Detecting Overlapping Communities from Local Spectral Subspaces

Kun He, Yiwei Sun
Huazhong University of Science and Technology
Wuhan 430074, China
Email: {brooklet60, yiweisun}@hust.edu.cn

David Bindel, John Hopcroft, Yixuan Li
Cornell University
Ithaca 14850, NY, USA
Email: {bindel, jeh, yli}@cs.cornell.edu

***Abstract*—Based on the definition of local spectral subspace, we propose a novel approach called *LOSP* for local overlapping community detection. Using the power method for a few steps, *LOSP* finds an approximate invariant subspace, which depicts the embedding of the local neighborhood structure around the seeds of interest. *LOSP* then identifies the local community expanded from the given seeds by seeking a sparse indicator vector in the subspace where the seeds are in its support.**

We provide a systematic investigation on *LOSP*, and thoroughly evaluate it on large real world networks across multiple domains. With the prior information of very few seed members, *LOSP* can detect the remaining members of a target community with high accuracy. Experiments demonstrate that *LOSP* outperforms the Heat Kernel and PageRank diffusions. Using *LOSP* as a subroutine, we further address the problem of *multiple membership identification*, which aims to find all the communities a single vertex belongs to. High F1 scores are achieved in detecting multiple local communities with respect to arbitrary single seed for various large real world networks.

I. Introduction

In recent years, there has been a growing interest in finding the local community structure in large networks [8]–[10], [16], [17]. Several seed set expansion based approaches have been demonstrated to be effective in identifying local community [9], [10], [15]. Starting from a few members shared by some domain expert, these algorithms attempt to uncover the remaining latent members. These known members are usually referred to as *seeds* in the literature, and the process of gradually growing the seed set into a larger set until the target community is revealed is called *seed set expansion*.

Seed set expansion can be applied to many real world scenarios. For instance, in political participation networks, one might discover a large political group from a small set of representative politicians [5]. In product co-purchase networks, sales websites may recommend potential products to the customer by expanding from a few purchased products. In biological networks representing the interactions among genes, biologists are likely to discover a set of genes that form a functionally similar unit starting from a few observed and well-studied genes.

Variants of random walk techniques have been extensively adopted as a subroutine for conducting seed set expansion [2], which can be seen as a localized graph diffusion process where the probability is distributed from the initial seeds to the surroundings. Most commonly seen approaches can be categorized into two types – PageRank-like diffusion [1], [9], [17], and Heat Kernel graph diffusion [3], [4], [8] – both of which will be discussed in detail below.

Spielman and Teng [13] use degree-normalized, personalized PageRank (DN PageRank) with respect to the start seed and truncate on small PageRank probability values, leading to the PageRank Nibble method [1]. The DN PageRank is adopted by several other PageRank-based clustering algorithms [2], [17], which are competitive with the METIS algorithm [7]. Kloumann and Kleinberg [9] evaluate different variations of PageRank, and find that the standard PageRank yields better performance than the DN PageRank.

The Heat Kernel method is another type of graph diffusion technique [3], [4], [8]. It involves the Taylor series expansion of the matrix exponential of the random walk transition matrix. Chung et al. analyze the property of this diffusion theoretically [3], and propose a randomized Monte Carlo method to estimate the diffusion [4]. Kloster et al. [8] propose a deterministic method that uses coordinate relaxation on an implicit linear system that estimates the Heat Kernel diffusion, and show that Heat Kernel outperforms the personalized PageRank with substantially higher F1 measures.

A stopping criterion for defining the community boundary is necessary for all seed set expansion methods, unless the size of the target community is known as a budget. Conductance has become a widely adopted metric in determining the boundary of a community [8], [15], [17]. For example, Yang and Leskovec [17] 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 accuracy. The Heat Kernel method [8] also adopts conductance as the stopping criterion for the local community.

Despite the success of many random walk based approaches which rely on the single probability vector after short random walks, very few have utilized the subspace formed by the random walk diffusion. Based on our previous investigation on local spectral clustering [10], we provide a systematic approach for finding small, overlapping communities using the subspace shaped by the dynamics of short random walks, which we call *LOSP (Local Spectral)*.

We first address problems concerning the local structure in large networks: How do we find a local community in time that is a function of the size of the target community rather than the size of the entire network? What constitutes a good stopping criterion when growing the seed set into a local community? How do the quality and quantity of the seed members affect

the performance of *LOSP*? We also consider *LOSP* when the domain expert poorly selects the initial seed members.

