F-measure	0.04	0.167	0.857
	Outlier	0	61	61
	Outlier F-measure	0	1.00	1.00

Table 3 Comparing Algorithm Accuracy of CONGO, CF and ONDOCS on the Football Dataset. (*The right cluster number is provided as a parameter for the CONGO algorithm.)

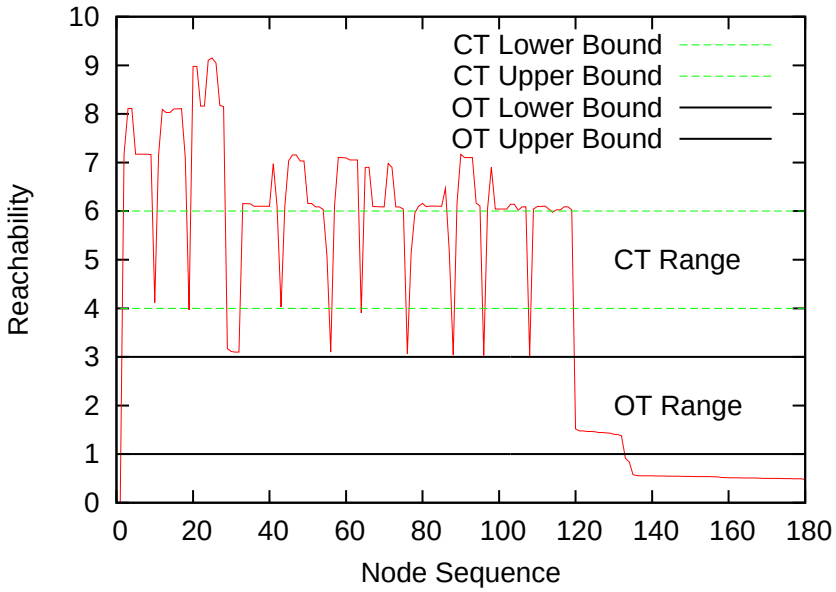


Fig. 4 Selecting CT and OT for ONDOCS

CT	OT = 2					OT	CT = 4.5				
	Cluster	Hub	H-FM	Outlier	O-FM		Cluster	Hub	H-FM	Outlier	O-FM
4.0	9	3	0.857	61	1.0	1.0	11	16	0.30	48	0.880
4.5	11	3	0.857	61	1.0	1.5	11	4	0.75	60	0.991
5.0	11	3	0.857	61	1.0	2.0	11	3	0.857	61	1.0
5.5	11	6	0.8	61	1.0	2.5	11	3	0.857	61	1.0
6.0	12	7	0.77	61	1.0	3.0	11	3	0.857	61	1.0

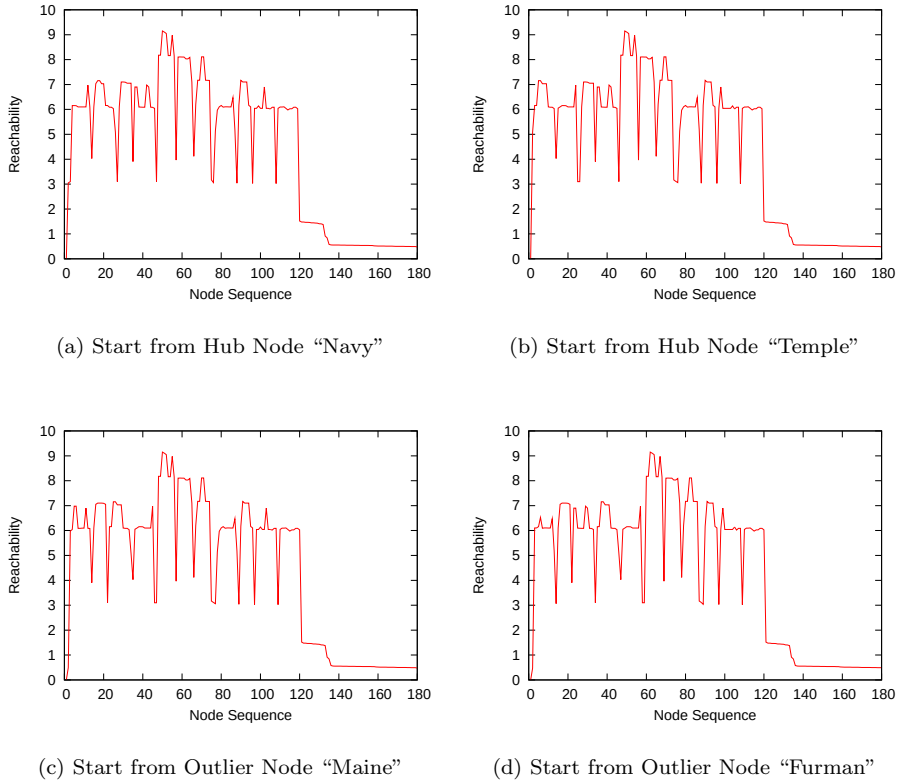
Table 4 Comparing ONDOCS Accuracy with Different CT and OT. (H-FM means F-measure for Hubs and O-FM means F-measure for Outliers.x)

