Detecting Communities in Large Networks by Iterative Local Expansion

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

Abstract

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, which means that there exists densely connected groups of vertices, with only sparser connections between groups. Identifying community structure in networks has attracted much research attention. However, most existing approaches require structure information of the graph in question to be completely accessible, which is impractical for some large networks, e.g., the World Wide Web (WWW). In this paper, we propose a community discovery algorithm for large networks that iteratively finds communities based on local information only. We compare our algorithm with previous global approaches to show its scalability. Experimental results on real world networks, such as the co-purchase network from Amazon, verify the feasibility and effectiveness of our approach.

1. Introduction

Many datasets can be represented as networks composed of vertices and edges. Examples include the World Wide Web (WWW) (e.g., the web page hyperlink network [1]), organization structures [2], academic collaboration records [3], [4], friendship network [5], biological networks (e.g., neural networks [6] and food webs [7]), and even political elections [8]. There are several definitions for communities in the network, e.g., a community 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 [9]. From that perspective, identifying communities can be seen as finding node clusters in a graph. In this paper, we define a community to be a social network partition such that entities within the same community share some common trait or proximity, judged by some predefined entity similarity or relationship metric. Identifying and locating entities in different communities is one of the main goals of the community mining research.

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 [10]; groups within social networks might correspond to social communities, which can be used to understand organization structures. Moreover, the community structure influence 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 analyses that focus on whole networks and ignore community structure may miss many interesting features [11]. For example, we may find that individuals within different community groups have a different mean number of contacts in some social networks: the individuals in one group might have many contacts with others while the others in another group might be more reticent. Examples of such social networks are reported in [12], [13] as 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 [14].

The problem of finding communities in social networks has been studied for decades. Recently, several quality metrics for community structure have been proposed [15], [16], [17]. Among them, modularity Q has proved to be the most accurate [18] and has been pursued by many researchers [19], [20], [21], [11], [22]. However, most of those approaches require knowledge of the entire graph structure. This constraint is problematic for networks which are either too large or too dynamic to know completely, e.g., the WWW. In spite of these limitations, finding local community structure would still be useful, albeit confined by the little accessible information of the network in question. For example, we might like to quantify the local communities of either a particular webpage given its link structure in the WWW, or a person given his social network in Facebook. Existing approaches [16], [19] 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). In this paper, we propose a new algorithm 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 with previous global approaches to evaluate its scalability and apply our approach on large real world networks to show its capability. In contrast to existing approaches, our approach 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 works in Section 2. Section 3 defines the problem and presents the local modularity metric. 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, naïvely applying traditional statistical inference procedures, which assume that instances are independent, can lead to inappropriate conclusions about the data [23]. 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 [24]. 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. 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, e.g., multilevel partitioning [25], k-partite graph partitioning [26], relational clustering [27], flow-based methods [10], information-theoretic methods [28] and spectral clustering [29]. 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* [30] and the *min-max cut* [31]. 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 [19], [33], [15], [16]. 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 [9], [34], [35], [36] have been proposed for overlapping structure. Recent work by Xu et al. [17] 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 [37] and Luo et al. [38] 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 [39], 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 [40] to measure

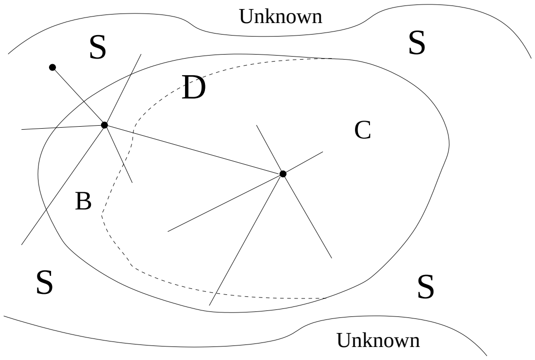

Figure 1. Local Community Definition

local structure, however, their method lacks an appropriate stopping criteria and thus still relies on arbitrary thresholds.

3. Preliminaries

As mentioned in the introduction, local communities are densely-connected node sets that are discovered and evaluated based only on local information, since global information is impossible to access. In this section, we first define the research problem of finding local communities in a network, then present a metric we adopt in our algorithm.

