Local Lanczos Spectral Approximation for Community Detection

Pan Shi[1], Kun He[1(✉)], David Bindel[2], John E. Hopcroft[2]

[1] Huazhong University of Science and Technology, Wuhan, China
{panshi,brooklet60}@hust.edu.cn
[2] Cornell University, Ithaca, NY, USA
{bindel,jeh}@cs.cornell.edu

Abstract. We propose a novel approach called the Local Lanczos Spectral Approximation (LLSA) for identifying all latent members of a local community from very few seed members. To reduce the computation complexity, we first apply a fast heat kernel diffusing to sample a comparatively small subgraph covering almost all possible community members around the seeds. Then starting from a normalized indicator vector of the seeds and by a few steps of Lanczos iteration on the sampled subgraph, a local eigenvector is gained for approximating the eigenvector of the transition matrix with the largest eigenvalue. Elements of this local eigenvector is a relaxed indicator for the affiliation probability of the corresponding nodes to the target community. We conduct extensive experiments on real-world datasets in various domains as well as synthetic datasets. Results show that the proposed method outperforms state-of-the-art local community detection algorithms. To the best of our knowledge, this is the first work to adapt the Lanczos method for local community detection, which is natural and potentially effective. Also, we did the first attempt of using heat kernel as a sampling method instead of detecting communities directly, which is proved empirically to be very efficient and effective.

Keywords: Community detection · Heat kernel · Local Lanczos method

1 Introduction

Community detection aims to find a set of nodes in a network that are internally cohesive but comparatively separated from the remainder of the network. In social networks, community detection is a classical and challenging problem which is very useful for analyzing the topology structure and extracting information from the network, and numerous algorithms and techniques have been proposed [12,34].

Most of the researchers have focused on uncovering the global community structure [28,1,13]. With the rapid growth of the network scale, global community detection becomes very costly or even impossible for very large networks.

The big data drives researchers to shift their attention from the global structure to the local structure [16,15]. How to adapt the existing effective methods initially designed for the global community detection in order to uncover the local community structure is a natural and important approach for the accurate membership identification from a few exemplary members. Several probabilty diffusion methods, PageRank [16], heat kernel [15] and spectral subspace approximation [14,22] are three main techniques for local community detection.

The Lanczos method [21] is a classic method proposed for calculating the eigenvalues, aka the spectra of a matrix. Through there exists some work using the Lanczos method for the spectral bisection [6], unlike other spectra calculation methods, the Lanczos method is seldom used for community detection and to the best of our knowledge, it has never been used for the local community detection. In this paper, we propose a novel approach called the Local Lanczos Spectral Approximation (LLSA) for local community detection. Specifically, we execute a few steps of Lanczos iteration to attain a local eigenvector that approximates the eigenvector of the transition matrix with the largest eigenvalue. Elements of this local eigenvector is a relaxed indicator for the affiliation probability of the corresponding nodes to the target community. As compared with other spectral approxiamtion methods, the Lanczos iterative method is efficient for computing the top eigen-pairs of large sparse matrices and it is space efficient, which is very helpful for large social networks which are usually sparse.

Our contributions include: (1) To the best of our knowledge, this is the first work to address local community detection by the Lanczos approximation. Also, we adapt the standard Lanczos method which is on a symmetric matrix to the unsymmetrical transition matrix directly. (2) Instead of using the heat kernel method to directly extract a community, we did the first attempt to leverage its very fast diffusion property to sample a localized subgraph to largely reduce the subsequent calculation. (3) Based on the Rayleigh quotient related to the conductance, we provide a theoretical base for the proposed LLSA method. (4) Experiments on five real-world networks as well as seven synthetic networks show that the proposed method considerably outperforms existing local community detection algorithms.

2 Related Work

2.1 Local Community Detection

Techniques for local community detection can be classified into three categories, namely the PageRank, heat kernel and local spectral methods. Other techniques like finding minimum cut [5,4,26] can also be used for local community detection.