We then tackle the problem of identifying all the communities a single vertex is in, which we refer to as the *multiple membership identification* problem. Identifying all local communities is useful in numerous applications. For instance, social network users may be interested in exploring all social groups an individual is in, and biologists would like to mine all GO (Gene Ontology) terms a gene serves.

LOSP is based on the classical spectral clustering method, which starts with computing the first d eigenvectors of the graph Laplacian or some matrix associated with the network. Each row in the eigenvector matrix corresponds to the embedding of a vertex in a d-dimensional space. These vectors are then clustered using k-means, resulting in disjoint communities. We make two fundamental changes:

1) **Define the local spectral subspace**. We calculate the first few eigenvectors by the power method, but instead of iterating to convergence, we iterate for a few steps such that the probability distribution of a random walk starting from the seeds reaches the local community but does not spread to the entire graph.

2) **Handle the overlapping situation**. Instead of clustering the row vectors in the d-dimensional subspace, we seek a minimum ℓ_1 norm indicator vector in the local spectral subspace where the initial seeds are in its support. Overlapping communities correspond to different seed sets.

We thoroughly analyze variations of *LOSP* and explain the intuition behind the method. Five community scoring functions are evaluated, including our newly proposed definitions of triad-participation-number (TPN) and normalized modularity. *LOSP* is robust with little fluctuations under different scoring functions, where conductance and TPN yield the best performance. We also investigate the structural properties of different seed sets, and find that low degree seeds and random seeds are essentially the same for real world networks in finding the local structure. The performance evaluation on various real world networks shows that *LOSP* outperforms the prevalent PageRank and Heat Kernel diffusion methods, and serves as a good subroutine for multiple membership identification.

II. Preliminaries

A. Problem Statement

Let $G = (V, E)$ be a connected, undirected graph with n vertices and m edges, and let $\mathcal{C}$ be a set of labeled communities. We address two basic questions: (1) Given a few exemplary members S in a target community $C_k \in \mathcal{C}$ where $|C_k| \ll n$, how to identify the remaining latent members in C_k? (2) How can we identify all the communities a single seed s belongs to?

B. Community Scoring Functions

To mathematically characterize a community-like subset of vertices, we adopt three commonly used scoring functions, modularity (Mod), conductance (Cond) and triad-participation-ratio (TPR). Additionally, we propose two new scoring functions, normalized modularity (nMod) and triad-participation-number (TPN).

Let c be the number of communities. Let n_k and e_{kk} be the number of vertices and edges within a community C_k, and d_k be the total degree of the internal vertices. The *modularity* [11] Q_k of a community C_k is $Q_k = \frac{e_{kk}}{m} - \left(\frac{d_k}{2m}\right)^2$.

Instead of using "minus" to define the modularity, we use "divide" to define the *normalized modularity* D. Since the coefficient $4m$ is a constant when evaluating in the same network, we simply define $D_k = \frac{e_{kk}}{d_k^2}$, which is insensitive to the network scale. We let $D_k = 0$ when there is only one isolated vertex in C_k.

Conductance measures the fraction of edges leaving the community [12]. When the community contains less than half the total edges in the network, the conductance of a community C_k is defined by $\Phi_k = \frac{d_k - 2e_{kk}}{d_k} = 1 - 2\frac{e_{kk}}{d_k}$.

Triad-Participation-Ratio (*TPR*) is the fraction of vertices in the community that belong to triads [17]. Based on TPR, we define another scoring function *Triad-Participation-Number* (*TPN*) as the average number of triads a vertex belongs to. TPR and TPN are based on the internal connectivity, while modularity, normalized modularity and conductance are based on both the internal and external connectivity.

C. Datasets

We consider five real-world network datasets available from the SNAP website[1].For each network, we adopt the top 5000 annotated communities with the highest quality evaluated with several metrics by Yang & Leskovec [17]. Table 1 summarizes the statistics on the datasets and the ground truth communities ($\mathcal{D}_{90\%}$ indicates 90-percentile effective diameter).

