

Krylov Subspace Approximation for Local Community Detection in Large Networks

KUN HE, Huazhong University of Science and Technology, China

PAN SHI, Huazhong University of Science and Technology, China

DAVID BINDEL, Cornell University, USA

JOHN E. HOPCROFT, Cornell University, USA

Community detection is an important information mining task to uncover modular structures in large networks. For increasingly common large network data sets, global community detection is prohibitively expensive, and attention has shifted to methods that mine local communities, *i.e.* identifying all latent members of a particular community from a few labeled seed members. To address such semi-supervised mining task, we systematically develop a local spectral subspace-based community detection method, called LOSP. We define a family of local spectral subspaces based on Krylov subspaces, and seek a sparse indicator for the target community via an ℓ_1 norm minimization over the Krylov subspace. Variants of LOSP depend on type of random walks with different diffusion speeds, type of random walks, dimension of the local spectral subspace and step of diffusions. The effectiveness of the proposed LOSP approach is theoretically analyzed based on Rayleigh quotients, and it is experimentally verified on a wide variety of real-world networks across social, production and biological domains, as well as on an extensive set of synthetic LFR benchmark datasets.

CCS Concepts: • **Computing methodologies** → **Spectral methods**; • **Information systems** → **Clustering and classification**; **Social networks**; *Web searching and information discovery*;

Additional Key Words and Phrases: Local community detection, spectral clustering, Krylov subspace, Rayleigh quotient, sparse linear coding

ACM Reference Format:

Kun He, Pan Shi, David Bindel, and John E. Hopcroft. 2019. Krylov Subspace Approximation for Local Community Detection in Large Networks. 1, 1 (June 2019), 30 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Community detection has arisen as one of the significant topics in network analysis and graph mining. Many problems in information science, social science, biology and physics can be formulated as problems of community detection. With the rapid growth of the network scale, however, exploring the *global* community structure [3, 39] becomes prohibitively expensive in such networks with millions or billions of nodes. While most of the time people are just interested in the local structure of the graph neighborhood. Hence, attention has shifted to methods that mine local community structure without processing the whole large network [23, 25, 28, 42, 50, 52].

Authors' addresses: Kun He, Huazhong University of Science and Technology, Wuhan, China, 430074, brooklet60@hust.edu.cn, kh555@cornell.edu; Pan Shi, Huazhong University of Science and Technology, Wuhan, China, 430074, panshi@hust.edu.cn; David Bindel, Cornell University, Ithaca, NY, USA, 14853, bindel@cs.cornell.edu; John E. Hopcroft, Cornell University, Ithaca, NY, USA, 14853, jeh@cs.cornell.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

XXXX-XXXX/2019/6-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

In many situations, instead of finding all communities in a large network in an unsupervised manner, we just want to quickly find communities from a small set of members labeled by a domain expert. For example, in political participation networks, one might discover the membership of a political group from a few representative politicians [47]; sales websites might use clusters in co-purchase networks to generate product recommendations for a customer based on previous purchases; and starting from a few well-studied genes, biologists may seek functionally similar genes via genetic interaction networks.

Communities in real-world networks are often small, comprising dozens or hundreds of members [20, 32]. Intuitively, the latent members in these small communities should be very close to any seed members chosen from the target communities. The *seed set expansion* approach to community detection [31, 39, 48] starts from “seed” nodes, or labeled members of a target community [27, 28, 52], and incrementally grows the set by locally optimizing a community scoring function.

A common theme in seed set expansion methods is to diffuse probability from the seeds. PageRank [5, 28], heat kernel [16, 27] and local spectral approximation [24, 33, 34] are three main techniques for the probability diffusion. Among these, the local spectral method is a newly proposed technique that exhibits high performance for the local community detection task. Motivated by standard spectral clustering methods that find disjoint global communities from the leading eigenvectors of the graph Laplacian, local spectral algorithms can handle local communities by seeking a sparse indicator vector that contains the seeds and lies in a local spectral subspace rather than in the global eigenspace. Moreover, starting from different seed sets, local spectral algorithms can uncover overlapping communities.

In this paper, we propose a local spectral (LOSP) algorithm to identify the members of small communities [24], and systematically build the LOSP family based on various Krylov subspace approximation. Theoretical analysis and empirical analysis are provided to show the soundness and effectiveness of the proposed approach. LOSP differs from standard spectral clustering in two ways:

- **Handling overlapping communities.** Standard spectral methods partition a graph into disjoint communities by k -means clustering in a coordinate system defined by eigenvectors or by recursive spectral bisection. We instead use ℓ_1 -norm optimization to search an (approximate) invariant subspace for an indicator vector for a sparse set containing the seeds. Overlapping communities correspond to different seed sets.
- **Defining a local spectral subspace.** To determine the local structure around the seeds of interest, we calculate a local approximate invariant subspace via a Krylov subspace associated with short random walks.

Starting from a few random seeds, we sample locally in the graph to get a comparatively small subgraph containing most of the latent members, so that the follow-up membership identification can focus on a local subgraph instead of the whole network. The LOSP methods are then used to extract the local community from the sampled subgraph via a Krylov subspace formed by short-random-walk diffusion vectors.

Our main contributions include:

- **Sampling to reduce the complexity.** We sample locally in the graph as a pre-processing to get a much smaller subgraph containing most of the latent members around the seeds. The follow-up calculation has a low complexity while maintaining an accurate covering on the target community. In this way, the proposed method is applicable to large-scale networks.
- **Building a rich LOSP family.** We systematically develop a family of local spectral methods based on random walk diffusion. We thoroughly investigate a rich set of diffusion methods; and we find that light lazy random walk, lazy random walk and personalized PageRank are

robust for different parameters, and outperform the standard random walk diffusions. In general, light lazy random walk performs slightly better. We also study the regular random walk diffusions spreading out from the seeds and the inverse random walk diffusions ending around the seeds.

- **Extensive demonstration of LOSP.** We provide a diverse set of computational experiments on 28 synthetic benchmark graphs and eight real-world social and biological networks to show that the proposed methods are much more accurate than previous local spectral approach LEMON [34], projected gradient descent algorithm PGDc-d [45], the well-known personalized PageRank diffusion algorithm pprpush [5] and heat kernel algorithm hk-relax [27].

This work is a significant extension and improvement based on our initial LOSP method [24]. By removing some costly seed-strengthen preprocessing and iterative post-processing of the initial LOSP [24], the current LOSP costs only one third of the running time of the initial LOSP after the sampling. We also improve the key component of LOSP based on the Krylov subspace, and extend it to an integrated LOSP family.

2 RELATED WORK

Communities are densely intra-linked components with sparser inter-connections, typically defined by means of metrics like modularity [37] or conductance [41]. There are many different research directions as well as approaches for the community detection [7, 18, 22, 40, 51]. Early works focused on global structure mining [3, 17, 39] while an increasing body of recent work focuses on local structure mining [28, 35, 45]. We will focus on seed set expansion methods that uncover a local community from a few seed members; these were initially designed for global community detection, but have since been extensively used in local community mining. We could regard global community detection as an unsupervised method for finding global community structure, and regard local community detection as a semi-supervised approach for mining local community structure supervised by the seed set in the target community we want to find.

2.1 Global Community Detection

Global community detection is an important topic in data mining. Global community detection methods include modularity optimization [9, 11, 36], stochastic block models [1], nonnegative matrix factorization [53, 55], spectral methods [38, 46], global seed set expansion [39, 43, 49], and many other techniques [12, 21]. In this subsection, we highlight a few lines of works in the literature that are related to our method.

Spectral-based method. Spectral method is a main technique for global community detection. Von Luxburg [46] introduces the most common spectral clustering algorithms based on different graph Laplacians, and discusses the advantages and disadvantages of different spectral clustering algorithms in details. Using spectral algorithms, Newman studies the three related problems of detection by modularity maximization, statistical inference or normalized-cut graph partitioning, and finds that with certain choices of the free parameters in the spectral algorithms the three problems are identical [38]. As the classic spectral clustering method could only find disjoint communities, Zhang *et al.* [54] propose a new algorithm to identify overlapping communities in small networks by combining a generalized modularity function, an approximated spectral mapping of network nodes into Euclidean space and fuzzy c -means clustering. Ali and Couillet [4] propose a spectral algorithm based on the family of “ α -normalized” adjacency matrices \mathbf{A} , which is in the type of $\mathbf{D}^{-\alpha}\mathbf{A}\mathbf{D}^{-\alpha}$ with \mathbf{D} the degree matrix, to find community structure in large heterogeneous networks.

Global seed set expansion. Many global community detection algorithms are based on seed set expansion. Clique Percolation [39], the most classic method, starts from maximal k -cliques and merges cliques sharing $k - 1$ nodes to form a percolation chain. OSLOM [31], starts with each node as the initial seed and optimizes a fitness function, defined as the probability of finding the cluster in a random null model, to join together small clusters into statistically significant larger clusters. Seed Set Expansion (SSE) [48, 49] identifies overlapping communities by expanding different types of seeds by a personalized PageRank diffusion. DEMON [17] and another independent work in [43] identify very small, tightly connected sub-communities, create a new network in which each node represents such a sub-community, and then identify communities in this meta-network. Belfin *et al.* [8] propose a strategy for locating suitable superior seed set by applying various centrality measures in order to find overlapping communities.

2.2 Local Community Detection

Local seed set expansion. Random walks have been extensively adopted as a subroutine for locally expanding the seed set [6], and this approach is observed to produce communities correlated highly to the ground-truth communities in real-world networks [2]. PageRank, heat kernel and local spectral diffusions are three main techniques for probability diffusion.

Spielman and Teng use the degree-normalized personalized [44] PageRank (DN PageRank) with truncation of small values to expand a starting seed. DN PageRank has been used in several subsequent PageRank-based clustering algorithms [6, 52], including the popular PageRank Nibble method [5]. However, a study evaluating different variations of PageRank finds that standard PageRank yields better performance than DN PageRank [28].

The heat kernel method provides another local graph diffusion. Based on a continuous-time Markov chain, the heat kernel diffusion involves the exponential of a generator matrix, which may be approximated via a series of expansion. Chung *et al.* have proposed a local graph partitioning based on the heat kernel diffusion [14, 15], and a Monte Carlo algorithm to estimate the heat kernel process [16]. Another approach is described in [27], where the authors estimate the heat kernel diffusion via coordinate relaxation on an implicit linear system; their approach uncovers smaller communities with substantially higher F_1 measures than those found through the personalized PageRank diffusion.

Spectral methods are often used to extract disjoint communities from a few leading eigenvectors of a graph Laplacian [26, 46]. Recently, there has been a growing interest in adapting the spectral approach to mine the local structure around the seed set. Mahoney, Orecchia, and Vishnoi [35] introduce a locally-biased analogue of the second eigenvector for extracting local properties of data graphs near an input seed set by finding a sparse cut, and apply the method to semi-supervised image segmentation and local community extraction. In [24, 33, 34], the authors introduce an algorithm to extract the local community by seeking a sparse vector from the local spectral subspaces using ℓ_1 norm optimization. They apply a power method for the subspace iteration using a standard random walk on a modified graph with a self loop on each node, which we call the light lazy random walk. They also apply a reseeding iteration to improve the detection accuracy.

Bounding the local community. All seed set expansion methods need a stopping criterion, unless the size of the target community is known in advance. Conductance is commonly recognized as the best stopping criterion [27, 48, 49, 52]. Yang and Leskovec [52] provide widely-used real-world datasets with labeled ground truth, and find that conductance and triad-partition-ratio (TPR) are the two stopping rules yielding the highest detection accuracy. He *et al.* [24] propose two new metrics, TPN and nMod, and compare them with conductance, modularity and TPR; and they show that conductance and TPN consistently outperform other metrics. Laarhoven and Marchiori [45]

study a continuous relaxation of conductance by investigating the relation of conductance with weighted kernel k -means.

Local seeding strategies. The seeding strategy is a key part of seed set expansion algorithms. Kloumann and Kleinberg [28] argue that random seeds are superior to high degree seeds, and suggest domain experts provide seeds with a diverse degree distribution. Our initial LOSP paper [24] compares low degree, random, high triangle participation (number of triangles inside the community containing the seed) and low escape seeds (judged by probability retained on the seeds after short random walks), and find all four types of seeds yield almost the same accuracy. Our initial LOSP work shows that low degree seeds spread out the probabilities slowly and better preserve the local information, and random seeds are similar to low degree seeds due to the power law degree distribution. High triangle participation seeds and low escape seeds follow another philosophy: they choose seeds more cohesive to the target community.

3 PRELIMINARIES

3.1 Problem Formulation

The local community detection problem can be formalized as follows. We are given a connected, undirected graph $G = (V, E)$ with n nodes and m edges. Let $\mathbf{A} \in \{0, 1\}^{n \times n}$ be the associated adjacency matrix, and \mathbf{D} the diagonal matrix of node degrees. Let S be the seed set of a few exemplary members in the target ground-truth community, denoted by a set of nodes T ($S \subset T$, $|T| \ll |V|$). And let $\mathbf{s} \in \{0, 1\}^{n \times 1}$ be a binary indicator vector representing the exemplary members in S . We are asked to identify the remaining latent members in the target community T .

3.2 Datasets

We consider four groups with a total of 28 synthetic datasets, five SNAP datasets in social, product, and collaboration domains, and three biology networks for a comprehensive evaluation on the proposed LOSP algorithms.

3.2.1 LFR Benchmark. For synthetic datasets, we use the LFR standard benchmark networks proposed by Lancichinetti *et al.* [29, 30]. The LFR benchmark graphs have a built-in community structure that simulates properties of real-world networks accounting for heterogeneity of node degrees and community sizes that follow power law distribution.