PageRank. The PageRank method is widely used for local community detection. Spielman and Teng [31] use degree-normalized, personalized PageRank (DN PageRank) with respect to the initial seeds and do truncation on small probability values. DN PageRank is adopted by several competitive PageRank based clustering algorithms [3,35], including the popular PageRank Nibble method [2].

Kloumann and Kleinberg [16] evaluate different variations of PageRank method and find that the standard PageRank yields higher performance than the DN PageRank.

Heat Kernel. The heat kernel method involves the Taylor series expansion of the exponential of the transition matrix. Chung [8,10] provides a theoretical analysis and a local graph partitioning algorithm based on heat kernel diffusion. Chung and Simpson [9] propose a randomized Monte Carlo method to estimate the diffusion speed, and Kloster and Gleich [15] propose a deterministic method that uses coordinate relaxation on an implicit linear system to estimate the heat kernel diffusion, and the heat value of each node represents the likelihood of affiliation.

Local Spectral. A third branch is to adapt the classic spectral method to locate the target community. Mahoney *et al.* [25] introduce a locally-biased analogue of the second eigenvector, the Fiedler vector associated with the algebraic connectivity, to extract local properties of data graphs, and apply the method for a semi-supervised image segmentation and a local community extraction by finding a sparse-cut around the seeds in small social networks. He *et al.* [14] and Li *et al.* [22] extract the local community by seeking a sparse vector from the local spectral subspaces using ℓ_1 norm optimization.

2.2 Lanczos Method

Many real world problems can be modeled as sparse graphs and be represented as matrices, and the eigenvalue calculation of the matrices is usually a crucial step for the problem solving. All the eigenpairs can be calculated by power method [29], SVD [32], or QR factorization [17]. However, these methods are intractable for large matrices due to the high complexity and memory consumption. As the Lanczos method can significantly reduce the time and space complexity, it is usually applied to large sparse matrices [27].

As a classic eigenvalue calculation method, the original Lanczos method [21] cannot hold the orthogonality of the calculated Krylov subspace and it is not widely used in practice. Paige [27] computes the eigenpairs for very large sparse matrices by an improved Lanczos method, as only a few iterations are typically required to get a good approximation on the extremal eigenvalues. After that, the Lanczos method becomes very attractive for large sparse matrix approximation. For example in the application of graph partitioning and image reconstruction, Barnes [6] illustrates that Lanczos method is an efficient implementation of the spectral bisection method; Wu *et al.* [33] propose an incremental bilinear Lanczos algorithm for high dimensionality reduction and image reconstruction; Bentbib *et al.* [7] illustrate that efficient image restoration can be achieved by Tikhonov regularization based on the global Lanczos method.

To the best of our knowledge, there is no Lanczos based algorithms for local community detection in the literature.

3 Local Lanczos Method

The local community detection problem can be formalized as follows. 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, $\mathbf{I}$ the identity matrix, and $\mathbf{e}$ the vector of all ones. Let $\mathbf{d} = \mathbf{Ae}$ be the vector of node degrees, and $\mathbf{D} = \mathrm{diag}(\mathbf{d})$ the diagonal matrix of node degrees. Let S be the set of a few exemplary members in the target community $T = (V_t, E_t)$ ($S \subseteq V_t \subseteq V$, $|V_t| \ll |V|$). Let $\mathbf{s} \in \{0,1\}^n$ 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.

There are three key steps in the proposed algorithm: heat kernel sampling, local Lanczos spectral approximation and community boundary truncation.

3.1 Local Heat Kernel Sampling

The heat kernel method [15] runs in linear time and is very fast for community detection on large networks. However, the detection accuracy is not high enough as compared with the local spectral method [14]. In this paper, we use the advantage of heat kernel's fast diffusion speed to do the sampling, using parameter settings such that the resulting subgraph is large enough to cover almost all members in the target community.

The Heat Kernel Diffusion. The heat kernel diffusion model spread the heat across a graph regarding the seed set as the persistent heat source.