Networks				Ground truth communities	
Name	**#Vertices**	**#Edges**	$\mathcal{D}_{90\%}$	**Avg. ± Std. Size**	**Avg. Cond.**
Amazon	334,863	925,872	15.0	13.49 ± 17.51	0.07
DBLP	317,080	1,049,866	8.0	22.44 ± 201.08	0.41
LiveJ	3,997,962	34,681,189	6.5	27.80 ± 58.04	0.39
YouTube	1,134,890	2,987,624	6.5	14.59 ± 60.46	0.80
Orkut	3,072,441	117,185,083	4.8	215.72 ± 320.55	0.73

TABLE I. **Statistics for real networks and their labeled communities.**

D. Evaluation via Ground Truth

We adopt precision, recall and F1 score to measure how close the community C expanded from a seed set S is to the target ground truth community T containing S. The precision and recall are defined as:

$$P(C,T) = \frac{|C \cap T|}{|C|}, R(C,T) = \frac{|C \cap T|}{|T|}.$$

The F1 score is the harmonic mean of precision and recall:

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

III. The Local Spectral Subspace

Let $\mathbf{A}$ be the adjacency matrix, and d the vector of vertex degrees. Let $\mathbf{D} = \mathrm{diag}(d)$ denote the diagonal matrix of degrees. Let $\mathbf{N_{rw}} = \mathbf{D}^{-1}\mathbf{A}$ be the transition matrix, and $\mathbf{N_{sym}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ the normalized adjacency matrix. $\mathbf{V_d} \in \mathbb{R}^{n \times d}$ indicates a matrix containing the first d eigenvectors of $\mathbf{N_{rw}}$ or $\mathbf{N_{sym}}$ as the columns.

[1] http://snap.stanford.edu

A. Defining the Local Spectral Subspace

Consider a short random walk starting from the known seed members. We approximate the first d eigenvectors to characterize the embedding of the local network structure surrounding the seeds.

Let $\mathbf{p^{(t)}}$ be a column vector specifying the probability mass of each vertex at step t. We define a basis for a local approximately-invariant subspace as follows:

1) The initial probability $\mathbf{p^{(1)}}$ is assigned by evenly distributing the probability among the seed members.
2) Conduct $d-1$ steps of the random walk $\mathbf{N_{rw}^T p^{(t)}} = \mathbf{p^{(t+1)}}$ to get the span of d successive probability vectors. Find their orthonormal basis $\mathbf{V_d^{(0)}}$, the initial invariant subspace approximation:
$$\mathbf{V_d^{(0)}} = \mathrm{orth}([\mathbf{p^{(1)}}, \mathbf{p^{(2)}}, ..., \mathbf{p^{(d)}}]).$$
3) Compute the orthonormal basis $\mathbf{V_d^{(k)}}$ by the subspace iteration:
$$\mathbf{V_d^{(k)}} = \mathrm{orth}(\mathbf{N_{rw}^T V_d^{(k-1)}}).$$

Here d and k are some modest parameters empirically determined in Section 4.3. The orthonormal basis $\mathbf{V_d^{(k)}}$ is what we call the *local spectral basis* and the subspace spanned by the basis is called the *local spectral subspace*.

B. Finding Local Overlapping Communities

With the local spectral basis $\mathbf{V_d^{(k)}}$, we look for row vectors in the subspace that are nearly in the same direction as the seeds. More precisely, we solve the following linear programming problem LP_1:

$$\min \quad |\mathbf{y}|_1 = \sum_{i=1}^{n} y_i$$
$$\textit{s.t. } (1)\ \exists \mathbf{x} \textit{ s.t. } \mathbf{y} = \mathbf{V_d^{(k)}}\mathbf{x},\ (2)\ \mathbf{y} \geq 0,\ (3)\ \mathbf{s}^T\mathbf{y} \geq 1.$$

Constraint (1) requires $\mathbf{y}$ be in the span of $\mathbf{V_d^{(k)}}$, and can be rewritten as $[\mathbf{V_d^{(k)}}, -\mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{0}$. Constraint (2) requires $\mathbf{y} \geq \mathbf{0}$ where y_i indicates the likelihood that vertex i belongs to the target community. Constraint (3) enforces that the seeds be in the support of sparse vector $\mathbf{y}$ where s is an indicator vector for the seed set.

We solve LP_1 and sort vertices according to their random walk probability scores in $\mathbf{y}$ in decreasing order. Then we select the top $\hat{n}$ vertices with the highest probabilities as the resulting community. One could simply select the top $|C|$ vertices if the size of the target community C is known as a budget, or use heuristics provided in Section 4.3 to determine $\hat{n}$ automatically.