3.1. Problem Definition

Suppose that in an undirected network G (directed networks are usually transformed to undirected ones first), we start with perfect knowledge of the connectivity of one node or some set of nodes, i.e., the known local portion of the graph, which we denote as D. This necessarily implies that we also have limited information for another shell node set S, which contains nodes that are adjacent to nodes in D but do not belong to D (note "limited" means that the complete connectivity information of any node in S is unknown). In such circumstances, the only way to gain additional information about the network G is to visit some neighbour nodes s_i of D (where $s_i \in S$) and obtain a list of adjacencies of s_i. As a result, s_i is removed from S and becomes a member of D while additional nodes may be added to S as neighbours of s_i. This typical one-node-at-one-step discovery process for local community detection is analogous to the method that is used by web crawling systems to explore the WWW. Furthermore, we define two subsets of D: the core node set C, where any node $c_i \in C$ has no outward links, i.e., all neighbours of c_i belong to D; and the boundary node set B, where any node $b_i \in B$ have 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 [40], [39], [37], [38], however, the metrics used in these approaches to discover and evaluate the local community are different.

3.2. Local Community Discovery Metrics

Clauset has proposed the local modularity R for the local community evaluation problem [37]. Intuitively, we hope that a community would have a sharp boundary which has fewer connections from the boundary to the unknown portion of the graph, while having a greater number of connections from the boundary nodes back into the local community. Therefore, a good measure could be of the sharpness of the boundary of a community, where boundary nodes have at least one neighbour outside the community. In other words, 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. Thus, R measures the fraction of those "inside-community" edges in all edges with one or more endpoints in B and community D is measured by the sharpness of the boundary given by B. By considering the fraction of internal boundary edges, R lies on the interval $0 < R < 1$. Additionally, this measure is independent of the size of the enclosed local community.

Similarly, Luo et al. later proposed the modularity M [38] 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 and is equivalent in some situations. 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 $R \geq 0.5$.

The metrics to evaluate local communities are straightforward. Several algorithms [37], [38] are proposed based on them to identify local communities. However, their performance relies on arbitrary parameters, such as the number of agglomerated nodes or a community threshold of the ratio of internal and external edges. Moreover, they usually focus on the first enclosing community and stop further identification, leaving other parts of the graph unexplored and the possibility of discovering other communities uncharted.

4. Our Approach

Existing metrics discussed in Section 3 are simple. However, 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 learn more knowledge about the structure is by expanding the community, by one node at a time. With all these limitations in mind, we present our algorithm.

Generally speaking, our algorithm consists of two steps. Given a node and its local information, our approach first identifies the local community for this node, and then iteratively applies the same procedure to cover the whole graph. In the following, we present these two steps and then discuss other advantages of our approach.

4.1. Identifying Local Community

We have introduced a metric to evaluate the quality of a local community in Section 3. The higher the R value is, the better a group can be considered as a community. Therefore, given a start node in a community, we could naturally optimize the R value to identify the local community structure. See Algorithm 1.

Algorithm 1 Local Community Identification Algorithm

```
Input: A social network $G$ and a start node $n_0$.
Output: A local community for $n_0$ with its quality score $R$.
1. Add $n_0$ to $D$ and $B$, add all $n_0$'s neighbours to $S$.
2. do
      for each $n_i \in S$ do
        compute $R'_i$
      end for
    Find $n_i$ with the maximum $R'_i$, breaking ties randomly
      Add $n_i$ to $D$
      Remove $n_i$ from $S$.
      Update $B$, $S$, $R$
   While ($R' > R$)
3. Return $D$ as $n_0$'s local community.
```

In Algorithm 1, we place the start node in the community, and its neighbour in the shell node set. At each step, the algorithm adds to the community the neighbour node that gives the largest increase of R, breaking ties randomly. We then update the community set, the boundary set, the shell node set and the R value. We continue this process until there are no candidate nodes that could give positive value to the community. Having $R = \frac{B_{in}}{B_{total}}$, we assume by merging node n_i, B_{in} will increase by Δ_{in}, which is the number of edges that connect from original community nodes to n_i; B_{total} will increase by Δ_{total}, which is the number of edges that connect from n_i to other nodes except ones within the community; B_{total} will also decrease by Δ', since merging n_i might change the boundary status of some community nodes, their connections will be taken off from B_{total}. Now, the computation of each R'_i can be done using the following expression.

$$\begin{aligned} R'_i &= R' - R \\ &= \frac{B_{in} + \Delta_{in} - \Delta'}{B_{total} + \Delta_{total} - \Delta'} - \frac{B_{in}}{B_{total}} \\ &= \frac{\Delta_{in} - \Delta_{total} * R - \Delta' * (1 - R)}{B_{total} + \Delta_{total} - \Delta'} \end{aligned}$$

At each step that merges n_i to the community, the algorithm needs to compute R' for every node in the shell node set to find out the one with the maximum increase, thus the complexity of each step is $O(d|S|)$, where d is the mean degree of the graph. If the size of the discovered local community is k, the complexity of the algorithm becomes $O(kd|S|)$. However, in real world 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 would be dominated by this time-consuming network information retrieval. Therefore, for real world problems the running time of this procedure is linear in the size of the community, i.e., $O(k)$.