completely different results and the impact can be perceived directly from the visualization. Furthermore, we see a community usually ends with a RS score between 3 and 5, thus we set $CT = 4.5$ so that all communities are separated. The range of possible thresholds are shown in the figure. Table 4 shows results of varying CT and OT in the range. As can be noticed, it is quite easy for one to select parameters given the network visualization, and the results are stable enough for a large range of parameters.

To evaluate how algorithms detect overlapping community structure, we provide the data to our algorithms in three different ways. At first, we give only 115 community nodes and connections between them, then we measure the accuracy of discovered communities by the ARI score based on the ground truth, which is the conference assignment. Then we add the 4 hubs and their connections into the network. Although these hubs clearly belong to multiple communities, we do not have exact ground truth for overlapping community structure, i.e., which communities these hubs should go. However, we do have ground truth for which nodes are hubs (outliers) and which are not. Therefore, we measure the accuracy of the output hubs and outliers by the F-measure score, which is defined as the harmonic mean of precision and recall. Finally we give the complete network with communities, hubs and outliers. Table 3 shows the experiment results for the three algorithms. As we can see, the CONGO algorithm always detects overlaps, even for the first network where there are only community nodes. Additionally, it requires the cluster number as the input parameter, which is usually unavailable for real world networks, and it still fails to find any outliers. The CF algorithm gives its best result when $k = 4$, where it detects all outliers and finds 12 clusters, which is very close to the truth. However, CF also finds hubs when there is no overlap and the accuracy of its overlap detection is low with only a 0.167 F-measure score. Our ONDOCS algorithm works the best overall. It finds all outliers and only detects hubs when there is indeed some overlap between communities. The hub detection accuracy is not perfect, however, when we look into the data, we find out that the only missing hub team (Temple) plays half of its games (6 out of 12) with teams from the Mid-American conference, which explains why it is classified into that community. Note that the result of our algorithm depends on two parameters (CT and OT), however, we believe that appropriate values are easy to find based on direct observation on network visualizations.

In ONDOCS, the node sequence might change if we choose different node n_{start} to start with. For previous experiments, we choose a community node to start the process. In Figure 5, visualizations that start from hub nodes and outlier nodes are shown. However, as we can see, a community, represented by a "mountain" curve, is found first. It is because our algorithm intends to visit the closest nodes in the sequence, which have higher RS scores, before nodes that are far away. Thus, no matter where the start node is, the closest community is found first, followed by other communities ordered by their RS values. Hubs are found as "valley" between communities.