C. An Alternative Definition of the Local Spectral Subspace

An alternative way to define the local spectral subspace is to use the normalized adjacency matrix $\mathbf{N_{sym}}$ instead of $\mathbf{N_{rw}}$. As $\mathbf{N_{sym}}$ is symmetric, the two matrices share the same set of real eigenvalues, with the eigenvectors of $\mathbf{N_{sym}}$ scaled by $\mathbf{D^{1/2}}$. Note that $(\mathbf{N_{sym}^T})^\mathbf{k} = \mathbf{D^{-1/2}}(\mathbf{N_{rw}^T})^\mathbf{k}\mathbf{D^{1/2}}$, and
$$(\mathbf{N_{sym}^T})^\mathbf{k}\mathbf{D^{-1/2}}\mathbf{p^{(1)}} = \mathbf{D^{-1/2}}(\mathbf{N_{rw}^T})^\mathbf{k}\mathbf{p^{(1)}}$$
for an initial probability $\mathbf{p^{(1)}}$. Thus the sequence of vectors generated with the random walk matrix is closely related to the sequence generated by the symmetrized matrix.

We know π where $\pi_i = d_i/2m$ indicates the stationary probability for the transition matrix $\mathbf{N_{rw}}$, i.e. $\mathbf{N_{rw}^T}\pi = \pi$. As
$$\mathbf{N_{sym}^T D^{-1/2}}\pi = \mathbf{D^{-1/2}N_{rw}^T}\pi = \mathbf{D^{-1/2}}\pi,$$
in the alternative definition of local spectral subspace, we defined another random walk by $\mathbf{p^{t+1}} = \mathrm{norm}(\mathbf{N_{sym}^T p^t})$, which converges to the stationary distribution π' where $\pi'_i = \frac{\sqrt{d_i}}{\sum \sqrt{d_i}}$.

IV. The Proposed Algorithm

The entire local spectral (*LOSP*) algorithm consists of several procedures that we describe in detail below.

A. Local Sampling

The 90-percentile effective diameter $\mathcal{D}_{90\%}$ in Table 1 shows that there is a small world phenomenon for most vertices. We apply a local sampling by breadth-first-search (BFS) to get a small subgraph covering most of the neighborhood vertices.

For each vertex in the seed set S, we first conduct a 2-step BFS (3-step BFS on Amazon due to its larger $\mathcal{D}_{90\%}$). Consider the set of the frontier vertices, and remove vertices whose outdegree is greater than 1000. Sort the remainder vertices in decreasing order according to their inward ratio (the fraction of inward edges to the BFS subgraph), and remove vertices after the first 1000. Then union the BFS subgraphs obtained from each seed.

For a set S with three random seeds, we generally produce a sampling subgraph of size thousands. The coverage ratio, defined as the fraction of ground truth vertices covered by the sampled network, can reach 90% to 99%, except on Orkut, where the ground truth communities are much larger.

The subnetwork is extracted in time $\Theta(d_{avg}^r|S|)$, where r is the number of BFS levels, d_{avg} is the average degree of the component, and $|S|$ is the initial number of seeds. As subsequent steps apply only to the sampled network, the complexity is reduced to time polynomial in the size of the sampled network.

B. Strengthen the Initial Seed Set

Larger seed sets can improve accuracy [9], but often few seeds are available. To compensate for the shortage of seeds, we provide a method to strengthen the initial seed set: find a shortest path for each pair of vertices, and add vertices on the path to the initial seed set if the length is no greater than a small number l. The intuition is that any two seeds in the same community must be related for some reason. They connect with each other either via a direct link or via some other intermediate vertices. In the latter case, those intermediate vertices bridging the seeds are very likely to be in the target community because they serve as the relational "relay" in order for the seeds to be in the same community. The shorter the path is, the more likely they are in the same community. We experimented with several values of l including $l = 1$, and find that $l = 4$ generally yields the best results overall datasets.

C. Initial Membership Identification

We use the strengthened initial seed set S as the input to the linear programming problem LP_1 to find an initial community. Three key parameters need to be determined.

Dimension of the subspace *d*. We observe that d is related to the number of local structures around the seeds. We calculate the overlapping membership om, the number of communities a vertex is in, for vertices belonging to at least one ground truth. om is as low as 1.2 for DBLP and is around 1.5 for other datasets. As there may be some other local structures out of the annotation, we choose $d = 3$, assuming there are three local structures on average. Our experiments also show that $d = 3$ yields the full potential.