The heat kernel diffusion vector is defined by

$$\mathbf{h} = e^{-t}\left[\sum_{k=0}^{\infty}\frac{t^k}{k!}(\mathbf{AD}^{-1})^k\right]\mathbf{p_0}, \tag{1}$$

where $\mathbf{p_0} = \mathbf{s}/|S|$ is the initial heat values on the source seeds. For simplicity of notation, let

$$\mathbf{h}_N = e^{-t}\left[\sum_{k=0}^{N}\frac{t^k}{k!}(\mathbf{AD}^{-1})^k\right]\mathbf{p_0} \tag{2}$$

indicate the sum of the first N terms.

In practice, we usually seek a vector $\mathbf{x}$ to approximate $\mathbf{h}$:

$$\|\mathbf{D^{-1}h} - \mathbf{D^{-1}x}\|_\infty < \epsilon. \tag{3}$$

Premultiplying e^t on both sides, we have

$$\|\mathbf{D^{-1}}e^t\mathbf{h} - \mathbf{D}^{-1}e^t\mathbf{x}\|_\infty < e^t\epsilon. \tag{4}$$

If for an integer N,

$$\|\mathbf{D^{-1}}e^t\mathbf{h} - \mathbf{D^{-1}}e^t\mathbf{h}_N\|_\infty < e^t\epsilon/2, \tag{5}$$

and $\mathbf{z} = e^t\mathbf{x} \approx e^t\mathbf{h}_N$ satisfies

$$\|\mathbf{D^{-1}}e^t\mathbf{h}_N - \mathbf{D^{-1}}\mathbf{z}\|_\infty < e^t\epsilon/2, \tag{6}$$

then by the triangle inequality, (4) holds, and then (3) holds.

Kloster and Gleich [15] propose a hk-relax algorithm to guarantee (5) by letting N be no greater than $2t\log(\frac{1}{\epsilon})$ and computing a vector $\mathbf{z}$ that satisfies (6), then use the heat values in $\mathbf{x}$ to identify memberships in the local community. We adapt their method to do the heat kernel sampling, shown in Algorithm 1.

Algorithm 1 The heat kernel sampling

Input: Graph $G = (V, E)$, seed set $S \subset V$, upper bound of the subgraph size N_1, heat kernel diffusion parameters t and ϵ

1: Start from S, calculate heat value vector $\mathbf{x}$ to approximate the heat kernel diffusion vector $\mathbf{h}$
2: Sort elements in $\mathbf{x}$ in decreasing order to get a vector $\tilde{\mathbf{x}}$
3: $G_s \leftarrow$ nodes corresponding to all the nonzero elements in $\tilde{\mathbf{x}}$
4: **if** $|G_s| > N_1$ **then**
5: $\quad G_s \leftarrow$ top N_1 nodes in G_s according to the heat value

Output: Sampled subgraph G_s

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 extract the local community from this comparatively small subgraph instead of the original large network. This pre-processing procedure runs in milliseconds in large networks with millions of nodes, and significantly reduces the computation cost for the follow-up fine tuning of the community detection.

3.2 Local Lanczos Spectral Approximation

In this subsection, we first provide the necessary theoretical base that finding a low-conductance community corresponds to finding the eigenvector of the transition matrix with the largest eigenvalue. Then we briefly introduce a variant of the Lanczos process on the Laplician matrix to calculate this eigenvector. Finally we propose a local Lanczos spectral approximation method to get a "local" eigenvector indicating the implicit topology structure of the network around the seeds, and provide an convergence analysis on the Lanczos iteration process.

