A Novel Method for Community Detection in Complex Network Using New Representation for Communities

Wang Yiwen and Yao Min

College of Computer Science, Zhejiang University, Hangzhou 310027, China

Abstract. During the recent years, community detection in complex network has become a hot research topic in various research fields including mathematics, physics and biology. Identifying communities in complex networks can help us to understand and exploit the networks more clearly and efficiently. In this paper, we investigate the topological structure of complex networks and propose a novel method for community detection in complex network, which owns several outstanding properties, such as efficiency, robustness, broad applicability and semantic. The method is based on partitioning vertex and degree entropy, which are both proposed in this paper. Partitioning vertex is a novel efficient representation for communities and degree entropy is a new measure for the results of community detection. We apply our method to several large-scale data-sets which are up to millions of edges, and the experimental results show that our method has good performance and can find the community structure hidden in complex networks.

Keywords: community detection, complex network, adjacency matrix.

1 Introduction

A network is a mathematical representation of a real-world complex system and is defined by a collection of vertices(nodes) and edges(links) between pairs of nodes. The modern science of networks has brought significant advances to our understanding of complex systems, which is an interdisciplinary endeavor, with methods and applications drawn from across the natural, social, and information sciences. There has been a surge of interest within the physics community in the properties of networks of many kinds, including the Internet, the world wide web, citation networks, transportation networks, software call graphs, email networks, food webs, and social and biochemical networks. Most of these networks are generally sparse in global yet dense in local. The view that networks are essentially random was challenged in 1999 when it was discovered that the distribution of number of links per node of many real networks is different from what is expected in random networks.

One of the most relevant features of graphs representing real systems is community structure. Communities, also called clusters or modules, are groups of

L. Cao et al. (Eds.): PAKDD 2011 Workshops, LNAI 7104, pp. 89–99, 2012.

vertices which probably share common properties or play similar roles within the graph. Such communities, can be considered as fairly independent compartments of a graph, playing a similar role like, e. g., the tissues or the organs in the human body. Detecting communities is of great importance in sociology, biology and computer science, in particular in behavior informatics [4], disciplines where systems are often represented as graphs. Identifying meaningful community structure in social networks is inherently a hard problem. Extremely large network size or sparse networks compound the difficulty of the task, despite the huge effort of a large interdisciplinary community of scientists working on it over the past few years.

Detecting communities is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs[7]. One important application of community detection is for web search. While web is constantly growing, web search is becoming more and more a complex and confusing task for the web user. The vital question is which the right information for a specific user is and how this information could be efficiently delivered, saving the web user from consecutive submitted queries and time-consuming navigation through numerous web results[8]. Most existing web search engines return a list of results based on the query without paying any attention to the underlying users interests. Web community detection is one of the important ways to enhance retrieval quality of web search engine. How to design one highly effective algorithm to partition network community with few domain knowledge is the key to network community detection.

2 Related Work

There have been many various approaches and algorithms to analyze the community structure in complex networks, which use methods and principles of physics, artificial intelligence, graph theory and so on. Most of the algorithms use adjacency matrix to present the complex network. Complex network theory uses the tools of graph theory and statistical mechanics to deal with the structure of relationships in complex systems. A network is defined as a graph $G = (N, E)$ where N is a set of nodes connected by a set of edges E. We will refer to the number of nodes as n and the number of edges as m. The network can also be defined in terms of the *adjacency matrix* $G = A$ where the elements of A are

$$A_{ij} = \begin{cases} 1 \textit{ if node } i \textit{ and node } j \textit{ are connected,} \\ 0 \textit{ otherwise.} \end{cases} \tag{1}$$

As discussed at length in two recent review articles [7,9] and references therein, the classes of techniques available to detect communities are both numerous and diverse; they include hierarchical clustering methods such as single linkage clustering, centrality-based methods, local methods, optimization of quality functions such as modularity and similar quantities, spectral partitioning, likelihood-based methods, and more.

The Kernighan-Lin algorithm [1] is one of the earliest methods proposed and is still frequently used, often in combination with other techniques. The authors were motivated by the problem of partitioning electronic circuits onto boards: the nodes contained in different boards need to be linked to each other with the least number of connections. The procedure is an optimization of a benefit function Q, which represents the difference between the number of edges inside the modules and the number of edges lying between them. But this algorithm need to know the size of two network communities, otherwise, this flaw enables it very difficult to actual application in network analysis.