Steps of the random walk *k*. k is related to the size and conductance of the target community. A larger or sparser community usually needs more steps to spread the information, and the lower conductance serves as a better bottleneck to prevent probability from leaking out. Experiments show the detection accuracy plateaus as k increases and $k = 3$ yields the full potential.

Boundary of the local community. As the global minimum of scoring functions might produce "cavemen-type" communities [6], we use a local optimum to decide the community size. For each set S_i consisting of the i vertices with the highest probability scores, we evaluate the quality via a candidate scoring function f. We truncate at the first local optimum of $f(S_i)$ to obtain the extracted community.

In minimizing a scoring function such as conductance, we increase the index i in the sorted $\mathbf{y}$ to find the first minimum. Since the function does not smoothly decrease to the minimum, we search for indices n_0 and $\hat{n}$ ($n_0 < \hat{n}$), such that function f starts increasing at $\hat{n}$ and the drop from $f(n_0)$ to $f(\hat{n})$ satisfies $f(n_0) \geq \gamma f(\hat{n})$. Experiments show that $\gamma = 1.7$ yields good results across all the datasets.

D. Fine-tuning the Local Community

After the initial membership identification, we further improve the detection quality by iteratively augmenting the seed set with the top elements in the sorted $\mathbf{y}$, and use the enlarged seed set to fine-tune the community. The intuition is that the top elements with high probabilities are likely to be in the target community, and augmenting the initial seed set would improve the detection accuracy by providing with more domain information.

Let the initial seed set $S_0 = S$. At each iteration t, we enlarge the current seed set S_t by combining with the top $|S| + \delta \cdot t$ candidate seeds[2]. Define the weight of each initial seed in S to be $w_1 = 1/|S|$, and the weight of each augmented seed in S_t/S_0 to be $w_2 = 0.5w_1$. Then feed the expanded seed set to the modified problem LP_2 by adding one more constraint on LP_1:

$$(4)\ \mathbf{s_t}^T\mathbf{y} \geq 1 + w_2 * (|S_t| - |S|)$$

where $\mathbf{s_t} \in \{0, 1\}^n$ is a binary indicator vector for the current seed set S_t. We halve the weight of the expanded seeds such that the initial members play a key role for the identification.

To complete the process, we track the value of the scoring function on the extracted community, and stop the iteration when the community quality starts to decline. Experiments show that the F1 scores increase considerably by 0.12 on average over the five real world networks.

V. Multiple Membership Identification

As individuals tend to be in multiple communities simultaneously, we further address the problem of finding the number of communities to which an individual is in and identifying those communities.

Regard the individual seed s as an "ego", and temporarily remove s from its ego network to get connected components sorted by their sizes in decreasing order, namely S_1, S_2, ..., S_q. Each $\{s\} \cup S_i$ is regarded as a candidate initial seed set. Iteratively start from each initial seed set, remove edges connecting s to other ego neighbors, and identify the corresponding community by using *LOSP* as a subroutine. Seed sets completely contained in previous communities will not be processed. In this way, we find a set of local overlapping communities for a single seed. The first is the community the seed is mostly attached to.

A crucial step is that we cut edges connecting s to other candidate seed sets so as to weaken the interference of different seed sets. If a candidate seed set still has very strong connections to the current target structure, it will be completely covered by the extracted community and will then be removed from the candidate seed list.

VI. Experimental Results

We implement *LOSP* in Matlab [3]and compare with state-of-the-art local community detection algorithms. For each of the datasets, we randomly select 500 annotated communities, and randomly pick one or three exemplary seeds from each target community. The mean and standard deviation of F1 scores over all trials are used for the evaluation and comparison.

A. Evaluation on LOSP

In this subsection, we investigate how the performance of *LOSP* is affected by the choice of: 1) graph Laplacian matrix, 2) community scoring function, 3) seed set structure, and 4) seed set size. We use the ground truth size in evaluating 1), 3), and 4) to factor out the impact of stopping criteria.