Theoretical Base. Let $\mathbf{L} = \mathbf{D_s} - \mathbf{A_s}$ be the Laplacian matrix of G_s where $\mathbf{A_s}$ and $\mathbf{D_s}$ denotes the adjacency matrix and the diagonal degree matrix of G_s. We define two normalized graph Laplacian matrices:

$$\mathbf{L_{rw}} = \mathbf{I} - \mathbf{N_{rw}} = \mathbf{D_s^{-1}}\mathbf{L},$$

$$\mathbf{L_{sym}} = \mathbf{I} - \mathbf{N_{sym}} = \mathbf{D_s}^{-\frac{1}{2}} \mathbf{L} \mathbf{D_s}^{-\frac{1}{2}},$$

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

For a community C, the conductance [30] of C is defined as

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

where $\overline{C}$ consists of all nodes outside C, $\text{cut}(C, \overline{C})$ denotes the number of edges between, and $\text{vol}(\cdot)$ calculates the "edge volume", i.e. for the subset of 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 $\mathbf{y} \in \{0,1\}^{n_s}$ be a binary indicator vector representing a small community C in the sampled graph G_s. Here for "small community", we mean $\text{vol}(C) \leq \frac{1}{2}\text{vol}(V_s)$. As $\mathbf{y^T D_s y}$ equals the total node degrees of C, and $\mathbf{y^T A_s y}$ equals two times the number of internal edges of C, the conductance $\Phi(C)$ could be written as a generalized Rayleigh quotient:

$$\Phi(C) = \frac{\mathbf{y^T L y}}{\mathbf{y^T D_s y}} = \frac{(\mathbf{D_s^{\frac{1}{2}} y})^\mathbf{T} \mathbf{L_{sym}} (\mathbf{D_s^{\frac{1}{2}} y})}{(\mathbf{D_s^{\frac{1}{2}} y})^\mathbf{T} (\mathbf{D_s^{\frac{1}{2}} y})}. \tag{7}$$

Theorem 1. *(Cheeger Inequality) Let λ_2 be the second smallest eigenvalue of $\mathbf{L_{sym}}$ for a graph G_s, then $\phi(G_s) \geq \frac{\lambda_2}{2}$, where $\phi(G_s) = \min_{\tilde{V} \subset V_s} \Phi(\tilde{V})$.*

The proof refers to [11], and we omit the details here. According to this theorem and the definition of $\Phi(C)$, we have $\frac{\lambda_2}{2} \leq \Phi(C) \leq 1$.

According to the Rayleigh-Ritz theorem [23], if we want to minimize the conductance $\Phi(C)$ by relaxing the indicator vector $\mathbf{y}$ to take arbitrary real values, then the scaled relaxed indicator vector $\mathbf{D_s^{\frac{1}{2}} y}$ should be the eigenvector of $\mathbf{L_{sym}}$ with the smallest eigenvalue 0, which is $\mathbf{D_s^{\frac{1}{2}} e}$.

We know that:

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

the relaxed indicator vector $\mathbf{y}$ should be the eigenvector of $\mathbf{L_{rw}}$ with the smallest eigenvalue. As $\mathbf{L_{rw}} = \mathbf{I} - \mathbf{N_{rw}}$, the eigenvalue decomposition of the Laplacian matrix is also closely related to the expansion of rapid mixing of random walks. As

$$\mathbf{L_{rw}} \mathbf{v} = (\mathbf{I} - \mathbf{N_{rw}}) \mathbf{v} = \lambda \mathbf{v} \quad \Leftrightarrow \quad \mathbf{N_{rw}} \mathbf{v} = (1 - \lambda) \mathbf{v},$$

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 the eigenvector of $\mathbf{N_{rw}}$ with the largest eigenvalue.

The largest eigenvalue of $\mathbf{N_{rw}}$ is 1 and the corresponding eigenvector is $\mathbf{e}$ [24], so the relaxed indicator vector $\mathbf{y} = \mathbf{e}$, corresponding to the whole graph with

zero conductance. This relaxed indicator vector $\mathbf{y}$ contains global information while the real solution of the indicator vector $\mathbf{y}$ reveals local property for a small community whose total degree is no greater than half of the total degree of the whole graph. As the Lanczos method is efficient for computing the top eigenpairs of large sparse matrices and it is space efficient, we propose a variant of Lanczos method on $\mathbf{N_{rw}}$ to get a "local" eigenvector indicating the latent local structure around the seeds.