We also apply our algorithm on other real world networks, including the Political Book network [32], the Mexican Politician network [33], the Dolphin network [3] and the Les Miserables network [34]. Although we do not have exact overlapping truth for these networks, approximate community structure information is provided by previous research. In the Political Books dataset, nodes represent political books sold by Amazon.com and edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these items" feature on Amazon. Nodes are manually labeled as "Liberal," "Neutral," or "Conservative" by Mark Newman [35]; In the Mexican Politicians dataset, edges indicate social re-



(a) Start from Hub Node "Navy"

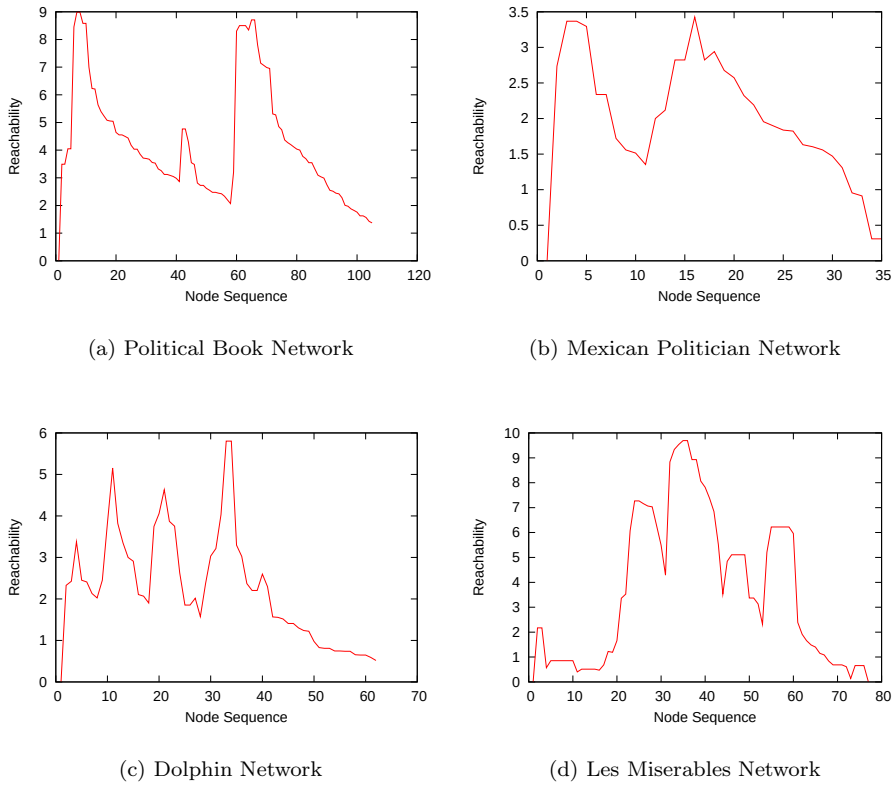
(b) Start from Hub Node "Temple"

(c) Start from Outlier Node "Maine"

(d) Start from Outlier Node "Furman"

Fig. 5 ONDOCS Visualizations with different starting nodes

lations between people and nodes represent politicians, who are classified based on their background as "Citizen" or "Military". The Dolphin Network gives the community structure of a group of bottle-nose dolphins. The network can be approximately divided into four main groups [3]. Finally, the Les Miserables network represents the coappearance network of characters in the novel Les Miserables. Note that for these datasets, we only have indefinite community information instead of perfect ground truth, which is the common case for overlapping community detection and evaluation. We show visualizations for these datasets generated by ONDOCS in Figure 6. One can see that the images correctly depict the approximate community information we have. Accurate CT and OT values should be easy to determine based on these figures. Also note that if the reachability plots are not clear for some datasets, the users may have problems selecting parameters. This could be the case when a large number of real communities exist, where the plot would present a jagged graph with many close peaks for a vague community structure. This is a limitation of the visualization and may be addressed by increasing the screen real-estate or a progressive hierarchical method, which selects parameters for each level of the community hierarchy. However, it is nevertheless reasonable to believe that other approaches with no visual data mining



(a) Political Book Network

(b) Mexican Politician Network

(c) Dolphin Network

(d) Les Miserables Network

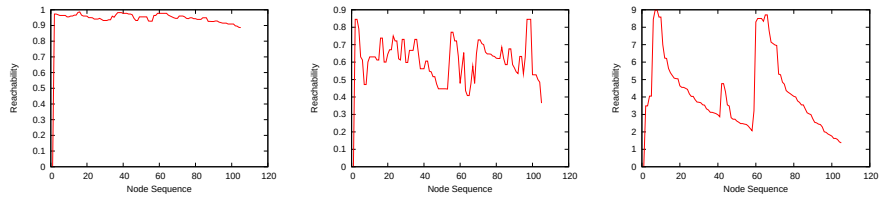
Fig. 6 Community Visualizations for Various Networks by ONDOCS

support, when faced with a large number of existing communities, would provide less information and do even worse in the mining process.

