Permanence and Community Structure in Complex Networks

TANMOY CHAKRABORTY, Indian Institute of Technology, Kharagpur
SRIRAM SRINIVASAN, University of Nebraska at Omaha
NILOY GANGULY and ANIMESH MUKHERJEE, Indian Institute of Technology
SANJUKTA BHOWMICK, University of Nebraska at Omaha

The goal of community detection algorithms is to identify densely connected units within large networks. An implicit assumption is that all the constituent nodes belong equally to their associated community. However, some nodes are more important in the community than others. To date, efforts have been primarily made to identify communities as a whole, rather than understanding to what extent an individual node belongs to its community. Therefore, most metrics for evaluating communities, for example modularity, are global. These metrics produce a score for each community, not for each individual node. In this article, we argue that the belongingness of nodes in a community is not uniform. We quantify the degree of belongingness of a vertex within a community by a new vertex-based metric called *permanence*.

The central idea of permanence is based on the observation that the strength of membership of a vertex to a community depends upon two factors (i) the extent of connections of the vertex within its community versus outside its community, and (ii) how tightly the vertex is connected internally. We present the formulation of permanence based on these two quantities. We demonstrate that compared to other existing metrics (such as modularity, conductance, and cut-ratio), the change in permanence is more commensurate to the level of perturbation in ground-truth communities. We discuss how permanence can help us understand and utilize the structure and evolution of communities by demonstrating that it can be used to – (i) measure the persistence of a vertex in a community, (ii) design strategies to strengthen the community structure, (iii) explore the core-periphery structure within a community, and (iv) select suitable initiators for message spreading.

We further show that permanence is an excellent metric for identifying communities. We demonstrate that the process of maximizing permanence (abbreviated as *MaxPerm*) produces meaningful communities that concur with the ground-truth community structure of the networks more accurately than eight other popular community detection algorithms. Finally, we provide mathematical proofs to demonstrate the correctness of finding communities by maximizing permanence. In particular, we show that the communities obtained by this method are (i) less affected by the changes in vertex ordering, and (ii) more resilient to resolution limit, degeneracy of solutions, and asymptotic growth of values.

CCS Concepts: • **Information systems** → **Clustering**; • **Mathematics of computing** → *Graph algorithms;*

Additional Key Words and Phrases: Permanence, community discovery, community evaluation metric, modularity

---

T. Chakraborty is financially supported by Google India PhD fellowship. S. Srinivasan is financially supported by UNO Graduate Research and Creative Activity Grant. S. Bhowmick is financially supported by University of Nebraska at Omaha, USA.
Authors' addresses: T. Chakraborty, N. Ganguly, and A. Mukherjee, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India; emails: {its_tanmoy, niloy, animeshm}@cse.iitkgp.ernet.in; S. Srinivasan and S. Bhowmick, Department of Computer Science, University of Nebraska at Omaha, Omaha, NE 68106; emails: {sriramsrinivas, sbhowmick}@unomaha.edu.

DOI: http://dx.doi.org/10.1145/2953883

ACM Reference Format:
Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. 2016. Permanence and community structure in complex networks. ACM Trans. Knowl. Discov. Data 11, 2, Article 14 (November 2016), 34 pages.
DOI: http://dx.doi.org/10.1145/2953883

1. INTRODUCTION

Community detection is the process of finding closely related groups of entities in a network. Complex networks, such as those arising in biology, social sciences, and epidemiology, represent systems of interacting entities. The entities are represented as vertices in the network and their pairwise interactions are represented as edges. A community, then, is a group of vertices that have more internal connections (i.e., connections to vertices within the group) than external connections (i.e., connections to vertices outside the group).

Most community detection algorithms are based on combinatorial optimization. The goal is to find the community assignment that leads to the optimal value of a specified network parameter, such as modularity [Newman and Girvan 2004] or conductance [Leskovec et al. 2009]. However, since many real-world communities are based on subjective measurements (as opposed to a formal mathematical definition), the validation of the results is done by comparing the obtained communities with known "ground-truth" communities. Very rarely do the obtained communities exactly match with ground-truth communities. Moreover, due to the phenomenon of resolution limit and degeneracy of solutions [Fortunato and Barthelemy 2007], the optimum parameter value sometimes produces intuitively incorrect solutions. As a result, community detection is an active area of research with new optimization metrics being regularly proposed [He et al. 2013; Yang and Leskovec 2012] that either produce more accurate results on a certain subclass of networks and/or can address some of the above discussed issues.

Almost all community detection metrics and related algorithms contain the implicit notion that all the vertices in a community belong *equally* to the community, i.e., the community membership is *homogeneous*. Therefore, the optimal score of the metric can only be obtained over the network as a whole. The information about how placement of vertices affects the community structure is lost using these current measurements. In order to include this important information, we have introduced a vertex-centric metric called *permanence* [Chakraborty et al. 2014].

The key idea in formulating permanence is as follows. Most optimization metrics are based on the total internal and external connections of the vertex. We posit that the distribution of the external connections of a vertex is equally important. In particular, our vertex assignment decisions are based not on the total number of external connections, but on *the maximum number of external connections to any single neighboring community*. If vertex v is in community S and vertex u is in community T and there exists an edge (v, u), then S and T are neighboring communities of each other.

To the best of our knowledge, we are the first to make this distinction between the total external connections and their distribution. Permanence of a vertex thus quantifies its propensity to remain in its assigned community and the extent to which it is "pulled" [Chakraborty et al. 2014] by the neighboring communities.

Permanence provides a *heterogeneous* vertex-centric measure (in the range 1 to -1) of the extent to which a vertex belongs to its community. A value of 1 indicates that a vertex is placed correctly in its community, and a value of -1 indicates the vertex does not belong to its assigned community. The permanence of the network is given by the average permanence of all the vertices of a network. It is easy to see that the more correctly the vertices are placed in their communities, the higher is the overall

permanence. Using this concept, we propose a new community detection algorithm, *MaxPerm,* based on optimizing the permanence of the network [Chakraborty et al. 2014].

In this article, after providing background information on the related work in community detection (Section 2) and the datasets that we used in our experiments (Section 3), we explain the rationale behind creating the permanence metric (Section 4). We show how change in permanence is more commensurate to the perturbation of ground-truth communities as compared to other competing metrics (Section 6) and present a community detection, *MaxPerm*, based on maximizing permanence (Section 8). We extend this earlier work with the following new contributions.

—*In-depth study of network parameters affecting the value of permanence:* We study how the distribution of different network parameters, such as connectivity and clustering coefficient affect the value of permanence (Sections 5 and 6).
—*Use of permanence to understand the structure of the network:* We show that permanence can provide a better understanding of the structure of the network, such as how to strengthen the communities and revealing the core periphery based on the community structure, as well as its use in applications such as vertex persistence in communities of evolving networks and identifying effective initiators for message spreading (Section 7).
—*In-depth analysis of communities obtained using MaxPerm:* The communities obtained by MaxPerm are, in general, of a smaller size than those obtained by other community detection methods. We study the structure of the communities obtained and show that these communities are actually well defined sub-communities (Section 8).
—*Algorithmic factors affecting results of MaxPerm:* We study how different algorithmic factors such as ordering of the vertices and selection of initial seed communities affects the results of MaxPerm (Sections 9 and 10).
—*Analytical proof on correctness:* We provide analytical results to demonstrate how finding communities by maximizing permanence reduces the existing limitations of community detection algorithms such as resolution limit, degeneracy of solutions, and asymptotic growth of values (Section 11).

We make our experimental codes available in the spirit of reproducible research: http://cnerg.org/permanence.

2. RELATED WORK

We present the ongoing research on two aspects of community detection (i) algorithms to detect community and (ii) metrics for evaluating the correctness of the obtained community.

2.1. Community Detection Algorithms

Most of the research in community detection algorithms are based on the idea that a community is a set of nodes that has more and/or better links between its members than with the remainder of the network. Work in this area encompasses many different approaches including, modularity optimization [Blondel et al. 2008; Clauset et al. 2004; Guimera and Amaral 2005; Newman 2004b, 2006], spectral graph-partitioning algorithm [Newman 2013; Richardson et al. 2009], clique percolation [Farkas et al. 2007; Palla et al. 2005], local expansion [Baumes et al. 2005; Lancichinetti et al. 2009], fuzzy clustering [Psorakis et al. 2011; Sun et al. 2011], link partitioning [Ahn et al. 2010; Evans and Lambiotte 2009], random-walk-based approach [De Meo et al. 2013; Pons and Latapy 2006], information theoretic approach [Rosvall and Bergstrom 2007, 2008], diffusion-based approach [Raghavan et al. 2007], significance-based approach

[Lancichinetti et al. 2010], and label propagation [Raghavan et al. 2007; Xie and Szymanski 2011, 2012].

However, most of these algorithms produce different community assignments if certain algorithmic factors, such as the order in which the vertices are processed, change. Lancichinetti and Fortunato [2012] proposed consensus clustering by re-weighting the edges based on how many times the pair of vertices were allocated to the same community, for different identification methods. Several pre-processing techniques [Bader et al. 2013; Riedy et al. 2011] have been developed to improve the quality of the solution. These methods form an initial estimate of the community allocation over a small percentage of the vertices and then refine this estimate over successive steps. Recently, Chakraborty et al. [2013] pointed out how vertex ordering influences the results of the community detection algorithms. They identified invariant groups of vertices (named as "constant communities") whose assignment to communities is not affected by vertex ordering.

2.2. Community Evaluation Metrics

Most community detection algorithms are based on optimizing a combinatorial metric. Examples of such metrics include conductance [Leskovec et al. 2009; Kannan et al. 2000; Shi and Malik 2000], cut-ratio [Fortunato 2010; Leskovec et al. 2010]; however, the most popular and widely accepted metric is modularity [Newman 2006; Newman and Girvan 2004]. It is defined as the difference (relative to the total number of edges) between the actual and the expected (in a randomized graph with the same number of nodes and the same degree sequence) number of edges inside a given community. Although initially defined for undirected and unweighted networks, the definition of modularity has been extended to capture community structure in weighted [Newman 2004a] and directed [Leicht and Newman 2008] networks.

It was also demonstrated that modularity suffers from a *resolution limit*, that is, by optimizing modularity we cannot find communities smaller than a threshold size [Fortunato and Barthelemy 2007] or weight [Berry et al. 2011]. The threshold depends on the total number (or total weight) of edges in the network and on the degree of interconnectedness between communities. Later, Good et al. [2010] also showed that optimizing modularity can lead to *degeneracy of solutions*, i.e., an exponential number of high (and nearly equal) modularity but structurally distinct solutions from a single graph. They also studied the (*asymptotic*) *growth* of modularity, showing that it depends strongly on both the size of the network and the number of modules it contains.

To address the resolution limit problem, multi-resolution versions of modularity [Arenas et al. 2008; Reichardt and Bornholdt 2006] were proposed to allow researchers to specify a tunable target resolution limit parameter. Lambiotte [2010] proposed different types of multi-resolution quality functions to tackle resolution limit problem. He et al. [2013] considered different community densities as good quality measures for community identification, which do not suffer from resolution limits. Furthermore, Lancichinetti and Fortunato [2011] stated that even those multi-resolution versions of modularity are inclined to merge the smallest well-formed communities and to split the largest well-formed communities. Recently, Chen et al. [2013] proposed *Modularity Density* metric to solve the problems raised by Lancichinetti and Fortunato [2011]. A detailed review can be found in Chakraborty et al. [2016a].

Metrics to compare with ground-truth communities: Although all these metrics mentioned above are useful in analytically evaluating a community structure, a stronger measure of correctness is to compare the obtained community structure with the actual known community structure (ground-truth) of a network. To compare these two community structures, different validation metrics have been proposed, such as

Table I. Properties of Real-World Networks

Network	n	e	$< k >$	k_{max}	c	n_c^{max}	n_c^{min}
Football	115	613	10.57	12	12	13	5
Railway	301	1,224	6.36	48	21	46	1
Coauthorship	103,677	352,183	5.53	1,230	24	14,404	34

n and e are the number of nodes and edges, c is the number of communities, $< k >$ and k_{max} are its average and maximum degree, n_c^{min} and n_c^{max} are the sizes of its smallest and largest communities, respectively.

Normalized Mutual Information (NMI) [Danon et al. 2005], Adjusted Rand Index (ARI) [Hubert and Arabie 1985], and Purity (PU) [Manning et al. 2008]. However, Orman et al. [2012] argued that these metrics are not completely relevant in the context of network analysis, because they ignore the network structure. They proposed the weighted versions of these measures where misplacing a high degree vertex would incur higher penalty compared to a low-degree vertex. In our experiments, we, therefore, also use the weighted versions of these measures, namely Weighted-NMI (*W-NMI*), Weighted-ARI (*W-ARI*), and Weighted-Purity (*W-PU*) (we refer the reader to the paper Orman et al. [2012] for the detailed descriptions of these metrics). Note that all the metrics are bounded between 0 (no matching) and 1 (perfect matching).

3. NETWORK DATASETS AND GROUND-TRUTH COMMUNITIES

We examine a set of artificially generated networks and three real-world complex networks whose underlying ground-truth community structures are known to us. The brief description of the used datasets and their ground-truth communities are mentioned below.

3.1. Synthetic Networks

We select the Lancichinetti-Fortunato-Radicchi (LFR) benchmark model [Lancichinetti and Fortunato 2009] to generate artificial networks with a community structure. The model allows us to control directly the following properties: number of nodes n, desired average degree k and maximal degree k_{max}, exponent γ for the degree distribution, exponent β for the community size distribution, and mixing coefficient μ. The parameter μ represents the desired average proportion of links between a node and the nodes located outside its community, called *inter-community links*. Unless otherwise stated, the LFR network is generated with the number of nodes (n) as 1,000, and μ is varied from 0.1 to 0.6. For the rest of the parameters, we use the default value of the parameters mentioned in the implementation[1] designed by Lancichinetti and Fortunato [2009].

