Community Detection based on Distance Dynamics

Junming Shao
University of Electronic
Science and Technology of
China, Chengdu, China
junmshao@uestc.edu.cn

Zhichao Han
University of Electronic
Science and Technology of
China, Chengdu, China
hanchao0202@gmail.com

Qinli Yang
University of Electronic
Science and Technology of
China, Chengdu, China
qinli.yang@uestc.edu.cn

Tao Zhou
University of Electronic
Science and Technology of
China, Chengdu, China
zhutou@ustc.edu

ABSTRACT

How can we uncover the natural communities in a real-world network that allows insight into its underlying structure and also potential functions? In this paper, we introduce a new community detection algorithm, called Attractor, which automatically spots communities in a network by examining the changes of "distances" among nodes (i.e. distance dynamics). The fundamental idea is to envision the target network as an adaptive dynamical system, where each node interacts with its neighbors. The interaction will change the distances among nodes, while the distances will affect the interactions. Such interplay eventually leads to a steady distribution of distances, where the nodes sharing the same community move together and the nodes in different communities keep far away from each other. Building upon the distance dynamics, Attractor has several remarkable advantages: (a) It provides an intuitive way to analyze the community structure of a network, and more importantly, faithfully captures the natural communities (with high quality). (b) Attractor allows detecting communities on large-scale networks due to its low time complexity ($O(|E|)$). (c) Attractor is capable of discovering communities of arbitrary size, and thus small-size communities or anomalies, usually existing in real-world networks, can be well pinpointed. Extensive experiments show that our algorithm allows the effective and efficient community detection and has good performance compared to state-of-the-art algorithms.

Keywords

Community detection; Interaction model; Network

1. INTRODUCTION

During the past decades, community detection [3] (also called graph clustering or graph partitioning) has attracted a lot of attention. Many approaches have been proposed to identify communities based on different criteria (e.g. *betweenness* [5], *normalized cut* [16], *modularity* [11], etc.), and each criterion comes to specific advantages and drawbacks. As an example, the wide-spread modularity based algorithms [11], only yield a good graph partitioning if the network follows the random null assumption that each node has the equal chance to link any other node of the network [4]. This assumption becomes unreasonable for large networks (usually called "resolution limit") as their connectivity patterns are usually in a local instead of a global fashion. Moreover, the growing large-scale networks in diverse fields are posing an increasing challenge for most established community detection algorithms. Therefore, how to identify the community structure in large-scale networks effectively and efficiently remains a big data mining task to date.

In this paper, instead of introducing a new user-defined criterion for community detection like *normalized cut* [16] or *modularity* [11], we consider the problem of community detection from a new point of view: **distance dynamics**. We will see this new viewpoint supplements an intuitive way to identify community structure, and has several attractive properties. But let us first illustrate the basic idea.

1.1 Basic Idea

From the view of sociology, a "community" can be perceived as a group of persons who are connected to each other by relatively durable social relations to form a tight and cohesive social entity, due to the presence of a "*unity of will*" or *"sharing common values"* [7]. It is thus curious for us to know whether the community structure can be automatically revealed by simulating the degree of cohesiveness of persons over time. Namely, we expect all persons in the same community gradually enhance the cohesiveness by influencing each other, and finally converge together (e.g. same opinion, common values, etc.). Inspired by such perception, we present a new method to shed light on the compartmental organization of a given network from the perspective of distance dynamics. The basic idea is to view a network as an adaptive dynamical system, and investigate its dynamics over time. Here, instead of exploiting the node dynamics like traditional dynamical systems in physics, we examine the changes of "distances" among nodes (i.e. distance dynamics) over time. Driven by the local topology-

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...$15.00
DOI: http://dx.doi.org/10.1145/2783258.2783301.

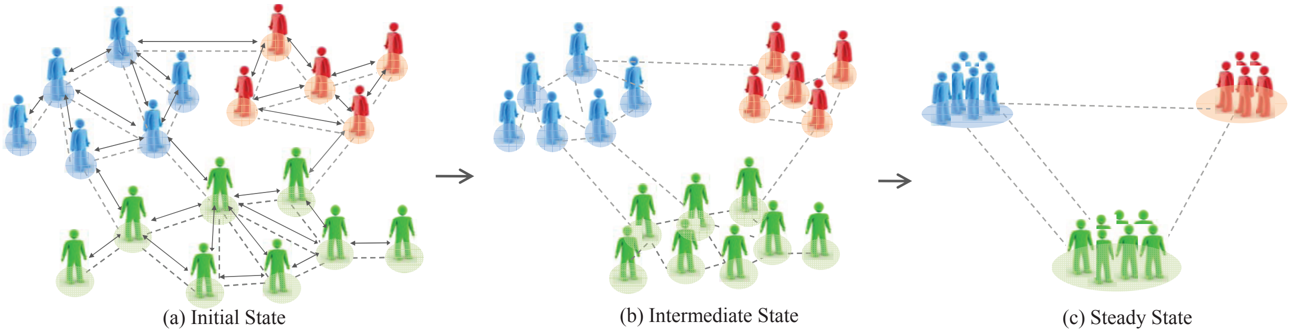


Figure 1: The illustration of community detection based on distance dynamics. (a) A social network, where the dashed lines indicate the relationships among persons, and arrows demonstrate the direct mutual interactions based on their relationships. (b) Relying on a proposed interaction model, the "distances" among people will change over time, where persons in the same community tend to gradually move together while people in different communities will keep far away from each other. (c) The steady state of persons in terms of the "distances": three intuitive communities.

driven interactions (cf. Section 3.3), the distances among nodes change gradually, and often exhibit two distinct ways as time evolves, where the distances among nodes in the same communities tend to decrease while those in different communities increase. Finally, all distances achieve a stable pattern. We call the stable pattern as an ***attractor***, a conceptual metaphor that all nodes attract their adjacent nodes, resulting in the nodes sharing the same communities move together while those in different communities keep far away from each other. Therefore, the community structure finally pops up automatically.