Lanczos Process. Based on a theoretical guarantee [18], there exists an orthogonal matrix $\mathbf{Q}$ and a tridiagonal matrix $\mathbf{T}$ such that

$$\mathbf{Q^T}(\mathbf{D_s^{-\frac{1}{2}}}\mathbf{A_s}\mathbf{D_s^{-\frac{1}{2}}})\mathbf{Q} = \mathbf{T}, \tag{8}$$

$$\mathbf{T} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n_s-1} \\ & & & \beta_{n_s-1} & \alpha_{n_s} \end{bmatrix}. \tag{9}$$

Designate the columns of $\mathbf{Q}$ by

$$\mathbf{Q} = \left[\mathbf{q_1} \mid \cdots \mid \mathbf{q_{n_s}}\right].$$

Let $\tilde{\mathbf{Q}} = \mathbf{D_s^{-\frac{1}{2}}}\mathbf{Q}$, so

$$\begin{aligned} \tilde{\mathbf{Q}} &= \left[\mathbf{D_s^{-\frac{1}{2}}}\mathbf{q_1} \mid \cdots \mid \mathbf{D_s^{-\frac{1}{2}}}\mathbf{q_{n_s}}\right] \\ &\triangleq \left[\tilde{\mathbf{q}}_\mathbf{1} \mid \cdots \mid \tilde{\mathbf{q}}_{\mathbf{n_s}}\right]. \end{aligned}$$

Eq. (8) can be rewritten as

$$\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{A_s}\tilde{\mathbf{Q}} = \mathbf{T}. \tag{10}$$

As $\mathbf{Q}$ is an orthogonal matrix,

$$\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^\mathbf{T} = \mathbf{D_s^{-\frac{1}{2}}}\mathbf{Q}\mathbf{Q}^\mathbf{T}\mathbf{D_s^{-\frac{1}{2}}} = \mathbf{D_s^{-1}}. \tag{11}$$

Premultiplying $\tilde{\mathbf{Q}}$ on both sides of Eq. (10), we have $\mathbf{D_s^{-1}}\mathbf{A_s}\tilde{\mathbf{Q}} = \tilde{\mathbf{Q}}\mathbf{T}$. Equating the columns in this equation, we conclude that for $k \in \{1, ..., n_s\}$,

$$\mathbf{D_s^{-1}}\mathbf{A_s}\tilde{\mathbf{q}}_\mathbf{k} = \beta_{k-1}\tilde{\mathbf{q}}_{\mathbf{k-1}} + \alpha_k\tilde{\mathbf{q}}_\mathbf{k} + \beta_k\tilde{\mathbf{q}}_{\mathbf{k+1}}, \tag{12}$$

by setting $\beta_0\tilde{\mathbf{q}}_\mathbf{0} \triangleq \mathbf{0}$, and $\beta_n\tilde{\mathbf{q}}_{\mathbf{n_s+1}} \triangleq \mathbf{0}$.

By the orthogonality of $\mathbf{Q}$, we have $\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_s}\tilde{\mathbf{Q}} = \mathbf{I}$. Premultiplying $\tilde{\mathbf{q}}_\mathbf{k}^\mathbf{T}\mathbf{D_s}$ on both sides of Eq. (12), the $\mathbf{D_s}$-inner product orthonormality of the $\tilde{\mathbf{q}}$-vectors implies

$$\alpha_k = \tilde{\mathbf{q}}_\mathbf{k}^\mathbf{T}\mathbf{A_s}\tilde{\mathbf{q}}_\mathbf{k}. \tag{13}$$

Let the vector $\tilde{\mathbf{r}}_\mathbf{k}$ be

$$\tilde{\mathbf{r}}_\mathbf{k} = \mathbf{D_s^{-1}}\mathbf{A_s}\tilde{\mathbf{q}}_\mathbf{k} - \alpha_k\tilde{\mathbf{q}}_\mathbf{k} - \beta_{k-1}\tilde{\mathbf{q}}_{\mathbf{k-1}}. \tag{14}$$