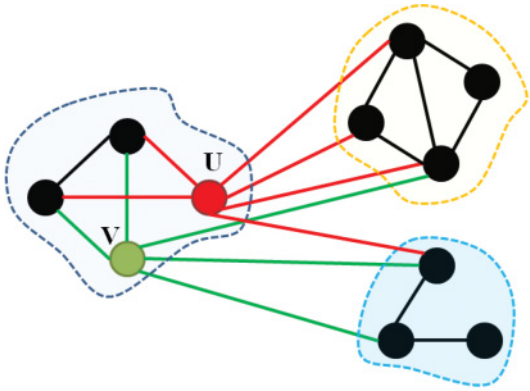
3.2. Real-World Networks

We use three real-world networks mentioned below whose ground-truth community structures are known *a priori*. The properties of these dataset are summarized in Table I.

Football network [Girvan and Newman 2002] contains the network of American football games between Division IA colleges during the regular season of Fall 2000. The vertices in the graph represent teams (identified by their college names) and edges represent regular season games between the two teams they connect. The teams are divided into conferences (indicating communities) containing around 8–12 teams each.

Railway network [Ghosh et al. 2011] consists of nodes representing stations, where two stations s_i and s_j are connected by an edge if there exists at least one train route such that both s_i and s_j are scheduled halts on that route. Here, the communities are

[1]https://sites.google.com/site/santofortunato/inthepress2.



Vertex	D(.)	I(.)	$E_{max}(.)$	$C_{in}(.)$	Perm(.)
U	6	2	3	1	0.11
V	5	2	2	1	0.20

Fig. 1. (Color online) Toy example depicting *permanence* of two vertices u and v. The communities are represented by broken lines.

states/provinces of India since the number of trains within each state is much higher than the trains in-between two states.

Coauthorship network [Chakrabort et al. 2013] is derived from the citation dataset.[2] Here, each node represents an author and an undirected edge between authors is drawn if the two authors collaborate at least once via publishing a paper. The communities are marked by the research fields since authors have a tendency to collaborate with other authors within the same field. Besides the aggregated network, we also create some intermediate networks mentioned in Table VII by cumulatively aggregating all the vertices and edges over each year, e.g., 1960–1971, 1960–1972, . . . , 1960–1980.

4. DEFINING PERMANENCE

In this section, we describe the permanence metric and the two primary concepts behind its formulation.

CONCEPT I: *A vertex should have more number of internal connections than the number of connections to any of the external neighboring communities.*

Most optimization metrics consider the *total number of external neighbors* of the vertex. However, in our earlier experiment [Chakraborty et al. 2014; Chakraborty 2015; Chakraborty et al. 2016b], we empirically demonstrated that a group of vertices are likely to be placed together so long as the number of internal connections is *larger* than the number of connections to *any one single external community*. In other words, a vertex that has connections to some external communities experiences a *separate* "pull" from each of these external communities. In formulating permanence, we consider the maximum pull, which is proportional to the maximum number of connections to an external community (see Figure 1).

CONCEPT II: *Within the substructure of a community, the internal neighbors of the vertex should be highly connected among each other.*

[2]http://cnerg.org/.

Most optimization metrics only consider the internal connections of a vertex within its own community. However, how strongly a vertex is connected also depends on whether its internal neighbors are connected with each other. To measure this connectedness of a vertex, we compute the clustering coefficient of the vertex with respect to its internal neighbors. For a vertex v belonging to community c, it is measured by the ratio between the actual number of edges among the neighbors (which also belong to c) of v and the total number of possible edges among the neighbors [Holland and Leinhardt 1971]. The higher this internal clustering coefficient, the more tightly the vertex is connected to its community (see Figure 1).

We combine these two criteria to formulate permanence of a vertex v, as follows:

$$Perm(v) = \left[\frac{I(v)}{E_{max}(v)} \times \frac{1}{D(v)}\right] - \left[1 - c_{in}(v)\right], \tag{1}$$

where $I(v)$ is the number of internal (in its own community) neighbors of v, $E_{max}(v)$ is the maximum number of connections of v to any one of the external communities, $D(v)$ is the degree of v and $c_{in}(v)$ is the clustering coefficient among the internal neighbors of v. Figure 1 presents a toy example to calculate the permanence of a vertex.

For vertices that do not have any external connections, $Perm(v)$ is considered to be equal to the internal clustering coefficient (i.e., $Perm(v) = c_{in}(v)$). If the number of internal connections, $I(v)$, is less than 2, we set the internal clustering coefficient, $c_{in}(v)$, to be 0. Therefore, for a vertex in a singleton community, $Perm(v) = 0$.

The maximum value of $Perm(v)$ is 1 and is obtained when vertex v is an internal node and part of a clique. The lower bound of $Perm(v)$ is close to -1. This is obtained when $I(v) \ll D(v)$ such that $\frac{I(v)}{D(v)E_{max}(v)} \approx 0$ and $c_{in}(v) = 0$. Therefore, for every vertex v, $-1 < Perm(v) \leq 1$. The permanence of a graph $G(V, E)$, where V is the set of vertices and E is the set of edges, is given by $Perm(G) = \frac{1}{|V|}\sum_{v \in V} Perm(v)$. For a graph $G(V, E)$, the range is $-1 < Perm(G) \leq 1$. $Perm(G)$ will be closer to 1 if the majority of vertices have high permanence that is the vertices are in well-defined communities. This can happen only if the network inherently possesses a strong community structure.

5. EFFECT OF INDIVIDUAL COMPONENTS ON PERMANENCE

In this section, we study the distribution of permanence values corresponding to the vertices in the graph based on their communities. We first compute the permanence of each vertex based on the ground-truth communities of the benchmark networks. We divide the permanence values ranging from -1 to 1 into 20 bins where the low (high) numbered bins contain nodes with lower (higher) permanence. We plot the bins on x-axis, and for each bin, on the y-axis, we plot the fraction of vertices whose permanence value falls in that bin. We observe in Figures 2(a) that this curve follows a Gaussian-like distribution, i.e., there are few vertices with very high or very low permanence values with a peak at the intermediate values. The peak shifts from left to right with the decrease of μ value in the LFR network (keeping the other parameters of LFR constant). The shift in the peak shows that as the communities get more well defined with the decrease of μ, most vertices move toward higher permanence. Figure 2(b) shows that Football network also follows similar kind of behavior, where most of the vertices fall in medium Perm range. However, for Railway and Coauthorship networks, the curve follows "U-shaped" pattern, indicating maximum vertices falling in either very low or very high permanence buckets.

This phenomenon indicates that the community structure is not very clear in these networks. Recall that we have computed communities from ground-truths. The high proportion of lower values indicates that the networks contain entities that are not

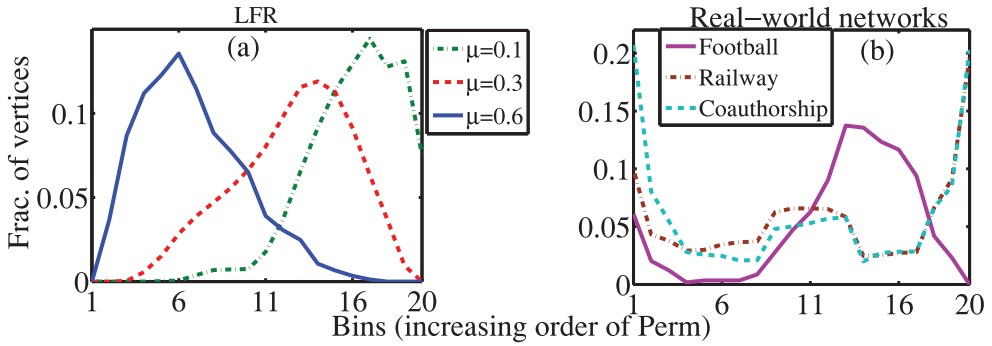


Fig. 2. (Color online) Distribution of the values of Perm for different networks. The value of Perm of vertices is equally divided into 20 buckets indicated in x-axis (bin 1: $-1 \leq Perm < -0.9, \ldots,$ bin 20: $0.9 \leq Perm \leq 1$).

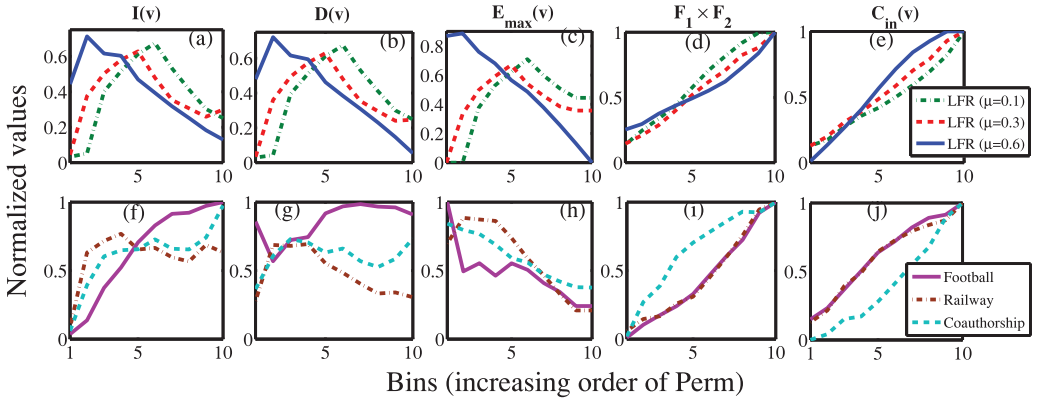


Fig. 3. (Color online) The relation of average Perm with (a) and (f) $< I(v) >$, average internal neighbors per vertex, (b) and (g) $< D(v) >$, average degree per vertex, (c) and (h) $< E_{max}(v) >$, average maximum connections to an external community per vertex, (d) and (i) average value of $F_1 \times F_2 = \frac{I(v)}{D(v)} \times \frac{1}{E_{max}(v)}$ per vertex, (e) and (j) $< C_{in}(v) >$, average internal clustering coefficient per vertex for LFR (upper panel) and real-world (lower panel) networks.

easy to classify, such as railway stations that are at the border of the states or authors who publish in multiple fields.

To understand the dependence on each component of the permanence equation (Equation (1)), we further plot permanence with respect to each individual component, i.e., $I(v)$, $D(v)$, $E_{max}(v)$, $C_{in}(v)$ and their combination in Figure 3. Figure 3(a) shows a decreasing trend of $I(v)$ with the increase of permanence for LFR, where the pattern is completely opposite for real-world networks as shown in Figure 3(f). The trend is almost similar for the relation between $D(v)$ and permanence in Figures 3(b) and (g). However, here most of the real-world networks except Football show similar pattern with that of LFR, where high degree nodes tend to exhibit low or medium permanence value. Furthermore, we plot the relation between permanence and $E_{max}(v)$ in Figures 3(c) and (h) and observe that though all the real-world networks show an inverse relation, for two LFR networks ($\mu = 0.3$ and $\mu = 0.6$), it initially increases and then starts decreasing. From these observations, one cannot find any universal relation as such among different networks. However, once we combine these factors together and plot the dependence between permanence and $\frac{I(v)}{D(v)} \times \frac{1}{E_{max}(v)}$ in Figures 3(d) and (i), we observe a consistent behavior for all the networks in that the value of the combination

tends to increase almost linearly with permanence. A similar trend is followed in Figures 3(e) and (j) wherein the value of internal clustering coefficient tends to increase with the increase of permanence. These results show that permanence depends on two factors – (i) the combined effect of $\frac{I(v)}{D(v)} \times \frac{1}{E_{max}(v)}$, and (ii) the value of $C_{in}(v)$.

6. EFFECT OF PERTURBATIONS ON GROUND-TRUTH COMMUNITIES

One of the crucial measures for an effective community scoring metric is how it behaves under different perturbations of the ground-truth community structure [Yang and Leskovec 2012]. The metric should be robust to small perturbations of the ground-truth communities, such as when groupings of nodes that differ very slightly from the original ground-truth grouping. Furthermore, the metric should also be sensitive to large perturbations. If the change is so large that the ground-truth structure dissolves to a random set of nodes, then the value of the scoring function should be low. In this section, we compare the change in value of permanence with three other community scoring metrics and demonstrate that among them permanence is both robust to noise and sensitive to large changes in the network.

6.1. Community Scoring Metrics

We consider the following community scoring metrics.

—*Modularity (Mod)*: Modularity [Newman 2006] is defined by the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random. Formally, for a given graph $G(V, E)$, it is quantified as follows:

$$Q = \frac{1}{2m} \sum_{u,v \in V} \left[A_{uv} - \frac{k_u k_v}{2m} \right] \delta(c_u, c_v), \tag{2}$$

where $m = |E|$, A_{uv} is the (u, v) entry of the adjacency matrix A, k_u is the degree of vertex u, c_u is the community of vertex u, and $\delta(c_u, c_v) = 1$ if u and v are in the same community or 0 otherwise.

—*Conductance (Con)*: Conductance [Leskovec et al. 2009] is the ratio between the number of edges inside the cluster and the number of edges leaving the cluster [Kannan et al. 2000; Shi and Malik 2000]. More formally, conductance $\Phi(S)$ of a set of nodes S is defined as follows:

$$\Phi(S) = \frac{C_S}{min(Vol(S), Vol(V \setminus S))}, \tag{3}$$

where C_S denotes the size of the edge boundary, $C_S = |(u, v) : u \in S, v \notin S|$, and $Vol(S) = \sum_{u \in S} d_u$ where d_u is the degree of vertex u.

—*Cut-ratio (Cut)*: Cut-ratio is a standard metric in graph clustering [Fortunato 2010; Leskovec et al. 2010], which is defined as the fraction of all possible edges leaving the cluster S. Formally, given an undirected graph $G(u, v)$, the cut-ratio $\theta(S)$ of a set of nodes S is defined as follows:

$$\theta(S) = \frac{C_S}{n_S(n - n_S)}, \tag{4}$$

where C_S is defined earlier, and $n_S = |S|$.

Note that the higher is the value of modularity, the better is the quality of the community structure; however, for conductance and cut-ratio, the opposite argument is applicable. Therefore, to make these two measures comparable to modularity and permanence, we measure (1-Con) and (1-Cut) for conductance and cut-ratio, respectively.