Building upon our proposed interaction model (cf. Section 3), the dynamics of each distance between two connecting nodes can be described in the following three phases: Firstly, each distance starts with an initial value. Secondly, as time evolves, each distance gradually shrink or stretch relying on its local topological structure. Finally, each distance converges (0 or 1 eventually). And as a result, the network will be naturally split into several distinct communities by simply removing the edges associated with distances equivalent to 1. To better illustrate the basic idea, let us take a social network as an example. Fig. 1 displays the distance dynamics of an artificial social network composing of a number of persons and a set of inter-relationships (dashed lines). In this network, there exist three groups (representing as cartoon people with different colors) based on their different hobbies. Supposing some new techniques have been plugged into a mobile phone, and persons in this network are discussing their cons and pros. We are interested in knowing how opinion disparities among persons evolve driven by the underlying structure over time. In the beginning, each person usually has their own ideas, and the disparities of opinions with his neighbors are thus different (First phase: initial opinion disparities among people, see Fig. 1(a)). Due to the influence from his/her known persons (i.e. persons having relationships), the disparities of opinions among these persons gradually change (increase or decrease) over time (Second phase: the simulation of distance dynamics, see Fig. 1(b)). Finally the opinion disparities of all people tend to converge, and three communities naturally pop up in terms of the "distances" among persons (Third phase: stable pattern, see Fig. 1(c)).

1.2 Contributions

By simulating the distance dynamics, Attractor has several attractive benefits for community detection in networks, most importantly:

- **Intuitive Community Detection**: Instead of optimizing user-specified measures, Attractor investigates the community structure in networks from a new point of view: *distance dynamics*. Building upon three proposed interaction patterns, Attractor allows automatically spotting communities intuitively, and more importantly, faithfully finding high-quality communities (cf. Fig. 5 - Fig. 9, Table 2 and 3).
- **Small Community and Anomaly Detection**: Relying on the local topology-driven dynamic interactions, the small communities or anomalies usually existing in large-scale networks can be well identified as Attractor allows discovering arbitrary-size communities (cf. Fig. 10, Table 2 and 3).
- **Scalability**: Thanks to the local interaction model, Attractor only needs to investigate the distances of linked nodes over time, which results in a relatively low time complexity of $O(|E|)$ (cf. Section 3.5, Fig. 11). This property of Attractor lends itself to handling large real-world networks.

The remainder of this paper is organized as follows: In the following section, we briefly survey related work. Section 3 presents our algorithm in detail. Section 4 contains an experimental evaluation. We finally conclude the paper in Section 5.

2. RELATED WORK

During the past several decades, many approaches have been established for community detection, such as [8], [16], [15] [11] etc. Due to space limitation, we only report the closest approaches from the literature. For detailed reviews of graph clustering, please refer to [14][3].

Cut-Criteria Based Community Detection. The cut-criterion based community detection algorithms refer to a class of widely used techniques which seek to partition a graph into disjoint subgraphs such that the number of "cuts" across the subgraphs is minimized. Wu and Leahy [19] have proposed a clustering method based on the *minimum-cut* criterion, where the cut between two subgraphs is computed as the total weights of the edges that have been removed. $k-$disjoint subgraphs are obtained by recursively finding the minimum cuts that bisect the existing segments. To avoid an unnatural bias towards splitting small-sized subgraphs based on the *minimum-cut* criterion, Shi and Malik [16] have proposed the popular *normalized cut*, to compute the cut cost as a fraction of the total edge connections to all the nodes in a graph. To optimize this criterion, a generalized eigenvalue decomposition was used to speed up computation time. In many cases, this class of graph clustering algorithms relying on the eigenvector decomposition of a similarity matrix is also called spectral clustering. Although this type of community detection usually allows identifying the communities with high quality, it is not capable of handling large-scale networks. In addition, it is a non-trivial task to determine the suitable number of communities without prior knowledge. Currently, another mainstream strategy to community detection is based on *modularity* criterion. Modularity is originally introduced to measure the quality of a division of the network according to the "*expected cut*". It is defined as the fraction of the edges that fall within groups minus the expected such fraction in an equivalent network with edges placed at random. In order to get a graph partition with high modularity, modularity-based community detection methods (e.g. [11, 2, 13]) use various kinds of techniques, such as greedy search and simulated annealing, to optimize the quality function. As mentioned in Section 1, studies have demonstrated that modularity-based community detection algorithms tend to fail on many real-world networks due to the "resolution limit". The situation becomes worse especially when the network size increases.

Large-Scale Network Handling. To identify communities on large-scale networks, many algorithms [9, 8, 18] have been proposed during past decades. Metis is a class of multi-level scalable partitioning techniques proposed by Karypis and Kumar [9], [8]. The graph clustering starts with constructing a sequence of successively smaller graphs, and a bisection of the coarsest graph is applied. Subsequently, a finer graph is generated in the next level based on the previous bisections. At each level, an iterative refinement algorithm such as Kernighan-Lin (KL) or Fiduccia-Mattheyses (FM) is used to further improve the bisection. A more robust overall multilevel paradigm has been introduced by Karypis and Kumar [8], which presents a powerful graph coarsening scheme. It uses simplified variants of KL and FM to speed up the refinement. Thanks to the multilevel graph construction, Metis allows scaling up large-scale networks, however, the quality of resulting communities may suffer in coarsening. The Markov Cluster algorithm (MCL) [18] is another popular algorithm used in life sciences based on the simulation of (stochastic) flow in graphs. MCL allows identifying high-flowing regions representing the communities using random walk.

In contrast to optimizing user-defined measures, here we provide a more intuitive way to investigate the community structure based on distance dynamics, which is not only capable of uncovering the high-quality communities, but also allows handling large-scale networks.

3. COMMUNITY DETECTION BASED ON DISTANCE DYNAMICS

3.1 Distance Dynamics versus User-defined C-ommunity Criteria

Currently, many criteria have been proposed to qualify community structure from different point of view, and each criterion has its own advantages and drawbacks. In this study, instead of introducing new user-defined criterion, we present a new community detection approach based on the distance dynamics. As stated in Section 1.1, the basic philosophy is to envision a network as a dynamic system, and dynamically investigate the distances among adjacent nodes to uncover its community structure. Compared to most existing algorithms, except for the vivid way to community discovery, the new viewpoint also has some additional desirable prosperities. (a) Exploring the distance dynamics provides an intuitive and comprehensive image to model the real-world network dynamics. For example, a community (e.g. friendship network) is usually established and intensified based on the relationships by interactions (e.g. social activities in friendship networks). (b) Without using user-defined measures, communities are discovered automatically driven by the intrinsic local topology of networks. (c) Insight into distance dynamics also offers a generalized way for network mining in metric space instead of vector space. This is quite beneficial to network analysis as the information of real-world networks we usually can gain is their connectivity patterns.

In the following, we start with some preliminary definitions, and then an interaction model is proposed in Section 3.3. Section 3.4 presents the algorithm Attractor in detail, and we analyze its time complexity in Section 3.5.