We adopt the same set of parameter settings used in [51] and generate four groups with a total of 28 LFR benchmark graphs. Table 1 summarizes the parameter settings we used, among which the mixing parameter μ has a big impact on the network topology. Parameter μ controls the average fraction of neighboring nodes that do not belong to any community for each node, two ranges of typical community size, big and small, are provided by b and s . Each node belongs to either one community or om overlapping communities, and the number of nodes in overlapping communities is specified by on . A larger om or on indicates more overlaps that are harder for the community detection task.

For four groups of configurations based on the community size and on , we vary om from 2 to 8 to get seven networks in each group, denoted as: **LFR_s_0.1** for $\{s : [10, 50], on = 500\}$, **LFR_s_0.5** for $\{s : [10, 50], on = 2500\}$, **LFR_b_0.1** for $\{b : [20, 100], on = 500\}$, and **LFR_b_0.5** for $\{b : [20, 100], on = 2500\}$. The average conductance for four groups of datasets are 0.522, 0.746, 0.497 and 0.733, respectively. We see more overlapping on the communities (a bigger on) leads to a higher conductance.

Table 1. Parameters for the LFR benchmarks.

Parameter	Description
$n = 5000$	number of nodes in the graph
$\mu = 0.3$	mixing parameter
$\bar{d} = 10$	average degree of the nodes
$d_{max} = 50$	maximum degree of the nodes
$s : [10, 50], b : [20, 100]$	range of the community size
$\tau_1 = 2$	node degree distribution exponent
$\tau_2 = 1$	community size distribution exponent
$om \in \{2, 3, \dots, 8\}$	overlapping membership
$on \in \{500, 2500\}$	number of overlapping nodes

3.2.2 Real-world Networks. We consider five real-world network datasets with labeled ground truth from the Stanford Network Analysis Project (SNAP)¹ and three genetic networks with labeled ground truth from the Isobase website².

- **SNAP:** The five SNAP networks, **Amazon**, **DBLP**, **LiveJ**, **YouTube**, **Orkut**, are in the domains of social, product, and collaboration [52]. For each network, we adopt the top 5000 annotated communities with the highest quality evaluated with 6 metrics [52]: Conductance, Flake-ODF, FOMD, TPR, Modularity and CutRatio. Our algorithm adopts the popular metric of conductance to automatically determine the community boundary. To make a fair comparison, we choose four state-of-the-art baselines that also adopt conductance to determine the community boundary.
- **Biology:** The three genetic networks from the Isobase website describe protein interactions. **HS** describes these interactions in humans, **SC** in *S. cerevisiae*, a type of yeast, and **DM** in *D. melanogaster*, a type of fruit fly. Such networks are interesting as communities may correspond to different genetic functions.

Table 2 summarizes the networks and their ground truth communities. We calculate the average and standard deviation of the community sizes, and the average conductance, where low conductance gives priority to communities with dense internal links and sparse external links. We also define and calculate the roundness of communities.

Definition 3.1. Roundness of a subgraph. The roundness of a subgraph $G' = (V', E')$ is the average shortest path among all pair-wise nodes divided by the longest shortest path in the subgraph.

The roundness value R is 1 for a clique, and $R = \frac{|V'|+1}{3(|V'|-1)} \approx \frac{1}{3}$ if the subgraph is a straight line. Because large roundness value indicates a “round” subgraph and small roundness value indicates a “long and narrow” subgraph, the roundness reveals some information on the topology structure of the subgraph. Table 2 shows that communities in the above real-world networks have an average roundness of about 0.67. If we normalize the roundness value from $[1/3, 1]$ to $[0, 1]$, then we get $R_{norm} = \frac{R-1/3}{1-1/3}$.

3.3 Evaluation Metric

For the evaluation metric, we adopt F_1 score to quantify the similarity between the detected local community C and the target ground truth community T . The F_1 score for each pair of (C, T) is

¹<http://snap.stanford.edu>

²<http://groups.csail.mit.edu/cb/mna/isobase/>

Table 2. Statistics for real-world networks and their ground truth communities.

Domain	Name	Network		Ground truth communities			
		# Nodes	# Edges	Avg. \pm Std. Size	Avg. Cond.	Roundness R	R_{norm}
Product	Amazon	334,863	925,872	13 \pm 18	0.07	0.69	0.54
	DBLP	317,080	1,049,866	22 \pm 201	0.41	0.74	0.61
Social	LiveJ	3,997,962	34,681,189	28 \pm 58	0.39	0.65	0.48
Social	YouTube	1,134,890	2,987,624	21 \pm 73	0.84	0.69	0.54
Social	Orkut	3,072,441	117,185,083	216 \pm 321	0.73	0.50	0.25
Biology	DM	15,294	485,408	440 \pm 2096	0.88	0.68	0.52
Biology	HS	10,153	54,570	113 \pm 412	0.88	0.58	0.37
Biology	SC	5,523	82,656	67 \pm 110	0.90	0.63	0.45

defined by:

$$F_1(C, T) = \frac{2|C \cap T|}{|C| + |T|}.$$

4 KRYLOV SUBSPACE CLUSTERING

It is well known that there is a close relationship between the combinatorial characteristics of a graph and the algebraic properties of its associated matrices [13]. In this section, we present a local community detection method of finding a linear sparse coding on the Krylov subspace, which is a local approximation of the spectral subspace. We first present the local sampling method starting from the seeds to reach a comparatively small subgraph G_s , and provide the theoretical base that finding a local community containing the seeds corresponds to finding a sparse linear coding in the invariant subspace spanned by the dominant eigenvectors of the transition matrix. Then we propose a family of local spectral subspace definitions based on various short random walk diffusions, and do local community detection by finding a sparse relaxed indicator vector that lies in a local spectral subspace representing the subordinate probability of the corresponding nodes.

4.1 Local Sampling

We first do a local sampling from the seeds for large networks to reduce the computational complexity. According to the small world phenomenon and “six degrees of separation”, most members should be at most two or three steps far away from the seed members if we want to identify a small community of size hundreds. Thus, we mainly use a few steps of Breadth-First Search (BFS) to expand nodes for the sampling method. To avoid BFS to expand too fast, we adopt a strategy to filter some very popular nodes during the BFS expansion. The detailed description for the sampling method is as follows.

Starting from each seed, we do a one-round BFS. The $BFS(\cdot)$ operation achieves a set of nodes containing the source node and its neighbor nodes by breadth-first search. For the current subgraph expanded by the BFS, we define the inward ratio of a node as the fraction of inward edges to the out-degree. If the sampled subgraph is no greater than the lower bound N_1 , then we do one more round of BFS to contain more neighbor nodes. To avoid the next round of BFS to expand too fast, we first filter low inward ratio frontier nodes which may contain some inactive nodes or very popular nodes. The $Filter(\cdot)$ operation chooses high inward ratio nodes until the total out-degree is no less than 3000. In the end, we union all BFS subgraphs obtained from each seed. If the amalgamated subgraph scale is larger than the upper bound N_2 , we then conduct a k -step short random walk from the seeds to remove some low probability nodes. Details of the sampling are as shown in Algorithm 1.

ALGORITHM 1: Sampling

Input: Graph $G = (V, E)$, seed set $S \subseteq V$, lower bound of sampled size from each seed N_1 , upper bound of the subgraph size N_2 , upper bound of BFS rounds t , and steps of random walks k for postprocessing

Output: Sampled subgraph $G_s = (V_s, E_s)$

$V_s \leftarrow S$

for each $s_i \in S$ **do**

$V_i \leftarrow \text{BFS}(s_i)$

$V'_i \leftarrow V_i$

while ($|V_i| < N_1$ and BFS rounds $\leq t$) **do**

$V'_i \leftarrow \text{Filter}(V'_i)$

$V'_i \leftarrow \text{BFS}(V'_i)$

$V_i = V_i \cup V'_i$

end

$V_s = V_s \cup V_i$

end

$G_s = (V_s, E_s)$ is the induced subgraph from V_s

if $|V_s| > N_2$ **then**

 Conduct a k -steps of random walk from S in G_s

$V_s \leftarrow N_2$ nodes with higher probability

$G_s = (V_s, E_s)$ is the induced subgraph from V_s

end

Denote the sampled subgraph as $G_s = (V_s, E_s)$ with n_s nodes and m_s edges in the following discussion. We then identify the local community from this comparatively small subgraph instead of the original large network. The complexity is only related to the degrees of the nodes and it is very quick in seconds for the datasets we considered. The sampling quality, evaluated by the coverage ratio of the labeled nodes, plays a key role for the follow-up membership identification. This pre-processing procedure significantly reduces the membership identification cost.

4.2 Spectra and Local Community

In this subsection, we provide the necessary theoretical base that finding a low-conductance community corresponds to finding a sparse indicator vector in the span of dominant eigenvectors of the transition matrix with larger eigenvalues.

Let $L = D_s - A_s$ be the Laplacian matrix of G_s where A_s and D_s denote the adjacency matrix and the diagonal degree matrix of G_s . We define two normalized graph Laplacian matrices:

$$L_{\text{rw}} = D_s^{-1}L = I - N_{\text{rw}}, \quad L_{\text{sym}} = D_s^{-\frac{1}{2}}LD_s^{-\frac{1}{2}} = I - N_{\text{sym}},$$

where I is the identity matrix, $N_{\text{rw}} = D_s^{-1}A_s$ is the transition matrix, and $N_{\text{sym}} = D_s^{-\frac{1}{2}}A_sD_s^{-\frac{1}{2}}$ is the normalized adjacency matrix.

For a community C , the conductance [41] of C is defined as

$$\Phi(C) = \frac{\text{cut}(C, \bar{C})}{\min\{\text{vol}(C), \text{vol}(\bar{C})\}},$$

where \bar{C} consists of all nodes outside C , $\text{cut}(C, \bar{C})$ denotes the number of edges between C and \bar{C} , and $\text{vol}(\cdot)$ calculates the “edge volume”, i.e. for the subset nodes, we count their total node degrees in graph G_s . Low conductance gives priority to a community with dense internal links and sparse external links.

Let $\mathbf{y} \in \{0, 1\}^{n_s \times 1}$ be a binary indicator vector representing a small community C in the sampled graph $G_s = (V_s, E_s)$. Here for “small community”, we mean $\text{vol}(C) \leq \frac{1}{2} \text{vol}(V_s)$. As $\mathbf{y}^T \mathbf{D}_s \mathbf{y}$ equals the total node degrees of C , and $\mathbf{y}^T \mathbf{A}_s \mathbf{y}$ equals two times the number of internal edges of C , the conductance $\Phi(C)$ could be written as a generalized Rayleigh quotient

$$\Phi(C) = \frac{\mathbf{y}^T \mathbf{L}_s \mathbf{y}}{\mathbf{y}^T \mathbf{D}_s \mathbf{y}} = \frac{(\mathbf{D}_s^{\frac{1}{2}} \mathbf{y})^T \mathbf{L}_{\text{sym}} (\mathbf{D}_s^{\frac{1}{2}} \mathbf{y})}{(\mathbf{D}_s^{\frac{1}{2}} \mathbf{y})^T (\mathbf{D}_s^{\frac{1}{2}} \mathbf{y})}.$$

THEOREM 4.1. *Let λ_2 be the second smallest eigenvalue of \mathbf{L}_{sym} , then conductance $\Phi(C)$ of a small community C in graph $G_s = (V_s, E_s)$ (“small” means $\text{vol}(C) \leq 0.5 \text{vol}(V_s)$) is bounded by*

$$\frac{\lambda_2}{2} \leq \Phi(C) \leq 1,$$

where $\text{vol}(C)$ denotes for all nodes inside $C \subseteq V_s$, we count the total degree in graph G_s .

The proof omits here, and we attach the details in Appendix A.

Let $\mathbf{L}_{\text{sym}} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$ be the eigendecomposition, where $\mathbf{Q} = [\mathbf{q}_1 \mid \cdots \mid \mathbf{q}_{n_s}]$ is an orthonormal matrix and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_s})$, $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_s}$. Then

$$\Phi(C) = \frac{(\mathbf{Q}^T \mathbf{D}_s^{\frac{1}{2}} \mathbf{y})^T \mathbf{\Lambda} (\mathbf{Q}^T \mathbf{D}_s^{\frac{1}{2}} \mathbf{y})}{(\mathbf{Q}^T \mathbf{D}_s^{\frac{1}{2}} \mathbf{y})^T (\mathbf{Q}^T \mathbf{D}_s^{\frac{1}{2}} \mathbf{y})}.$$

Let $x_i = \mathbf{q}_i^T \mathbf{D}_s^{\frac{1}{2}} \mathbf{y}$ be the projection of $\mathbf{D}_s^{\frac{1}{2}} \mathbf{y}$ on the i th eigenvector \mathbf{q}_i of \mathbf{L}_{sym} , we have

$$\Phi(C) = \frac{\sum_{i=1}^{n_s} \lambda_i x_i^2}{\sum_{i=1}^{n_s} x_i^2} = \sum_{i=1}^{n_s} w_i \lambda_i, \quad (1)$$

where $w_i = \frac{x_i^2}{\sum_{i=1}^{n_s} x_i^2}$ is the weighting coefficient of the eigenvalues. If $\Phi(C)$ is close to the smallest eigenvalue λ_1 , then most of the weight on average must be on the eigenvalues close to λ_1 .