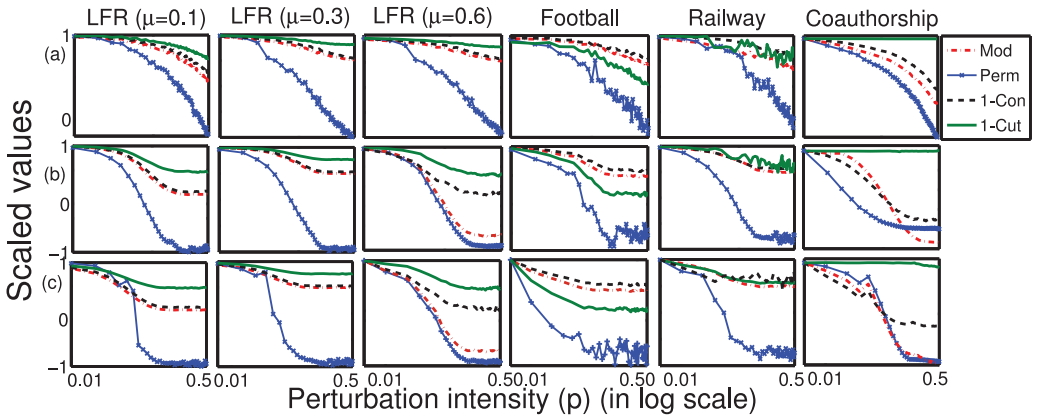


Fig. 4. (Color online) Change in the value of the scoring functions with the increase of perturbation intensity p in (a) edge-based, (b) random, and (c) community-based perturbation strategies. The values are normalized by the maximum value obtained from each function.

6.2. Perturbation Strategies

Given a graph $G =< V, E >$ and *perturbation intensity* p, we restructure the ground-truth community by applying a perturbation strategy. We experiment with the three perturbation strategies as proposed in Yang and Leskovec [2012]. We designate a given ground-truth community as S and the rest of the network as S'.

(1) *Edge-based perturbation*: We select an inter-community edge (u, v) where $u \in S$ and $v \in S'$ (where $S \neq S'$) and assign u to S' and v to S. We continue this process for $p \cdot |E|$ iterations. This strategy preserves the size of S, but certain vertices within the ground-truth community may become disconnected.
(2) *Random perturbation*: We pick two random nodes $u \in S$ and $v \in S'$ (where $S \neq S'$) that may not be connected by an edge and then swap their memberships. We continue this process for $p \cdot |V|$ iterations. Random perturbation maintains the size of S, but the community may have disconnected vertices.
(3) *Community-based perturbation*: This perturbation is similar to the edge-based strategy. However, each community S is perturbed one by one for $p \cdot |S|$, until the nodes of the community are swapped with nodes outside the community. This process is repeated for all the communities separately.

We perturb networks using these perturbation strategies for values of p ranging between 0.01 and 0.5. We compute how the perturbations, as given by the value of p, affect the values of modularity, permanence, 1-Con, and 1-Cut. For small values of p, small change of the scoring function is desirable. This indicates that the scoring function is robust to noise. For high perturbation, which is larger values of p, the communities become more random. Therefore, the values should drop significantly.

6.3. Experimental Results

Figure 4 shows the results of our experiments. To compare equitably across the different scoring functions, we scale the values of each parameter by normalizing with the maximum value obtained from that function. For all three strategies, the scoring function values decrease with the increase of p. Of the three methods community-based produces the fastest degradation, followed by random perturbation and edge-based perturbation is the slowest. However, once p has reached a certain threshold, the decrease

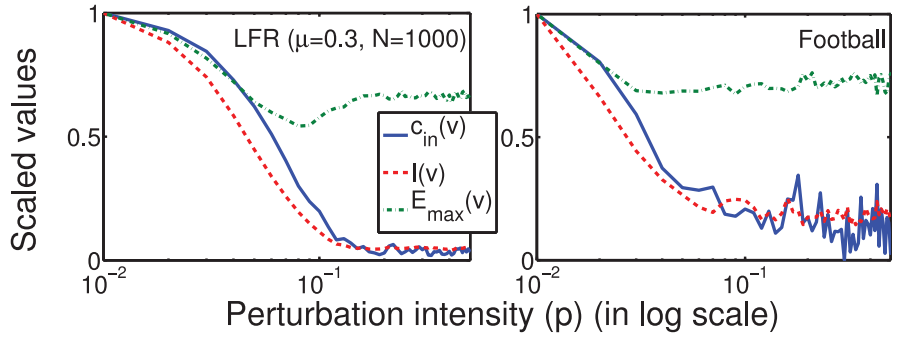


Fig. 5. (Color online) Change in the average values of internal degree $I(v)$, maximum external connections $E_{max}(v)$, and internal clustering-coefficient $c_{in}(v)$ of vertices of two representative networks with the increase of perturbation intensity in random perturbation strategy.

is much faster in permanence, whereas other scoring functions do not always show this sensitivity to large perturbation.

In order to further observe how perturbation affects each of the three major components of the permanence metric, namely the internal degree $I(v)$, the maximum external connections $E_{max}(v)$, and the internal clustering-coefficient $c_{in}(v)$, we further measure the change of their individual values as a function of p. Figure 5 shows the rate of these changes for random perturbation. The most sensitive components of permanence are the internal degree and the average internal clustering-coefficient of vertices. These values tend to be comparatively stable for small perturbations, but degrades significantly as p increases. This provides another justification for incorporating the internal clustering-coefficient as a penalty factor in the formulation of permanence.

7. IMPLICATIONS OF PERMANENCE

We have shown in Chakraborty et al. [2014], using a rank correlation approach, that permanence is a better quality scoring metric as compared to modularity, conductance, and cut-ratio. In Section 6, we demonstrated how permanence is robust to small perturbations and sensitive to larger ones. In this section, we analyze the characteristics of permanence from different perspectives of community structure, based on their known ground-truth structure.

7.1. Measuring Persistence of a Vertex in Its Community

We observe, using the metadata of the co-authorship networks, that the permanence of a vertex is proportional to its persistence, i.e., how long a vertex remains in an evolving community. In the original publication dataset [Chakrabort et al. 2013] as mentioned in Section 3, each scientific article is categorized into one of the 24 research fields (such as Algorithms, Programming Languages, AI, and the like). We tag each author by the field in which she has published maximum papers. Each field corresponds to a community [Chakrabort et al. 2013]. Essentially, we intend to measure the persistence of an author in her own community in terms of her *research age*. For this, we define *research age* (ξ) of an author in a field/community in two different ways as follows.

Definition 1 (*Collective Research Age* (ξ_c)). The collective research age $\xi_c^f(a)$ of an author a in a field/community f is defined by the total number of distinct years author a has published at least one paper in the field f.

Definition 2 (*Discounted Research Age* (ξ_d)). The discounted research age $\xi_d^f(a)$ of an author a in a field/community f is defined by the total number of distinct years

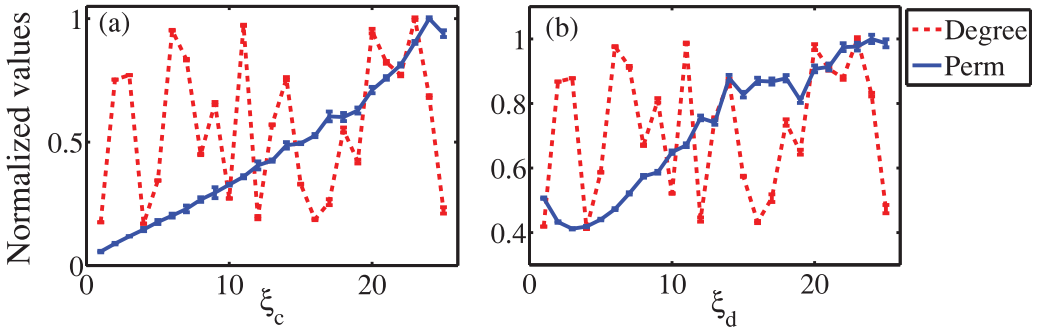


Fig. 6. Changes of average degree and average permanence (with variance) of authors with the increase in (a) collective research age ξ_c, and (b) discounted research age ξ_d.

author a has published at least one paper in the field f, where each year is linearly penalized by the number of its immediate consecutive preceding years when she has not published a single paper on f. The linear penalty is introduced to bring in the effect of "consistency break" in the publication career of an author. Ideally, an author who is publishing consistently in a field should be more persistent than an author publishing in stretches with intermediate gaps. The penalty is used to significantly put more emphasis on the former case than the latter case.

For example, let us assume that an author a has published papers in the following years: 1960, 1965, 1966, 1967, and 1970. Therefore, $\xi_c^f(a) = 5$ and $\xi_d^f(a) = (1+1/4+1+1+1/2) = 3.75$ (year 1965 is penalized by its previous four consecutive unproductive years, similarly for 1970). We then plot the average degree and the average permanence of authors against two types of research ages in Figure 6. We observe that though there is almost no correlation between the average degree and the research age of an author, the permanence value of an author is almost linearly correlated with the research age. This evidence essentially leads to the the following conclusions: (i) permanence of a vertex is a suitable way of representing its persistence in its own community, which cannot be derived from the degree of a vertex, (ii) since the result in Figure 6 is reported over all the authors in different communities, we can also compare the extent of persistence of two vertices belonging to two different communities, i.e., the same permanence value of two vertices in two different communities indicates the equal extent of persistence in their corresponding communities.

7.2. Strengthening the Community Structure

The value of permanence of a vertex signifies its propensity to remain in its own community. Therefore, vertices having low permanence in a community are loosely connected to the community. We explore whether we can strengthen the community structure by deleting vertices with low permanence. Note that when a vertex is deleted from its community, it would also affect the permanence value of the remaining vertices. Therefore, we rank[3] the vertices at the very beginning based on permanence and do not consider further changes in permanence during the deletion. Then, in each step, we measure the quality of the cluster by *edge-density* (the ratio between the actual number of edges and the expected number of edges in that cluster).

For each community, we remove top $n\%$ low-ranked vertices based on permanence and measure the percentage change of edge-density (averaged over all the communities) due to this removal. One can observe in Figure 7 that the edge-density increases with

[3]We use dense ranking scheme to rank the authors.

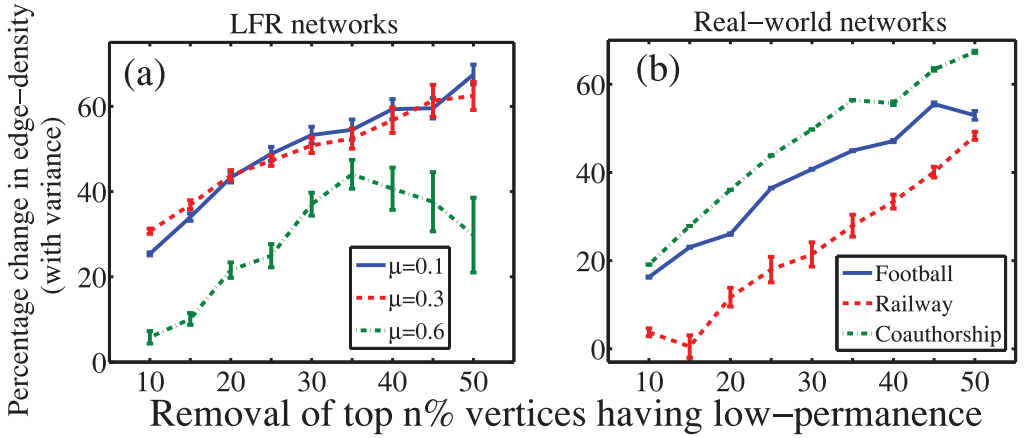


Fig. 7. (Color online) Percentage change in edge-density after removing top $n\%$ (varies from 10% to 50%) low-ranked vertices based on permanence. Each point in this plot is averaged over all the communities, and therefore, the variance is also plotted.

the increase in n. Although overall there is increase in edge-density, we notice that in LFR ($\mu = 0.6$), the edge-density starts decreasing after removing 35% of vertices. The reason might be described as follows. In LFR ($\mu = 0.6$), vertices generally exhibit small permanence value due to the overall inferior community structure. The range of permanence values of vertices in each community of LFR ($\mu = 0.6$) is also not high as compared to the same for LFR ($\mu = 0.1$). Therefore, removing 35% of vertices from LFR ($\mu = 0.6$) might result in the removal of vertices having relatively high ranking based on permanence. On the other hand, since the range is high for LFR ($\mu = 0.1$) and LFR ($\mu = 0.3$), the same extent of deletion of vertices might not affect the high-ranked vertices in the community. However, for these networks, such decrease in edge-density can also be observed for higher extent of deletion (usually beyond 50%, not shown in Figure 7).

7.3. Heterogeneity and Core-Periphery Organization of Community Structure

Although it is implicitly assumed that all the constituent members of a community belong to the community equally, this is not true in reality. Within a community, the extent of involvement and activity may not be same for all members – permanence can capture this *heterogeneity*. The permanence of a node v belonging to a community c indicates the extent to which the node belongs to the community. With this value, several inferences can be drawn about the communities present. For instance, it inherently creates a gradation/ranking of the constituent vertices in a community. This ranking may be important in many cases – for example, in exploring the core-periphery structure of a community.

To explore the relation of permanence of a vertex with its position vis-a-vis core of a community, we use *farness centrality* (d) proposed in Yang and Leskovec [2014] as a measure to locate the position of a vertex within a community. In order to measure farness centrality for each community, we construct the induced subgraph constituting all the nodes in the community and measure average shortest path for each vertex within this subgraph. Thus, the lower is the value of d for a vertex, the closer the vertex is to the core part of the community.[4] We plot average permanence of vertices as a function of farness centrality in Figure 8. We observe that for both LFR and

[4]Farness centrality is just the reverse of closeness centrality in a connected component.