3.2 Preliminaries

For the purpose of community detection, some necessary definitions are first introduced.

Definition 1 (UNDIRECTED GRAPH) Let $G = (V, E, W)$ be an undirected graph, where V is the set of nodes, E is the set of edges and W is the corresponding set of weights. $e = \{u, v\} \in E$ indicates a connection between the nodes u and v. $w(u, v)$ represents the weight of edge e. $\forall e = \{u, v\} \in E, w(u, v) = 1$, in case of unweighted graph.

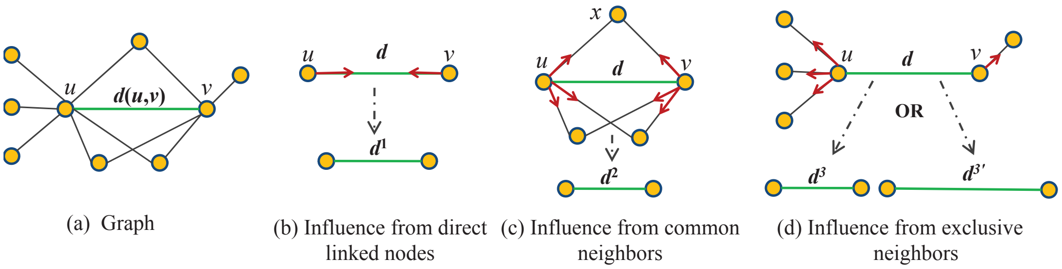
Definition 2 (NEIGHBORS OF NODE u) Given an undirected graph $G = (V, E, W)$, the neighborhood of a node $u \in V$ is the set $\Gamma(u)$ containing node u and its adjacent nodes.

$$\Gamma(u) = \{v \in V | \{u, v\} \in E\} \cup \{u\} \tag{1}$$

Based on the two definitions, we further use the popular Jaccard distance [6] to quantify the initial distance between two adjacent nodes. Selecting this measure mainly has two reasons. First, Jaccard distance provides an intuitive way to characterize the node similarity. Generally, the more common neighbors two nodes have, the more similar they are. Secondly, Jaccard distance is computed in a local fashion and is thus time efficient.

Definition 3 (JACCARD DISTANCE) Given an undirected graph $G = (V, E, W)$, the Jaccard distance of two nodes u and v is defined as:

$$d(u, v) = 1 - \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{2}$$



(a) Graph (b) Influence from direct linked nodes (c) Influence from common neighbors (d) Influence from exclusive neighbors

Figure 2: Illustration of the change of node distances influencing by three distinct interaction patterns.

For weighted graph, the Jaccard distance of two nodes u and v is further extended as:

$$d(u,v) = 1 - \frac{\sum_{x \in \Gamma(u) \cap \Gamma(v)} (w(u,x) + w(v,x))}{\sum_{\{x,y\} \in E; x,y \in \Gamma(u) \cup \Gamma(v)} w(x,y)} \tag{3}$$

3.3 Local Interaction Model

To uncover the community structure in networks based on distance dynamics, we should build up a suitable interaction model. Therefore, the interaction range and interaction patterns need to be first considered.

Interaction Range. In order to identify the community structure in networks, the exploring of local topology is essential. Thus, instead of observing the collective interactions, we focus on the distance dynamics in a local way. Obviously, the intrinsic connections (edges) of real-world networks gives a natural way to model the interaction range. Precisely, for each node, it naturally interacts with its adjacent nodes.

Interaction Patterns. After specifying the interaction range, the next crucial step is to determine the interaction patterns among nodes to simulate the distance dynamics. Formally, let $e = \{u, v\} \in E$ be an edge between two adjacent nodes u and v, and $d(u,v)$ is its initial distance. Obviously, any change of the distance $d(u,v)$ actually results from the variation of node u and node v. In fact, there are three distinct scenarios that allows influencing the distance $d(u,v)$, relying on its local topological structure (see Fig. 2). In the following, we will elaborate how the distance changes in the three different scenarios, respectively.

Pattern 1: Influence from direct linked nodes. The distance $d(u,v)$ between node u and node v, is obviously influenced by the two direct linked nodes u and v. Through mutual interactions, the one node attracts the other to move towards itself, and thus result in the decrease of $d(u,v)$ (see Fig. 2 (b)). Like a friendship network, each people affects their known people, and tends to increase their cohesiveness gradually (i.e. the "distance" will decrease). Formally, to characterize the change of the distance $d(u,v)$, we define DI, indicating the influence from the interactions of direct linked nodes, as follows:

$$DI = -\left(\frac{f(1 - d(u,v))}{deg(u)} + \frac{f(1 - d(u,v))}{deg(v)} \right) \tag{4}$$

where $deg(u)$ is the degree of the node u, $f(\cdot)$ is a coupling function and $sin(\cdot)$ is used in this study. $1 - d(u,v)$ indicates the similarity between u and v, and the more similar the two nodes are, the higher influence between each other they will have. The term $\frac{1}{deg(u)}$ is called normalization factor, which is used to consider the different influences between linked nodes with diverse degrees. Namely, the nodes with more links are harder to be influenced comparing to the nodes with less links. Take instructor network as an example. One supervisor usually links to many students while one student only connects to his supervisor. In this situation, the supervisor may have a high influence on each student while the influence for supervisor from each student is relatively low.

Pattern 2: Influence from common neighbors Here we consider the second scenario: the influence from the common neighbors $CN = (\Gamma(u) - u) \cap (\Gamma(v) - v)$ of nodes u and v (Fig. 2(c)). As the common neighbors have both links with the two nodes u and v, they attract the two nodes and thus result in the change of the distance $d(u,v)$. Specifically, each common neighbor attracts both node u and node v to move towards itself, and thus leads to the decrease of the distance $d(u,v)$ (See Fig. 2(c)). Formally, we define the change of $d(u,v)$ from the influence of common neighbors, CI, as follows:

$$CI = -\sum_{x \in CN} \left(\frac{1}{deg(u)} \cdot f\big(1 - d(x,u)\big) \cdot (1 - d(x,v)) + \frac{1}{deg(v)} \cdot f\big(1 - d(x,v)\big) \cdot (1 - d(x,u)) \right) \tag{5}$$

Here the two terms $(1 - d(x,v))$ and $(1 - d(x,u))$ for each common neighbor are used to further quantify the degree of influence compared to the influence from direct linked nodes. For example, considering a common neighbor x interacting with node u (see Fig. 2(c)), if x and v are more similar, the influence from x on u is more similar to the influence from v. Theoretically, once the similarity between x and v equals one (i.e. they can be viewed as the same node), the influence of the node x on the distance $d(u,v)$ simply transfers into the first pattern.