THEOREM 4.2. *Let ϵ be a small positive real number, if $\Phi(C) < \lambda_1 + \epsilon$, then for any positive real number t ,*

$$\sum_{i: \lambda_i < \lambda_1 + t\epsilon} w_i > 1 - \frac{1}{t}.$$

PROOF. By Eq. (1) we have

$$\begin{aligned} \Phi(C) &= \sum_{i: \lambda_i < \lambda_1 + t\epsilon} w_i \lambda_i + \sum_{j: \lambda_j \geq \lambda_1 + t\epsilon} w_j \lambda_j \\ &\geq \lambda_1 \sum_{i: \lambda_i < \lambda_1 + t\epsilon} w_i + (\lambda_1 + t\epsilon) \sum_{j: \lambda_j \geq \lambda_1 + t\epsilon} w_j \\ &= \lambda_1 + t\epsilon \sum_{j: \lambda_j \geq \lambda_1 + t\epsilon} w_j. \end{aligned}$$

As $\Phi(C) < \lambda_1 + \epsilon$, we get

$$\lambda_1 + t\epsilon \sum_{j: \lambda_j \geq \lambda_1 + t\epsilon} w_j < \lambda_1 + \epsilon.$$

Therefore,

$$\sum_{j: \lambda_j \geq \lambda_1 + t\epsilon} w_j < \frac{1}{t}, \quad \sum_{i: \lambda_i < \lambda_1 + t\epsilon} w_i > 1 - \frac{1}{t}.$$

□

Note that $\lambda_1 = 0$ for the Laplacian matrix \mathbf{L}_{sym} [46], however, Theorem 4.2 holds for any real number λ_1 . Theorem 4.2 indicates for a low conductance $\Phi(C)$ close to the smallest value λ_1 , the

smaller eigenvalues of \mathbf{L}_{sym} contribute most of the weights. As $w_i = \frac{x_i^2}{\sum_{i=1}^{n_s} x_i^2}$, if we want to find a low conductance $\Phi(C)$, w_i and hence x_i should be larger on smaller eigenvalues λ_i ($1 \leq i \leq k \ll n_s$). As $x_i = \mathbf{q}_i^T \mathbf{D}_s^{\frac{1}{2}} \mathbf{y}$, a larger x_i indicates a smaller angle between $\mathbf{D}_s^{\frac{1}{2}} \mathbf{y}$ and eigenvector \mathbf{q}_i . When we relax \mathbf{y} from $\{0, 1\}^{n_s \times 1}$ to $[0, 1]^{n_s \times 1}$, the relaxed scaled indicator vector $\mathbf{D}_s^{\frac{1}{2}} \mathbf{y}$ should be well approximated by a linear combination of the dominant eigenvectors with smaller eigenvalues. As

$$\mathbf{L}_{\text{rw}} \mathbf{v} = \lambda \mathbf{v} \quad \Leftrightarrow \quad \mathbf{L}_{\text{sym}}(\mathbf{D}_s^{\frac{1}{2}} \mathbf{v}) = \lambda(\mathbf{D}_s^{\frac{1}{2}} \mathbf{v}),$$

where \mathbf{v} is a nonzero vector. It shows that the relaxed indicator vector \mathbf{y} should be well approximated by a linear combination of the dominant eigenvectors of \mathbf{L}_{rw} with smaller eigenvalues. Also,

$$\mathbf{L}_{\text{rw}} \mathbf{v} = (\mathbf{I} - \mathbf{N}_{\text{rw}}) \mathbf{v} = \lambda \mathbf{v} \quad \Leftrightarrow \quad \mathbf{N}_{\text{rw}} \mathbf{v} = (1 - \lambda) \mathbf{v}, \quad (2)$$

it follows that \mathbf{L}_{rw} and \mathbf{N}_{rw} share the same set of eigenvectors and the corresponding eigenvalue of \mathbf{N}_{rw} is $1 - \lambda$ where λ is the eigenvalue of \mathbf{L}_{rw} . Equivalently, the relaxed indicator vector \mathbf{y} should be well approximated by a linear combination of the eigenvectors of \mathbf{N}_{rw} with larger eigenvalues.

This leads to our idea of finding a relaxed sparse indicator vector \mathbf{y} containing the seeds in the span of the dominant eigenvectors with larger eigenvalues of \mathbf{N}_{rw} .

$$\begin{aligned} \min \quad & \|\mathbf{y}\|_1 = \mathbf{e}^T \mathbf{y} \\ \text{s.t.} \quad & (1) \mathbf{y} = \mathbf{V} \mathbf{u}, \\ & (2) y_i \in [0, 1], \\ & (3) y_i \geq \frac{1}{|S|}, i \in S, \end{aligned} \quad (3)$$

where the column vectors of \mathbf{V} are formed by the dominant eigenvectors of \mathbf{N}_{rw} with larger values and \mathbf{e} is the vector of all ones. \mathbf{u} is the coefficient vector, and y_i denotes the i th element of the relaxed indicator vector \mathbf{y} . The constraint $\mathbf{y} = \mathbf{V} \mathbf{u}$ indicates that \mathbf{y} lies in the eigenspace spanned by the column vectors of \mathbf{V} . Vector \mathbf{y} indicates the local community with low conductance containing the labeled seeds.

4.3 Krylov Subspace Approximation

4.3.1 Variants of Random Walk Diffusion. Instead of using the eigenvalue decomposition on \mathbf{N}_{rw} , we consider short random walks for the probability diffusion starting from the seed set to get the “local spectral subspace”. We define several variants of the spectral diffusion based on different transition matrices for the random walks.

1) *Standard Random Walk (SRW)* uses the transition matrix \mathbf{N}_{rw} for the probability diffusion.

$$\mathbf{N}_{\text{rw}} = \mathbf{D}_s^{-1} \mathbf{A}_s. \quad (4)$$

2) *Light Lazy Random Walk (LLRW)* retains some probability at the current node for the random walks.

$$\mathbf{N}_{\text{rw}} = (\mathbf{D}_s + \alpha \mathbf{I})^{-1} (\alpha \mathbf{I} + \mathbf{A}_s), \quad (5)$$

where $\alpha \in N^{0+}$. $\alpha = 0$ degenerates to the standard random walk and $\alpha = 1, 2, 3, \dots$ corresponds to a random walk in the modified graph with 1, 2, 3, ... loops at each node.

3) *Lazy Random Walk (LRW)* is defined by

$$\begin{aligned} \mathbf{N}_{\text{rw}} &= (\mathbf{D}_s + \alpha \mathbf{D}_s)^{-1} (\alpha \mathbf{D}_s + \mathbf{A}_s) \\ &= \frac{\alpha}{1 + \alpha} \mathbf{I} + \frac{1}{1 + \alpha} \mathbf{D}_s^{-1} \mathbf{A}_s, \end{aligned} \quad (6)$$

where $\alpha \in [0, 1]$. E.g. $\alpha = 0.1$ corresponds to a random walk that always retains $\frac{0.1}{1+0.1}$ probability on the current node during the diffusion process. $\alpha = 0$ degenerates to the standard random walk.

4) *Personalized PageRank (PPR)* is defined by

$$\mathbf{N}_{\text{rw}} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{D}_s^{-1} \mathbf{A}_s, \quad (7)$$

where $\alpha \in [0, 1]$ and S the diagonal matrix with binary indicators for the seed set S . E.g. $\alpha = 0.1$ corresponds to a random walk that always retains 10% of the probability on the seed set. $\alpha = 0$ is the standard random walk.

4.3.2 Regular and Inverse Random Walks. Based on the above random walk diffusion definition, one step of random walk is defined as $N_{\text{rw}}^T \mathbf{p}$ for a probability column vector \mathbf{p} , and the probability density for a random walk of length k is given by a Markov chain:

$$\mathbf{p}_k = N_{\text{rw}}^T \mathbf{p}_{k-1} = (N_{\text{rw}}^T)^k \mathbf{p}_0, \quad (8)$$

where \mathbf{p}_0 is the initial probability density evenly assigned on the seeds.

We can also define an “inverse random walk”:

$$\mathbf{p}_k = N_{\text{rw}} \mathbf{p}_{k-1} = (N_{\text{rw}})^k \mathbf{p}_0. \quad (9)$$

Here \mathbf{p}_k indicates a probability density such that the probability concentrates to the seed set as \mathbf{p}_0 after k steps of short random walks. The value of \mathbf{p}_k also shows a snapshot of the probability distribution for the local community around the seed set, and follow-up experiments also demonstrates the effectiveness of the “inverse random walk”, which has a slightly lower accuracy as compared with the “regular random walk”.

4.3.3 Local Spectral Subspace. Then we define a local spectral subspace as a proxy of the invariant subspace spanned by the leading eigenvectors of N_{rw} . The local spectral subspace is defined on an order- d Krylov matrix and k is the number of diffusion steps:

$$\mathbf{V}_d^{(k)} = [\mathbf{p}_k, \mathbf{p}_{k+1}, \dots, \mathbf{p}_{k+d-1}]. \quad (10)$$

Here k and d are both some modest numbers. Then the Krylov subspace spanned by the column vectors of $\mathbf{V}_d^{(k)}$ is called the local spectral subspace, denoted by $\mathcal{V}_d^{(k)}$.

In the following discussion, we provide some theoretical analysis to relate the spectral property to the local spectral subspace $\mathcal{V}_d^{(k)}$.

LEMMA 4.3. *Let $G_s = (V_s, E_s)$ be a connected and non-bipartite graph with n_s nodes and m_s edges, N_{rw} (defined by Eq. (4)) the transition matrix of G_s with eigenvalues $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n_s}$, and the corresponding normalized eigenvectors are $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$. Then $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$ are linearly independent, and*

$$1 = \sigma_1 > \sigma_2 \geq \dots \geq \sigma_{n_s} > -1, \quad \mathbf{u}_1 = \frac{\mathbf{e}}{\|\mathbf{e}\|_2},$$

where \mathbf{e} is a vector of all ones.

PROOF. By Eq. (2), we know that L_{rw} and N_{rw} share the same eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$ and the corresponding eigenvalues of L_{rw} are $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_s}$ where $\lambda_i = 1 - \sigma_i$ ($1 \leq i \leq n_s$).

According to Proposition 3 of [46], L_{sym} and L_{rw} share the n_s non-negative eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_s}$ and the corresponding eigenvectors of L_{sym} are $D_s^{\frac{1}{2}} \mathbf{u}_1, D_s^{\frac{1}{2}} \mathbf{u}_2, \dots, D_s^{\frac{1}{2}} \mathbf{u}_{n_s}$.

From Theorem 8.1.1 of [19], there exists an orthogonal matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{n_s}]$ such that

$$\mathbf{Q}^T L_{\text{sym}} \mathbf{Q} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_s}).$$

It shows that $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{n_s}$ are linearly independent eigenvectors of L_{sym} , so $D_s^{\frac{1}{2}} \mathbf{u}_1, D_s^{\frac{1}{2}} \mathbf{u}_2, \dots, D_s^{\frac{1}{2}} \mathbf{u}_{n_s}$ are linearly independent. As G_s is a connected graph, $D_s^{\frac{1}{2}}$ is invertible, and it is obvious that $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$ are linearly independent.

Additionally, as $L_{\text{rw}} \frac{\mathbf{e}}{\|\mathbf{e}\|_2} = \mathbf{0}$, we have $1 - \sigma_1 = \lambda_1 = 0$ and $\mathbf{u}_1 = \frac{\mathbf{e}}{\|\mathbf{e}\|_2}$, so $\sigma_1 = 1$.

As G_s is a connected and non-bipartite graph, by Lemma 1.7 of [13], we have $1 - \sigma_2 = \lambda_2 > 0$ and $1 - \sigma_{n_s} = \lambda_{n_s} < 2$. Therefore, $\sigma_2 < 1$ and $\sigma_{n_s} > -1$. \square

THEOREM 4.4. Let $G_s = (V_s, E_s)$ be a connected and non-bipartite graph with n_s nodes and m_s edges, when $k \rightarrow \infty$, \mathbf{p}_k defined by Eq. (9) converges to $\alpha_1 \mathbf{u}_1$ where α_1 is the nonzero weighting portion of \mathbf{p}_0 on the eigenvector \mathbf{u}_1 .

PROOF. By Lemma 4.3, $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$ are linearly independent normalized eigenvectors of \mathbf{N}_{rw} , so there exist $\alpha_1, \alpha_2, \dots, \alpha_{n_s}$ such that $\mathbf{p}_0 = \sum_{i=1}^{n_s} \alpha_i \mathbf{u}_i$.

As $\mathbf{u}_1 = \frac{\mathbf{e}}{\|\mathbf{e}\|_2}$ and \mathbf{p}_0 is the initial probability density evenly assigned on the seeds, \mathbf{u}_1 and \mathbf{p}_0 are not orthogonal. It shows that α_1 is the nonzero weighting portion of \mathbf{p}_0 on the eigenvector \mathbf{u}_1 . Then

$$\mathbf{p}_k = (\mathbf{N}_{\text{rw}})^k \mathbf{p}_0 = \sum_{i=1}^{n_s} \alpha_i \sigma_i^k \mathbf{u}_i = \sigma_1^k \sum_{i=1}^{n_s} \alpha_i \left(\frac{\sigma_i}{\sigma_1}\right)^k \mathbf{u}_i.$$

Since $1 = \sigma_1 > \sigma_2 \geq \dots \geq \sigma_{n_s} > -1$ and $\alpha_1 \neq 0$, for all $i = 2, 3, \dots, n_s$, we have

$$\lim_{k \rightarrow \infty} \left(\frac{\sigma_i}{\sigma_1}\right)^k = 0,$$

and

$$\lim_{k \rightarrow \infty} \mathbf{p}_k = \lim_{k \rightarrow \infty} \alpha_1 \sigma_1^k \mathbf{u}_1 = \alpha_1 \mathbf{u}_1.$$

□

Obviously, Lemma 4.3 and Theorem 4.4 also hold for \mathbf{N}_{rw} based on Eq. (5) or Eq. (6), which is the transition matrix of modified graph with a weighting loop at each node.