In general, it is uncommon to know the number of clusters in which the graph is split, or other indications about the membership of the vertices. In such cases, hierarchical clustering algorithms [2] is proposed, that reveal the multilevel structure of the graph. Hierarchical clustering is very common in social network analysis, biology, engineering, marketing.

Spectral clustering is able to separate data points that could not be resolved by applying directly k-means clustering. The first contribution on spectral clustering was a paper by Donath and Hoffmann [3], who used the eigenvectors of the adjacency matrix for graph partitions. The Laplacian matrix is by far the most used matrix in spectral clustering.

Above algorithms, divisive method and agglomerative method should be introduced. GN algorithm [5], which is one kind of divisive methods, through removing the edge one by one which weight is highest(this weight is the number of short-path passing through each node in the network) until entire network is divided into more and more small part. But GN algorithm has no function to measure the quality of one community partition.

The first problem in graph clustering is to look for a quantitative definition of community. No definition is universally accepted. As a matter of fact, the definition often depends on the specic system at hand and/or application one has in mind [16]. The most popular quality function for community detection is the modularity of Newman and Girvan [6]. It is based on the idea that a random graph is not expected to have a cluster structure, so the possible existence of clusters is revealed by the comparison between the actual density of edges in a subgraph and the density one would expect to have in the subgraph if the vertices of the graph were attached regardless of community structure. This expected edge density depends on the chosen null model,i. e. a copy of the original graph keeping some of its structural properties but without community structure. Modularity can then be written as follows

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(C_i, C_j) \tag{2}$$

where the sum runs over all pairs of vertices, A is the adjacency matrix, m the total number of edges of the graph, and P_{ij} represents the expected number of edges between vertices i and j in the null model. The δ-function yields one if vertices i and j are in the same community ($C_i = C_j$), zero otherwise.

Modularity is by far the most used and best known quality function , otherwise a number of Modularity-based methods have been proposed in recent years. But this measure essentially compares the number of links inside a given module with the expected values for a randomized graph of the same size and same degree sequences.

The current algorithms are successful approaches in community detection. However there are some drawbacks of current algorithms.
1. Most of these algorithms have time complexities that make them unsuitable for very large networks.
2. Some algorithms have data structures like matrices, which are hard to implement and use in very large networks.
3. Some algorithms also need some priori knowledge about the community structure like number of communities, where it is impossible to know these values in real-life networks.

3 Proposed Method

In this section, we introduce a novel method for community detection in complex network, which owns several outstanding properties, such as efficiency, robustness, broad applicability and semantic. The method proposed is based on partitioning vertex and degree entropy. Partitioning vertex is a new efficient representation for communities and degree entropy is a new measure for the results of community detection.

3.1 Partitioning Vertex

There are a number of recent studies focused on community detection in complex network, but all of them use a naive representation for communities in which each community is assigned to precisely a unique number. For example, a network consists 9 vertices is divided into 3 communities. The result of community detection is {1(2), 2(2), 3(1), 4(1), 5(1), 6(2), 7(3), 8(3), 9(3)}. So community 1 contains vertices {3, 4, 5}, community 2 contains vertices {1, 2, 6}, community 3 contains vertices {7, 8, 9}. We must record each vertex is assigned to which community when using naive representation and there two critical defects. One is that it is supposed to lose effectiveness and occupy too much space when the number of vertices comes to millions or even larger. The other defect is that naive representation does no good to understand the structure of network better.

For the two reasons above, we proposed a novel representation for communities: partitioning vertex. If we choose n vertices in the network as partitioning vertices, such as $v_1, v_2, ..., v_n$, there are totaly 2^n corresponding communities. Each community can be represented as a n-bit binary string, and each bit stands for that every corresponding partitioning vertex is connected with the community or not. For example, as shown in Figure. 1, the network has 12 vertices and two partitioning vertices is selected, vertex 9 and vertex 10. The whole network is divided into 4 communities, which are community 00(not connected with vertex 9 and vertex 10) with vertices{11, 12}, community 01(connected with vertex 9,

not connected with vertex 10) with vertices{1, 2, 3, 0}, community 10(connected with vertex 10) with vertices {6, 7, 8, 9}, community 11(connected with vertex 9 and vertex 10) with vertices{4, 5}.