Pattern 3: Influence from exclusive neighbors: The third interaction pattern happens when there exists some neighbors exclusively belong to node u or v, $EN(u) = \Gamma(u) - \Gamma(u) \cap \Gamma(v)$, $EN(v) = \Gamma(v) - \Gamma(u) \cap \Gamma(v)$, respectively. Although, like pattern 1 and pattern 2, each exclusive neighbor of u attracts u to move close to itself, there is no knowledge whether node u is attracted to move closer to node v or attracted to move far away from v (see Fig. 2(d)). To determine the positive or negative influence of exclusive neighbors on the distance, a similarity-based heuristic strategy is proposed. The basic philosophy is to investigate whether each exclusive neighbor of node u is similar with node v, and vice versa. If the exclusive neighbor of node u is similar with node v, the movement of node u towards exclusive neighbor results in the decrease of the distance $d(u,v)$. Sim-

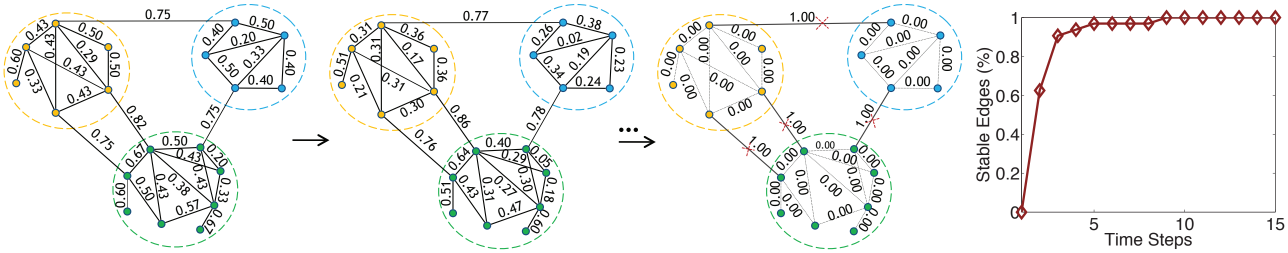
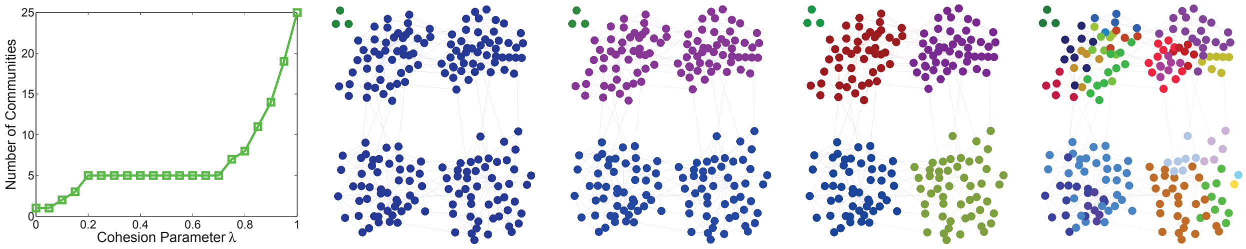


Figure 3: Illustration of the distance dynamics. (a) The graph representation of the social network of Fig. 1(a), where the numbers on edges indicate the initial distances among connected nodes. (b) The updated node distances after one time step. (c) The final state of the network.



(a) λ vs #C (b) $\lambda = 0.1$ (c) $\lambda = 0.15$ (d) $\lambda \in$ [0.2 - 0.7] (e) $\lambda = 1$

Figure 4: The sensitivity of cohesion parameter λ on community detection.

ilarly, If the exclusive neighbor is not similar with node v, the movement of node u towards the exclusive neighbor will lead to the opposite effect: moving far away from the node v. Therefore, here we introduce a cohesion parameter λ, to determine the underlying influence as follows. The cohesion parameter λ will be further discussed in Section 3.4.

$$\rho(x,u) = \begin{cases} \big(1-d(x,v)\big) & \big(1-d(x,v)\big) \geq \lambda \\ \big(1-d(x,v)\big) - \lambda & \text{otherwise} \end{cases} \tag{6}$$

where $\rho(x,u)$ characterizes the degree of positive or negative influence on the distance $d(u,v)$. Then, the change of $d(u,v)$ influencing by exclusive neighbors, EI, is defined as follows:

$$EI = -\sum_{x \in EN(u)} \left(\frac{1}{deg(u)} \cdot f\big(1-d(x,u)\big) \cdot \rho(x,u) \right) - \sum_{y \in EN(v)} \left(\frac{1}{deg(v)} \cdot f\big(1-d(y,v)\big) \cdot \rho(y,v) \right) \tag{7}$$

Finally, by considering the three interaction patterns together, the dynamics of the distance $d(u,v)$ between nodes u and v over time is govern by:

$$d(u,v,t+1) = d(u,v,t) + DI(t) + CI(t) + EI(t) \tag{8}$$

where $d(u,v,t+1)$ is the renewed distance at time step $t+1$. $DI(t)$, $CI(t)$ and $EI(t)$ characterize the changes of distance from the direct linked nodes, common neighbors and exclusive neighbors, respectively.

3.4 The Attractor Algorithm

In this section, we present the Attractor algorithm in detail.

Dynamic Interaction. Building upon the interaction model (cf. Eq. (8)), the distance dynamics can be simulated, which mainly involves the following steps:

1. At initial time ($t = 0$), without any interaction, each edge is associated with an initial distance. Here, the initial value is computed according to the Jaccord distance with Definition 2 or Definition 3.
2. As time evolves, relying on local intrinsic topological structure, the dynamics of each distance is simulated according to the three proposed interaction patterns (Eq.(4), Eq.(5) and Eq.(7)). Thanks to the topological-driven influences, the distances among nodes sharing the same community tend to decrease while those in different communities increase gradually.
3. Finally, all distances will converge, and the communities can be easily obtained by removing the edges with maximal distances (i.e. $d(u,v)$ =1).

For illustration, Fig. 3(a)-(c) shows three states for the social network of Fig. 1 from $t = 0$ to $t = 9$ during the local dynamic interaction process. $T = 0$ indicates the initial distances among connected nodes (Fig. 3(a)). From that moment on, each node interacts with its neighbors and influences the corresponding distances based on the proposed interaction model (cf. Eq. 8), and the new distances after one time step are further illustrated in Fig. 3(b). After nine steps, all distances converge, either 0 or 1, and three communities are naturally identified by cutting out all edges with distances equaling to 1.