By Theorem 4.4, we have the following corollary.

COROLLARY 4.5. Suppose matrix \mathbf{N}_{rw} defined by Eq. (7) has n_s eigenvalues $\mu_1, \mu_2, \dots, \mu_{n_s}$ with an associated collection of linearly independent eigenvectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n_s}\}$. Moreover, we assume that $|\mu_1| > |\mu_2| \geq \dots \geq |\mu_{n_s}|$. Then we have

$$\lim_{k \rightarrow \infty} \mathbf{p}_k = \lim_{k \rightarrow \infty} \beta_1 \mu_1^k \mathbf{v}_1,$$

where \mathbf{p}_k defined by Eq. (9) and β_1 is the weighting portion of \mathbf{p}_0 on the eigenvector \mathbf{v}_1 .

Corollary 4.5 indicates that \mathbf{p}_k converges to $\beta_1 \mathbf{v}_1$ if $\mu_1 = 1$ and $\beta_1 \neq 0$.

Below, we provide some discussion on the convergence of \mathbf{p}_k defined by Eq. (8). Firstly, we give the following theorem.

THEOREM 4.6. Every real square matrix \mathbf{X} is a product of two real symmetric matrices, $\mathbf{X} = \mathbf{Y}\mathbf{Z}$ where \mathbf{Y} is invertible.

We will not include the proof of Theorem 4.6 here. The interested reader is referred to [10].

By Theorem 4.6, we have $\mathbf{N}_{\text{rw}} = \mathbf{P}\mathbf{U}$ where \mathbf{P} and \mathbf{U} are symmetric matrices, and \mathbf{P} is invertible. Then,

$$\mathbf{N}_{\text{rw}} = \mathbf{P}\mathbf{U} = \mathbf{P}(\mathbf{U}\mathbf{P})\mathbf{P}^{-1} = \mathbf{P}(\mathbf{P}\mathbf{U})^T\mathbf{P}^{-1} = \mathbf{P}\mathbf{N}_{\text{rw}}^T\mathbf{P}^{-1}. \quad (11)$$

According to Eq. (11), we have

$$\mathbf{N}_{\text{rw}}\mathbf{v} = \lambda\mathbf{v} \Leftrightarrow \mathbf{N}_{\text{rw}}^T(\mathbf{P}^{-1}\mathbf{v}) = \lambda(\mathbf{P}^{-1}\mathbf{v}), \quad (12)$$

where \mathbf{v} is a nonzero vector. It shows that \mathbf{N}_{rw} and \mathbf{N}_{rw}^T share the same set of eigenvalues and the corresponding eigenvector of \mathbf{N}_{rw}^T is $\mathbf{P}^{-1}\mathbf{v}$ where \mathbf{v} is the eigenvector of \mathbf{N}_{rw} .

By Lemma 4.3, we know that \mathbf{N}_{rw} defined by Eq. (4) has n_s linearly independent normalized eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{n_s}$. Then by Eq. (12), \mathbf{N}_{rw}^T has n_s linearly independent eigenvectors $\mathbf{P}^{-1}\mathbf{u}_1, \mathbf{P}^{-1}\mathbf{u}_2, \dots, \mathbf{P}^{-1}\mathbf{u}_{n_s}$. By Theorem 4.4, we have the following theorem.

THEOREM 4.7. Let $G_s = (V_s, E_s)$ be a connected and non-bipartite graph with n_s nodes and m_s edges, when $k \rightarrow \infty$, \mathbf{p}_k defined by Eq. (8) converges to $\gamma_1 \mathbf{P}^{-1}\mathbf{u}_1$ where γ_1 is the nonzero weighting portion of \mathbf{p}_0 on the eigenvector $\mathbf{P}^{-1}\mathbf{u}_1$.

The proof of Theorem 4.7 is similar to Theorem 4.4, hence we omit the details here.

Obviously, Theorem 4.7 also holds for N_{rw} based on Eq. (5) or Eq. (6), which is the transition matrix of modified graph with a weighting loop at each node.

By Eq. (12) and Theorem 4.7, we have the following corollary.

COROLLARY 4.8. *Suppose matrix N_{rw} defined by Eq. (7) has n_s eigenvalues $\mu_1, \mu_2, \dots, \mu_{n_s}$ with an associated collection of linearly independent eigenvectors $\{v_1, v_2, \dots, v_{n_s}\}$. Moreover, we assume that $|\mu_1| > |\mu_2| \geq \dots \geq |\mu_{n_s}|$. Then we have*

$$\lim_{k \rightarrow \infty} p_k = \lim_{k \rightarrow \infty} \delta_1 \mu_1^k P^{-1} v_1,$$

where p_k defined by Eq. (8) and δ_1 is the weighting portion of p_0 on the eigenvector $P^{-1} v_1$ of N_{rw}^T .

Corollary 4.8 indicates that p_k converges to $\delta_1 P^{-1} v_1$ if $\mu_1 = 1$ and $\delta_1 \neq 0$.

When $k \rightarrow \infty$, Theorem 4.4 and Corollary 4.5 state that the local spectral subspace $\mathcal{V}_d^{(k)}$ built on “inverse random walk” approaches the eigenspace associated with eigenvector of N_{rw} with the largest eigenvalue, and Theorem 4.7 and Corollary 4.8 indicate that the local spectral subspace $\mathcal{V}_d^{(k)}$ built on “regular random walk” approaches the eigenspace associated with eigenvector of N_{rw}^T with the largest eigenvalue. Our interest now, though, is not in the limiting case when k is large, but for a much more modest number of diffusion steps to reveal the local property around the seeds. Based on different local spectral diffusions in Eq. (4) - Eq. (7), we have a set of local spectral subspace definitions. Based on Eq. (8) and Eq. (9), we have two sets of local spectral subspace definitions.

Experiments in Section 5 show that the definitions on N_{rw}^T , which is on the regular random walk, is considerably better than that on N_{rw} for the detection accuracy. However, the definitions on N_{rw} corresponding to the inverse random walk also show high accuracy as compared with the baselines. Our results show that the local approximation built on “regular random walk” shows higher accuracy than that built on “inverse random walk”.

4.4 Local Community Detection

We modify the optimization problem shown in Eq. (3) by relaxing each element in the indicator vector y to be nonnegative, and approximating the global spectral subspace by the local spectral subspace. Thus, we seek a relaxed sparse vector in the local spectral subspace by solving a linear programming problem.

$$\begin{aligned} \min \quad & \|y\|_1 = e^T y \\ \text{s.t.} \quad & (1) \ y = V_d^{(k)} u, \\ & (2) \ y \geq 0, \\ & (3) \ y_i \geq \frac{1}{|S|}, \ i \in S. \end{aligned} \tag{13}$$

This is an ℓ_1 norm approximation for finding a sparse linear coding that indicates a small community containing the seeds with y in the local spectral subspace spanned by the column vectors of $V_d^{(k)}$. The i th entry y_i indicates the likelihood of node i belonging to the target community.

1) If $|T|$ is known, we then sort the values in y in non-ascending order and select the corresponding $|T|$ nodes with the higher belonging likelihood as the output community.

2) If $|T|$ is unknown, we use a heuristic to determine the community boundary. We sort the nodes based on the element values of y in the decreasing order, and find a set S_{k^*} with the first k^* nodes having a comparatively low conductance. Specifically, we start from an index k_0 where set S_{k_0} contains all the seeds. We then generate a sweep curve $\Phi(S_k)$ by increasing index k . Let k^* be

the value of k where $\Phi(S_k)$ achieves a first local minimum. The set S_{k^*} is regarded as the detected community.

We determine a local minima as follows. If at some point k^* when we are increasing k , $\Phi(S_k)$ stops decreasing, then this k^* is a candidate point for the local minimum. If $\Phi(S_k)$ keeps increasing after k^* and it eventually becomes higher than $\beta\Phi(S_{k^*})$, then we take k^* as a valid local minimum. We experimented with several values of β on a small trial of data and found that $\beta = 1.02$ gives good performance across all the datasets.

Denote the corresponding local community detection methods based on Eq. (4) - Eq. (7) as: LRw (LOSP based on Standard Random Walk), LLi (LOSP based on Light Lazy Random Walk), LLa (LOSP based on Lazy Random Walk) and LPr (LOSP based on PPR).

5 EXPERIMENTS AND RESULTS

We implement the family of local spectral methods (LOSPs) in Matlab^{3,4} and thoroughly compare them with state-of-the-art localized community detection algorithms on the 28 LFR datasets as well as the 8 real-world networks across multiple domains. For the 5 SNAP datasets, we randomly locate 500 labeled ground truth communities on each dataset, and randomly pick three exemplary seeds from each target community. For the 28 LFR datasets and the 3 Biology datasets, we deal with every ground truth community and randomly pick three exemplary seeds from each ground truth community. We pre-process all real-world datasets by sampling, and apply the local spectral methods for each network.

5.1 Statistics on Sampling

In order to evaluate the effectiveness of the sampling method in Algorithm 1, we empirically set $(N_1, N_2, k) = (300, 5000, 3)$ to control the subgraph size and experiment with the upper bound of BFS rounds t from 1 to 5 to extract different sampling rate on Amazon, as shown in Table 3. For notations, the coverage indicates the average fraction of ground truth covered by the sampled subgraph, and n_s/n indicates the sampling rate which is the average fraction of subgraph size as compared with the original network scale.

Table 3 shows that there is a 7.2% significant improvement on coverage when we increase the upper bound of BFS rounds t from 1 to 2, but there is only 0.4% improvement when t continue increases from 2 to 5. On the other hand, the sampling rate is only 0.1% and the sampling procedure is very fast in 0.730 seconds for $t = 2$. For these reasons, we set $(N_1, N_2, t, k) = (300, 5000, 2, 3)$ to trade off among the coverage, sampling rate and running time for the sampling method in our experiments.

Table 4 provides statistics on real-world networks for the sampling method in Algorithm 1. For SNAP datasets, our sampling method has a high coverage with reasonable sample size, covering about 96% ground truth with a small average sampling rate of 0.1%, and the sampling procedure is within 14 seconds. For the Biology networks, which are comparatively denser, the sampled subgraph covers about 91% ground truth with a relatively high sampling rate, and the sampling procedure is very fast in less than 0.2 seconds.

5.2 Parameter Setup

To remove the impact of different local spectral methods in finding a local minimum for the community boundary, we use the ground truth size as a budget for parameter testing on the family of LOSP methods. When we say LOSP, we mean the family of LOSP defined on the N_{rw}^T Krylov

³https://github.com/PanShi2016/LOSP_Plus

⁴https://github.com/JHL-HUST/LOSP_Plus

Table 3. Test parameter t (upper bound of BFS rounds) for the sampling on Amazon.

Statistics	Upper bound of BFS rounds				
	1	2	3	4	5
Coverage	0.918	0.990	0.991	0.993	0.994
n_s	13	34	70	184	312
n_s/n	0.00004	0.00010	0.00021	0.00055	0.00093
Time (s)	0.374	0.730	2.208	4.361	7.028

Table 4. Statistics on average values for the sampling on real-world networks.

	Networks	Coverage	n_s	n_s/n	Time (s)
SNAP	Amazon	0.990	34	0.0001	0.730
	DBLP	0.980	198	0.0006	0.720
	LiveJ	1.000	629	0.0002	19.050
	YouTube	0.950	3237	0.0028	3.760
	Orkut	0.870	4035	0.0013	44.430
	Average	0.958	1627	0.001	13.738
Biology	DM	0.910	2875	0.1880	0.256
	HS	0.876	2733	0.2692	0.125
	SC	0.947	3341	0.6049	0.076
	Average	0.911	2983	0.3540	0.152

subspace, which is the normal case for random walk diffusion. A comparison in subsection 5.3 will show that in general, LOSEP defined on \mathbf{N}_{rw}^T Krylov subspace outperforms that on \mathbf{N}_{rw} Krylov subspace with respect to the accurate detection.

Dimension of the subspace and diffusion steps. For local spectral subspace, we need to choose some modest numbers for the step k of random walks and the subspace dimension d such that the probability diffusion does not reach the global stationary. We did a small trial parameter study on all datasets, and found that $d = 2$ and $k = 2$ perform the best in general.

Parameters for the random walk diffusion. We thoroughly evaluate different spectral diffusion methods on all datasets, as shown in Fig. 1 and Fig. 2. The three columns correspond to light lazy random walk, lazy random walk and personalized pagerank with different α parameters. All three variants degenerate to the standard random walk when $\alpha = 0$. The results show that light lazy random walk, lazy random walk and personalized pagerank are robust for different α parameters. The personalized pagerank declines significantly when $\alpha = 1$ as all probability returns to the original seed set.

During the probability diffusion, light lazy random walk and lazy random walk always retain a ratio of probability on the current set of nodes to keep the detected structure to be “local”. The personalized pagerank always returns a ratio of probability from the current set of nodes to the seed set. Instead of retaining some probability distribution on the current set of nodes, the personalized

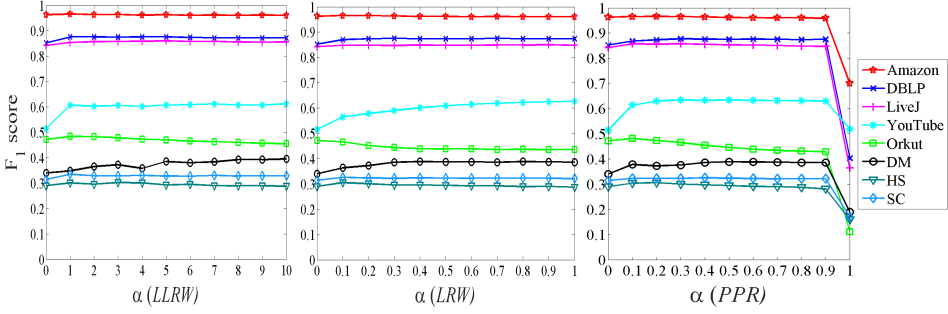


Fig. 1. Evaluation of different diffusion parameters α on real-world datasets (Defined on N_{rw}^T Krylov subspace, community size truncated by truth size). The three diffusions are robust for different α parameters, except for $\alpha = 1$ on personalized pagerank, in which case all probability returns to the original seed set.

pagerank “shrinks” some probability to the original seed set. Such process also wants to keep the probability distribution “local” but it is not continuous as compared with the previous two methods.