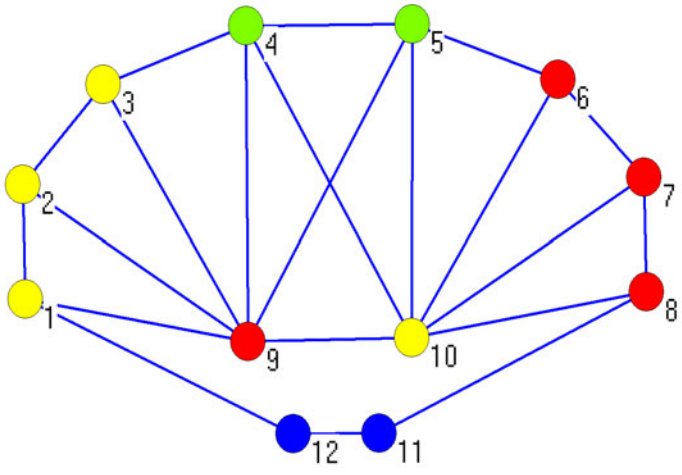

Fig. 1. Example for Partitioning Vertex

There are two outstanding advantages when using partitioning vertex to present communities. One is that you do not have to record each vertex is assigned to which community, only the selected partitioning vertices are needed. So the community detection can be changed to find out partitioning vertices, which is much easier than finding out every vertex is assigned to which community. The other advantage is that every community itself can give information about why the vertices are clustered. The vertices in the same community are all connected with some partitioning vertex or not. In reality, the vertices of complex network have their own specific meanings, and the edges between vertices are also rich in semantic information. When using semantic information in the network and partitioning vertices, we can tell what the semantic information of communities, which help us understand the network better.

It is obvious that a community is no restricted when using a naive representation. But what about the partitioning vertex way? The only difference is that the structurally equivalent vertices(have the same relationships to all other vertices) are in the same community. It is suitable in community detection, for the structurally equivalent vertices probably share common properties or play similar roles in the network.

3.2 Degree Entropy

If we adopt partitioning vertex to present communities, the community detection is changed to find out suitable partitioning vertices. And the most important

thing is how to measure the quality of community detection when using partitioning vertex. We proposed degree entropy here to cover it, which is a quantitative function as shown below.

$$DE(v_i) = (degree(v_i)/m) * log_2(1/(degree(v_i)/m)) \tag{3}$$

Here, v_i stands for a vertex in a network, and $degree(v_i)$ stands for the degree of vertex(the number of vertices that are connected with the vertex), m stands for the number of edges in the network. When we say vertex A is connected with vertex B, we get the information that the edge connected vertex A and vertex B is one of the edges connected with vertex B. And the degree entropy is designed to quantitatively determine information. When the degree entropy of vertex B is larger, we get more information. In order to quantitatively determine the information of communities when using partitioning vertices, we definite the degree entropy of community as shown below.

$$DE(c_i) = \sum_{i}^{n}(DE(pv_i)) \tag{4}$$

The degree entropy of community is a sum of degree entropy of every partitioning vertices. Here, c_i stands for a community in a network, n stands for the number of partitioning vertices, $DE(pv_i)$ stands for the degree entropy of partitioning vertex i. We use the degree entropy for community to measure the quality of community detection. So the purpose of community detection is maximizing the degree entropy, as larger degree entropy implies more information.

3.3 The Method

The degree distribution was one of the most popular issues of concern in complex network. More importantly, the scientists find that the degree distribution of complex network obey the power law or just called scale-free property. Scale-free property means that there some vertices owning quite high degree, which is suitable for using partitioning vertices to present communities. So we adopt partitioning vertices in the community detection in complex network, and the community detection is changed to select partitioning vertices.