Normalized Matrices. $\mathbf{N_{rw}}$ and $\mathbf{N_{sym}}$ are related to the normalized graph Laplacian $\mathbf{L_{rw}} = \mathbf{I} - \mathbf{N_{rw}}$ and $\mathbf{L_{sym}} = \mathbf{I} - \mathbf{N_{sym}}$. In the classical spectral method, as the degrees are very broadly distributed in most real world networks, it is preferable to use normalized spectral clustering, and in the normalized case to use $\mathbf{L_{rw}}$ rather than $\mathbf{L_{sym}}$ [14]. Table 2 shows the average F1 scores using the two different local spectral subspaces defined by $\mathbf{N_{rw}}$ and $\mathbf{N_{sym}}$, respectively. $\mathbf{N_{rw}}$ performs better than $\mathbf{N_{sym}}$ on average.

Community Scoring Functions. Different community scoring functions can be adapted to *LOSP* in determining the local community structure. Fig. 1 shows the average F1 scores and standard deviations of the resulting communities with

[2]We expand the candidates by $\delta = 5$ at each iteration in the experiments.

[3]https://github.com/KunHe2015/LOSP/

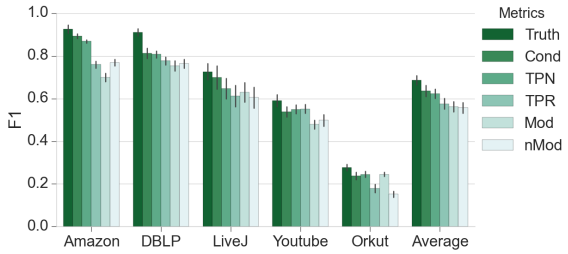


Fig. 1. **Evaluation on different community scoring functions.**

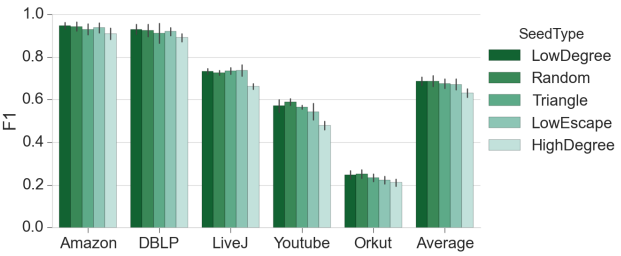


Fig. 2. **Evaluation on different seed structure.**

	Amazon	DBLP	LiveJ	YouTube	Orkut	Avg.
$\mathbf{N_{rw}}$	0.938	0.911	0.726	0.591	0.261	0.685
$\mathbf{N_{sym}}$	0.920	0.845	0.751	0.531	0.277	0.659

TABLE II. **Evaluation of LOSP with different Laplacian matrices.**

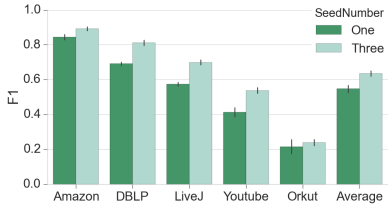


Fig. 3. **Evaluation on different seed set size (with Cond. scoring function).**

different scoring functions. For reference, "Truth" is obtained using the ground truth community size. *LOSP* is robust with low variance in general. Conductance and TPN consistently outperform other metrics. Note that we work with the sampled subgraph when calculating the scoring functions for candidate communities, and all the metrics are calculated within the scope of sampled graph.

Seed Structure. To understand how the structure of the initial seed set affects the accuracy of the resulting community, we quantitatively evaluate the performance of five different seeding strategies. *Random seeding* randomly picks $|S|$ vertices from the target community C. *High (low) degree seeding* randomly picks $|S|$ vertices with degree ranked in the top (bottom) one third. *High triangle participation seeding* sorts vertices according to the number of triangles they belong to in C, and randomly picks seeds ranked in the top one third. *Low escape seeding* sorts vertices basing on the probability reserved after a short random walk starting from each vertex, and randomly picks seeds in the top one third.

Fig. 2 shows that low degree, random, triangle (high triangle participation) and low escape yield almost the same accuracy on average. Due to the power law degree distribution, most of the vertices are of low degree. And this explains why low degree seeding and random seeding yield almost the same performance. Low degree seeds spread out the probabilities slowly and better preserve the local structure. High triangle participation seeds and low escape seeds follow another philosophy in that they choose seeds more cohesive to the target community, resulting in high quality output.

High degree seeds are inferior to the previous four seed structures in that the probabilities spread out very quickly from these popular individuals, causing less accuracy. Also, high degree vertices behave as the "hubs" while low degree vertices have more loyalty to the target community.