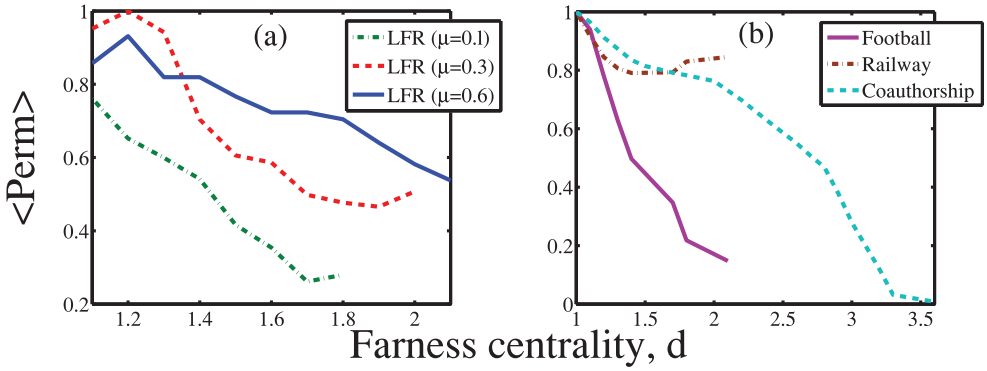


Fig. 8. (Color online) Community-wise average permanence, $< Perm >$ of vertices as a function of farness centrality d for LFR and real-world networks.

Table II. Average of the Assortativity Scores, $< r >$ (Degree Based and *Perm* Based) of the Communities Per Network

$< r >$	LFR ($\mu = 0.1$)	LFR ($\mu = 0.3$)	LFR ($\mu = 0.6$)
Degree based	0.088	0.108	0.082
Perm based	0.520	0.551	0.430

$< r >$	Football	Railway	Coauthorship
Degree based	−0.105	0.153	0.155
Perm based	0.747	0.531	0.489

real-world networks, average permanence decreases with the distance from the center of the community. Therefore, the value of permanence can act as a strong indicator of the position of the vertex in the community.

The next investigation reveals the manner in which the permanence value of vertices decreases from the core. A smooth decrease in value would indicate that the nodes in a community are arranged in layers with each layer of vertices roughly having similar permanence. In order to understand the mixing pattern of vertices, we measure permanence-based assortativity (r)[5] Newman [2003] to observe the preference for a network's nodes in a community c to attach to other nodes that have nearly similar permanence. We divide the permanence values into 20 bins so that nodes within a bin are considered to have equivalent permanence values, and then measure r of a network. For comparison, we also measure degree-based assortativity of vertices in each network. We observe in Table II that both synthetic and real-world networks are highly assortative in terms of permanence, rather than in terms of degree. This result indeed indicates that, in general, a community is organized into several layers, wherein each layer is composed of vertices exhibiting similar permanence, and vertices tend to be highly connected *within each layer* than across different layers.

7.4. Initiator Selection for Message Spreading

Message spreading is one of the challenging problems in complex networks and distributed systems [Chierichetti et al. 2010]. Starting with one source node/initiator having a message, the protocol proceeds in a sequence of synchronous rounds with the goal of delivering it to every node in the network. At every timestep, each node in the system having the message communicates with one node (not having the message)

[5]Assortativity (r) lies between -1 and 1. When $r = 1$, the network is said to have perfect assortative patterns, when $r = 0$ the network is non-assortative, while at $r = -1$ the network is completely disassortative.

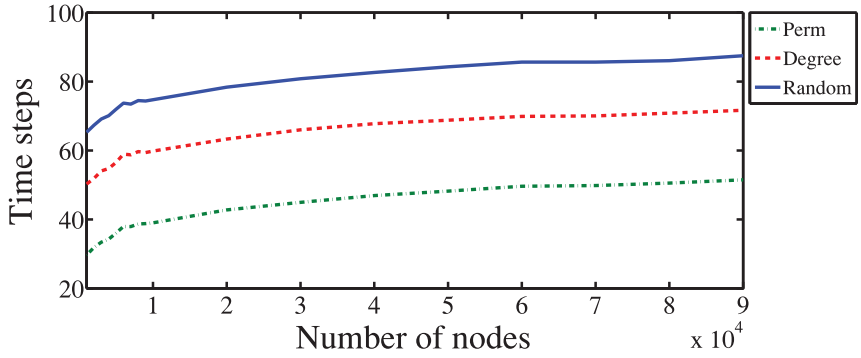


Fig. 9. (Color online) Number of timesteps required to broadcast a message in the LFR network by varying the number of nodes.

in its neighborhood and transfers the message. The algorithm terminates when all the nodes in the system have received the message. A fundamental issue in message spreading is the selection of initiators. Selecting initiators based on the degree leads to faster spreading (requires less steps in average) of message than the random node selection [Demers et al. 1987]. Since vertices with higher permanence form the core of the community, we posit that initiator selection based on permanence would help in faster dissemination of the message. Note that the message spreading algorithms are based on only the local view of the vertices; therefore, global methods such as those described in Kempe et al. [2003] will not be applicable under this formulation.

To validate this hypothesis, we consider the LFR network and vary the number of nodes from 10,000 to 90,000, keeping the other parameters constant (see Section 3). We select multiple initiators by picking one node per community present in the ground-truth structure based on the following criteria separately: (i) random, (ii) highest degree, and (iii) highest permanence. For each network configuration, we run 500 simulations and report in Figure 9 the average number of timesteps required for the message to reach all the nodes in the network. We observe that the permanence-based initiator selection from ground-truth communities requires minimum timesteps to spread the message compared to the degree-based selection.

8. COMMUNITY DETECTION USING PERMANENCE MAXIMIZATION

We now present an algorithm for detecting communities by maximizing the permanence of the network. Our algorithm, *MaxPerm* (pseudocode in Algorithm 1), finds high permanence partitions of large networks using a greedy agglomerative approach, similar to the methods used in Blondel et al. [2008] and Clauset et al. [2004].

Initially, the vertices are assigned to random connected subgraphs as their initial seed community. At each iteration, a vertex is moved from one community to another if its permanence increases. This process is continued for several iterations until the value of the permanence of the network is unchanged. Although convergence to a fixed value is not theoretically guaranteed, we have observed that on test cases the algorithm converges within 10 iterations. As in the case of modularity-maximization methods, we observe that creation of appropriate seed communities can help improve the quality of community detection. We shall discuss this issue later in Section 10.

8.1. Computational Complexity and Strategies for Improvement

The computational complexity of the algorithm is as follows. The most expensive part in computing the permanence of a vertex v is the internal clustering co-efficient, $c_{in}(v)$. Given the degree of vertex v is $D(v)$, computing $c_{in}(v)$ takes time $O(D(v)^2)$. For each

ALGORITHM 1: MaxPerm: Community Detection Using *Max*imizing *Perm*anence

```
Data: A graph G(V, E)
Result: Permanence of G; Detected communities
Each vertex is assigned to its seed community;
Set value of maximum iteration as maxIt ;
vertices ← |V|;
Sum ← 0;
Old_Sum ← −1;
Itern ← 0;
while Sum ≠ Old_Sum and Itern < maxIt do
    Itern ← Itern + 1;
    Old_Sum ← Sum;
    Sum ← 0;
    forall v ∈ V do
        //  Compute current permanence of v
        cur_p ← Perm(v);
        if cur_p == 1 then
            Sum ← Sum + cur_p;
            continue;;
        end
        cur_p_neig ← 0;
        Neig(v)=set of neighbors of v;
        forall u ∈ Neig(v) do
            //  Compute current permanence of u
            cur_p_neig ← cur_p_neig + Perm(u);
        end
        // Comm(v) is the set of neighboring communities of v
        forall C ∈ Comm(v) do
            Move v to community C;
            // Compute permanence of v in community C
            n_p ← Perm(v) ;
            // Neighbors of v are affected for this movement
            n_p_neig ← 0;
            forall u ∈ Neig(v) do
                //  Compute new permanence of u
                n_p_neig ← n_p_neig + Perm(u);
            end
            if (cur_p < n_p) and (cur_p_neig < n_p_neig) then
                cur_p ← n_p;
            else
                Replace v to its original community;
            end
        end
        Sum ← Sum + cur_p;
    end
end
Netw_perm = Sum/vertices ;                                  // Permanence of G
Return Netw_perm;
```

vertex, we compute the permanence for its own and each of its neighboring communities. Let the number of neighboring communities of vertex v at iteration k be $C_k(v)$. Let the total number of iterations required by the algorithm be $maxIt$. Therefore, the time to execute $MaxPerm$ is $\sum_{k=1}^{k=maxIt} \sum_{v=1}^{v=|V|} (C_k(v)O(D(v)^2))$.

Let d_{max} be the maximum degree of the network. We also note that the maximum number of communities that a node can belong to is $d_{max} + 1$. The upper bound for $MaxPerm$ is $O(maxIt \cdot |V| \cdot d_{max}^3)$. Since only a few nodes of the network has the highest degree, in practice, the time is much lower than the value given by this upper bound.

The execution time can be further reduced by a few simple strategies. First, instead of recomputing $Perm(v)$, for each community, we can store the number of edges each vertex has in each of its neighboring communities. We update these values only when a vertex or its neighbor changes communities. Second, since we want the permanence to increase if a vertex changes communities, the only communities to consider for relocation are those with both high internal degree and high clustering coefficient. By computing permanence for only communities that satisfy these criteria, we can reduce the computation time. Both these strategies require us to keep track of the communities for neighbors of the vertex, for each vertex. Together, this requires extra storage of order $O(|V|D(v)) \approx O(E)$.

8.2. Baseline Community Detection Algorithms

There exist numerous community detection algorithms, which differ in the way they define the community structure. Here, we select the following set of algorithms and categorize them according to the principle they use to identify communities as per [Orman et al. 2012].

(1) *Modularity-based approaches*: We select three modularity optimization algorithms, namely *FastGreedy* approach [Newman 2004b], *Louvain* [Blondel et al. 2008], and *CNM* [Clauset et al. 2004], which differ in the way they perform this optimization.
(2) *Node similarity-based approaches*: This category deals with the notion that a community is viewed as a group of nodes that are similar to each other, but dissimilar from the rest of the network. It includes *WalkTrap* [Pons and Latapy 2006] that is built on the notion that random walks tend to get trapped into a community.
(3) *Compression-based approaches*: These approaches assume the community structure as a set of regularities in the network topology, which can be used to represent the whole network in a more compact way than the whole adjacency matrix. The best community structure is supposed to be the one maximizing compactness while minimizing information loss. The quality of the representation is assessed through measures derived from information theory. Two popular such algorithms are *InfoMod* [Rosvall and Bergstrom 2007] and *InfoMap* [Rosvall and Bergstrom 2008].
(4) *Significance-based approaches*: According to these approaches, a community structure can be expected under certain circumstances, but groups of densely connected nodes can also appear only by chance. *Order Statistics Local Optimization Method (OSLOM)* [Lancichinetti et al. 2010] is a local optimization method applied to measure the statistical significance of individual communities.
(5) *Diffusion-based approaches*: These approaches rely on the assumption that information is more efficiently exchanged between nodes of the same community. In *Community Overlap Propagation Algorithm (COPRA)* [Raghavan et al. 2007], the information takes the form of a label, and the propagation mechanism relies on a vote between neighbors. Communities are then obtained by considering groups of nodes with the same label.

Each algorithm is used with its default parameters. Although the algorithms OSLOM and COPRA are suitable for overlapping community detection, these are used here to detect mutually exclusive communities (i.e., non-overlapping communities) by setting *a priori* the number of overlapping nodes as zero.

Table III. Differences of MaxPerm with the Other Algorithms for Different Networks

Networks	Louvain	FastGreedy	CNM	WalkTrap	Infomod	Infomap	COPRA	OSLOM
LFR ($\mu = 0.1$)	0.00	0.00	0.14	0.00	0.06	0.00	0.11	0.00
LFR ($\mu = 0.3$)	0.00	0.87	0.40	0.00	0.08	0.00	0.02	0.00
LFR ($\mu = 0.6$)	−0.75	0.02	−0.13	−0.50	−0.20	−0.72	−0.09	−0.68
Football	0.02	0.01	0.30	0.02	0.01	0.00	0.03	0.01
Railway	0.14	0.37	0.20	0.02	0.19	0.02	0.01	0.11
Coauthorship	0.00	0.14	0.05	0.02	−0.04	−0.02	0.09	0.09

Each value is obtained by averaging the values of all six validation metrics. The expanded results are shown in appendix. Positive differences indicate the improvement of our algorithm.

8.3. Validation Measures

A stronger test of the correctness of the community detection algorithm, however, is by comparing the obtained community with a given ground-truth structure. We use three standard validation metrics, namely NMI [Danon et al. 2005], ARI [Hubert and Arabie 1985], and PU [Manning et al. 2008] to measure the accuracy of the detected communities with respect to the ground-truth community structure. Orman et al. [2012] argue that these metrics ignore the network structure and propose the weighted versions of these measures where misplacing a high degree vertices would incur higher penalty. We therefore also use the weighted versions of these measures, namely *W-NMI*, *W-ARI*, and *W-PU*. All these metrics are bounded between 0 (no matching) and 1 (perfect matching).

8.4. Performance Analysis

Table III shows results of the improvement of our method (as differences between the NMI of a baseline algorithm to MaxPerm) compared to the algorithms given in Section 8.2 and averaged over all the validation metrics. The detailed results of improvement in terms of each validation measure separately are shown in Table VIII of Appendix.

For the LFR ($\mu = 0.1$) network, MaxPerm is as efficient as Louvain, WalkTrap, Infomap, and OSLOM (and achieves an average accuracy of 0.95), which is followed by FastGreedy, Infomod, COPRA, and Clauset-Newman-Moore (CNM). For the LFR ($\mu = 0.3$) network, MaxPerm once again seems to be comparable in performance to Louvain, WalkTrap, Infomap, and OSLOM (and achieves average accuracy of 0.86), which is followed by COPRA, Infomod, CNM, and FastGreedy. However, MaxPerm does not work well for the LFR ($\mu = 0.6$) network. In this case, Louvain outperforms other competing algorithms with the average accuracy of 0.53, which is followed by Infomap, OSLOM, WalkTrap, Infomod, CNM, MaxPerm, and FastGreedy. We hypothesize that maximizing permanence performs better at identifying communities from networks which actually possess a modular structure. If the communities are not that well-separated as in LFR ($\mu = 0.6$), more singleton communities are formed and the permanence value tends to degrade. We shall address this issue further at the end of this section.