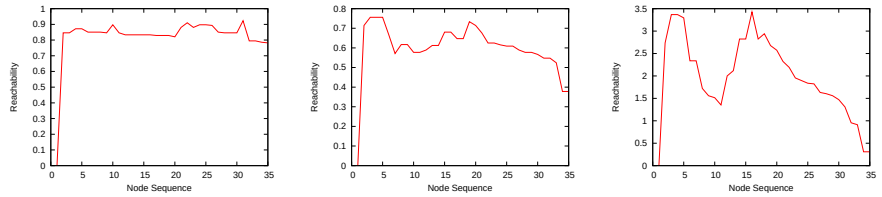
5.3 Comparing Metrics within ONDOCS

We have reviewed previous community mining metrics (Q and S) and proposed our relational metric R. We then evaluated them from a theoretical perspective. Here we apply these three metrics to measure the similarity between two nodes in our ONDOCS system and compare the images generated for several real world datasets respectively in order to further evaluate the effectiveness of the metrics.

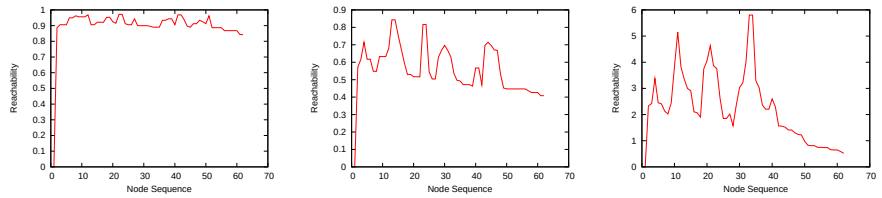
The visualizations for four different datasets based on metrics Q, S and R are shown in Figure 7(a) to 7(l) respectively (s is set to 2 for all metrics). We see that the plots using the R metric accurately depict the network structure since they match the vague community information that we have for those datasets. On the other hand, visualizations using the S metric are ambiguous and the community structure is hard to read. Also note that the R visualizations provide a much wider range for the user to observe accurate CT and OT values to detect the right number of communities



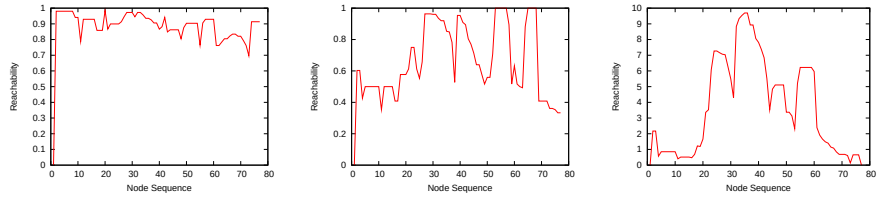
(a) Political Book by Q (b) Political Book by S (c) Political Book by R



(d) Mexican Politics by Q (e) Mexican Politics by S (f) Mexican Politics by R



(g) Dolphin Network by Q (h) Dolphin Network by S (i) Dolphin Network by R



(j) Les Miserables by Q (k) Les Miserables by S (l) Les Miserables by R

Fig. 7 Comparing Metric Q, S and R with ONDOCS Visualizations

than the S visualizations. Finally, visualizations based on the Q metric do not show any community structure. The reason is that Q does not consider local structure thus similarity scores of all node pairs are smaller than and close to 1 after node ordering, which makes the plots into a nearly-horizontal line.