Seed Set Size. Fig. 3 shows the average F1 scores of the communities by starting from one or three random seed(s). Due to space limitation, we only show the result with conductance as the scoring functions. We see that *LOSP* is able to achieve high accuracy even with a single seed.

B. Comparisons with Local Algorithms

We compare *LOSP* with the state-of-the-art local community detection algorithms. The results of applying the best two scoring functions, TPN and conductance, are reported.

We randomly pick one seed from each target community, and compare the average F1 scores with Heat Kernel[4] [8] under the same condition. The authors compared their results with pprpush [1], and Heat Kernel far outperforms pprpush by almost doubling the F1 scores. We then randomly pick three seeds from each target community, and compare *LOSP* with LEMON [10] that reports result using conductance as scoring function and starts from three random seeds. Table 3 shows that *LOSP* apparently outperforms Heat Kernel and LEMON. Fig. 4 illustrates examples of the detected local communities starting from three random seeds.

Algorithm	Amazon	DBLP	LiveJ	YouTube	Orkut	Avg.
one seed:						
LOSP_Truth	0.864	0.734	0.686	0.476	0.247	0.601
LOSP_Cond	0.845	0.691	0.674	0.413	0.216	0.568
LOSP_TPN	0.722	0.686	0.669	0.406	0.224	0.542
HeatKernel	0.712	0.378	0.553	0.098	0.320	0.412
three seeds:						
LOSP_Truth	0.938	0.911	0.726	0.591	0.261	0.685
LOSP_Cond	0.893	0.812	0.699	0.538	0.234	0.635
LOSP_TPN	0.868	0.807	0.646	0.550	0.231	0.620
LEMON_Cond	0.910	0.525	-	0.190	0.170	-

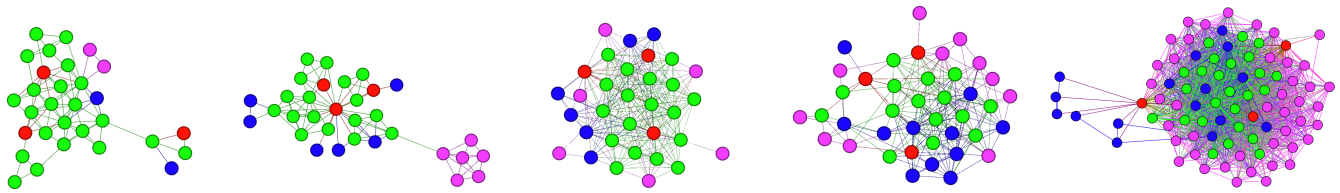
TABLE III. **Comparison with local algorithms.**

C. All Local Membership Identification

To reveal all communities to which a single seed s belongs, we group the vertices in the ground truth according to their overlapping memberships om, randomly pick 500 vertices in each group (pick all if the vertices in a group is less than 500) and find all their membership communities.

We preprocess the ground truth by removing identical copies of ground truth communities, and obtain 1517, 4959, 4703, 4771 and 4885 ground truth communities for the five datasets. The ground truth communities of Amazon form a typical hierarchical dendrogram, those of LiveJ also reveal hierarchical structure. Others form overlapping relationships.

[4]https://gist.github.com/dgleich/cf170a226aa848240cf4. We choose the best variation hk-relax.

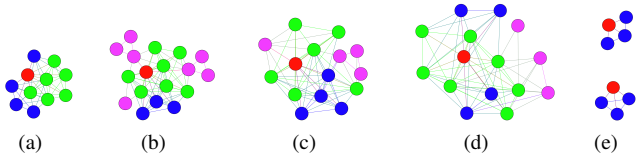


(a) Amazon (b) DBLP (c) LiveJ (d) Youtube (e) Orkut

Fig. 4. **An example of the detected communities from three random seeds (the red). The green are the intersection of detected community and the ground truth, while the pink and blue are the remainders of the ground truth and false positives.**