In the following discussion, we set $\alpha = 1$ for light lazy random walk and lazy random walk, and set $\alpha = 0.1$ for personalized pagerank.

5.3 Evaluation on Local Spectral Methods

To remove the impact of different methods in finding a local minimum for the community boundary, we use the ground truth size as a budget for the proposed four LOSP variants: LRw (standard), LLi (light lazy), LLa (lazy) and LPr (pagerank). We first compare the four LOSP variants defined on the standard N_{rw}^T Krylov subspace, then compare the general performance on subspaces defined on either N_{rw}^T or N_{rw} .

Evaluation on variants of N_{rw}^T Krylov subspace. Fig. 3 illustrates the average detection accuracy on the eight real-world datasets. LLi, LPr, LLa and LRw achieve almost the same performance on almost all datasets. One exception is on YouTube that LLi, LLa and LPr considerably outperform LRw. LLi achieves slightly better performance on four out of eight real-world networks.

Fig. 4 illustrates the average detection accuracy on the four sets with a total of 28 LFR networks. For $on = 500$, Fig. 4 (a) and Fig. 4 (c) show that LLi, LPr and LRw achieve almost the same performance and outperform LLa on average. For $on = 2500$, Fig. 4 (b) and Fig. 4 (d) show that LLi, LPr and LLa achieve almost the same performance and outperform LRw on average. On both cases, LLi, LOSP with light lazy, demonstrates slightly higher accuracy on LFR datasets.

For the running time on the sampled subgraphs of all datasets, LLi, LPr, LLa and LRw take almost the same time, and run within 1.1 seconds, as shown in Fig. 5 and Fig. 6.

Comparison on N_{rw}^T and N_{rw} Krylov subspace. We compare the average F_1 score on each group of networks: SNAP, Biology and the four groups of LFR. Table 5 shows the comparison of detection accuracy where the community is truncated on truth size for subspace definitions on N_{rw}^T and N_{rw} respectively (The best three values on each row appear in bold). In general, N_{rw}^T outperforms N_{rw} , especially on real-world datasets. When comparing with the LOSP variant defined on N_{rw} , the four variants defined on N_{rw}^T have about 15% or 5% higher F_1 scores on SNAP and Biology respectively. As for the synthetic LFR datasets, in general, N_{rw}^T performs slightly better than N_{rw} on the small ground truth communities, while N_{rw} performs slightly better than N_{rw}^T on the big ground truth communities. Among all the variants of LOSP, LLi on N_{rw}^T is always on top three for all datasets. In summary, LLi on N_{rw}^T performs better than other variants of LOSP.

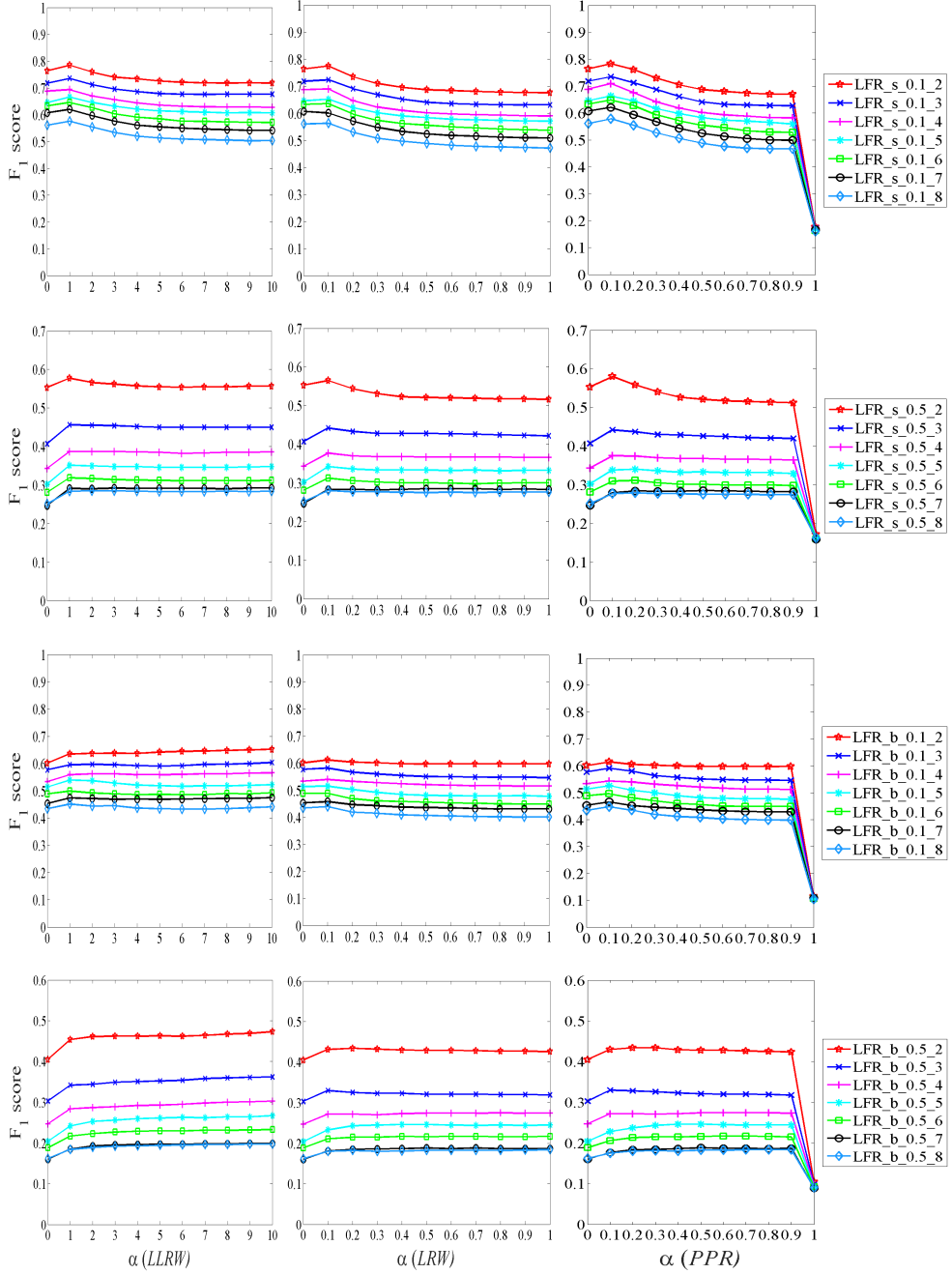


Fig. 2. Evaluation of different diffusion parameters α on LFR datasets (Defined on N_{rw}^T Krylov subspace, community size truncated by truth size). The three diffusions are robust for different α parameters, except for $\alpha = 1$ on personalized pagerank, in which case all probability returns to the original seed set.

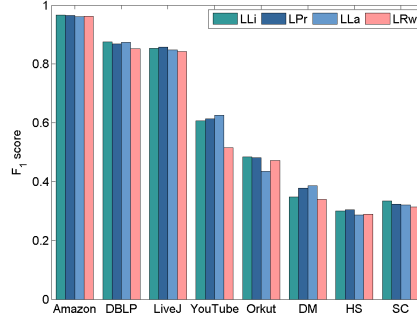
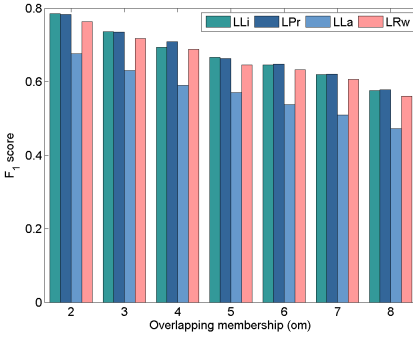
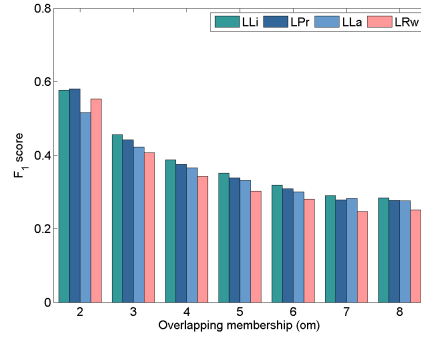


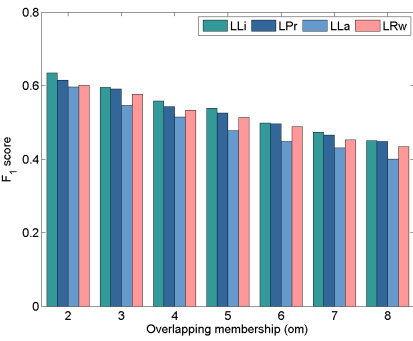
Fig. 3. Accuracy evaluation of LOSP on real-world networks (Defined on N_{rw}^T Krylov subspace, community size truncated by truth size). LOSPs based on the four diffusions show similar accuracy over all datasets. LLi, LOSP with light lazy, demonstrates slightly higher accuracy on half of the datasets.



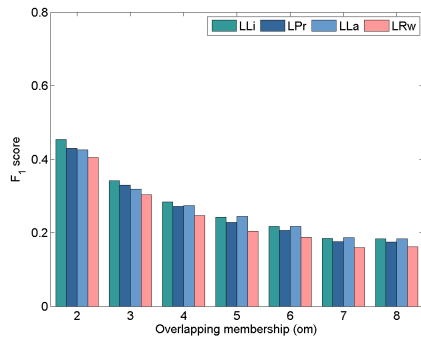
(a) LFR_s_0.1



(b) LFR_s_0.5



(c) LFR_b_0.1



(d) LFR_b_0.5

Fig. 4. Accuracy evaluation of LOSP on LFR networks (Defined on N_{rw}^T Krylov subspace, community size truncated by truth size). The accuracy decays when the overlapping membership, om , increases from 2 to 8. LOSPs based on the four diffusions show similar accuracy over all datasets. LLi, LOSP with light lazy, demonstrates slightly higher accuracy.

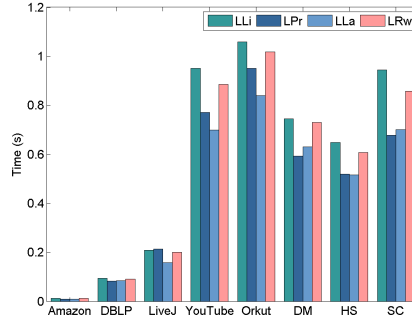
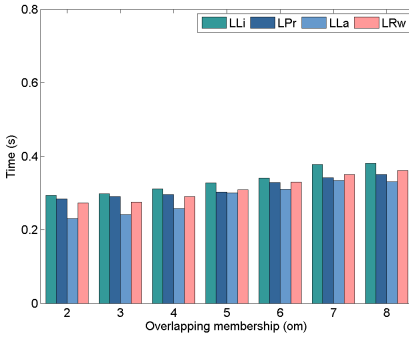
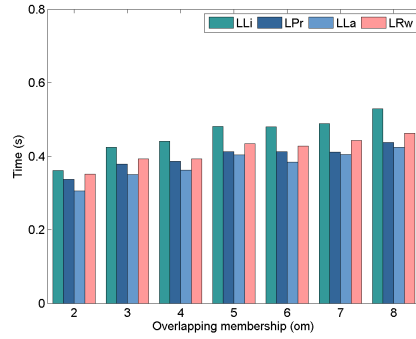


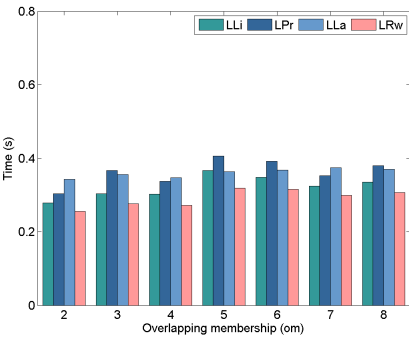
Fig. 5. Running time evaluation of LOSEP on real-world networks (Defined on N_{rw}^T Krylov subspace, community size truncated by truth size). LOSEPs based on the four diffusions show similar running time over all datasets, which are within 1.1 seconds.



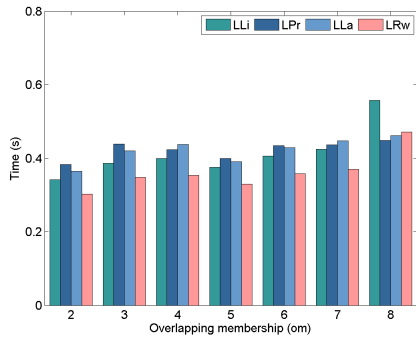
(a) LFR_s_0.1



(b) LFR_s_0.5



(c) LFR_b_0.1



(d) LFR_b_0.5

Fig. 6. Running time evaluation of LOSEP on LFR networks (Defined on N_{rw}^T Krylov subspace, community size truncated by truth size). LOSEPs based on the four diffusions show similar running time over all datasets, which are around 0.4 seconds.