As the degree entropy of community is introduced to measure the quality of community detection, we must select suitable partitioning vertices to maximize the degree entropy of community. The number of partitioning vertices must be fixed first, such as n, for lager degree entropy of community can be obtained just only using one more partitioning vertex. So which n vertices in a network should be selected to obtain the maximum degree entropy of community? Since the degree entropy of community is a sum of degree entropy of every partitioning vertex, we only need to select n vertices with largest degree entropy. The degree entropy changes with the value of $(degree(v_i)/m)$ is shown in Figure. 2. And the largest degree entropy is obtained when the value of $(degree(v_i)/m)$ is $1/e$, where e is the base of natural logarithm and approximately equal to 2.718281828.

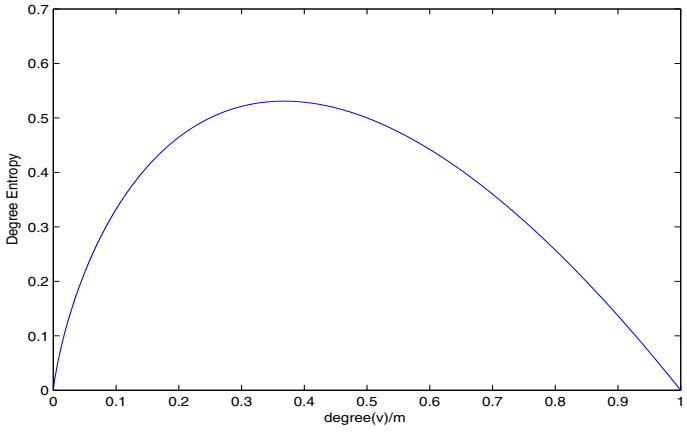

Fig. 2. Degree Entropy changes with the value of degree(v)/m, m stands for the number of edges in the network, degree(v) stands for the degree of vertex

We can select n vertices with largest degree entropy as partitioning vertices according to Figure. 2. In most cases, the value of $(degree(v_i)/m)$ is smaller than $1/e$, where degree entropy is a monotonically increasing function of $(degree(v_i)/m)$. Since m is a constant stands for the number of edges in a network, degree entropy a monotonically increasing function of $degree(v_i)$ too. So we can just select the top degree n vertices as partitioning vertices.

The main steps of our proposed method can be described as follows:

Step1: Set the only parameter n, the number of partitioning vertices. n partitioning vertices can divide the network into most 2^n communities. More communities can be obtained when using larger n. The n is set according to meet the need of user.
Step2: Calculate the degree of every vertex in the network. Our method owns broad applicability for the only information needed is degree.
Step3: Choose the top degree n vertices as partitioning vertices.

4 Experiments

In this section we apply our method to various of networksand all the graphs are drawn with a free software Pajek [10]. It is difficult to compare our method with other methods since there is no universally accepted measure for the results of community detection. The method is tested on the Zachary Karate Club [11], Java Compile-Time Dependency [12], High Energy Particle Physics (HEP) literature [13] and Stanford web graph [14] networks. In each case we find that our method reliably detects the structures.

4.1 Zachary Karate Club

Zachary Karate Club is one of the classic studies in social network analysis. Over the course of two years in the early 1970s, Wayne Zachary observed social interactions between the members of a karate club at an American university. He built network of connections with 34 vertices and 78 edges among members of the club based on their social interactions. By chance, a dispute arose during the course of his study between the clubs administrator and the karate teacher. As a result, the club splits into two smaller communities with the administrator and the teacher being as the central persons accordingly. Figure. 3 shows the detected two communities by our method which are mostly matched with the result of Zacharys study(only 3 vertices are different). The number of partitioning vertices is set to 1 as Zachary Karate Club is known to divided into 2 communities. The selected partitioning vertex is 34, which exactly stands for the club's administrator.