4.2. Iterative Local Expansion

Algorithm 1 is for identifying a local community for a specific set of starting nodes, however, we could apply this algorithm iteratively to cover the whole graph. In other words, instead of one-node-at-one-step, we expand as one-community-at-one-step. See Algorithm 2.

Algorithm 2 Iterative Expansion Algorithm

```
Input: A social network $G$ and a start node $n_0$.
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 ($|S| != 0$)
      Randomly pick one node $n_i \in S$.
      Apply algorithm 1 to find a local community $l_i$ for $n_i$.
      Remove nodes in $S$ that are covered by $l_i$.
      Update $S$ by neighbours of $l_i$ that are not covered yet.
4. $m$ local communities $l_0, l_1, l_2 ...$, $m$ could be given as a stop parameter.
```

In algorithm 2, we recursively apply the local community identification algorithm to expand the community structure. Every time we find a local community, we update the shell

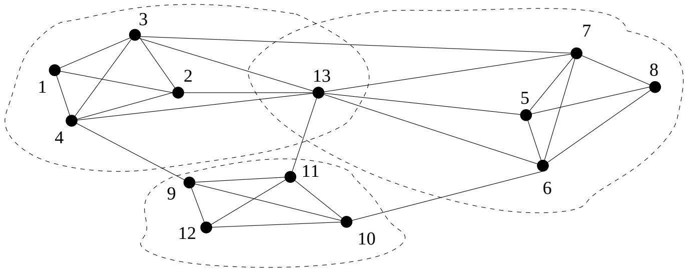

Figure 2. An Example for Overlapping Communities

node set, which is actually a set of nodes whose community information is still unclear. 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 have multiple starting nodes, where several local community identification procedures start simultaneously from different locations of the network. Obviously, the complexity of the Algorithm 2 is still $O(kd|S|)$.

4.3. Detecting Overlapping Communities

As previously noted, in real world network, one entity usually belongs to multiple communities. However, most of the existing approaches cannot identify overlapping communities. Fortunately, our approach is able to discover overlapping communities even though we do not specifically focus on finding such community property. For example, in Figure 2 we have a simple network with 13 nodes. It is easy to identify that nodes 1 to 4, 5 to 8 and 9 to 12 are three local communities since they are cliques. However, node 13 seems to belong to two communities at the same time, since it connects to 3 of the 4 nodes in both communities. While other algorithms might mistakenly classify node 13 to only one community, our approach could detect this overlap without requiring any arbitrary parameters. Assume we start from node 1, the discovered local community is nodes 1 to 4 and node 13. The algorithm then randomly turns to node 9 and discovers the community for nodes 9 to 12. Finally, it identifies the community of nodes 5 to 8 and node 13. The fact that node 13 is already classified into another community does not affect the decision of our algorithm, which is made based on the available local network structure.

5. Experiment Results

In this section, we apply our iterative local expansion algorithm to detect communities on various real world social networks. Danon et al. [18] found that the modularity method outperformed all other methods for community detection of which they were aware, in most cases by an impressive margin, thus maximization of the modularity to be perhaps the definitive state of the art method of community detection. Therefore, we compared our approach with a hierarchical clustering algorithm FastModularity [19], which uses Newman's modularity to measure community structure, to show the scalability on large networks. We then apply our algorithm on the co-purchase network of Amazon to show its effectiveness. All the experiments were conducted on a PC with a 3.0 GHz Xeon processor and 4GB of RAM.

5.1. Scalability

To evaluate the scalability, we apply our algorithm and FastModularity on several real world networks. Table 1 shows the source of each network, its statistics and the execution time. From the table, we can see that our algorithm runs measurably faster than FastModularity overall. Since the complexity of our approach is $O(kd|S|)$, our algorithm performs better in sparser networks. For example, it is faster for the PGP network and blogs2 network, where the mean degree is only about 2, and spends more time on dense networks, such as the word association network and cond-mat network. Note that while FastModularity requires complete network structure information, our algorithm starts with local information only, then expands to the whole available graph, thus it is more practical for huge networks. Also note that another community detection algorithm to possibly compare with is SCAN [17], however, the performance of SCAN relies on input parameters, which are very sensitive and extremely hard to determine for real world networks, especially when the global network information is not available.