For *Football* network, MaxPerm achieves highest average accuracy of 0.86 with Infomap, followed by FastGreedy, Infomod, and OSLOM at the second position, Louvain and WalkTrap at the third position, COPRA and CNM at the fourth and fifth positions, respectively. For *Railway* network, MaxPerm completely dominates others with the average accuracy of 0.78, which is followed by COPRA, Infomap, WalkTrap, OSLOM, Louvain, CNM, and FastGreedy. Once again, MaxPerm shows moderate performance for *Coauthorship* network and seems to be as good as Louvain (achieves average accuracy of 0.34). Although MaxPerm seems to be superior than FastGreedy, CNM, WalkTrap, COPRA, and OSLOM, they are dominated by two information-theoretic approaches (Infomod and Infomap).

Table IV. Average Pairwise Similarities between Outputs of The Community Detection Algorithms on Different LFR Networks

Validation measure	LFR ($\mu = 0.1$)	LFR ($\mu = 0.3$)	LFR ($\mu = 0.6$)
NMI	0.95	0.82	0.53
ARI	0.98	0.79	0.48
PU	0.99	0.85	0.56
W-NMI	0.94	0.85	0.54
W-ARI	0.97	0.78	0.50
W-PU	0.98	0.83	0.57

To summarize our algorithm is competitive with other standard algorithms, except for LFR($\mu = 0.6$) and coauthorship networks. In order to understand this behavior, we further look at the community structure of these two networks individually.

8.5. Analyzing the Community Structure of LFR ($\mu = 0.6$)

To understand why MaxPerm is not as competitive for LFR ($\mu = 0.6$), we study the quality of the ground-truth communities. We observe that the average internal clustering coefficient in the network decreases with increase in μ. The value is 0.78 for LFR ($\mu = 0.1$), it reduces to 0.36 for LFR ($\mu = 0.6$). Moreover, 97% of vertices in ground-truth communities of LFR ($\mu = 0.6$) have less internal connections than the external connections. In contrast, LFR ($\mu = 0.1$) and LFR ($\mu = 0.3$) have almost no such nodes. This indicates that the LFR ($\mu = 0.6$) network does not have modular structure.

To further validate this hypothesis, we measure the similarity of the communities obtained by different community detection algorithms using the validation measures. The results in Table IV clearly show that the values of the validation metrics decrease with the increase in μ. This is because as μ increases, the communities in the LFR network become more fuzzy and the consensus between the outputs of different algorithms dilutes. The results of a good community detection algorithm should reflect such absence of modular structure in the network (hence show poor performance). In the absence of a modular structure, the permanence-based algorithm tends to detect more singleton communities rather than arbitrarily assigning vertices into communities.

8.6. Analyzing the Community Structure of Coauthorship Network

To explain the results of MaxPerm obtained from coauthorship network, we analyze the metadata of the communities. The titles and the abstract written by the authors in each community obtained by MaxPerm show that our method splits large ground-truth communities into denser submodules.

This phenomenon is more prominent in older research areas such as algorithms and theory, databases, etc. These submodules are actually the subfields (sub-communities) of a field (community) in computer science domain. Few examples of such sub-communities obtained from our algorithm are noted in Table VI. Thus, our algorithm, in addition to identifying well-defined communities, is also able to unfold the hierarchical organization of a network (see Section 8.7 for more discussion).

8.7. Detection of Small-Sized Communities

Many optimization algorithms tend to ignore smaller size communities and combine them to produce larger communities. This phenomenon is known as the "resolution limit" problem. Here, we provide experimental results to show that MaxPerm can mitigate the effects of resolution limit (analytical proof is given in Section 11).

In our test suite, we observe that all the competing algorithms produce larger sized communities as compared to those obtained by permanence. Figure 10 shows the community size distribution of the ground-truth structure and that obtained from other community detection algorithms. We observe that with the increase of

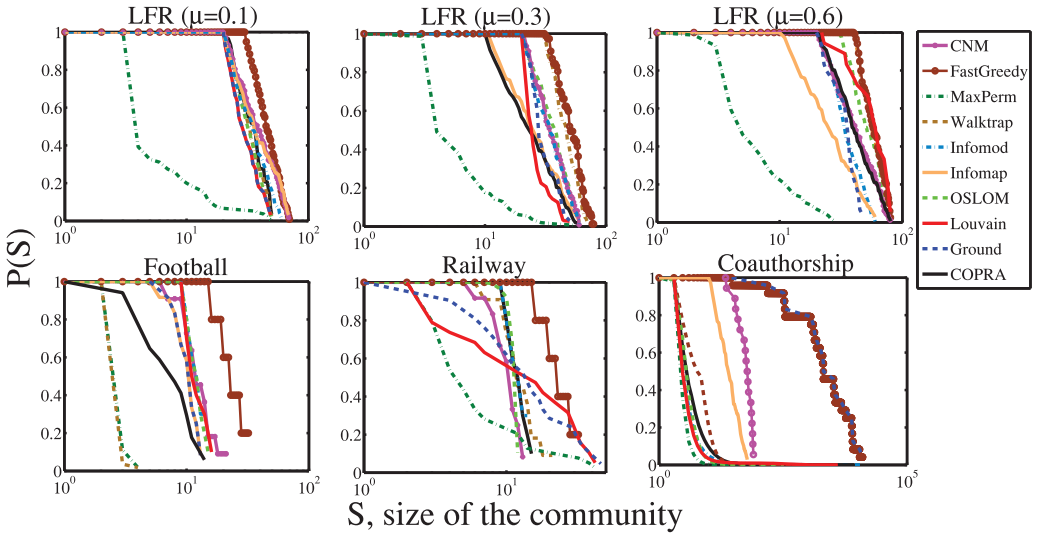


Fig. 10. (Color online) Distribution of the community size obtained from the ground-truth structure vis-a-vis other community detection algorithms.

μ in LFR network, the size distributions obtained from ground-truth and Louvain, respectively, start separating, whereas the pattern obtained from MaxPerm remains almost the same in that most of the communities are small in size. Interestingly, for coauthorship network even if most of the communities in ground-truth are large in size, the communities obtained from Louvain and MaxPerm are almost of similar size.

We, therefore, investigate whether these smaller size communities are arbitrary or actually represent different sub-communities of a large community present in the ground-truth structure. As shown in Table VI, for the co-authorship network, the smaller communities actually represented sub-disciplines of a larger academic field. However, such metadata are not available for all networks. To answer this question, we construct a bipartite network consisting of the communities obtained from the algorithm A (denoted as set C_A) as vertices in one partition and the ground-truth communities (denoted as set C_G) as vertices in another partition. We create the edges $C_A \times C_G$ with edge weights derived as follows: The weight of the edge connecting $c_a \in C_A$ and $c_g \in C_G$ is measured by the fraction of vertices in c_a that are also part of c_g. We only consider edges with non-zero weights. If a detected community is mostly subsumed by one ground-truth community, it produces very high edge-weight (ranging from 0.8–1) and very small edge-weight (0–0.2), whereas medium edge-weight (0.4–0.7) indicates that the detected community is equally absorbed in multiple ground-truth communities. For example, assume that c_a contains 10 nodes that are distributed into two ground-truth communities c_g^1 and c_g^2 in two different ways: (*Case 1.*) eight vertices of c_a are in c_g^1 and two are in c_g^2, (*Case 2.*) four vertices of c_a are in c_g^1 and six are in c_g^2. Therefore, in Case 1, the edge weights are 0.8 and 0.2 for $(c_a \rightarrow c_g^1)$ and $(c_a \rightarrow c_g^2)$, respectively, whereas in Case 2, the edge weights are 0.4 and 0.6 for $(c_a \rightarrow c_g^1)$ and $(c_a \rightarrow c_g^2)$, respectively.

We construct such weighted bipartite graphs separately for all the algorithms. In Figure 11, we divide the edge-weights into 10 buckets such that bucket 1 corresponds to higher edge weight. Then, in y-axis, we plot the fraction of edges falling in each bucket. We observe that though the proportion of edges for baseline algorithms is higher in medium weight zone, for MaxPerm most of the edges either fall in higher

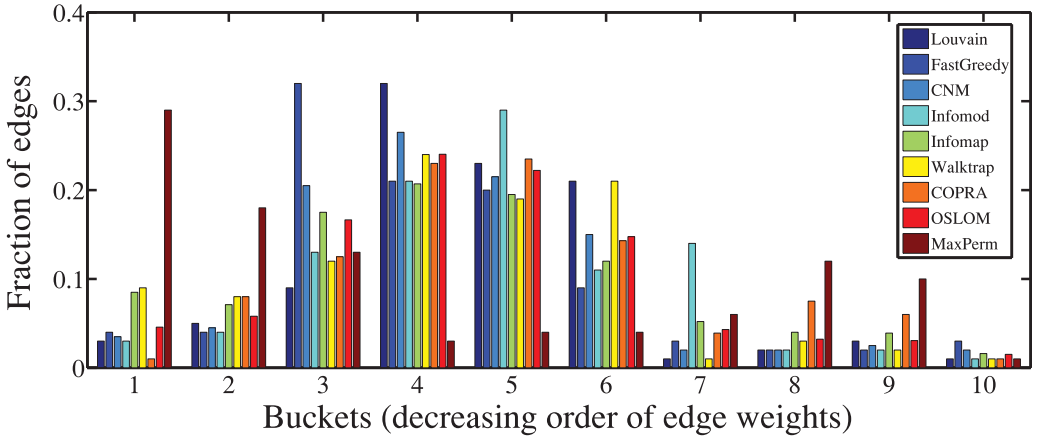


Fig. 11. (Color online) Fraction of edges of the bipartite network with a specific weight. The range of the edge-weight is divided into 10 buckets (bucket 1: 0.9–1, bucket 2: 0.8–0.89, and so on). The bipartite network corresponds to the coauthorship network.

Table V. Size of The Largest Communities Obtained from Different Community Detection Algorithms and Their Similarities with The Ground-Truth Structure

Size of the large communities

	LFR ($\mu = 0.1$)	LFR ($\mu = 0.3$)	LFR ($\mu = 0.6$)	Football	Railway	Coauthorship
Ground-truth	63	49	42	12	13	12,674
Louvain	65	62	57	24	17	1,254
FastGreedy	78	95	76	18	4	9,875
CNM	91	86	72	32	32	11,251
Walktrap	71	83	65	15	14	8,620
Infomod	65	61	46	16	4	324
Infomap	65	59	48	16	4	357
COPRA	54	56	76	20	10	465
OSLOM	57	42	87	18	12	732
MaxPerm	60	49	40	13	13	318

Similarity of largest community obtained from the algorithm with that of the ground-truth

	LFR ($\mu = 0.1$)	LFR ($\mu = 0.3$)	LFR ($\mu = 0.6$)	Football	Railway	Coauthorship
Louvain	0.89	0.70		0.41	0.87	0.70
FastGreedy	0.51	0.32	0.39	0.65	0.52	0.39
CNM	0.82	0.52	0.76	0.31	0.71	0.66
Walktrap	0.88	0.51	0.73	0.57	0.75	0.64
Infomod	0.90	0.79	0.82	0.86	0.84	0.78
Infomap	0.90	0.74	**0.83**	0.86	0.85	0.78
COPRA	0.79	0.67	0.70	0.78	0.52	0.59
OSLOM	0.81	0.81	0.73	0.72	0.68	0.61
MaxPerm	**0.95**	**1**	**0.83**	**0.92**	**0.87**	**0.79**

The bold font in Table V indicates the highest accuracy among the competing algorithms for each dataset.

weighted buckets or lower weighted buckets. This indicates that the communities obtained by MaxPerm are indeed subgroups within one larger community, rather than being scattered across multiple communities.

We also observe that despite finding small communities, the largest size community obtained by MaxPerm best corresponds to the largest ground-truth community. In Table V, we show for all the networks that the size of the largest communities detected by the other algorithms is much larger than the size of the largest community present

Table VI. Example of Communities and Sub-Communities Obtained from Coauthorship Network using The MaxPerm Algorithm

Communities	Sub-communities
Algorithms and theory	Theory of computation; Formal methods; Information & coding theory; Computational geometry; Data structure;
Databases	Models; Query optimization; Database languages; storage; Performance; security, and availability

in the ground-truth structure. We also measure the maximum similarity (using the Jaccard coefficient) between the largest-size community detected by each algorithm and the communities in ground-truth structure and notice that MaxPerm is able to detect largest size community that is most similar to the ground-truth structure (see Table V). These experimental results indicate that MaxPerm is more effective in reducing the effect of resolution limit.

9. EFFECT OF VERTEX ORDERING

Most of the community detection algorithms attempt to optimize certain functions (such as modularity) and therefore are heavily dependent on the order in which vertices are processed. This is an important source of concern among researchers on how to reconcile results where the final outcome can change due to the mere change in vertex ordering [Seifi et al. 2013; Lancichinetti and Fortunato 2012; Delvenne et al. 2010; De Meo et al. 2013; Chakraborty et al. 2013]. In this direction, Chakraborty et al. [2013] show that despite such fluctuations in the final outcome, there exist few invariant groups of vertices in a network that always remain together, and they are known as "constant communities." Furthermore, they study the change in community structure based on the number of constant communities by a metric, called *sensitivity* (ϕ), which is measured as the ratio of the number of constant communities to the total number of vertices. For a particular network, if the value of ϕ for an algorithm remains consistent over different vertex orderings, the algorithm would be qualified to be resilient to the effect of vertex ordering.

We plot the value of sensitivity over different vertex orderings for each algorithm in Figure 12. In x-axis, we plot the number of different permutations of the vertices. For fair comparison, we normalize the sensitivity values by the minimum value for each algorithm and plot it in y-axis so that the sensitivity profiles of all the algorithms start from 1. For a particular network, the lower the value of sensitivity of an algorithm across different perturbations is, the better is the algorithm resilient to the initial vertex ordering. There are two consequences of this result (i) for a specific LFR network, say LFR ($\mu = 0.3$), we can observe that MaxPerm remains almost consistent in terms of sensitivity over different iterations, which is in most cases followed by Infomod, Infomap, and Louvain. COPRA and OSLOM perform worst among the others. (ii) Across different LFR networks, we observe that with the increase of μ, the performances of all the algorithms start deteriorating. The reason could be that with the increase of μ, communities in the LFR network become fuzzier, and therefore multiple community partitions can be equally good. Similar result is observed across different real-world networks where the algorithms tend to be largely insensitive for coauthorship network due to the lack of clear separation between communities.