If $\tilde{\mathbf{r}}_\mathbf{k}$ is nonzero, then by Eq. (12) we have

$$\tilde{\mathbf{q}}_{\mathbf{k+1}} = \tilde{\mathbf{r}}_\mathbf{k}/\beta_k. \tag{15}$$

With the "canonical" choice such that $\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_s}\tilde{\mathbf{Q}} = \mathbf{I}$,

$$\beta_k = \|\mathbf{D_s^{\frac{1}{2}}}\tilde{\mathbf{r}}_\mathbf{k}\|_2. \tag{16}$$

For any unit vector $\mathbf{q_1}$, let $\beta_0 = 1$, $\tilde{\mathbf{q}}_\mathbf{0} = 0$, and $\tilde{\mathbf{r}}_\mathbf{0} = \mathbf{D_s}^{-\frac{1}{2}}\mathbf{q_1}$. Start from $k = 1$, we could iteratively calculate the entries of α_k, β_k in $\mathbf{T}$ until $k = n_s$. Meanwhile, $\tilde{\mathbf{Q}}$ is also obtained during the iteration.

Spectral Calculation via Lanczos Process. Let $\mathbf{v}$ be the eigenvector of $\mathbf{N_{rw}}$ with the largest eigenvalue λ, we know

$$\mathbf{N_{rw}v} = \mathbf{D_s^{-1}A_sv} = \lambda\mathbf{v}.$$

Premultiplying $\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_s}$ on both sides, we get

$$\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{A_sv} = \lambda\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_sv}. \tag{17}$$

According to Eq. (11), $\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_s} = \mathbf{I}$. Then by Eq. (10), the left hand side of Eq. (17) equals

$$\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{A_s}\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_sv} = \mathbf{T}\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_sv}.$$

Let $\mathbf{u} = \tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_sv}$, we get

$$\mathbf{Tu} = \lambda\mathbf{u}. \tag{18}$$

On the other hand, premultiplying $\tilde{\mathbf{Q}}$ and postmultiplying $\mathbf{u}$ on both sides of Eq. (8), we have

$$\mathbf{N_{rw}}\tilde{\mathbf{Q}}\mathbf{u} = \lambda\tilde{\mathbf{Q}}\mathbf{u},$$

so λ is also the largest eigenvalue of $\mathbf{T}$. As

$$\mathbf{v} = \tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^\mathbf{T}\mathbf{D_sv} = \tilde{\mathbf{Q}}\mathbf{u}, \tag{19}$$

we can calculate $\mathbf{v}$ by calculating the eigenvector $\mathbf{u}$ of $\mathbf{T}$ with the largest eigenvalue λ.

Local Lanczos Spectral Approximation. Instead of using the eigenvalue decomposition to get the "global spectra", He *et al.* [14] use short random walks starting from the seed set to get a local proxy for the eigenvectors of $\mathbf{N_{rw}}$, which they call the "local spectra". Here we consider a novel way based on the Lanczos method [27] to approximate the eigenvector of $\mathbf{N_{rw}}$ with the largest eigenvalue. A few steps of the Lanczos iteration lead to the local approximation of this eigenvector.