Detection of small communities or Anomalies. In real-world networks, there usually exist many communities with various sizes. Especially in large-scale networks, the size of a large fraction of communities is usually small [1]. However, for many traditional community detection algorithms, such as Modularity or Ncut, they tend to partition the whole network into relatively equal-size groups with cluster size being no less than $\sqrt{n}$ (n is the number of nodes in a network) [1], and fail to find small communities due to the problem called "resolution limit" [4]. For attractor, as it sim-

Algorithm 1 Attractor

```
1: Input: $G = (V, E, W)$, $\lambda$
2: // Initialization of distances
3: for each edge $e = \{u, v\} \in E$ do
4:     compute the initial distance $d_e^0$ using Eq. (3);
5:     for each node $x \in EN(u)$ do
6:         compute the distance $d_{ux}^0$ using Eq. (3);
7:     end for
8:     for each node $y \in EN(v)$ do
9:         compute the distance $d_{vy}^0$ using Eq. (3);
10:    end for
11: end for
12: // Dynamic Interaction
13: Flag = TRUE;
14: while Flag do
15:     Flag = FALSE;
16:     for each edge $e = \{u, v\} \in E$ do
17:         if $0 < d_e^t < 1$ then
18:             Compute $DI_e^t$, $CI_e^t$, $EI_e^t$ using Eq. (4), (5), (7);
19:             $\Delta d_e^t = DI_e^t + CI_e^t + EI_e^t$;
20:             if $\Delta d_e^t \neq 0$ then
21:                 // compute the renewable distance over time
22:                 $d_e^{t+1}$ =$d_e^t$ + $\Delta d_e^t$;
23:                 if $d_e^{t+1} > 1$ then
24:                     $d_e^{t+1} = 1$;
25:                 end if
26:                 if $d_e^{t+1} < 0$ then
27:                     $d_e^{t+1} = 0$;
28:                 end if
29:                 Flag = TRUE;
30:             end if
31:         end if
32:     end for
33: end while
34: //Find communities
35: for each edge $e = \{u, v\} \in E$ do
36:     if $d_e^{t+1} = 1$ then
37:         remove the edge $e$ from the network;
38:     end if
39: end for
40: find the resulting components (communities) $C$;
41: Output: $C$;
```

ulates the distance dynamics and does not rely on any user-defined criterion, it allows intuitively finding the intrinsic communities with arbitrary size in networks. Therefore, it also provides a promising way to handle anomalies/outliers. In this scenario, anomalies are interpreted as the noisy nodes or unusual nodes isolated from all other nodes over time, and finally pop out automatically.

Cohesion parameter λ. For the Attractor algorithm, the cohesion parameter λ is used to determine the positive or negative interaction influence on the distances from exclusive neighbors (see Eq. (6)). Generally, with the higher value of λ, it yields more communities while produces bigger communities with lower value of λ. By modulating the cohesion parameter λ, Attractor allows analyzing the community structure from coarse to fine. Moreover, λ is informative and is easy to tune compared to the algorithms requiring to specify the number of clusters. Fig. 4(a) plots the finding number of communities with different λ ranging from 0 to 1 on a synthetic network. From this plot, we can see that Attractor allows yielding perfect partitioning with the parameter λ on a long stable range (0.2 - 0.7). The clustering results with respect to distinct parameters are further illustrated in Fig. 4(b) to Fig. 4(e). Extensive experiments further demonstrate Attractor is not sensitive to clustering results and usually produces a good result within the range $\lambda = [0.4, 0.6]$. Finally, the Pseudocode of Attractor is given in Algorithm 1.

3.5 Complexity Analysis

To investigate the distance dynamics, the initial distance of any two linked nodes in a network is required, and thus the time computation is $O(|E|)$. Moreover, for the local dynamic interaction, Attractor also needs to compute the corresponding jaccard distances for exclusive neighbors (Algorithm 1(Line 5-10)). The time complexity is $O(k \cdot |E|)$, where k is approximately the average number of exclusive neighbors of two linked nodes. During the local dynamic interaction process, as all distances have already existed, Attractor only needs to recall these distances at previous time stamp without any distance computation, and thus the time complexity is $O(T \cdot |E|)$. Totally, the time complexity is $O(|E| + k \cdot |E| + T \cdot |E|)$, where T is the number of time steps. In most cases, T is small with $3 \leq T \leq 50$.

4. EXPERIMENTS

In this section, we evaluate our proposed algorithm Attractor on synthetic as well as real-world networks to demonstrate its benefits.

Selection of comparison methods. To evaluate the performance of Attractor, we compare it to several representatives of community detection algorithms.

Ncut [16] is a well-known algorithm for graph clustering by optimizing the *normalized cut* criterion. As the eigenvalue decomposition is applied to speed up finding the optimal cut, it is also usually called as spectral clustering.

Modularity [11] is the current most popular community detection algorithm based on the *modulairty* measure, which uses the expected cut to measure clustering quality.

Metis [8] is a very fast graph clustering approach for large networks via multi-level partitioning and parallelized implementation.

MCL [18] is a popular algorithm used in life sciences based on the simulation of (stochastic) flow in graphs.

Louvain [2] is another well-known *modularity* based algorithm. Compared with the algorithm *Modularity* proposed by Newman [11], it allows for hierarchical community detection and has lower time complexity.

Infomap [13] envisions community detection problem as a coding problem, and aims at finding the optimal partitions based on minimum description length principle.

For all experiments, without further statement, Ncut and Metis specify the cluster number $K = |C|$, $|C|$ is the true number of classes of the network if the ground truth is available. MCL takes the default inflation parameter ($i = 2.0$) as suggested by authors [18]. For Louvain and Infomap, the default parameters are used. We set the cohesion parameter $\lambda = 0.5$ for Attractor as default parameter. All experiments have been performed on a workstation with 3.4 GHz CPU and 32.0 GB RAM.

Evaluation Matrices. To extensively compare different community detection algorithms with respect to effectiveness, we evaluate the clustering results in two ways.