10. EFFECT OF SEED COMMUNITY SELECTION

The first step before starting the iterations to maximize permanence is to place the vertices in initial (seed) communities. The importance of seed communities in detecting

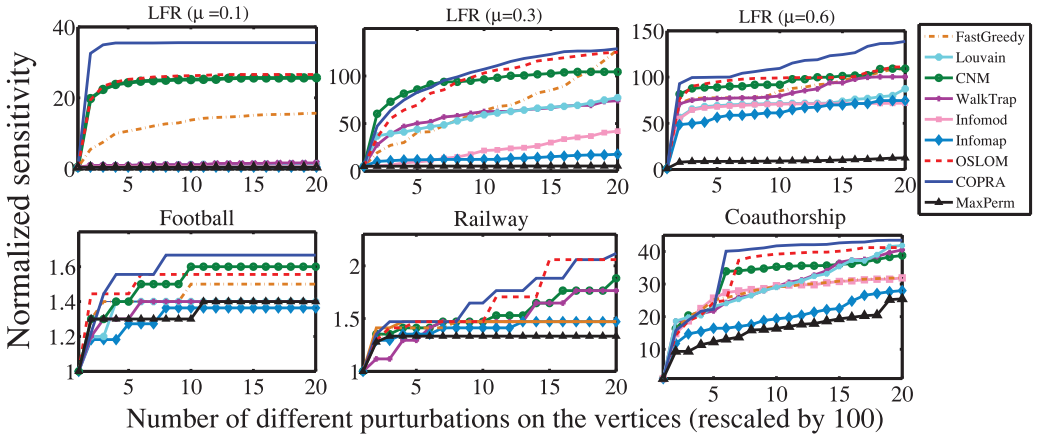


Fig. 12. Sensitivity of each algorithm across 2,000 permutations. The x-axis is rescaled by a constant factor of 100. Y-axis indicates the value of sensitivity as it changes over the permutations. For better visualization, we rescale the value of sensitivity with the minimum value for each algorithm so that the sensitivity of all the algorithms always starts from 1.

communities has been studied for other metrics as in Riedy et al. [2011]. In this section, we explore how it affects the MaxPerm algorithm.

We note that vertices in seed communities should consist of connected subgraphs. If the vertices in the communities are not connected to each other, then their initial permanence will be zero (because $I(v)$ is zero), and moving to a neighboring community does not improve this value. We consider three seed selection strategies as follows.

—*Pairwise*: Two vertices are assigned to the same seed community if they are connected by an edge. If we encounter a vertex whose neighbors have all been already assigned to communities, that unmatched vertex is kept as a singleton. This is the fastest out of the three seeding methods.
—*High Degree*: We first order the vertices in the decreasing order of degree. The vertex with the highest degree and its neighbors are assigned to the same community. We continue combining each unassigned vertex in the sorted list and its unassigned neighbors into a community. This seeding is based on maximizing $I(v)$ for the high degree vertices.
—*High CC*: We order the vertices in the decreasing order of clustering coefficient, and similar to the *high degree*, and combine the vertices with high clustering coefficient and their neighbors in a community. This seeding is based on maximizing $c_{in}(v)$ and is the most expensive of the three methods.

We test the seeding strategies on the four networks (LFR with $\mu = 0.1, 0.3$, Football and Railway) on which MaxPerm consistently outperformed the other algorithms. As can be seen in Figure 13, the best accuracy comes from using the high degree strategy. The reason for this is that pairwise depends on the vertex ordering, and the pairings can change depending on how vertices are numbered. High CC is too restrictive, because once a vertex is in a tightly coupled group, there is less chance for it to migrate to a larger (if slightly) less tightly coupled group. Thus, maximizing permanence using high clustering coefficient tends to fall into local minima. In High Degree, the groupings are less random compared to Pairwise, and they also provide more flexibility for vertices to migrate between communities as compared to High CC. We believe that this is the reason behind High Degree providing the best accuracy.

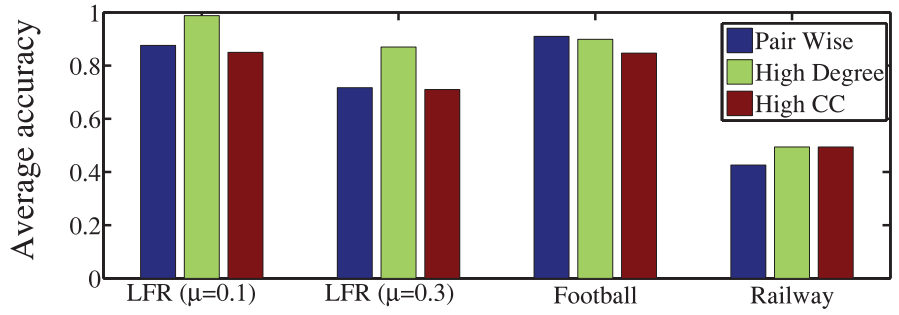


Fig. 13. Average accuracy of community detection results with different seeding techniques. Seeding based on high degree gives the highest accuracy.

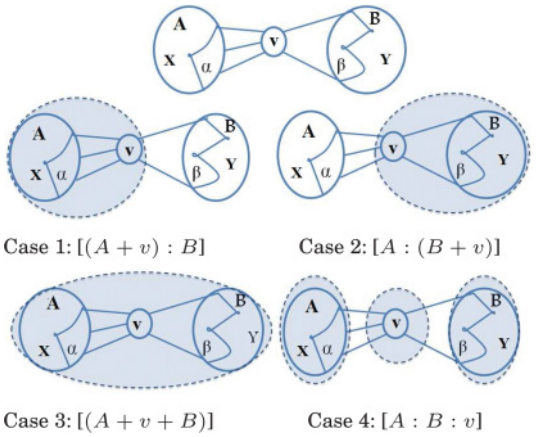


Fig. 14. (Color online) Illustrative example explaining four possible cases of community assignment of vertex v.

11. HANDLING LIMITATIONS OF MODULARITY MAXIMIZATION ALGORITHMS

In this section, we analytically show how finding communities by maximizing permanence can reduce the effects of some of the common issues in community detection including (a) resolution limit, (b) degeneracy of solution, and (c) dependence on the size of the graph.

We illustrate our proof using a simple example of two communities A and B connected by one vertex v (as shown in Figure 14). There is no edge between the communities A and B, except through the vertex v. This simple example covers many of the scenarios wherein problems due to degeneracy of solutions or resolution limits arise. For example, by considering A and B to be cliques, and v to be a vertex within the clique A, we can form a subgraph from the circle of cliques as shown in Figure 15(a). This figure is a common example to show the existence of resolution limit [Fortunato and Barthelemy 2007]. In a similar manner, by considering A and B as single vertices, we can obtain the subgraphs of a grid as shown in Figure 15(b). A grid is an example of a network where multiple solutions can occur.

11.1. Terminology and Theorems

Let vertex v be connected to α (β) nodes in community A (B), and these α (β) nodes form the set N_α (N_β). The number of vertices in community A is $(x + \alpha)$, and the number of vertices in community B is $(y + \beta)$. Let the average internal degree (connections to

Fig. 15. (a) A cycle of m identical k-cliques each having k vertices and connected by single edges; (b) a 5×5 grid network.

internal neighbors only) of a vertex $a \in N_\alpha$ and a vertex $b \in N_\beta$, before v is added, be I_α and I_β, respectively. Let the average internal clustering coefficient[6] of the neighboring nodes in communities A and B be C_A and C_B, respectively.

If v is added to communities $A(B)$, then the average internal clustering coefficient of v becomes C_A^v (C_B^v), respectively, and the average internal clustering coefficients of the nodes in $N_\alpha(N_\beta)$ become C^α (C^β). We will use these average values to approximate the permanence measure.

We assume that the communities A and B are tightly connected internally such that the values of C_A and C_B are very high (at least greater than 0.5). We note that the values C^α (C^β) will depend on the connections of v to the communities and the connections of the vertices in N_α and N_β.

To simplify this, we will consider two special cases. One case is when the nodes in the community are tightly connected and adding v does not significantly change the internal clustering coefficient. In this case, we assume $C^\alpha = C^A$ and $C^\beta = C^B$. The other case is when v is added that no new connections are formed among the neighbors of v, but the internal degree increases by 1. Therefore, $C^\alpha = C_A \frac{(I_\alpha - 1)}{(I_\alpha + 1)}$ (similarly, $C^\beta = C_B \frac{(I_\beta - 1)}{(I_\beta + 1)}$).

The combination of communities A, B, and the vertex v can have four cases (see Figure 14) as follows.

—*Case 1*: v joins with community A only. We denote this configuration as $[(A + v) : B]$, and its total permanence as $P_{(A+v):B}$. We assume that the combined permanence of all nodes $x \notin (N_\alpha \cup N_\beta \cup v)$ as P_x. This value will not be affected due to the reassignments. Therefore, the total permanence is the sum of the following factors: P_x, $[\alpha C^\alpha]$ (for the nodes in N_α connected to v), $[\frac{\alpha}{(\alpha+\beta)\beta} - (1 - C_A^v)]$ (for vertex v) and $[\beta(\frac{I_\beta}{I_\beta+1} - (1 - C_B))]$ (for the nodes in N_β).

$$P_{(A+v):B} = P_x + \alpha C^\alpha + \frac{\alpha}{(\alpha+\beta)\beta} - \left(1 - C_A^v\right) + \beta\left(\frac{I_\beta}{I_\beta + 1} - (1 - C_B)\right).$$

—*Case 2:* v joins with community B only. We denote this configuration as $[(A : (v + B)]$, and its total permanence as $P_{A:(v+B)}$. The value of this total permanence is the sum of the following factors: P_x, $[\alpha(\frac{I_\alpha}{I_\alpha+1} - (1 - C_A))]$ (for the nodes in N_α), $[\frac{\beta}{(\alpha+\beta)\alpha} - (1 - C_B^v)]$ (for vertex v) and $[\beta C^\beta]$ (for the nodes in N_β connected to v).

[6]Note that internal clustering coefficient of v is obtained by considering the ratio of the existing connections and the total number of possible connections among the *internal neighbors* of v.

$$P_{A:(v+B)} = P_x + \alpha\left(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)\right) + \frac{\beta}{(\alpha+\beta)\alpha} - \left(1-C_B^v\right) + \beta C^\beta.$$

—*Case 3: A, B,* and v merge together. We denote this configuration as $[(A+v+B)]$, and its total permanence as $P_{(A+v+B)}$. The value of this total permanence is the sum of the following factors: P_x, $[\alpha C^\alpha]$ (for the nodes in N_α), $[\frac{\alpha(\alpha-1)C_A^v+\beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)}]$ (for vertex v) and $[\beta C^\beta]$ (for the nodes in N_β connected to v).

$$P_{(A+v+B)} = P_x + \alpha C^\alpha + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^\beta.$$

—*Case 4: A, B,* and v remain as separate communities. We denote this configuration as $[(A:v:B)]$, and its total permanence as $P_{(A:v:B)}$. The value of this total permanence is the sum of the following factors: P_x, $[\alpha(\frac{I_\alpha}{I_\alpha+1} - (1-C_A))]$ (for the nodes in N_α), 0 (for vertex v), and $[\beta(\frac{I_\beta}{I_\beta+1} - (1-C_B))]$ (for the nodes in N_β).

$$P_{(A:v:B)} = P_x + \alpha\left(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)\right) + \beta\left(\frac{I_\beta}{I_\beta+1} - (1-C_B)\right).$$

We present a set of theorems as to when these conditions will occur. By using these theorems, we can analytically show that degeneracy of solutions and resolution limit is reduced when maximizing permanence.

LEMMA 1. *Given* $C^\alpha = C_A$ *and* $C^\beta = C_B$, *let* $Z1 = \frac{\alpha-\beta}{\alpha\beta} + (C_A^v - C_B^v) + (\frac{\alpha}{I_\alpha+1} - \frac{\beta}{I_\beta+1})$. *The assignment* $[(A+v):B]$ *will have a higher permanence than* $[A:(v+B)]$, *if* $Z1 > 0$ *and a lower permanence if* $Z1 < 0$.

Given $C^\alpha = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}$ *and* $C^\beta = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$, *let* $Z2 = \frac{\alpha-\beta}{\alpha\beta} + (C_A^v - C_B^v) + (\frac{\alpha(C_A+1)}{I_\alpha+1} - \frac{\beta(C_B+1)}{I_\beta+1})$. *The assignment* $[(A+v):B]$ *will have a higher permanence than* $[A:(v+B)]$, *if* $Z2 > 0$ *and a lower permanence if* $Z2 < 0$.