6 Conclusions

In this paper, we first propose a general definition of communities in social networks and a list of requirements for a good similarity metric to detect those communities. We analyze existing metrics based on those criteria and then propose a new similarity metric R which satisfies all of those requirements. A visual data mining approach for overlapping community detection in networks is then proposed based on metric R. The method first generates lists of nodes, ordered by their reachability scores. Network visualizations are then provided to help the user determine important parameters. Finally, overlapping community structure, i.e., communities, hubs and outliers, are extracted based on these parameters. Experiment results show that our approach not only scales well for large networks, but also achieves a high accuracy for real world networks. Unlike previous approaches, our method only detects overlap when overlap exists. Moreover, appropriate parameters are easy to obtain by means of visual data mining. The effectiveness of R over previous metrics are also confirmed by comparing ONDOCS visualizations.

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).

References

1. Gregory, S.: An algorithm to find overlapping community structure in networks. In: PKDD, pp. 91–102 (2007)
2. 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)
3. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004)
4. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: Scan: a structural clustering algorithm for networks. In: KDD, pp. 824–833 (2007)
5. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**, 814 (2005)
6. Gregory, S.: A fast algorithm to find overlapping communities in networks. In: PKDD, pp. 408–423 (2008)
7. Nepusz, T., Petróczi, A., Negyessy, L., Bazso; F.: Fuzzy communities and the concept of bridgeness in complex networks. Physical Review E **77, 016107** (2008)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE. Trans. on Pattern Analysis and Machine Intelligence (2000)
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. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E **69** (2004)

11. Danon, L., Duch, J., Diaz-Guilera, A., Arenas, A.: Comparing community structure identification. J. Stat. Mech p. P09008 (2005)
12. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**, 066,111 (2004)
13. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E **74** (2006)
14. Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Phys. Rev. E **72**, 027,104 (2005)
15. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: SIAM (2005)
16. Guimera, R., Amaral, L.A.N.: Functional cartography of complex metabolic networks. Nature **433**, 895–900 (2005)
17. Wei, F., Wang, C., Ma, L., Zhou, A.: Detecting overlapping community structures in networks with global partition and local expansion. In: APWeb, pp. 43–55 (2008)
18. Li, X., Liu, B., Yu, P.S.: Discovering overlapping communities of named entities. In: PKDD, pp. 593–600 (2006)
19. Baumes, J., Goldberg, M.K., Magdon-Ismail, M.: Efficient identification of overlapping communities. In: ISI, pp. 27–36 (2005)
20. 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)
21. Ankerst, M., Keim, D.A.: Visual data mining (Tutorial at SIAM Int. Conf on Data Mining 2003)
22. Ankerst, M., Elsen, C., Ester, M., Kriegel, H.P.: Visual classification: an interactive approach to decision tree construction. In: KDD, pp. 392–396 (1999)
23. Ankerst, M., Ester, M., Kriegel, H.P.: Towards an effective cooperation of the user and the computer for classification. In: KDD, pp. 179–188 (2000)
24. Han, J., Cercone, N.: Ruleviz: a model for visualizing knowledge discovery process. In: KDD, pp. 244–253 (2000)
25. Teoh, S.T., Ma, K.L.: Paintingclass: interactive construction, visualization and exploration of decision trees. In: KDD, pp. 667–672 (2003)
26. Zaïane, O.R., Foss, A., Lee, C.H., Wang, W.: On data clustering analysis: Scalability, constraints, and validation. In: PAKDD, pp. 28–39 (2002)
27. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: SIGMOD, pp. 49–60 (1999)
28. Gnuplot: http://www.gnuplot.info/
29. Yip, K.Y., Ng, M.K.: Harp: A practical projected clustering algorithm. IEEE TKDE **16**(11), 1387–1397 (2004)
30. Boguñá, M., Pastor-Satorras, R., Díaz-Guilera, A., Arenas, A.: Models of social networks based on social distance attachment. Phys. Rev. E **70**(5), 056,122 (2004)
31. Newman, M.E.J.: The structure of scientific collaboration networks. In: PNAS USA, 98:404-409 (2001)
32. Krebs, V.: http://www.orgnet.com/
33. Pajek: http://vlado.fmf.uni-lj.si/pub/networks/pajek/
34. Knuth, D.E.: The stanford graphbase: A platform for combinatorial computing (Addison-Wesley, Reading, MA (1993))
35. Newman, M.: http://www-personal.umich.edu/∼mejn/netdata/