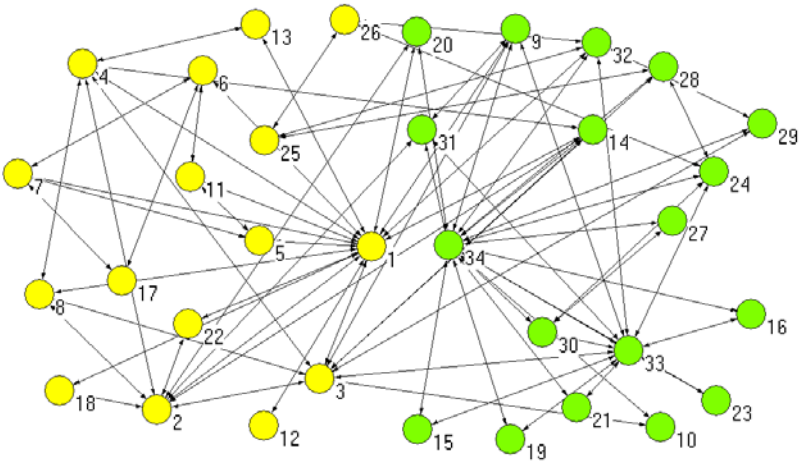

Fig. 3. Community Detection Results of Zachary Karate Club

4.2 Java Compile-Time Dependency

The Java compile-time dependency graph consists of nodes representing Java classes and directed edges representing compile-time dependencies between two classes. For example, the class java.util.Calendar depends (among others) on the class java.util.Hashtable. The data provided contains the dependencies for all classes under the java.* packages for JDK 1.4.2. Dependencies between a provided class and one outside the java.* realm are not shown. Figure. 4 shows the original Java compile-time dependency graph and Figure. 5 shows the detected four communities by our method. There are 1538 vertices and 8032 arcs in the Java compile-time dependency network. The number of partitioning vertices is set to 2 and the whole network is divided into four communities.

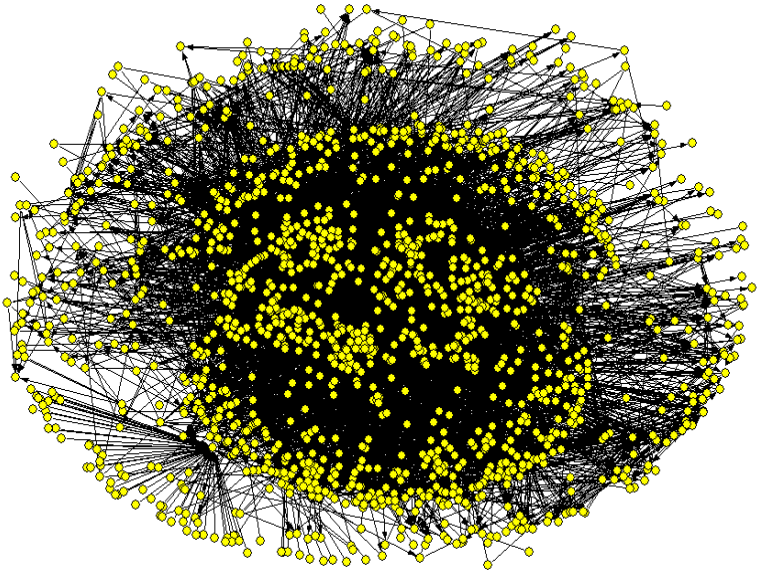

Fig. 4. Java Compile-Time Dependency

Two selected partitioning vertices are vertex 5 (java.lang.String) and vertex 20 (java.lang.Object). The semantic information for different communities is shown as the following Table. 1. The most important property of our method is semantic. The communities detected by our method own their semantic information, which helps people to understand the structure of complex network much better.

Table 1. Semantic Information for Experiment of Fig. 5

Community	Color	Semantic Information
00	Yellow	not depend on
01	Green	depend on java.lang.String
10	Red	depend on java.lang.Object
11	Blue	depend on both java.lang.String and java.lang.Object

4.3 HEP Literature and Stanford Web Graph

An efficient community detection method must can deal with large-scale networks. In this section, our method is tested on HEP literature network and Stanford web graph.The HEP literature network is a citation data from KDD Cup 2003, a knowledge discovery and data mining competition held in conjunction with the Ninth Annual ACM SIGKDD Conference. The Stanford Linear Accelerator Center SPIRES-HEP database has been comprehensively cataloguing the HEP literature online since 1974, and indexes more than 500,000 high-energy physics related articles including their full citation tree. It is a directed