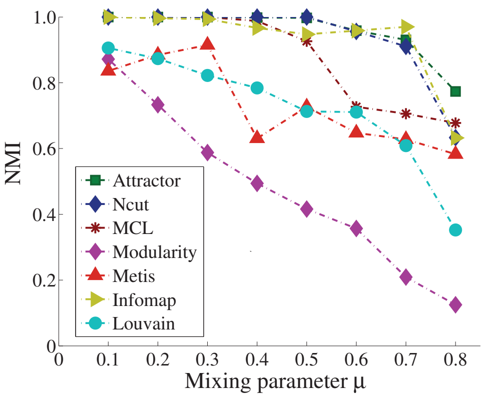


Figure 5: The performance of different algorithms on the LFR benchmark networks by varying the number of inter-cluster edges.

- *Networks with class label.* For networks whose communities are already known, the performance is directly measured by three widely used evaluation measures: *Normalized Mutual Information (NMI)* [17], *Adjusted Rand Index (ARI)* [12] and *Cluster Purity*.
- *Networks without class label.* Since the ground truth of the community structure is unknown, it is a non-trivial task to compare the performance of distinct algorithms in an objective way. In order to evaluate the quality of communities produced by different algorithms reasonably, two popular internal measures *modularity* [11] and *normalized cut (ncut)* [16] have been applied in this study, although they are somehow tailored for *modularity*-criterion or *cut*-criterion based algorithms.

4.1 Synthetic networks

In this section, we first generate several synthetic networks featuring distinct characteristics to compare the performance of various community detection algorithms. For fair comparison and to make the synthetic networks to be more consistent with the real-world networks, the LFR benchmark networks [10] have been applied, where the distributions of degree and community size of networks can be easily controlled. To increase the complexity of networks, the *mixing parameter* μ [10], defined as the fraction of links of each node outside its community, is used to control the difficulty of community separation.

Noise Edge: First, we evaluate how well the different graph clustering algorithms allow detecting communities by varying their inter-cluster edges. The inter-cluster edges, which we call noise edges, are added into the network to hamper community separation. We fix node average degree and community size, and change the *mixing parameter* μ from 0.1 to 0.8 to generate a serial of networks with different inter-cluster edges. All networks consist of 2000 nodes with the average degree $k = 20$.

With the increase of *mixing parameter*, the performance (measured by NMI) of all five approaches is shown in Fig. 5. We can see that the algorithms of Attractor, Ncut and MCL almost achieve the perfect clusterings by adding inter-cluster edges with the *mixing parameter* up to 0.4 (40% edges of each node links to other communities). Their performances begin to decrease with more and more inter-edges added into the network, and Attractor is more robust to these noise edges. For Modularity and Louvain, they are more sensitive to these noise edges, and their performances are not comparable with other five algorithms on these networks. Regarding the Metis algorithm, its performance is fluctuated and starts to decrease dramatically as soon as more inter-edges are added (with $\mu = 0.4$). The performance of infomap is surprising, since its performance decreases first and then increases with μ ranging from 0.5 to 0.7.

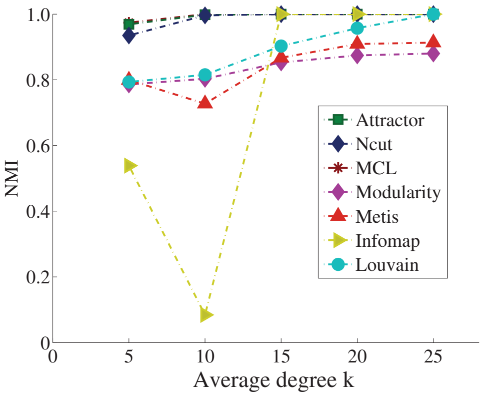


Figure 6: Performance of different algorithms on the LFR benchmark networks by varying community density using the average degree $<k>$.

Table 1: Statistics of real-world data sets, where AD: average degree; CC: clustering coefficient.

Data Sets	|V|	|E|	#Class	AD	CC
Zarachy	34	78	2	4.588	0.571
Football	115	613	11	10.661	0.403
Polbooks	105	441	3	8.400	0.488
Amazon	334863	925872	151037	5.530	0.397
Collaboration	9875	25973	-	5.260	0.312
Friendship	58228	214078	-	7.353	0.172
Road	1088092	1541898	-	2.834	0.047

Community Density: Next, we evaluate how the algorithms respond to the networks with different average degrees, which we call community density. Here we fix the inter-cluster edges ($\mu = 0.1$), and change the average degree k from 5 to 25 to see the influence of community density on the performance of these algorithms. Fig. 6 shows that Attractor, MCL and Ncut yield good results for all these networks, while the performances of Metis and Modularity are a bit worse. We can see that Attractor, Metis and Ncut allows correctly finding the good communities even with low community density ($k = 5$). For Louvain, Metis and Modularity, they are more sensitive to the community density on these synthetic networks. As to Infomap, it show an abnormal performance when average degree k is 10.

4.2 Real World Data

In this section, we evaluate the performances of different community detection algorithms on real-world networks which are all publicly available from the UCI network data repository (`https://networkdata.ics.uci.edu/index.php`) and Stanford large network dataset collection (`http://snap.`

Table 2: The performance of different graph clustering algorithms on labeled real-world networks.

	Zarachy			Football			Polbooks			Amazon		
	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.	NMI	ARI	Pur.
Attractor	**0.859**	**0.939**	**1.000**	**0.923**	**0.897**	**0.930**	**0.559**	**0.680**	**0.857**	**0.931**	**0.580**	**0.998**
Ncut	0.833	0.882	0.970	**0.923**	**0.897**	**0.930**	0.534	0.645	0.829	–	–	–
Modulairty	0.577	0.680	0.970	0.596	0.474	0.574	0.508	0.638	0.838	–	–	–
Metis	0.836	0.882	0.970	0.393	0.095	0.339	0.502	0.516	0.781	0.761	0.092	0.989
MCL	0.833	0.882	0.970	**0.923**	**0.897**	**0.930**	0.455	0.594	**0.857**	0.902	0.490	0.991
Louvain	0.524	0.541	**1.000**	0.858	0.807	0.870	0.440	0.537	**0.857**	0.738	0.384	0.384
Infomap	0.593	0.702	0.971	0.906	0.857	0.904	0.476	0.646	0.848	0.209	0.009	0.077

Table 3: The performance of different algorithms on large real-world networks without class information.