Table 5. Comparison of average F_1 score for subspaces on N_{rw}^T and N_{rw} (Truncated on truth size). Among all the variants of LOSP, LLi on N_{rw}^T is always on top three for all datasets.

Datasets	Subspace on N_{rw}^T				Subspace on N_{rw}			
	LLi	LPr	LLa	LRw	LLi	LPr	LLa	LRw
SNAP	0.757	0.757	0.749	0.729	0.578	0.601	0.594	0.556
Biology	0.328	0.335	0.331	0.315	0.287	0.284	0.246	0.266
LFR_s_0.1	0.675	0.677	0.570	0.660	0.673	0.660	0.600	0.628
LFR_s_0.5	0.381	0.371	0.356	0.341	0.373	0.351	0.364	0.317
LFR_b_0.1	0.536	0.526	0.488	0.515	0.568	0.550	0.516	0.523
LFR_b_0.5	0.273	0.260	0.265	0.238	0.272	0.254	0.272	0.230

Table 6. Comparison of average F_1 score for subspaces on N_{rw}^T and N_{rw} (Truncated on local minimal conductance). Among all the variants of LOSP, LLi on N_{rw}^T is always on top three for all datasets.

Datasets	Subspace on N_{rw}^T				Subspace on N_{rw}			
	LLi	LPr	LLa	LRw	LLi	LPr	LLa	LRw
SNAP	0.620	0.604	0.599	0.593	0.518	0.514	0.526	0.503
Biology	0.182	0.171	0.206	0.150	0.146	0.160	0.166	0.135
LFR_s_0.1	0.574	0.561	0.443	0.554	0.576	0.546	0.518	0.539
LFR_s_0.5	0.334	0.291	0.265	0.232	0.332	0.311	0.337	0.217
LFR_b_0.1	0.387	0.365	0.296	0.358	0.406	0.348	0.353	0.393
LFR_b_0.5	0.198	0.169	0.168	0.134	0.217	0.186	0.214	0.131

It is interesting that the results on N_{rw} is reasonably good. Even if we use the first local conductance to do the truncation, as shown in Table 6 (The best three values on each row appear in bold), the accuracy decays by about 5% to 15% as compared with that of the truncation on truth size (in Table 5). Nevertheless, the results of Table 6 are still much better than that of state-of-the-art local community detection algorithms, heat kernel (HK) and pagerank (PR), whose results are listed in Table 7 and Table 9.

Comparison on Krylov subspace and eigenspace for N_{rw} . In order to evaluate the effectiveness of Krylov subspace approximation, we compare Krylov subspace with the actual eigenspace. To make a fair comparison, we choose LRw defined on N_{rw} Krylov subspace and adopt our method on the actual eigenspace associated with two leading eigenvectors of standard transition matrix N_{rw} with larger eigenvalues, denoted as GLOSP (GLOBAL Spectral).

Fig. 7 illustrates the average detection accuracy on real-world networks and LFR datasets. For real-world networks, Fig. 7 (a) shows that LRw considerably outperforms GLOSP. For LFR datasets, Fig. 7 (b) illustrates that LRw is much better than GLOSP on each dataset. These experiments show that it is not good if we use the actual eigenspace that embodies the global structure, even on the comparatively small subgraph.

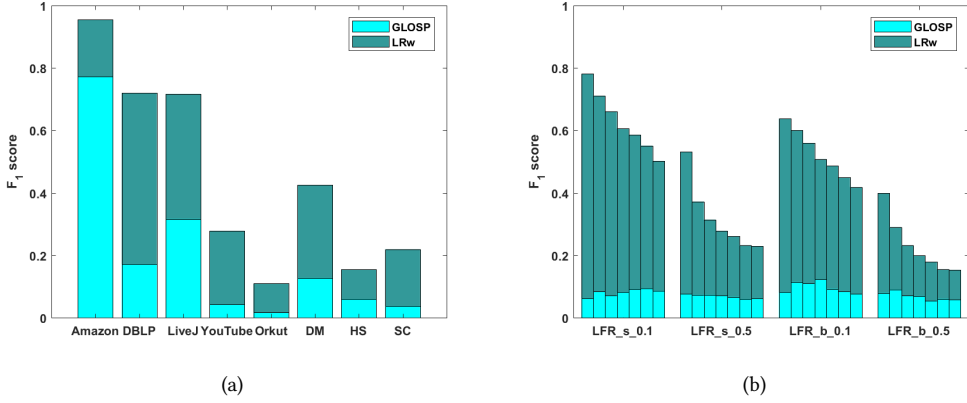


Fig. 7. Accuracy comparison with GLOSP based on actual eigenspace on real-world networks and LFR datasets (Community size truncated by truth size). In (b), the overlapping membership (om) starts from 2 to 8 for each group of LFR datasets. Note that the bars for GLOSP and LRw both start from zero to the height.

5.4 Final Comparison

For the final comparison, we find a local minimum of conductance to automatically determine the community boundary. Subsection 5.3 shows that LLI, LOSP with light lazy, is on average the best LOSP method defined on N_{rw}^T Krylov subspace on all datasets. In this subsection, we just compare LLI defined on the N_{rw}^T Krylov subspace with four state-of-the-art local community detection algorithms, LEMON [34], PGDc-d [45], HK for hk-relax [27] and PR for pprpush [5], which also use conductance as the metric to determine the community boundary. LEMON is a local spectral approach based on normalized adjacency matrix iteration, PGDc-d is a projected gradient descent algorithm for optimizing σ -conductance, and HK is based on heat kernel diffusion while PR is based on the pagerank diffusion. To make a fair comparison, we use the default parameter settings for baselines as they also test on SNAP datasets, and run the five algorithms on the same three seeds randomly chosen from the ground truth communities.

5.4.1 Comparison on Real-world Datasets. For each of the real-world networks, Table 7 shows the average detection accuracy (The best value appears in bold in each row) and the average running time, Table 8 shows the average community size and the average conductance of the detected communities.

For the SNAP datasets in product, collaboration and social domains, LLI yields considerably higher accuracy on the first three datasets (Amazon, DBLP, and LiveJ), and yields slightly lower accuracy on YouTube and Orkut. To have a better understanding on the results, we compare the property of the detection with the property of the ground truth communities, as shown in Table 2.

- For the first three datasets, the average size of the ground truth is small between 10 to 30, and the conductance is diverse (very low in Amazon and with reasonable value around 0.4 in DBLP and LiveJ). In general, the size and conductance of our detected communities are closest to the ground truth. This may explain why our detection accuracy is considerably higher than the baselines.
- For the last two datasets YouTube and Orkut, the average conductance of ground truth is very high with 0.84 and 0.73, respectively, and the size is diverse (very small of 21 on average in YouTube and large of 216 on average in Orkut). We found much larger communities with close conductance on YouTube (0.736) and Orkut (0.791). PGDc-d found much smaller communities

with close conductance on YouTube, and HK found reasonable-size communities with lower conductance on Orkut. This may explain why our detection accuracy on YouTube and Orkut is slightly lower than PGDc-d and HK, respectively.

For the three Biology datasets, LLi outperforms HK and PR, and yields slightly lower accuracy compared with LEMON and PGDc-d. Also, the conductance of the communities detected by LEMON (around 0.88) and PGDc-d (around 0.98) are very close to that of the ground truth (around 0.88, as shown in Table 2), compared with LLi (around 0.74), HK (around 0.37) and PR (around 0.20). As for the community size, we detected larger communities as compared to the ground truth, LEMON and PGDc-d detect much smaller communities, while HK and PR detect even larger communities.

For the running time, as shown in Table 7, LLi is fast in 4 seconds on SNAP and in 16 seconds on Biology datasets. The four baselines, LEMON, PGDc-d, HK and PR, are faster in less than 1.2 seconds. This may be due to that LLi is implemented in Matlab while the most baselines are implemented in C++.

Table 7. Comparison on detection accuracy and running time with baselines on real-world networks.

Datasets	F_1 score					Time (s)				
	LLi	LEMON	PGDc-d	HK	PR	LLi	LEMON	PGDc-d	HK	PR
SNAP	Amazon	0.800	0.723	0.576	0.751	0.531	0.011	0.039	0.058	0.008 0.015
	DBLP	0.779	0.590	0.523	0.413	0.388	0.099	0.271	0.063	0.025 0.075
	LiveJ	0.797	0.519	0.388	0.573	0.525	0.756	0.553	1.759	0.029 0.264
	YouTube	0.445	0.351	0.447	0.091	0.143	9.268	1.458	0.329	0.038 0.955
	Orkut	0.279	0.096	0.117	0.357	0.303	12.235	1.328	3.212	0.027 1.392
	Average	0.620	0.456	0.410	0.437	0.378	4.474	0.730	1.084	0.025 0.540
Biology	DM	0.221	0.322	0.274	0.219	0.035	33.535	1.151	0.007	0.015 1.394
	HS	0.206	0.200	0.213	0.025	0.020	3.823	0.462	0.003	0.071 1.048
	SC	0.120	0.239	0.238	0.036	0.034	11.119	0.950	0.002	0.034 0.979
	Average	0.182	0.254	0.242	0.093	0.030	16.159	0.854	0.004	0.040 1.140

Table 8. Comparison on community size and conductance with baselines on real-world networks.

Datasets	Size					Conductance				
	LLi	LEMON	PGDc-d	HK	PR	LLi	LEMON	PGDc-d	HK	PR
SNAP	Amazon	8	13	3	48	4485	0.300	0.850	0.605	0.042 0.030
	DBLP	10	20	4	87	9077	0.396	0.559	0.751	0.110 0.114
	LiveJ	35	26	3	119	512	0.305	0.637	0.817	0.083 0.086
	YouTube	588	18	12	122	13840	0.736	0.748	0.723	0.175 0.302
	Orkut	1733	9	3	341	1648	0.791	0.934	0.976	0.513 0.546
	Average	475	17	5	143	5912	0.506	0.746	0.774	0.185 0.216
Biology	DM	992	19	3	606	15120	0.772	0.909	0.980	0.571 0.181
	HS	465	13	5	4360	10144	0.758	0.836	0.966	0.229 0.080
	SC	1623	14	3	2604	2673	0.696	0.903	0.990	0.322 0.343
	Average	1027	15	4	2523	9312	0.742	0.883	0.979	0.374 0.201

Table 9. Comparison on detection accuracy and running time with baselines on LFR networks.

Datasets		F_1 score					Time (s)				
		LLi	LEMON	PGDc-d	HK	PR	LLi	LEMON	PGDc-d	HK	PR
LFR	s_0.1_2	0.673	0.458	0.261	0.051	0.025	0.918	1.282	0.207	0.025	0.692
	s_0.1_3	0.621	0.452	0.261	0.028	0.025	0.969	1.031	0.201	0.025	0.693
	s_0.1_4	0.583	0.445	0.263	0.027	0.024	1.022	1.470	0.201	0.025	0.690
	s_0.1_5	0.555	0.436	0.258	0.044	0.024	1.123	1.814	0.202	0.026	0.675
	s_0.1_6	0.548	0.435	0.255	0.026	0.024	1.269	1.666	0.207	0.025	0.682
	s_0.1_7	0.536	0.441	0.260	0.041	0.026	1.342	1.703	0.197	0.025	0.677
	s_0.1_8	0.503	0.429	0.252	0.029	0.024	1.371	1.502	0.205	0.025	0.678
	Average	0.574	0.442	0.259	0.035	0.025	1.145	1.495	0.203	0.025	0.684
	s_0.5_2	0.496	0.389	0.266	0.019	0.020	1.232	1.222	0.209	0.028	0.680
	s_0.5_3	0.388	0.310	0.253	0.020	0.022	1.626	1.160	0.212	0.026	0.686
	s_0.5_4	0.336	0.295	0.257	0.019	0.020	1.708	1.032	0.232	0.028	0.683
	s_0.5_5	0.313	0.281	0.261	0.018	0.019	1.949	1.052	0.217	0.028	0.687
	s_0.5_6	0.284	0.268	0.257	0.017	0.018	1.899	0.896	0.214	0.027	0.690
	s_0.5_7	0.260	0.249	0.257	0.017	0.018	1.914	0.928	0.216	0.027	0.694
	s_0.5_8	0.260	0.247	0.261	0.016	0.017	2.037	0.939	0.216	0.025	0.696
	Average	0.334	0.291	0.259	0.018	0.019	1.766	1.033	0.217	0.027	0.688
	b_0.1_2	0.461	0.279	0.140	0.094	0.040	1.089	0.924	0.189	0.026	0.737
	b_0.1_3	0.417	0.260	0.135	0.064	0.043	1.253	0.779	0.189	0.025	0.720
	b_0.1_4	0.392	0.268	0.141	0.092	0.039	1.424	0.931	0.195	0.025	0.717
	b_0.1_5	0.361	0.255	0.134	0.073	0.042	1.595	0.963	0.194	0.025	0.713
	b_0.1_6	0.374	0.272	0.144	0.060	0.038	1.640	0.923	0.194	0.025	0.718
	b_0.1_7	0.353	0.267	0.143	0.047	0.039	1.608	1.019	0.207	0.025	0.714
	b_0.1_8	0.349	0.269	0.143	0.051	0.038	1.597	0.902	0.209	0.025	0.721
	Average	0.387	0.267	0.140	0.069	0.040	1.458	0.920	0.197	0.025	0.720
	b_0.5_2	0.309	0.207	0.135	0.040	0.042	1.579	0.793	0.210	0.025	0.718
	b_0.5_3	0.236	0.197	0.135	0.039	0.041	1.858	0.896	0.205	0.025	0.706
	b_0.5_4	0.198	0.176	0.138	0.035	0.037	2.026	0.883	0.200	0.025	0.701
	b_0.5_5	0.177	0.154	0.135	0.033	0.036	1.818	0.949	0.205	0.025	0.689
	b_0.5_6	0.170	0.151	0.141	0.031	0.033	2.079	1.029	0.216	0.025	0.705
	b_0.5_7	0.150	0.144	0.134	0.031	0.033	2.123	1.142	0.200	0.025	0.704
	b_0.5_8	0.149	0.149	0.137	0.030	0.032	2.116	0.968	0.203	0.025	0.703
	Average	0.198	0.168	0.136	0.034	0.036	1.943	0.951	0.206	0.025	0.704

5.4.2 Comparison on LFR Datasets. For each of the synthetic LFR datasets (Properties described in subsection 3.2.1. Recall that the ground truth communities are in size [10,50] for **s** and [20,100] for **b**), Table 9 shows the average detection accuracy (The best value appears in bold in each row) and the average running time, Table 10 further shows the average community size and the average conductance of the detected communities.