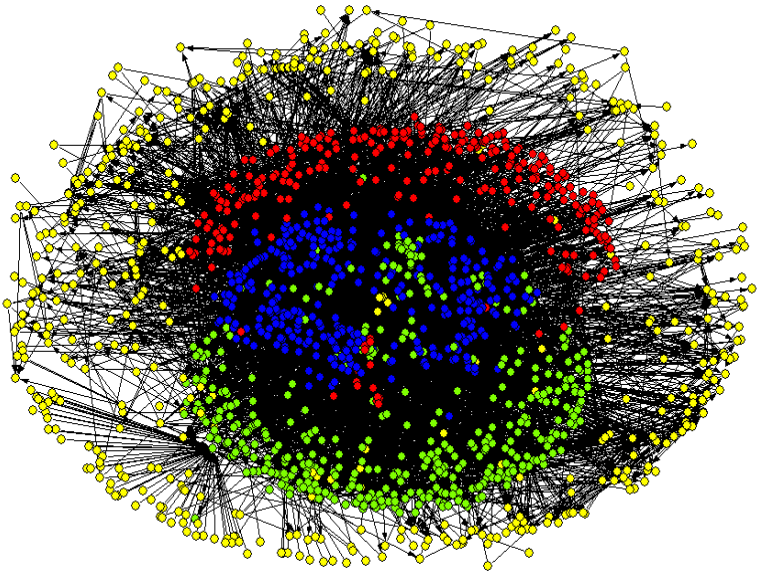

Fig. 5. Community Detection Results of Java Compile-Time Dependency

network with 27240 vertices and 342437 arcs. The units names are the arXiv IDs of papers; the relation is X cites Y. Stanford web graph was obtained from a September 2002 crawl (281903 pages, 2382912 links). The matrix rows represent the inlinks of a page, and the columns represent the outlinks. Suitable visualizations of the two graphs are difficult for they own too many vertices and edges. We choose fast Modurlity algorithm [15] which proposed by Newman to compare cost time with our method. The results is shown in Table. 2.

Table 2. Cost Time (seconds)

Network	Number of Vertices	Number of Edges	Our Method	Fast Modurlity
Java Dependency	1539	8032	0.091	4.2
HEP literature	27240	342437	2.7	failed
Stanford web graph	281903	2382912	3.8	failed

5 Conclusion

In this paper, we have mainly proposed an novel method for community detection in complex network. Identifying communities in complex networks can help us to understand and exploit the networks more clearly and efficiently. The method is based on partitioning vertex and degree entropy. Partitioning vertex is a novel efficient representation for communities and degree entropy is a new measure for the results of community detection. The method owns several outstanding

properties, such as efficiency, robustness, broad applicability and semantic. We apply our method to several large-scale data-sets which are up to millions of edges, and the experimental results show that our method has good performance and can find the community structure hidden in complex networks.

For the future work, we will continue our research by focusing on the applying our method to online social networks, such as Twitter, Facebook and Myspace. We will also search for more refined theoretical models to describe the structure of complex network.

References

1. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Syst. Tech. J. 49, 291–307 (1970)
2. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer, Berlin (2001)
3. Donath, W., Hoffman, A.: Lower bounds for the partitioning of graphs. IBM J. Res. Dev. 17(5), 420–425 (1973)
4. Cao, L.: In-depth Behavior Understanding and Use: the Behavior Informatics Approach. Information Science 180(17), 3067–3085 (2010)
5. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl Acad. 99, 7821–7826 (2002)
6. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. PNAS 99, 7821–7826
7. Fortunato, S.: Community detection in graphs. ArXiv:0906.0612v2 physics.soc-ph (January 25, 2010)
8. Garofalakis, J., Giannakoudi, T., Vopi, A.: Personalized Web Search by Constructing Semantic Clusters of User Profiles. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 238–247. Springer, Heidelberg (2008)
9. Porter, M.A., Onnela, J.-P., Mucha, P.J.: Communities in networks. Notices of the American Mathematical Society 56, 1082–1097, 1164-1166 (2009)
10. `http://vlado.fmf.uni-lj.si/pub/networks/pajek/default.htm`
11. Zachary, W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
12. `http://gd2006.org/contest/details.php`
13. `http://vlado.fmf.uni-lj.si/pub/networks/data/hep-th/hep-th.htm`
14. Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H.: Exploiting the Block Structure of the Web for Computing PageRank. Preprint (March 2003)
15. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E 69, 066133 (2004)
16. Fortunato, S.: Community detection in graphs. Physics Reports 486, 75–174 (2010)