5.2. Discovering Communities in Amazon Copurchase Network

While these networks provide diverse testbeds for scalability evaluation, it is also desirable to interpret 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 to the recommendation network of Amazon.com, collected in January 2006 [38]. 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 by customers, as indicated by the "customers who bought this book also bought these items" feature on Amazon. There are 585,283 nodes and 3,448,754 undirected edges in this network with a mean degree of 5.89. Similar datasets have been used for testing in previous works [37], [38].

Table 2 shows four local community examples of our result and their start items. The first community only has

Datasets	Vertices	Edges	Mean Degree	Runtime / s	
				FastModularity [19]	Our Algorithm
football [17]	180	787	4.17	< 1 s	< 1 s
blogs [41]	3,982	6,803	1.71	9 s	1 s
PGP [42]	10,680	24,316	2.28	28 s	2 s
word_association [35]	7,207	31,784	4.41	38 s	35 s
blogs2 [41]	30,557	82,301	2.69	201 s	67 s
cond-mat [43]	27,519	116,181	4.22	226 s	130 s

Table 1. Results on Real World Networks

five nodes, originated at the book *Aesop's Fables*. It naturally includes other fairy tale books, such as the book of *The 1001 Nights*. The second community includes 197 books, most of them focus on the topic of the great author William Shakespeare. Similarly, we have another 28-node-community about the legendary German musician Beethoven. Finally, the fourth community includes 101 books of war, such as Civil War, World War I and II. Note that many other community detection algorithms, e.g., FastModularity, 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 book *The Musician's Soul: A Journey Examining Spirituality for Performers, Teachers, Composers, Conductors, and Music Educators* is found in a community originated from the book *Classical Music in America: A History of Its Rise and Fall* and another community originated from the book *Choral Masterworks: A Listener's Guide*. Another example is the book *Letters of Wolfgang Amadeus Mozart*; it belongs to the community of the book *Beethoven* and the community of the book *Mozart: A Cultural Biography*. We could justify there is indeed some overlap by the book names and history knowledge.

6. Conclusions

In this paper, we propose an iterative local expansion approach to detect communities for large networks. While previous approaches may have problems with huge networks when the global structure information is unavailable, our method tackles the problem by evaluating the community structure by a local metric and then repeats that procedure to generate communities to cover the whole network, without requiring any arbitrary parameters. We have tested our algorithm on the Amazon co-purchase network to evaluate its accuracy. We have also compared its performance with previous approaches on real world networks to show its scalability. Experimental results confirm the effectiveness of our approach.

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.

References

[1] J. Kleinberg, "Authoritative sources in a hyperlinked environment," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

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

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

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

[5] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *KDD*, 2006, pp. 44–54.

[6] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[7] R. J. Williams and N. D. Martinez, "Simple rules yield complex food webs," *Nature*, vol. 404, pp. 180–183, 2000.

[8] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 u.s. election: divided they blog," in *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, 2005, pp. 36–43.

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

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

	Items (Books) in the Local Communities
start	*Aesop’s Fables (Oxford World’s Classics)*
1	The Complete Fables (Penguin Classics)
2	Fables: Babrius and Phaedrus (Loeb Classical Library No. 436)
3	Fables of Aesop According to Sir Roger L’Estrange ...
4	Aesop’s Fables (Puffin Classics)
5	The Book of the Thousand Nights and One Night
start	*Shakespeare After All*
1	The Age of Shakespeare (Modern Library Chronicles)
2	A Year in the Life of William Shakespeare: 1599
3	Shakespeare: The Invention of the Human
4	Essential Shakespeare Handbook
5	Imagining Shakespeare
6	Shakespeare’s Language
7	Shakespeare’s Words: A Glossary and Language Companion
8	Shakespeare: Modern Essays in Criticism
...	...
197	Shakespeare and the Bible
start	*Beethoven*
1	Late Beethoven: Music, Thought, Imagination
2	Letters of Wolfgang Amadeus Mozart
3	Beethoven and His Nine Symphonies
4	Beethoven: The Music and the Life
5	Beethoven: The Man and the Artist, As Revealed in His Own Words
6	Beethoven: Impressions by His Contemporaries
...	...
28	Beethoven’s Ninth: A Political History
start	*1776*
1	A Traveler’s Guide to D-Day and the Battle for Normandy
2	A Tour of the Bulge Battlefield
3	A Traveler’s Guide to the Battle for the German Frontier
4	Michelin Road to Liberty Map
5	A Short History of the Civil War at Sea
...	...
101	The 25 Best Civil War Sites: The Ultimate Traveler’s Guide ...