PROOF. Here, we are comparing between Case 1 and Case 2. The difference in total permanence between these two assignments by assuming $C^\alpha = C_A$ and $C^\beta = C_B$ is

$$\begin{aligned} P_{(A+v):B} - P_{A:(v+B)} &= \frac{\alpha}{(\alpha+\beta)\beta} + C_A^v + \beta\left(\frac{I_\beta}{I_\beta+1} - 1\right) \\ &\quad - \left(\alpha\left(\frac{I_\alpha}{I_\alpha+1} - 1\right) + \frac{\beta}{(\alpha+\beta)\alpha} + C_B^v\right) \\ &= \frac{\alpha-\beta}{\alpha\beta} + \left(C_A^v - C_B^v\right) + \left(\frac{\alpha}{I_\alpha+1} - \frac{\beta}{I_\beta+1}\right). \end{aligned}$$

The difference in total permanence between these two assignments by assuming $C^\alpha = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}$ and $C^\beta = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$ is

$$\begin{aligned} P_{(A+v):B} - P_{A:(v+B)} &= \frac{\alpha}{(\alpha+\beta)\beta} + C_A^v + \beta\left(\frac{I_\beta}{I_\beta+1} - 1\right) \\ &\quad - \left(\alpha\left(\frac{I_\alpha}{I_\alpha+1} - 1\right) + \frac{\beta}{(\alpha+\beta)\alpha} + C_B^v\right) \\ &= \frac{\alpha-\beta}{\alpha\beta} + \left(C_A^v - C_B^v\right) + \left(\frac{\alpha(C_A+1)}{I_\alpha+1} - \frac{\beta(C_B+1)}{I_\beta+1}\right). \end{aligned}$$

If this difference is greater than zero, then $[(A+v) : B]$ will have a higher permanence. If the difference is less than zero, then $[A : (v + B)]$ will have higher permanence. □

LEMMA 2. *Joining v to community A gives higher permanence than merging the communities A, B, and v, if (i) $C^\beta = C_B$, and $X > 0$ and (ii) if $C^\beta = C_B\frac{I_\beta-1}{I_\beta+1}$ and $X+\beta C_B\frac{2}{I_\beta+1} > 0$, where $X = \frac{\alpha}{(\alpha+\beta)\beta} - \frac{\beta}{I_\beta+1} - 1 + \frac{\beta(\beta-1)(C_A^v+C_B^v)}{(\alpha+\beta)(\alpha+\beta-1)} + \frac{2\alpha\beta C_A^v}{(\alpha+\beta)(\alpha+\beta-1)}$*

PROOF. We are comparing Case 1 and Case 3 and in this case $C^\beta = C_B$. The difference in total permanence is

$$\begin{aligned}
P_{(A+v):B} - P_{(A+v+B)} &= \frac{\alpha}{(\alpha+\beta)\beta} - 1 + C_A^v + \beta\left(\frac{I_\beta}{I_\beta+1} - 1 + C_B\right) \\
&\quad - \left(\frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^\beta\right). \\
&\quad \text{Substituiting } C_B \text{ with } C^\beta \\
&= \frac{\alpha}{(\alpha+\beta)\beta} - 1 + C_A^v - \frac{\beta}{I_\beta+1} \\
&\quad - \left(\frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)}\right) \\
&= \frac{\alpha}{(\alpha+\beta)\beta} - 1 - \frac{\beta}{I_\beta+1} \\
&\quad + \frac{\beta(\beta-1)(C_A^v+C_B^v)}{(\alpha+\beta)(\alpha+\beta-1)} + \frac{2\alpha\beta C_A^v}{(\alpha+\beta)(\alpha+\beta-1)}.
\end{aligned}$$

Now we consider the case where $C^\beta = C_B\frac{I_\beta-1}{I_\beta+1}$. The difference in total permanence is

$$\begin{aligned}
P_{(A+v):B} - P_{(A+v+B)} &= \frac{\alpha}{(\alpha+\beta)\beta} - 1 + C_A^v + \beta\left(\frac{I_\beta}{I_\beta+1} - 1 + C_B\right) \\
&\quad - \left(\frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^\beta\right). \\
&\quad \text{Substituiting } C^\beta = C_B\frac{I_\beta-1}{I_\beta+1} \\
&= \frac{\alpha}{(\alpha+\beta)\beta} - 1 - \frac{\beta}{I_\beta+1} + \beta C_B\frac{2}{I_\beta+1} \\
&\quad + \frac{\beta(\beta-1)(C_A^v+C_B^v)}{(\alpha+\beta)(\alpha+\beta-1)} + \frac{2\alpha\beta C_A^v}{(\alpha+\beta)(\alpha+\beta-1)}. \quad \square
\end{aligned}$$

LEMMA 3. *If $C^\alpha = C_A$ and $C^\beta = C_B$, then the communities will merge (i.e., $[(A+v+B)]$), rather than remaining separate (i.e., $[A : B : C]$).*

If $C^\alpha = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)} C^\beta = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$, then the communities will merge if:

$$\frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} > \alpha\frac{(2C_A-1)}{I_\alpha+1} + \beta\frac{(2C_B-1)}{I_\beta+1}$$

PROOF. We are comparing Case 3 and Case 4, and the case $C^\alpha = C_A$ and $C^\beta = C_B$ The difference in total permanence is

$$\begin{aligned} P_{(A+v+B)} - P_{(A:v:B)} &= \alpha C^\alpha + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^\beta \\ &\quad - \left(\alpha\left(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)\right) + \beta\left(\frac{I_\beta}{I_\beta+1} - (1-C_B)\right)\right) \\ &= \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} \\ &\quad + \frac{\alpha}{I_\alpha+1} + \frac{\beta}{I_\beta+1}. \end{aligned}$$

This value is always positive so the communities will merge.

We now consider the case where $C^\alpha = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)} C^\beta = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$ permanence is

$$\begin{aligned} P_{(A+v+B)} - P_{(A:v:B)} &= \alpha C^\alpha + \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} + \beta C^\beta \\ &\quad - \left(\alpha\left(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)\right) + \beta(\frac{I_\beta}{I_\beta+1} - (1-C_B))\right) \\ &= \frac{\alpha(\alpha-1)C_A^v + \beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} \\ &\quad - \alpha\frac{(2C_A-1)}{I_\alpha+1} - \beta\frac{(2C_B-1)}{I_\beta+1}. \quad \square \end{aligned}$$

LEMMA 4. *If $C^\alpha = C_A$ and $C^\beta = C_B$, then the communities will remain separate (i.e., $[A : v : B]$) rather than v joining with community A (i.e., $[(A+v) : B]$), if $\alpha(\frac{1}{I_\alpha+1} + \frac{1}{(\alpha+\beta)\beta}) + (C_A^v - 1) < 0$.*

Otherwise, if $C^\alpha = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}$; $C^\beta = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$, then the communities will remain separate if $\alpha(\frac{1-2C_A}{I_\alpha+1} + \frac{1}{(\alpha+\beta)\beta}) + (C_A^v - 1) < 0$.

PROOF. We are comparing Case 1 and Case 4 for the case $C^\alpha = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}; C^\beta = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$. The difference in total permanence is

$$\begin{aligned} P_{(A+v):B} - P_{(A:v:B)} &= \alpha C^\alpha + \frac{\alpha}{(\alpha+\beta)\beta} - \left(1 - C_A^v\right) \\ &\quad - \left(\alpha\left(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)\right)\right) \\ &= \alpha\left(\frac{1-2C_A}{I_\alpha+1}\right) + \frac{\alpha}{(\alpha+\beta)\beta} + (C_A^v - 1). \end{aligned}$$

If we consider the case $C^\alpha = C_A$ and $C^\beta = C_B$, then

$$\begin{aligned} P_{(A+v):B} - P_{(A:v:B)} &= \alpha C^\alpha + \frac{\alpha}{(\alpha+\beta)\beta} - \left(1 - C_A^v\right) \\ &\quad - \left(\alpha\left(\frac{I_\alpha}{I_\alpha+1} - (1-C_A)\right)\right) \\ &= \alpha\left(\frac{1}{I_\alpha+1} + \frac{1}{(\alpha+\beta)\beta}\right) + (C_A^v - 1). \quad \square \end{aligned}$$

11.2. Mitigation of Issues in Modularity Maximization

Using the above lemmas, we can determine the conditions for which a particular assignment (of the four possible ones) will give the highest permanence. Using these conditions, we shall show how permanence overcomes three major shortcomings of modularity maximization.

(i) *Degeneracy of solution* is a problem where a community scoring metric (e.g., modularity) admits multiple distinct high-scoring solutions and typically lacks a clear global maximum, thereby, resorting to tie-breaking [Good et al. 2010]. Consider the case wherein the vertex v has equal connections to groups A and B; therefore, $\alpha = \beta$ and the neighbors of v are not connected to each other, i.e., $C^{\alpha} = C_A\frac{(I_\alpha-1)}{(I_\alpha+1)}; C^{\beta} = C_B\frac{(I_\beta-1)}{(I_\beta+1)}$. Since the neighbors are not connected, therefore, $C_A^v = C_B^v = 0$.

In this case, the condition in Lemma 11.4 becomes $P_{(A+v):B} - P_{(A:v:B)} = \alpha\frac{1-2C_A}{I_\alpha+1} + \frac{1}{2\alpha} - 1$ Because the values of C_A range from 0 to 1, the value of $(1-2C_A)$ is negative. Moreover $\frac{1}{2\alpha}$ is less than 1. Therefore, the value of $P_{(A+v):B} - P_{(A:v:B)}$ is negative, indicating that permanence is higher if v, A and B form separate communities.

According to Lemma 3, the communities will merge if $\frac{\alpha(\alpha-1)C_A^v+\beta(\beta-1)C_B^v}{(\alpha+\beta)(\alpha+\beta-1)} > \alpha\frac{(2C_A-1)}{I_\alpha+1} + \beta\frac{(2C_B-1)}{I_\beta+1}$. Since $C_A^v = C_B^v = 0$, the left-hand side is 0. Therefore, permanence is higher if v, A, and B form separate communities.

Therefore, *if v has equal number of connections to each community, and the neighbors of v are not connected, then v will remain as singleton, rather than arbitrarily joining any of its neighbor groups.*

(ii) *Resolution limit* is a problem where communities of certain small size are merged into larger ones [Fortunato and Barthelemy 2007; Good et al. 2010]. One of the classic examples where modularity cannot identify communities of small size is a cycle of m cliques (see Figure 15(a)). Here, maximum modularity is obtained if two neighboring cliques are merged.

In the case of permanence, we can determine that whether two communities A and B would merge (as in modularity) or whether v would join community A (we select A as the community to explain the case, but similar analysis can also be done for the case when v joins B).

We assume that the communities A and B are tightly connected such that $C_A > 0.5$ and $C_B > 0.5$. We assume v is tightly connected to group A such that $C_A^v \approx 1$ and connected by one edge to group B ($\beta = 1$) such that $C^{\beta} = C_B\frac{I_\beta-1}{I_\beta+1}$.

From Lemma 2, we have $P_{(A+v):B} - P_{(A+v+B)} = \frac{\alpha}{(\alpha+1)} - 1 - \frac{1}{I_\beta+1} + \frac{2C_B}{I_\beta+1} + \frac{2}{\alpha+1}$ which is equal to $\frac{1}{(\alpha+1)} + \frac{2C_B-1}{I_\beta+1}$. Since $C_B > 0.5$, therefore, the value is positive. This indicates that permanence is higher if v joins group A rather than if the three groups merge.

Note that this result is independent of the size of the communities A and B. This phenomenon highlights that, in general, *if v is very tightly connected to a community and very loosely connected to another community, highest permanence is obtained when v joins the community to which it is more connected.*

(iii) *Asymptotic growth of value* of a metric implies a strong dependence on both the size of the network and the number of modules the network contains [Good et al. 2010]. Rewriting Equation (1), we get the permanence of the entire network G as follows: $Perm(G) = \frac{1}{|V|}\sum_{v\in V}[\frac{I(v)}{D(v)E_{max}(v)}] - \frac{1}{|V|}\sum_{v\in V}[(1-c_{in}(v))]$. We can notice that most of the parameters in the above formula are independent of the symmetric growth of network size and the number of communities. Table VII illustrates the property from a real-life example of coauthorship network where the modularity increases

Table VII. Change in Modularity, Permanence, and the Other Network Parameters with the (Near-)Symmetric Growth of Coauthorship Network as Discussed in Section 3.2

Coauthorship	Network properties	N	964	1,515	1,991	2,681	3,386	4,836	6,284	7,814	9,001	10,386
		C	24	24	24	24	24	24	24	24	24	24
		$\frac{I}{D}$	0.082	0.095	0.093	0.091	0.089	0.104	0.111	0.112	0.115	0.113
		$\frac{1}{E_{max}(v)}$ $(\times 10^{-4})$	3.8	3.2	2.9	3.9	2.8	2.11	2.39	2.92	2.69	3.22
		$1 - c_{in}(v)$	0.239	0.248	0.246	0.251	0.251	0.260	0.265	0.269	0.270	0.274
		CD	74.30	80.30	90.34	98.18	102.68	118.68	118.72	123.29	110.22	123.292
	Modularity		0.369	0.374	0.395	0.392	0.421	0.422	0.465	0.471	0.493	0.501
	Permanence		0.094	0.092	0.092	0.096	0.095	0.095	0.097	0.097	0.097	0.098

N: Number of nodes, C: number of communities, I: internal degree, D: degree, $c_{in}(v)$: clustering coefficient of v with respect to its internal neighbors, $E_{max}(v)$: maximum external connectivity of v, CD: average intra-community density (number of edges normalized by the number of nodes). The consistency of four network parameters indicates symmetric growth of the network in different instantiations.

with increase in the size of the network, whereas permanence remains almost constant.

12. CONCLUSION AND FUTURE WORK

In this article, we present a new vertex-centric community quality metric, called *permanence*, that unlike other metrics considers both the connection density among internal neighbors and the distribution of external connectivity of a vertex. We empirically demonstrated on synthetic and real-world networks that permanence is an effective community evaluation metric compared to other well-known approaches such as modularity, conductance, and cut-ratio. We also showed how permanence is appropriately sensitive to the fluctuations of community structure. Further experiments on characterizing permanence revealed that – (i) permanence can measure the persistence of a vertex in its own community, and (ii) one can strengthen the community structure by suitably removing nodes with low permanence value. Finally, we developed a new community detection algorithm by maximizing permanence – MaxPerm that has a much superior performance compared to state-of-the art algorithms on most datasets. Moreover, MaxPerm detects more efficient and realistic community structure – (i) the obtained communities are highly connected, irrespective of the size of the communities, as a result of which one is able to detect small and even singleton communities; (ii) the communities obtained by MaxPerm are less affected by the initial vertex ordering.

The proposed metric calls for deeper levels of investigation. More algorithms and datasets from diverse areas need to be selected to reinforce the robustness of our proposed metric. Since permanence is a local metric, one immediate direction would be to discover local community boundary for a particular seed node. We intend to extend permanence metric to enable evaluation of the quality of overlapping community structures and to weighted and directed networks. Overall, we believe that this metric will help in formulating a strong theoretical foundation in the identification and evaluation of various types of community structures where the ground-truth is not known.

APPENDIX

In this appendix, we expand Table III. In this table, the differences between the results obtained from MaxPerm and all the other algorithms are shown in terms of six validation measures for all the networks.