Datasets	om	1	2	3	4	5	Avg.
Amazon	Truth	0.850	0.756	0.661	0.583	0.439	0.659
	Cond	0.811	0.720	0.580	0.501	0.408	0.604
	TPN	0.824	0.746	0.600	0.506	0.382	0.616
DBLP	Truth	0.753	0.650	0.612	0.558	0.513	0.6172
	Cond	0.740	0.624	0.579	0.509	0.448	0.580
	TPN	0.694	0.616	0.592	0.489	0.434	0.565
LiveJ	Truth	0.671	0.530	0.435	0.287	0.180	0.421
	Cond	0.601	0.445	0.356	0.225	0.166	0.359
	TPN	0.591	0.45	0.325	0.229	0.127	0.345
Youtube	Truth	0.411	0.377	0.313	0.226	0.163	0.298
	Cond	0.429	0.340	0.254	0.136	0.132	0.258
	TPN	0.389	0.345	0.236	0.151	0.120	0.248
Orkut	Truth	0.315	0.251	0.201	0.142	0.097	0.201
	Cond	0.224	0.166	0.180	0.135	0.09	0.160
	TPN	0.214	0.166	0.176	0.129	0.090	0.155
All	Avg.	0.568	0.479	0.407	0.320	0.253	0.405

TABLE IV. **F1 scores for multiple membership identification.**



(a) (b) (c) (d) (e)

Fig. 5. **An example of all local membership identification. We find six small communities. (a) to (d) are related to the four ground truth.**

Table 4 shows the average F1 scores of $om \in \{1, 2, 3, 4, 5\}$ by using ground truth size or the best two scoring functions. The average F1 scores decline for larger om, coinciding with the intuition that multiple memberships make the detection task harder. On average, we achieve a high F1 score of 0.40 over all datasets and different community boundary strategies. Fig. 5 illustrates all the detected local communities of an exemplary vertex in four annotated communities from DBLP. Note that our method is different from sweeping the function curve to find multiple local minima [17], which essentially finds hierarchical communities.

VII. Conclusion

We define a local invariant subspace spanned by a few leading approximate eigenvectors of the neighborhood subgraph around the seeds, and present a systematic and effective approach called *LOSP* for finding local overlapping communities. We defined two new community scoring functions, triad-participation-number (TPN) and normalized modularity, for the stopping rules of community boundary. The structural properties of different seed sets are discussed.

We demonstrate that *LOSP* outperforms the state-of-the-art local diffusion methods in real world networks across multiple domains, and reduces the complexity by running in time and space polynomial in the scale of the local structure. We further effectively find all overlapping local communities for a single vertex by applying *LOSP* as a subroutine.

Acknowledgment

Supported by US Army Research Office W911NF-14-1-0477, NSF of China (61472147) and Hubei (2015CFB566).

References

[1] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *FOCS*, pages 475–486, 2006.

[2] R. Andersen and K. J. Lang. Communities from seed sets. In *WWW*, pages 223–232. ACM, 2006.

[3] F. Chung. The heat kernel as the pagerank of a graph. *PNAS*, 104(50):19735C19740, 2007.

[4] F. Chung and O. Simpson. Solving linear systems with boundary conditions using heat kernel pagerank. *Algorithms and Models for the Web Graph*, page 203C219, 2013.

[5] V. R. K. G. Ingmar Weber and A. Batayneh. Secular vs. Islamist polarization in Egypt on twitter. In *ASONAM*, pages 290–297, 2013.

[6] U. Kang and C. Faloutsos. Beyond 'caveman communities': Hubs and spokes for graph compression and mining. In *ICDM*, pages 300–309, 2011.

[7] G. Karypis and V. Kumar. Metis-unstructured graph partitioning and sparse matrix ordering system. version 2.0, 1995.

[8] K. Kloster and D. F. Gleich. Heat kernel based community detection. In *KDD*, pages 24–27, 2014.

[9] I. M. Kloumann and J. M. Kleinberg. Community membership identification from small seed sets. In *KDD*, pages 1366–1375, 2014.

[10] Y. Li, K. He, D. Bindel, and J. Hopcroft. Uncovering the small community structure in large networks: A local spectral approach. In *WWW*, pages 658–668, 2015.

[11] M. E. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, 2000.

[13] D. A. Spielman and S. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*, pages 81–90, 2004.

[14] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[15] J. J. Whang, D. F. Gleich, and I. S. Dhillon. Overlapping community detection using seed set expansion. In *CIKM*, pages 2099–2108, 2013.

[16] Y. Wu, R. Jin, J. Li, and X. Zhang. Robust local community detection: on free rider effect and its elimination. In *VLDB*, pages 798–809, 2015.

[17] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *ICDM*, pages 10–13, 2012.