Let $\mathbf{q_1}$ be the normalized indicator vector for the seed set, set $\beta_0 = 1$, $\tilde{\mathbf{q}}_\mathbf{0} = 0$, and $\tilde{\mathbf{r}}_\mathbf{0} = \mathbf{D_s}^{-\frac{1}{2}}\mathbf{q_1}$. By k steps of Lanczos iteration, we could get the first k by k submatrix of $\mathbf{T}$, denoted by $\mathbf{T_k}$. Correspondingly, let the first k columns of $\tilde{\mathbf{Q}}$ be a matrix $\tilde{\mathbf{Q}}_\mathbf{k}$. Let the eigenvectors of $\mathbf{T_k}$ with larger eigenvalues be a matrix $\mathbf{U_k}$. According to Eq. (19), the columns of $\mathbf{V_k} = \tilde{\mathbf{Q}}_\mathbf{k}\mathbf{U_k}$ approximate the eigenvectors of $\mathbf{N_{rw}}$ with larger eigenvalues. The first column of $\mathbf{V_k}$ approximates the eigenvector of $\mathbf{N_{rw}}$ with the largest eigenvalue, which is the indicator vector $\mathbf{y}$ we want to find.

The Local Lanczos Spectral Approximation (LLSA) procedure on the sampled graph is summarized in Algorithm 2. The slowest step is Step 4 for calculating $\tilde{\mathbf{q}}_\mathbf{k}, \alpha_k, \tilde{\mathbf{r}}_\mathbf{k}$, and β_k. It requires $O(Kn_s^2)$ time to implement the Lanczos iteration, where K is the steps of Lanczos iteration and n_s is number of nodes in graph G_s. Also, note that the Lanczos iteration requires only a few vectors of intermediate storage.

Algorithm 2 Local Lanczos Spectral Approximation

Input: $G_s = (V_s, E_s)$, maximum iteration steps K, initial vector $\mathbf{q_1}$

1: Initialize $k = 0, \beta_0 = 1, \mathbf{q_0} = \mathbf{0}, \tilde{\mathbf{r}}_\mathbf{0} = \mathbf{D_s^{-\frac{1}{2}}} \mathbf{q_1}$
2: **while** $(k < K)$ **do**
3: $\quad k = k + 1$
4: $\quad$ Calculate $\tilde{\mathbf{q}}_\mathbf{k}, \alpha_k, \tilde{\mathbf{r}}_\mathbf{k}, \beta_k$ by Eq. (15),(13), (14), (16)
5: Let $\mathbf{T_k}$ be the first $k \times k$ entries of $\mathbf{T}$ in Eq. (9)
6: Get the eigenvector $\mathbf{u}$ of $\mathbf{T_k}$ with the largest eigenvalue λ

Output: $\mathbf{y} = \tilde{\mathbf{Q}}_\mathbf{k}\mathbf{u}$

Convergence Analysis. Here we provide an analysis on the convergence of the Lanczos process, i.e. the approximation gap between the local eigenvector which indicates the local structure around the seeds and the global eigenvector of the graph.

By Eq. (12) and Eq. (14), we conclude that for $1 \leq k < n_s$,

$$\mathbf{N_{rw}}\tilde{\mathbf{Q}}_\mathbf{k} = \mathbf{D_s^{-1}}\mathbf{A_s}\tilde{\mathbf{Q}}_\mathbf{k} = \tilde{\mathbf{Q}}_\mathbf{k}\mathbf{T_k} + \tilde{\mathbf{r}}_\mathbf{k}\mathbf{e_k^T}, \tag{20}$$

where $\mathbf{e_k}$ is the kth unit vector with unity in the kth element and zero otherwise.

Let $\mathbf{u}$ be the eigenvector of $\mathbf{T_k}$ with the largest eigenvalue λ, postmultiplying $\mathbf{u}$ on both sides of Eq. (20), we have

$$\mathbf{N_{rw}}\tilde{\mathbf{Q}}_\mathbf{k}\mathbf{u} = \tilde{\mathbf{Q}}_\mathbf{k}\mathbf{T_k}\mathbf{u} + \tilde{\mathbf{r}}_\mathbf{k}\mathbf{e_k^T}\mathbf{u}. \tag{21}$$