Table VIII. Differences of MaxPerm with the Other Algorithms in Terms of the Validation Metrics

Type	Validation metrics	Louvain	FastGreedy	CNM	WalkTrap	Infomod	Infomap	COPRA	OSLOM
L	NMI	0.14; 0.00; –0.78	0.00; 0.81; -0.02	0.07; 0.24; –0.25	0.00; 0.00; –0.13	0.04; 0.05; –0.78	0.00;0.01;0.12	0.08; 0.09; –0.78	0.00; 0.01; 0.12
	ARI	0.00; –0.02; –0.76	0.00; 0.98; 0.03	0.24; 0.59; –0.10	0.00; 0.01; –0.52	0.11; 0.13; –0.05	0.00; –0.01;–0.95	0.16; 0.01; 0.02	0.00; –0.01; –0.86
	PU	0.00; 0.00; –0.72	0.00; 0.86; 0.04	0.12; 0.41; –0.13	0.00; –0.01; –0.58	0.08; 0.09; –0.11	0.00; 0.00;–0.83	0.09; –0.01; 0.06	0.00; 0.01; -0.81
	W-NMI	0.00; 0.00; –0.78	0.00; 0.81; 0.01	0.06; 0.23; –0.10	0.00; 0.00; –0.58	0.02; 0.04; –0.08	0.00;0.00;–0.94	0.07; 0.03; 0.04	0.00; 0.00; –0.88
	W-ARI	0.00; 0.02; –0.72	0.00; 0.93; 0.07	0.23; 0.54; –0.04	0.00; –0.01; –0.52	0.05; 0.08; –0.02	0.00;–0.01; –0.88	0.16; –0.01; 0.10	0.00; –0.01; –0.82
	W-PU	0.00; 0.00; –0.79	0.00; 0.86; 0.00	0.11; 0.39; –0.20	0.00; 0.00; –0.67	0.05; 0.09; –0.18	0.00; 0.00;–0.89	0.09; 0.00; 0.01	0.00; 0.00; –0.88
	Avg.	**0.02; 0.00; –0.75**	**0.00; 0.87; 0.02**	**0.14; 0.40; –0.13**	**0.00; 0.00; –0.50**	0.06; 0.08; –0.20	**0.00; 0.00; –0.72**	**0.11; 0.02; –0.09**	**0.00; 0.00; –0.68**
R	NMI	0.01; 0.37; 0.03	0.00; 0.14; 0.13	0.22; 0.07; 0.02	0.01; 0.11; 0.01	0.01; 0.25; –0.02	0.01; 0.06; –0.06	0.02; 0.05; 0.14	0.03; 0.16; 0.09
	ARI	0.00; 0.04; –0.08	0.00; 0.36; 0.13	0.42; –0.09; 0.02	0.03; –0.07; 0.08	0.03; 0.05; –0.01	0.04; –0.06;–0.05	0.09; –0.06; 0.12	–0.06; 0.01; 0.05
	PU	–0.01; 0.08; –0.10	–0.01; 0.41; 0.13	0.26; 0.11; 0.12	0.05; 0.07; 0.05	0.02; 0.13; –0.01	–0.02; 0.06;–0.06	0.04; 0.06; 0.07	–0.04; 0.11; 0.06
	W-NMI	0.07; 0.27; 0.05	0.02; 0.56; 0.18	0.37; 0.10; 0.05	0.05; 0.14; 0.07	0.05; 0.43; –0.06	0.03; 0.17;–0.01	0.04; 0.14; 0.02	0.03; 0.28; 0.12
	W-ARI	0.02; 0.05; 0.05	0.00; 0.31; 0.11	0.34; –0.15; 0.02	0.02; –0.11; 0.05	0.02; 0.17; –0.06	0.00;–0.07; –0.01	0.01; –0.10; 0.09	0.00; 0.05; 0.11
	W-PU	0.01; 0.05; 0.02	0.06; 0.44; 0.14	0.21; –0.06; 0.06	–0.05; –0.04; –0.15	–0.05; 0.12; –0.12	–0.06;–0.05; 0.05	0.00; –0.05; 0.08	–0.06; 0.00; 0.10
	Avg.	**0.02; 0.14; 0.00**	**0.01; 0.37; 0.14**	**0.30; 0.00; 0.05**	**0.02; 0.02; 0.02**	**0.01; 0.19; –0.04**	**0.00; 0.02; –0.02**	**0.03; 0.01; 0.09**	**–0.01; 0.11; 0.09**

Positive differences indicate the improvement of our algorithm. The rows indicated by "L" show values obtained from the LFR graphs with μ =0.1, 0.3, and 0.6, respectively (from left to right and separated by semicolons). The rows indicated by "R" show values for football, railway, and coauthorship networks (from left to right and separated by semicolons). The average improvements over different validation measures are shown in Rows 8 and 15 for LFR and real-world networks, respectively

The bold font in Table VIII indicates the average accuracy of each algorithm over different validation measures.

REFERENCES

Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466, (August 2010), 761–764.

A. Arenas, A. Fernández, and S. Gómez. 2008. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics* 10, 5 (2008), 053039.

David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner (Eds.). 2013. Graph partitioning and graph clustering. In *Proceedings of the10th DIMACS Implementation Challenge Workshop.* Contemporary Mathematics, vol. 588. American Mathematical Society.

Jeffrey Baumes, Mark Goldberg, and Malik Magdon-Ismail. 2005. Efficient identification of overlapping communities. In *Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics (ISI'05)*. Springer-Verlag, Berlin, 27–36.

Jonathan W. Berry, Bruce Hendrickson, Randall A. LaViolette, and Cynthia A. Phillips. 2011. Tolerating the community detection resolution limit with edge weighting. *Physical Review E* 83, 5 (May 2011), 056119.

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

Tanmoy Chakrabort, Sandipan Sikdar, Vihar Tammana, Niloy Ganguly, and Animesh Mukherjee. 2013. Computer science fields as ground-truth communities: Their impact, rise and fall. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'13)*. ACM, New York, NY, 426–433.

Tanmoy Chakraborty. 2015. Leveraging disjoint communities for detecting overlapping community structure. *Journal of Statistical Mechanics: Theory and Experiment* 2015, 5 (2015), P05017.

Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. 2016a. Metrics for community analysis: A survey. *CoRR* abs/1604.03512 (2016).

Tanmoy Chakraborty, Suhansanu Kumar, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. 2016b. GenPerm: A unified method for detecting non-overlapping and overlapping communities. *CoRR* abs/1604.03454 (2016).

Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Sanjukta Bhowmick, and Animesh Mukherjee. 2013. Constant communities in complex networks. *Scientific Reports* 3, (May 2013). DOI:10.1038/srep01825

Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. 2014. On the permanence of vertices in network communities. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, NY, 1396–1405. DOI:http://dx.doi.org/10.1145/2623330.2623707

Mingming Chen, Tommy Nguyen, and Boleslaw Szymanski. 2013. A new metric for quality of network community structure. *ASE Human Journal* 1, 4 (2013), 226–240.

Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010. Rumour spreading and graph conductance. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1657–1663.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical Review E* 70, 6 (2004), 066111.

L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. 2005. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment* 9, (2005), P09008.

Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2013. Enhancing community detection using a network weighting strategy. *Journal of Information Science* 222, (Feb. 2013), 648–668.

J.-C. Delvenne, S. N. Yaliraki, and M. Barahona. 2010. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences* 107, 29 (2010), 12755–12760.

Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1987. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. New York, 1–12.

T. S. Evans and R. Lambiotte. 2009. Line graphs, link partitions, and overlapping communities. *Physical Review E* 80, 1 (July 2009), 016105.

Illés Farkas, Dániel Ábel, Gergely Palla, and Tamás Vicsek. 2007. Weighted network modules. *New Journal of Physics* 9, 6 (2007), 180.

Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3–5 (2010), 75–174.

Santo Fortunato and M. Barthelemy. 2007. Resolution limit in community detection. *Proceedings of the National Acadamy of Sciences of the United States of America (PNAS)* 104, 1 (2007), 36–41.

Saptarshi Ghosh, Avishek Banerjee, Naveen Sharma, Sanket Agarwal, and Niloy Ganguly. 2011. Statistical analysis of the indian railway network: A complex network approach. *Acta Physica Polonica B Proceedings Supplement* 4, (March 2011), 123–137.

M. Girvan and M. E. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (June 2002), 7821–7826.

B. H. Good, Y. A. De Montjoye, and A. Clauset. 2010. Performance of modularity maximization in practical contexts. *Physical Review E* 81, 4 (2010), 046106.

Roger Guimera and Luis A. Nunes Amaral. 2005. Functional cartography of complex metabolic networks. *Nature* 433, 7028 (Feb. 2005), 895–900.

Dongxiao He, Dayou Liu, Weixiong Zhang, Di Jin, and Bo Yang. 2013. Discovering link communities in complex networks by exploiting link dynamics. *CoRR* abs/1303.4699 (2013).

Paul W. Holland and Samuel Leinhardt. 1971. Transitivity in structural models of small groups. *Small Group Research* 2, 2 (1971), 107–124.

L. Hubert and P. Arabie. 1985. Comparing partitions. *Journal of Classification* 2, 1 (1985), 193–218.

R. Kannan, S. Vempala, and A. Veta. 2000. On clusterings-good, bad and spectral. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS'00)*. IEEE Computer Society, Washington, DC, 367.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, New York, NY, 137–146. DOI: http://dx.doi.org/10.1145/956750.956769

Renaud Lambiotte. 2010. Multi-scale modularity in complex networks. In *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'10)*. IEEE, 546–553.

Andrea Lancichinetti and Santo Fortunato. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E* 80, 1 (July 2009), 016118.

Andrea Lancichinetti and Santo Fortunato. 2011. Limits of modularity maximization in community detection. *Physical Review E* 84, (2011), 066122.

Andrea Lancichinetti and Santo Fortunato. 2012. Consensus clustering in complex networks. *Scientific Reports* 2 (2012). DOI: 10.1038/srep00336

Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11, 3 (2009), 033015.

Andrea Lancichinetti, Filippo Radicchi, Jose J. Ramasco, and Santo Fortunato. 2010. Finding statistically significant communities in networks. *CoRR* abs/1012.2363 (2010).

E. A. Leicht and M. E. J. Newman. 2008. Community structure in directed networks. *Physical Review Letters* 100, 11 (March 2008), 118703.

Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 2009. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* 6, 1 (2009), 29–123.

Jure Leskovec, Kevin J. Lang, and Michael Mahoney. 2010. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM, New York, NY, 631–640.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY.

M. E. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (June 2006), 8577–8582.

M. E. J. Newman. 2003. Mixing patterns in networks. *Physical Review E* 67, 2 (Feb. 2003), 026126. DOI: http://dx.doi.org/10.1103/PhysRevE.67.026126

M. E. J. Newman. 2004a. Analysis of weighted networks. *Physical Review E* 70, 5 (Nov. 2004), 056131.

M. E. J. Newman. 2004b. Fast algorithm for detecting community structure in networks. *Physical Review E* 69, 6 (June 2004), 066133.

M. E. J. Newman. 2013. Community detection and graph partitioning. *CoRR* abs/1305.4974 (2013).

M. E. J. Newman and M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, (2004) 026113.

Günce Keziban Orman, Vincent Labatut, and Hocine Cherifi. 2012. Comparative evaluation of community detection algorithms: A topological approach. *CoRR* abs/1206.4987 (2012).

Gergely Palla, Imre Dernyi, Ills Farkas, and Tams Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435, 7043 (June 2005), 814–818.

Pascal Pons and Matthieu Latapy. 2006. Computing communities in large networks using random walks. *Journal of Graph Algortihms and Applications* 10, 2 (2006), 191–218.

Ioannis Psorakis, Stephen Roberts, Mark Ebden, and Ben Sheldon. 2011. Overlapping community detection using Bayesian non-negative matrix factorization. *Physical Review E* 83, 6 (June 2011), 066114.

Usha N. Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 3 (Sep. 2007), 036106.

J. Reichardt and S. Bornholdt. 2006. Statistical mechanics of community detection. *Phys. Rev. E* 74, 1 (2006), 016110.

Thomas Richardson, Peter J. Mucha, and Mason A. Porter. 2009. Spectral tripartitioning of networks. *Physical Review E* 40, (2009), 027104.

Jason Riedy, David A. Bader, Karl Jiang, Pushkar Pande, and Richa Sharma. 2011. *Detecting Communities from Given Seeds in Social Networks*. Technical Report GT-CSE-11-01. Georgia Institute of Technology.

M. Rosvall and C. T. Bergstrom. 2007. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences* 104, 18 (2007), 7327.

Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123.

Massoud Seifi, Ivan Junier, Jean-Baptiste Rouquier, Svilen Iskrov, and Jean-Loup Guillaume. 2013. Stable community cores in complex networks. In *Complex Networks*, Ronaldo Menezes, Alexandre Evsukoff, and Marta C. Gonzlez (Eds.), Studies in Computational Intelligence, vol. 424. Springer, Berlin, 87–98. DOI:http://dx.doi.org/10.1007/978-3-642-30287-9_10

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (Aug. 2000), 888–905. DOI:http://dx.doi.org/10.1109/34.868688

Peng-Gang Sun, Lin Gao, and Shan Shan Han. 2011. Identification of overlapping and non-overlapping community structure by fuzzy clustering in complex networks. *Information Sciences* 181, 6 (2011), 1060–1071.

Jierui Xie and Boleslaw K. Szymanski. 2011. Community detection using a neighborhood strength driven label propagation algorithm. *CoRR* abs/1105.3264 (2011).

Jierui Xie and Boleslaw K. Szymanski. 2012. Towards linear time overlapping community detection in social networks. *CoRR* abs/1202.2465 (2012).

Jaewon Yang and Jure Leskovec. 2012. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (MDS'12)*. ACM, New York, NY, 3:1–3:8.

Jaewon Yang and Jure Leskovec. 2014. Overlapping communities explain core-periphery organization of networks. *Proceedings of IEEE* 102, (2014), 1892–1902.

Received July 2015; revised April 2016; accepted June 2016