	Collaboration			Friendship			Amazon			Road		
	#C	*mod.*	*ncut*	#C	*mod.*	*ncut*	#C	*mod.*	*ncut*	#C	*mod.*	*ncut*
Attractor	1384	0.579	1179	8045	0.421	7325	23825	0.741	10811	59919	0.856	25055
Metis	1384	0.309	4217	8045	0.138	53984	23825	0.451	47336	59919	0.673	31542
MCL	2093	0.537	2103	13788	0.319	36723	46557	0.623	47488	86745	0.810	25065
Louvain	475	0.768	10.120	746	0.684	38.340	240	0.926	9.617	492	0.989	2.032
Infomap	456	0.722	5.470	572	0.439	4.104	12	0.4224	0.1249	208	0.660	6.088

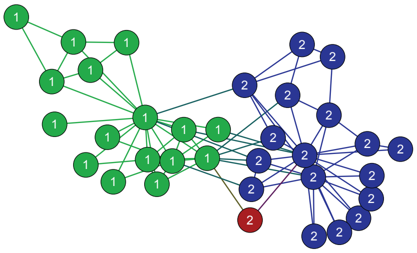


Figure 7: Attractor on karate club network. Colors of nodes indicate different detected communities.

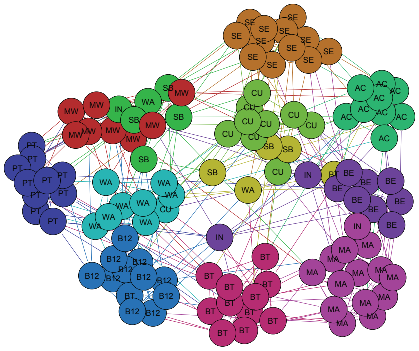


Figure 8: Attractor on American football network.

stanford.edu/data/). The statistics of seven networks are summarized in Table 1.

(1). Networks with class information

We first investigate the networks for which the ground truth of community structure is already known. The external measures such as *NMI*, *ARI* and *purity* are reported.

Zachary's karate club network: The famous network, derived by Zachary's observation about a karate club, reflects the friend relationship among these members. Specially, the network could be divided into two communities, which reflects the disagreement between the administrator and the instructor. Fig. 7 shows that Attractor identifies the communities with a high degree of success (with high values of NMI, ARI and Purity), and outperforms other comparing algorithms (Table 2). Specifically, two communities are successfully found, except one member is viewed as noise (node '10'). It is also interesting to observe that this member is located between two communities, and links with the hub nodes of two communities, respectively. In real-world scenario, it is also difficult to determine its community belonging to. Actually, it is more likely to assign this node to both communities, which is the overlapping clustering that we will not discuss in this study. For comparing algorithms of MCL and Ncut, they also achieve a good performance, and most members are correctly grouped. However, for the algorithms of Modularity, Louvain and Infomap, many members are wrongly grouped, which result in a relatively low values of NMI. The performance of the different algorithms is summarized in Table 2.

American college football: The network derived from the American football games of the schedule of Division I during regular season Fall 2000, where 115 vertices in the graph represent teams, and 613 edges represent regular-season games between the two teams they connect. The teams are divided into 12 conferences containing around 8-12 teams each, and thereby the real community structure is already known. Fig. 8 plots the communities which are detected by Attractor. It is interesting to note that Attractor automatically finds 12 communities with high quality (NMI = 0.923, ARI = 0.897, Purity = 93.0%). From this Figure, we can observe that most of teams are correctly assigned into corresponding communities. Ncut and MCL find the similar community structure as Attractor. For Metis, Modularity, Louvain and Infomap, however, they are difficult to discover the natural community structure (Table 2).

Books about US politics: This network, derived from the politic books about US politics published around the time of the 2004 presidential election, consists of 105 nodes and 441 edges. Nodes represent books sold by the online bookseller *Amazon.com*. Edges represent frequent co-purchasing of books by the same buyers. Each book is labeled

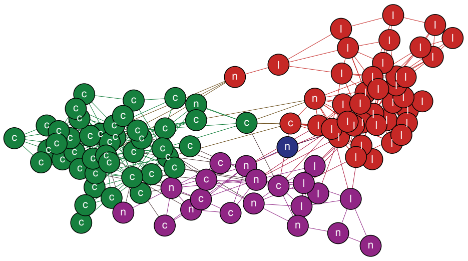


Figure 9: Attractor on political book network.

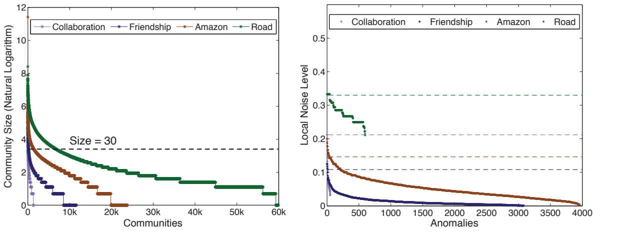
with 'l', 'n', or 'c' to indicate whether they are "liberal", "neutral", or "conservative", based on Newman's reading of the descriptions and reviews of the books posted on *Amazon*. Attractor allows a good grouping these books into there categories, where two clusters well represent the corresponding liberal and conservative books, respectively (Fig. 9 and Table 2). For algorithms of Modularity, Metis and Ncut, they yield comparable results, while MCL and Infomap produce a relatively bad grouping on this network.

Amazon network: This network consists of 334,863 nodes and 925,872 edges, and each node represents a product on the Amazon website. Each product is categorized to corresponding community based on its category provided by Amazon, and the top 5,000 communities with highest quality were investigated in [20]. Due to the high time and space complexity of eigenvalue decomposition in Ncut and Modularity, they cannot handle this network. Relying on the partial ground-truth communities (top 5,000 communities), Attractor obtains the best community quality comparing to other algorithms with high measures (NMI = 0.931, ARI = 0.580, Purity = 0.998) (Table 2). MCL algorithm allows producing a reasonable result (NMI = 0.902). However, for Metis, Louvain and Infomap, they tend to fail, especially for the algorithm Informap (NMI = 0.209). Moreover, for comprehensive evaluation, all results of the five algorithms are also evaluated by the internal criteria of *modularity* and *ncut* (see Table 3).

(2). Networks without class information

In this section, due to the time complexity of Ncut and Modularity, we limit the comparison to the clustering algorithms Metis, MCL, Louvain and Infomap on large-scale networks without class ground truth (Table 3). As there exist no convincing measures for the unlabeled network, we use *Modularity* and *Ncut* to evaluate these algorithms in an informative way.