For the four groups with a total of 28 LFR datasets, LLi clearly outperforms the state-of-the-art baselines, LEMON, PGDc-d, HK and PR, as evaluated by F_1 score in Table 9. On further analysis of each group, shown in Table 10, we see that:

Table 10. Comparison on community size and conductance with baselines on LFR networks.

Datasets		Size					Conductance				
		LLi	LEMON	PGDc-d	HK	PR	LLi	LEMON	PGDc-d	HK	PR
LFR	s_0.1_2	13	20	3	2279	2327	0.557	0.537	0.938	0.276	0.269
Size:	s_0.1_3	12	17	3	2356	2317	0.598	0.563	0.945	0.289	0.281
[10, 50]	s_0.1_4	11	20	3	2354	2318	0.619	0.534	0.954	0.294	0.286
	s_0.1_5	11	19	3	2363	2328	0.636	0.556	0.957	0.300	0.290
Cond.:	s_0.1_6	34	20	3	2387	2302	0.627	0.548	0.954	0.295	0.294
0.522	s_0.1_7	22	20	3	2349	2272	0.644	0.555	0.959	0.300	0.297
	s_0.1_8	39	19	3	2371	2278	0.666	0.568	0.968	0.305	0.302
	Average	20	19	3	2351	2306	0.621	0.552	0.954	0.294	0.288
	s_0.5_2	49	14	3	2420	2313	0.741	0.643	0.956	0.333	0.324
Size:	s_0.5_3	76	15	3	2456	2281	0.789	0.675	0.966	0.348	0.346
[10, 50]	s_0.5_4	121	15	3	2459	2258	0.801	0.695	0.972	0.359	0.357
	s_0.5_5	129	16	3	2463	2246	0.814	0.699	0.978	0.362	0.363
Cond.:	s_0.5_6	210	16	3	2384	2242	0.813	0.714	0.981	0.355	0.368
0.746	s_0.5_7	213	17	3	2374	2231	0.817	0.715	0.986	0.358	0.371
	s_0.5_8	220	17	3	2304	2230	0.812	0.721	0.985	0.368	0.374
	Average	145	16	3	2409	2257	0.798	0.695	0.975	0.355	0.358
	b_0.1_2	16	16	3	2258	2344	0.666	0.660	0.977	0.303	0.286
Size:	b_0.1_3	15	14	3	2368	2340	0.688	0.686	0.975	0.307	0.295
[20, 100]	b_0.1_4	22	16	3	2267	2343	0.702	0.656	0.981	0.308	0.295
	b_0.1_5	25	18	3	2354	2334	0.716	0.659	0.976	0.301	0.300
Cond.:	b_0.1_6	50	17	3	2374	2323	0.698	0.647	0.974	0.296	0.301
0.497	b_0.1_7	29	17	3	2417	2326	0.710	0.638	0.980	0.296	0.301
	b_0.1_8	40	16	3	2374	2330	0.714	0.664	0.980	0.312	0.304
	Average	28	16	3	2344	2334	0.699	0.659	0.978	0.303	0.297
	b_0.5_2	33	14	3	2465	2299	0.792	0.711	0.982	0.343	0.336
Size:	b_0.5_3	106	16	3	2430	2280	0.808	0.695	0.986	0.340	0.355
[20, 100]	b_0.5_4	237	16	3	2440	2252	0.813	0.711	0.990	0.353	0.362
	b_0.5_5	184	18	3	2473	2242	0.815	0.692	0.989	0.373	0.365
Cond.:	b_0.5_6	196	15	3	2425	2240	0.817	0.733	0.991	0.359	0.368
0.733	b_0.5_7	247	19	3	2415	2224	0.810	0.696	0.994	0.363	0.369
	b_0.5_8	268	17	3	2406	2221	0.804	0.720	0.994	0.363	0.370
	Average	182	16	3	2436	2251	0.808	0.708	0.989	0.356	0.361

- **LFR_s_0.1.** The size of the ground truth is in [10, 50], and the average conductance of the ground truth is 0.522. The average community size of LLi over the seven LFR_s_0.1 datasets is 20, while the baselines detect either much smaller communities in size 3 (PGDc-d) or much larger communities in size 2300 (HK and PR). Due to the difference on the detected community size, LLi has a stable conductance of around 0.62 while the baselines have a high conductance of around 0.95 (PGDc-d) or a low conductance of around 0.29 (HK and PR).
- **LFR_s_0.5.** The size of the ground truth is in [10, 50], and the average conductance of the ground truth is 0.746. The average community size of LLi is considerably larger in 145, and the average conductance of 0.798 is close to that of the ground truth. LEMON and PGDc-d achieve

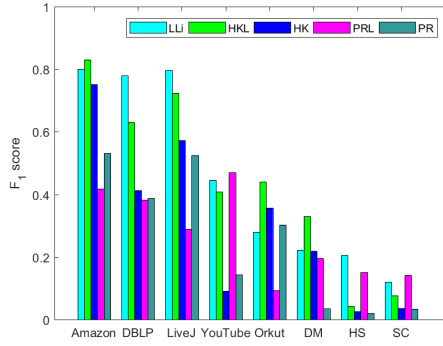
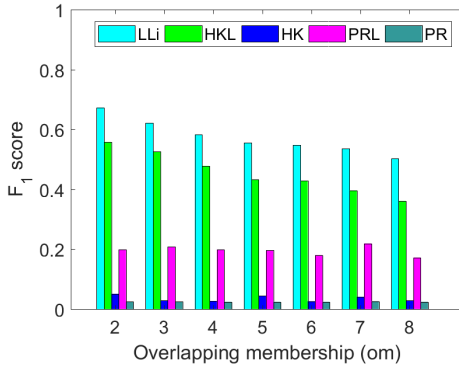
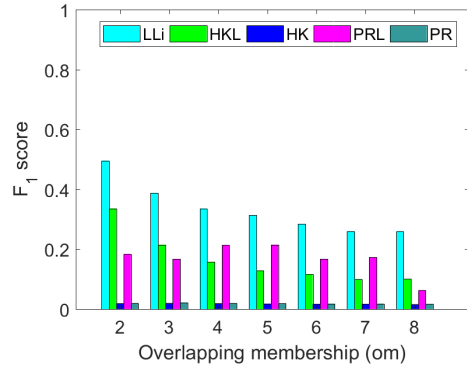


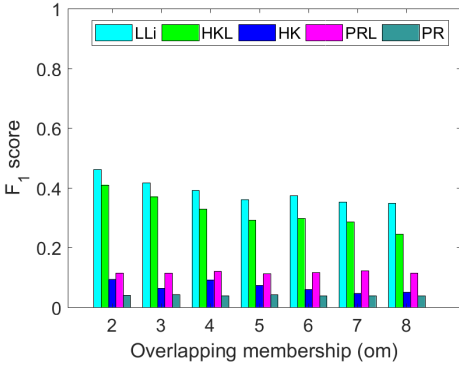
Fig. 8. Accuracy comparison with HKL and PRL on real-world networks (Community size truncated by local minimal conductance).



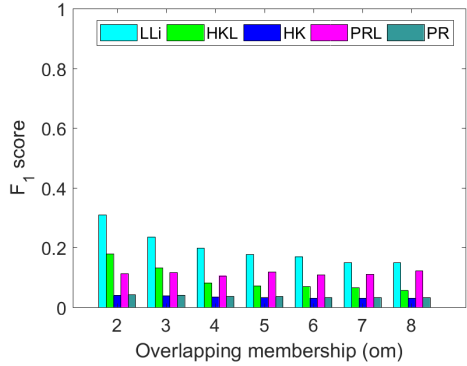
(a) LFR_s_0.1



(b) LFR_s_0.5



(c) LFR_b_0.1



(d) LFR_b_0.5

Fig. 9. Accuracy comparison with HKL and PRL on LFR datasets (Community size truncated by local minimal conductance).

smaller communities with higher conductance. HK and PR find much larger communities, leading to apparently smaller conductance.

- **LFR_b_0.1.** The size of the ground truth is in [20, 100], and the average conductance of the ground truth is 0.497. The average community size of LLi is suitable in 28. The baselines detect much smaller communities of size around 3 (PGDc-d) or much larger communities of size around 2300 (HK and PR). Due to the difference on the detected community size, LLi has a stable conductance of around 0.70, while the baselines have either much higher conductance of around 0.98 (PGDc-d) or much lower conductance of around 0.30 (HK and PR).
- **LFR_b_0.5.** The size of the ground truth is in [20, 100], and the average conductance of the ground truth is 0.733. The average community size of LLi is considerably larger in 182, and the average conductance of 0.808 is close to that of the ground truth. LEMON and PGDc-d achieve smaller communities with higher conductance. HK and PR find much larger communities, leading to apparently smaller conductance.

It is reasonable that the detection accuracy decays on graphs where there exist more overlappings indicated by higher *om* and *on*. LLi is adaptive to find suitable size of communities for different configurations, and substantially outperforms the baselines. LEMON and PGDc-d tend to find much-smaller-size communities with higher-conductance, while HK and PR tend to find much-larger-size communities with lower-conductance and their detection accuracy is very low in less than 0.1 on average on each of the four groups of datasets.

For the running time, as shown in Table 9, LLi is fast in 1 to 2 seconds while the four baselines, LEMON, PGDc-d, HK and PR, are slightly faster in less than 1.5 seconds. This may due to the implementation difference on programming language.

5.4.3 More Comparison on the variants of HK and PR. In order to evaluate the effectiveness of using the local minimal conductance for determining the community boundary, we compare LLi with the modification versions of HK and PR, based on local minimal conductance rather than global minimal conductance, denoted by HKL and PRL.

Fig. 8 and Fig. 9 illustrate the average detection accuracy on real-world networks and LFR datasets, respectively. For real-world networks, Fig. 8 shows that HKL yields higher accuracy on each dataset compared with HK, while PRL achieves lower accuracy compared with PR on Amazon, DBLP, LiveJ and Orkut. On average, LLi still outperforms HKL and PRL. For LFR datasets, Fig. 9 shows that LLi outperforms HKL and PRL on each LFR network. HKL and PRL yield considerably higher accuracy on each LFR dataset compared with HK and PR, respectively. These experiments show that it is useful for improving performance based on local minimal conductance truncation.

5.4.4 Summary on the Proposed Algorithm. For all comparisons of LLi, LOSP with light lazy, with four baselines, we see that LLi is adaptive for different configurations of the synthetic LFR datasets. LLi could find small communities with dozens of members for *s*: [10,50] and medium communities with hundreds of members for *b*: [20,100].

For well-defined community structure (low conductance, less overlapping, reasonably small), LLi has a high detection accuracy (**LFR_s_0.1**, **LFR_b_0.1**). When the overlapping membership (*om*) or the number of overlapping nodes (*on*) increases, the community structure becomes more mixed, and the detection accuracy decays. Nevertheless, LLi always yields the best accuracy for different parameter settings. By comparison, LEMON always finds communities of size around 20, PGDc-d always finds communities of size less than 10, and HK as well as PR always find large communities of size around 2300. Thus, they are not very scalable to networks with diverse community structure.

Comparisons on real-world datasets also show that LLi is very good for finding small communities with reasonably low conductance (Amazon, DBLP and LiveJ). If the community structure is not very clear, for example, with high conductance of around 0.90 on the Biology datasets, LLi tends to

find larger communities using longer time due to the sweep search, and it does not yield the best accuracy.

Therefore, our method can be used for large-scale real-world complex networks, especially when the communities are in reasonable size of no greater than 500, and the network has a reasonable clear community structure.

6 CONCLUSION

This paper systematically explores a family of local spectral methods (LOSP) for finding members of a local community from a few randomly selected seed members. Based on a Krylov subspace approximation, we define “approximate eigenvectors” for a subgraph including a neighborhood around the seeds, and describe how to extract a community from these approximate eigenvectors. By using different seed sets that generate different subspaces, our method is capable of finding overlapping communities. Variants of LOSP are introduced and evaluated. Four types of random walks with different diffusion speeds are studied, regular random walk and inverse random walk are compared, and analysis on the link between Krylov subspace and eigenspace is provided. For this semi-supervised learning task, LOSP outperforms prior state-of-the-art local community detection methods in social and biological networks as well as synthetic LFR datasets.

A THE PROOF OF THEOREM 4.1

Before given the proof of Theorem 4.1, we first give the following theorem.