Table 2. A Local Community Example for the Amazon Network.

[11] M. E. J. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E*, vol. 74, 2006.

[12] S. O. Aral, J. P. Hughes, B. Stoner, W. Whittington, H. H. Handsfield, R. M. Anderson, and K. K. Holmes, “Sexual mixing patterns in the spread of gonococcal and chlamydial infections,” *American Journal of Public Health*, vol. 89, pp. 825–833, 1999.

[13] G. P. Garnett, J. P. Hughes, R. M. Anderson, B. P. Stoner, S. O. Aral, W. L. Whittington, H. H. Handsfield, and K. K. Holmes, “Sexual mixing patterns of patients attending sexually transmitted diseases clinics,” *Sexually Transmitted Diseases*, vol. 23, pp. 248–257, 1996.

[14] S. Gupta, R. M. Anderson, and R. M. May, “Networks of sexual contacts: Implications for the pattern of spread of hiv,” *AIDS*, vol. 3, pp. 807–817, 1989.

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

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

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

[18] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, “Comparing community structure identification,” *J. Stat. Mech*, p. P09008, 2005.

[19] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Phys. Rev. E*, vol. 70, p. 066111, 2004.

[20] J. Duch and A. Arenas, “Community detection in complex networks using extremal optimization,” *Phys. Rev. E*, vol. 72, p. 027104, 2005.

[21] R. Guimera and L. A. N. Amaral, “Functional cartography of complex metabolic networks,” *Nature*, vol. 433, pp. 895–900, 2005.

[22] S. White and P. Smyth, “A spectral clustering approach to finding communities in graphs,” in *Proceedings of the 5th SIAM International Conference on Data Mining*, 2005.

[23] D. Jensen, “Statistical challenges to inductive inference in linked data,” 1999.

[24] M. E. J. Newman, “Modularity and community structure in networks,” *PROC.NATL.ACAD.SCI.USA*, vol. 103, 2006.

[25] G. Karypis and V. Kumar, “Multilevel k-way partitioning scheme for irregular graphs,” *Journal of Parallel and Distriuted Computing*, vol. 48, no. 1, pp. 96–129, 1998.

[26] B. Long, X. Wu, Z. Zhang, and P. S. Yu, “Unsupervised learning on k-partite graphs,” in *KDD*, 2006, pp. 317–326.

[27] B. Long, Z. Zhang, and P. S. Yu, “A probabilistic framework for relational clustering,” in *KDD*, 2007, pp. 470–479.

[28] I. S. Dhillon, S. Mallela, and D. S. Modha, “Information-theoretic co-clustering,” in *KDD*, 2003, pp. 89–98.

[29] J. Ruan and W. Zhang, “An efficient spectral algorithm for network community discovery and its applications to biological and social networks,” in *ICDM*, 2007, pp. 643–648.

[30] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[31] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, “A min-max cut algorithm for graph partitioning and data clustering,” in *ICDM ’01: Proceedings of the 2001 IEEE International Conference on Data Mining*, 2001, pp. 107–114.

[32] J. Scott, “Social network analysis: A handbook,” Sage, London 2nd edition(2000).

[33] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” in *Proceedings of the National Academy of Science USA, 99:8271-8276*, 2002.

[34] T. Nepusz, A. Petroczi, L. Negyessy, and F. Bazso, “Fuzzy communities and the concept of bridgeness in complex networks,” *Physical Review E*, vol. 77, 2008.

[35] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, pp. 814–818, 2005.

[36] S. Zhang, R. Wang, and X. Zhang, “Identification of overlapping community structure in complex networks using fuzzy c-means clustering,” *Physica A*, vol. 374, pp. 483–490, 2007.

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

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

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

[40] J. P. Bagrow, “Evaluating local community methods in networks,” *J.STAT.MECH.*, p. P05001, 2008.

[41] S. Gregory, “A fast algorithm to find overlapping communities in networks,” in *PKDD*, 2008, pp. 408–423.

[42] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, “Models of social networks based on social distance attachment,” *Phys. Rev. E*, vol. 70, no. 5, p. 056122, 2004.

[43] M. E. J. Newman, “The structure of scientific collaboration networks,” in *PNAS USA, 98:404-409*, 2001.