Hepth collaboration network: The network is a collaboration network of 9,875 authors working on the theory of high energy physics. Attractor identifies 1384 communities, which results in *modularity* = 0.579 and *ncut* = 1179. On the data set, Metis (K = 1384) and MCL also yield a good partitioning while its performance is worse than Attractor in terms of the two measures (Table 3). If we just look at *modularity* and *ncut*, Louvain and Infomap seem much better than Attractor, MCL and Metis. However, the reason is that the two algorithms only yield few communities, naturally lead to better values of *modularity* and *ncut*, based on the definitions. If we regress out the effect of different number of communities, Attractor actually obtains better performance.



(a) Small communities (b) Anomalies

Figure 10: Evaluation of small communities and anomalies.

Brightkite friendship network: The graph is a location-based friendship network consisting of 58,228 nodes and 214,078 undirected edges. *Attractor* finds 8045 communities and shows a clear advantage over two other algorithms based on the two measures. For Metis (K=8045), many friends seem to be incorrectly grouped, which result in a low value of *modularity* = 0.138. As usual, Louvain and Infomap, the two algorithms tend to produce small number of communities, and many small-sized communities cannot be detected. The reason is might due to the "resolution limit".

Pennsylvania road network: This network reflects road structure of Pennsylvania, where nodes represent intersections or the endpoints and edges represent the roads connecting these intersections or endpoints. Here, we set the parameter $\lambda = 0.6$ for Attractor and $i = 1.4$ for MCL as the default values of the two algorithms cannot result in a good results due to the very sparsity of the network. Attractor finally identifies 59,919 clusters with *modularity* = 0.856 and *ncut* = 25055. MCL achieves the comparable performance and is better than the algorithm Metis. Louvain and Infomap only find small number of equal-size communities (492 and 208, respectively).

In total, the experiments on all real-world networks demonstrate that Attractor not only allows extracting meaningful communities in networks with class label (with highest performance in terms of all measures), but also scales up large-scale networks and yields a good graph partitioning in terms of the internal (*modularity* and *ncut*) and external measures (NMI, ARI and Purity) (see Table 2, Table 3).

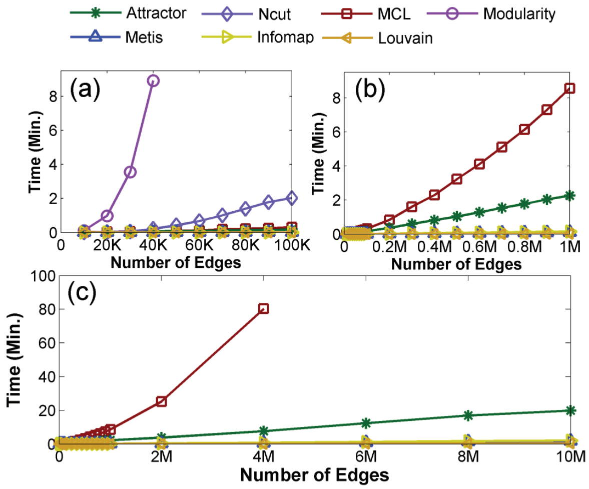


Figure 11: The runtime of the different algorithms.

4.3 Small Community and Anomaly Detection

In this section, we evaluate whether Attractor allows identifying small meaningful communities and anomalies. Fig. 10(a) plots the distribution of community size for the four large real-world networks, and we can see that Attractor can find many small communities with different size. To demonstrate the potential high-quality of the communities, we further examine the quality of the resulting small communities (size≤30) on Amazon network as it has the ground truth for the top 5,000 communities. It is interesting to note that the1458 small communities (size≤30) result in high values of NMI = 0.941, ARI = 0.637 and Purity = 0.989, which shows the desirable property of small community detection.

Moreover, to check whether the detected anomalies are the potential noisy/unusual nodes, we evaluate the **local noise level** of each node, which is defined as the fraction of node degree over the number of all links of its neighbors. Fig. 10(b) depicts the local noise level for all resulting anomalies compared to the average noise level (indicated by dashed lines) by Attractor on the four real-world networks, providing a potential evidence for the effective anomaly detection.

4.4 Runtime

To assess the scalability of Attractor with respect to network size, we generate several benchmark networks [10] with different edge sizes ranging from ten thousand to ten million by fixing the average node degree $k = 20$. Fig. 11 shows the running time for different graph clustering algorithms. We can observe that Attractor is faster than Modularity, Ncut and MCL since its time complexity is linear against to $|E|$. However, Attractor is a bit slower than scalable community detection algorithms Metis, Louvain and Infomap. Although the three algorithms are much faster than Attractor, they suffer in the quality of resulting communities.

5. CONCLUSIONS

In this paper, we introduce a new community detection algorithm, called Attractor, to automatically uncover community structure in networks based on distance dynamics. Extensive experiments further demonstrate that Attractor allows finding communities in small to large size networks with high quality, and also shows attractive benefits compared to several state-of-the-art methods. In future work, we plan to focus on exploring large network abstraction and visualization based on the intuitive dynamic interaction model.

6. ACKNOWLEDGMENTS

The research was supported partially by the National Natural Science Foundation of China (Grant No. 61403062, 61433014), Fundamental Research Funds for the Central Universities (Grant No. ZYGX2014J053, ZYGX2014J091) and the Postdoctoral Science Foundation of China (Grant No. 2014M552344).

7. REFERENCES

[1] Nir Ailon, Yudong Chen, and Huan Xu. Breaking the small cluster barrier of graph clustering. In *ICML*, pages 995–1003, 2013.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[3] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[4] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *PNAS*, 104(1):36–41, 2007.

[5] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[6] Christian Hennig and Bernhard Hausdorf. Design of dissimilarity measures: A new dissimilarity between species distribution areas. In *Data Science and Classification*, pages 29–37. Springer, 2006.

[7] Paul James, Yaso Nadarajah, Karen Haive, and Victoria Stead. Sustainable communities, sustainable development: Other paths for papua new guinea. *Hawaiian Journal of History*, 48, 2014.

[8] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[9] George Karypis and Vipin Kumar. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1):96–129, 1998.

[10] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[11] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[12] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[13] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 105(4):1118–1123, 2008.

[14] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[15] Junming Shao, Xiao He, Qinli Yang, Claudia Plant, and Christian Boehm. Robust synchronization-based graph clustering. In *PAKDD*, pages 249–260, 2013.

[16] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[17] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[18] Stijn Van Dongen. A cluster algorithm for graphs. *Report-Information systems*, (10):1–40, 2000.

[19] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE TPAMI*, 15(11):1101–1113, 1993.

[20] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *IEEE ICDM*, pages 745–754, 2012.