THEOREM A.1. (Courant-Fischer Formula) Let \mathbf{H} be an $n \times n$ symmetric matrix with eigenvalues $\lambda_1^{(H)} \leq \lambda_2^{(H)} \leq \dots \leq \lambda_n^{(H)}$ and corresponding eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$. Then

$$\begin{aligned}\lambda_1^{(H)} &= \min_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{H} \mathbf{x} = \min_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T \mathbf{H} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \\ \lambda_2^{(H)} &= \min_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x} \perp \mathbf{v}_1}} \mathbf{x}^T \mathbf{H} \mathbf{x} = \min_{\substack{\mathbf{x} \neq \mathbf{0} \\ \mathbf{x} \perp \mathbf{v}_1}} \frac{\mathbf{x}^T \mathbf{H} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \\ \lambda_n^{(H)} &= \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{H} \mathbf{x} = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T \mathbf{H} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}.\end{aligned}$$

We will not include the proof of the Courant-Fischer Formula here. The interested reader is referred to [19].

Let \mathbf{L}_{sym} be the normalized graph Laplacian matrix of G_s with eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_s}$ and corresponding eigenvectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{n_s}$. According to Proposition 3 of [46], \mathbf{L}_{sym} has n_s non-negative eigenvalues. As $\mathbf{L}_{\text{sym}}(\mathbf{D}_s^{-\frac{1}{2}} \mathbf{e}) = \mathbf{0}$ where \mathbf{e} is the vector of all ones, we have $\lambda_1 = 0$,

$$\mathbf{q}_1 = \frac{\mathbf{D}_s^{-\frac{1}{2}} \mathbf{e}}{\|\mathbf{D}_s^{-\frac{1}{2}} \mathbf{e}\|_2}.$$

By Theorem A.1, we have

$$\lambda_2 = \min_{\substack{\mathbf{x} \neq \mathbf{0} \\ \mathbf{x} \perp \mathbf{q}_1}} \frac{\mathbf{x}^T \mathbf{L}_{\text{sym}} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \min_{\substack{\mathbf{z} \neq \mathbf{0} \\ \mathbf{z} \perp \mathbf{D}_s \mathbf{e}}} \frac{\mathbf{z}^T \mathbf{L} \mathbf{z}}{\mathbf{z}^T \mathbf{D}_s \mathbf{z}} = \min_{\substack{\mathbf{z} \neq \mathbf{0} \\ \mathbf{z} \perp \mathbf{D}_s \mathbf{e}}} \frac{\sum_{i \sim j} (z_i - z_j)^2}{\sum_i d_i z_i^2}, \quad (14)$$

where $\mathbf{z} = \mathbf{D}_s^{-\frac{1}{2}} \mathbf{x}$, d_i is the degree of the i th node and $\sum_{i \sim j}$ denotes the sum over all unordered pairs $\{i, j\}$ for which i and j are adjacent.

Proof of Theorem 4.1. Let $\mathbf{z} = \mathbf{y} - \sigma \mathbf{e}$, where $\mathbf{y} \in \{0, 1\}^{n_s \times 1}$ is a binary indicator vector representing community C in graph G_s , \mathbf{e} the vector of all ones, and $\sigma = \frac{\text{vol}(C)}{\text{vol}(V_s)}$.

We can check that $\mathbf{z} \perp \mathbf{D}_s \mathbf{e}$:

$$\mathbf{z}^T \mathbf{D}_s \mathbf{e} = \mathbf{y}^T \mathbf{D}_s \mathbf{e} - \sigma \mathbf{e}^T \mathbf{D}_s \mathbf{e} = \text{vol}(C) - \frac{\text{vol}(C)}{\text{vol}(V_s)} \text{vol}(V_s) = 0.$$

We also know

$$\mathbf{z}^T \mathbf{L} \mathbf{z} = (\mathbf{y} - \sigma \mathbf{e})^T \mathbf{L} (\mathbf{y} - \sigma \mathbf{e}) = \mathbf{y}^T \mathbf{L} \mathbf{y}.$$

It remains to compute

$$\begin{aligned} \mathbf{z}^T \mathbf{D}_s \mathbf{z} &= (\mathbf{y} - \sigma \mathbf{e})^T \mathbf{D}_s (\mathbf{y} - \sigma \mathbf{e}) \\ &= \mathbf{y}^T \mathbf{D}_s \mathbf{y} - 2\sigma \mathbf{y}^T \mathbf{D}_s \mathbf{e} + \sigma^2 \mathbf{e}^T \mathbf{D}_s \mathbf{e} \\ &= \text{vol}(C) - 2\sigma \text{vol}(C) + \sigma^2 \text{vol}(V_s) \\ &= \frac{\text{vol}(C) \text{vol}(V_s - C)}{\text{vol}(V_s)}. \end{aligned}$$

By Eq. (14), we have

$$\lambda_2 \leq \frac{\mathbf{z}^T \mathbf{L} \mathbf{z}}{\mathbf{z}^T \mathbf{D}_s \mathbf{z}} = \frac{\mathbf{y}^T \mathbf{L} \mathbf{y} \cdot \text{vol}(V_s)}{\text{vol}(C) \text{vol}(V_s - C)}. \quad (15)$$

As the larger value of $\text{vol}(C)$ and $\text{vol}(V_s - C)$ is at least half of $\text{vol}(V_s)$,

$$\lambda_2 \leq 2 \frac{\mathbf{y}^T \mathbf{L} \mathbf{y}}{\min(\text{vol}(C), \text{vol}(V_s - C))} = 2 \frac{\mathbf{y}^T \mathbf{L} \mathbf{y}}{\text{vol}(C)} = 2 \frac{\mathbf{y}^T \mathbf{L} \mathbf{y}}{\mathbf{y}^T \mathbf{D}_s \mathbf{y}} = 2\Phi(C).$$

Therefore,

$$\frac{\lambda_2}{2} \leq \Phi(C).$$

And it is obvious that

$$\Phi(C) = \frac{\mathbf{y}^T \mathbf{L} \mathbf{y}}{\mathbf{y}^T \mathbf{D}_s \mathbf{y}} \leq 1. \quad (16)$$

□

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (61772219, 61702473, 61572221).

REFERENCES

- [1] Emmanuel Abbe. 2017. Community detection and stochastic block models: recent developments. *arXiv preprint arXiv:1703.10146* (2017).
- [2] Bruno Abrahao, Sucheta Soundarajan, John E. Hopcroft, and Robert Kleinberg. 2014. A separability framework for analyzing community structure. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8, 1 (2014), 5.
- [3] Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466, 7307 (2010), 761–764.
- [4] Hafiz Tiomoko Ali and Romain Couillet. 2017. Improved spectral community detection in large heterogeneous networks. *The Journal of Machine Learning Research* 18, 1 (2017), 8344–8392.
- [5] Reid Andersen, Fan Chung, and Kevin Lang. 2006. Local graph partitioning using PageRank vectors. In *FOCS*. 475–486.
- [6] Reid Andersen and Kevin J Lang. 2006. Communities from seed sets. In *WWW*. ACM, 223–232.
- [7] Seung-Hee Bae, Daniel Halperin, Jevin D. West, Martin Rosvall, and Bill Howe. 2017. Scalable and Efficient Flow-Based Community Detection for Large-Scale Graph Analysis. *TKDD* 11, 3 (2017), 32:1–32:30.
- [8] RV Belfin, Piotr Bródka, et al. 2018. Overlapping community detection using superior seed set selection in social networks. *Computers & Electrical Engineering* (2018).
- [9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.
- [10] A. J. Bosch. 1986. The factorization of a square matrix into two symmetric matrices. *The American Mathematical Monthly* 93, 6 (1986), 462–464.
- [11] Jie Cao, Zhan Bu, Guangliang Gao, and Haicheng Tao. 2016. Weighted modularity optimization for crisp and fuzzy community detection in large-scale networks. *Physica A: Statistical Mechanics and its Applications* 462 (2016), 386–395.

- [12] Jinxin Cao, Di Jin, Liang Yang, and Jianwu Dang. 2018. Incorporating network structure with node contents for community detection on large networks using deep learning. *Neurocomputing* 297 (2018), 71–81.
- [13] Fun Chung. 1997. *Spectral graph theory*. American Mathematical Soc.
- [14] Fan Chung. 2007. The heat kernel as the PageRank of a graph. *PNAS* 104, 50 (2007), 19735–19740.
- [15] Fun Chung. 2009. A local graph partitioning algorithm using heat kernel PageRank. *Internet Mathematics* 6, 3 (2009), 315–330.
- [16] Fan Chung and Olivia Simpson. 2013. Solving linear systems with boundary conditions using heat kernel PageRank. In *Algorithms and Models for the Web Graph (WAW)*. 203–219.
- [17] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. 2012. Demon: a local-first discovery method for overlapping communities. In *KDD*. ACM, 615–623.
- [18] Santo Fortunato and Claudio Castellano. 2012. Community Structure in Graphs. *Computational Complexity* (2012), 490–512.
- [19] Gene H. Golub and Charles F. Van Loan. 1996. *Matrix computations* (3. ed.). Johns Hopkins University Press.
- [20] Yu Han and Jie Tang. 2015. Probabilistic community and role model for social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 407–416.
- [21] Dongxiao He, Xinxin You, Zhiyong Feng, Di Jin, Xue Yang, and Weixiong Zhang. 2018. A Network-Specific Markov Random Field Approach to Community Detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- [22] Kun He, Yingru Li, Sucheta Soundarajan, and John E. Hopcroft. 2018. Hidden community detection in social networks. *Inf. Sci.* 425 (2018), 92–106.
- [23] Kun He, Pan Shi, John E Hopcroft, and David Bindel. 2016. Local spectral diffusion for robust community detection. In *Twelfth Workshop on Mining and Learning with Graphs*.
- [24] Kun He, Yiwei Sun, David Bindel, John E. Hopcroft, and Yixuan Li. 2015. Detecting overlapping communities from local spectral subspaces. In *ICDM*. 769–774.
- [25] Lucas G. S. Jeub, Prakash Balachandran, Mason A. Porter, Peter J. Mucha, and Michael W. Mahoney. 2015. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E* 91, 1 (2015), 012821.
- [26] Ravi Kannan, Santosh Vempala, and Adrian Vetta. 2000. On clusterings - good, bad and spectral. In *FOCS*. 367–377.
- [27] Kyle Kloster and David F. Gleich. 2014. Heat kernel based community detection. In *KDD*. ACM, 1386–1395.
- [28] Isabel M. Kloumann and Jon M. Kleinberg. 2014. Community membership identification from small seed sets. In *KDD*. ACM, 1366–1375.
- [29] 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 (2009), 016118.
- [30] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Physical Review E* 78, 4 (2008), 046110.
- [31] Andrea Lancichinetti, Filippo Radicchi, Jose J. Ramasco, and Santo Fortunato. 2011. Finding statistically significant communities in networks. *PLoS ONE* 6, 4 (2011), e18961.
- [32] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 2008. Statistical properties of community structure in large social and information networks. In *WWW*. 695–704.
- [33] Yixuan Li, Kun He, David Bindel, and John E. Hopcroft. 2015. Uncovering the small community structure in large networks. In *WWW*. 658–668.
- [34] Yixuan Li, Kun He, Kyle Kloster, David Bindel, and John E. Hopcroft. 2018. Local Spectral Clustering for Overlapping Community Detection. *TKDD* 12, 2 (2018), 17:1–17:27.
- [35] Michael W. Mahoney, Lorenzo Orecchia, and Nisheeth K. Vishnoi. 2012. A local spectral method for graphs: with applications to improving graph partitions and exploring data graphs locally. *The Journal of Machine Learning Research* 13, 1 (2012), 2339–2365.
- [36] Mark EJ Newman. 2004. Fast algorithm for detecting community structure in networks. *Physical review E* 69, 6 (2004), 066133.
- [37] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582.
- [38] M. E. J. Newman. 2013. Spectral methods for network community detection and graph partitioning. *Physical Review E* 88, 4 (2013), 042822.
- [39] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435, 7043 (2005), 814–818.
- [40] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. 2012. Community detection in social media. *Data Mining and Knowledge Discovery* 24, 3 (2012), 515–554.
- [41] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.

- [42] Pan Shi, Kun He, David Bindel, and John E Hopcroft. 2017. Local lanczos spectral approximation for community detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 651–667.
- [43] Sucheta Soundarajan and John E. Hopcroft. 2015. Use of Local Group Information to Identify Communities in Networks. *TKDD* 9, 3 (2015), 21:1–21:27.
- [44] Danial A. Spielman and Shanghua Teng. 2004. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*. 81–90.
- [45] Twan van Laarhoven and Elena Marchiori. 2016. Local network community detection with continuous optimization of conductance and weighted kernel K-means. *The Journal of Machine Learning Research* 17 (2016), 5148–5175.
- [46] Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17, 4 (2007), 395–416.
- [47] Ingmar Weber, Venkata R. Kiran Garimella, and Alaa Batayneh. 2013. Secular vs. Islamist polarization in Egypt on Twitter. In *ASONAM*. 290–297.
- [48] Joyce J. Whang, David F. Gleich, and Inderjit S. Dhillon. 2013. Overlapping Community Detection Using Seed Set Expansion. In *CIKM*. 2099–2108.
- [49] Joyce Jiyoung Whang, David F. Gleich, and Inderjit S. Dhillon. 2016. Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion. *IEEE Trans. Knowl. Data Eng.* 28, 5 (2016), 1272–1284.
- [50] Yubao Wu, Ruoming Jin, Jing Li, and Xiang Zhang. 2015. Robust local community detection: on free rider effect and its elimination. In *VLDB*. 798–809.
- [51] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. 2013. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (CSUR)* 45, 4 (2013), 43.
- [52] Jaewon Yang and Jure Leskovec. 2012. Defining and Evaluating Network Communities based on Ground-truth. In *ICDM*. 745–754.
- [53] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 587–596.
- [54] Shihua Zhang, Rui-Sheng Wang, and Xiang-Sun Zhang. 2007. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications* 374, 1 (2007), 483–490.
- [55] Yu Zhang and Dit-Yan Yeung. 2012. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 606–614.