Let $\mathbf{y} = \tilde{\mathbf{Q}}_\mathbf{k}\mathbf{u}$, the approximated residual value can be calculated as

$$\|\mathbf{r}\|_\mathbf{2} = \|\mathbf{N_{rw}}\mathbf{y} - \lambda\mathbf{y}\|_\mathbf{2} = \|\mathbf{N_{rw}}\tilde{\mathbf{Q}}_\mathbf{k}\mathbf{u} - \lambda\tilde{\mathbf{Q}}_\mathbf{k}\mathbf{u}\|_\mathbf{2}. \tag{22}$$

By Eq. (21), Eq. (22) can be modified as

$$\|\mathbf{r}\|_\mathbf{2} = \|\mathbf{N_{rw}}\tilde{\mathbf{Q}}_\mathbf{k}\mathbf{u} - \tilde{\mathbf{Q}}_\mathbf{k}\mathbf{T_k}\mathbf{u}\|_\mathbf{2} = \|\tilde{\mathbf{r}}_\mathbf{k}\mathbf{e_k^T}\mathbf{u}\|_\mathbf{2}. \tag{23}$$

Furthermore, by Eq. (15),

$$\|\mathbf{r}\|_\mathbf{2} = \|\beta_k\tilde{\mathbf{q}}_\mathbf{k+1}\mathbf{e_k^T}\mathbf{u}\|_\mathbf{2} = \|\beta_k\mathbf{D_s^{-\frac{1}{2}}}\mathbf{q_{k+1}}\mathbf{e_k^T}\mathbf{u}\|_\mathbf{2} = \beta_k|u_k| \cdot \|\mathbf{D_s^{-\frac{1}{2}}}\mathbf{q_{k+1}}\|_\mathbf{2}, \tag{24}$$

where u_k is the kth (last) term of eigenvector $\mathbf{u}$.

As $\mathbf{q_{k+1}}$ is a unit vector, according to the Rayleigh-Ritz theorem [23],

$$\|\mathbf{D_s^{-\frac{1}{2}}}\mathbf{q_{k+1}}\|_\mathbf{2} \leq \max_{\|\mathbf{x}\|_\mathbf{2}=\mathbf{1}} \|\mathbf{D_s^{-\frac{1}{2}}}\mathbf{x}\|_\mathbf{2} = d_{min}^{-\frac{1}{2}}, \tag{25}$$

where d_{min} denotes the minimum degree of the nodes in graph G_s. $d_{min}^{-\frac{1}{2}}$ is also the largest eigenvalue of the diagonal matrix $\mathbf{D_s^{-\frac{1}{2}}}$.

By Eq. (24) and Eq. (25), we have

$$\|\mathbf{r}\|_\mathbf{2} \leq \beta_k d_{min}^{-\frac{1}{2}}|u_k|. \tag{26}$$

Generally, the higher the value of k is, the smaller the residual value $\|\mathbf{r}\|_\mathbf{2}$ is. And we need to use a small iteration step to find the "local" eigenvector. Experimental analysis in Section 4 shows a suitable value for the iteration step is around 4 or 5.

3.3 Community Boundary Truncation

The value of the kth element of $\mathbf{y}$ indicates how likely node k belongs to the target community. We use a heuristic similar to [35] to determine the community boundary.

We sort the nodes based on the element values of $\mathbf{y}$ 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 eventually becomes higher than $\alpha\Phi(S_{k^*})$, then we take k^* as a valid local minimum. We experimented with several values of α on a small trial of data and found that $\alpha = 1.03$ gives good performance across all the datasets.

The overall Local Lanczos Spectral Approximation (LLSA) algorithm is shown in Algorithm 3.

Algorithm 3 The overall LLSA algorithm

Input: $G = (V, E)$, seed set $S \subseteq V$
1: Get sampled subgraph $G_s = (V_s, E_s)$ by Algorithm 1
2: Calculate vector $\mathbf{y}$ by Algorithm 2
3: Sort nodes by the decreasing value of elements in $\mathbf{y}$
4: Find k_0 where S_k contains all the seeds
5: For $k = k_0 : n_s$, compute the conductance $\Phi(S_k)$: $\Phi_k = \Phi(S_k = \{v_i | i \leq k$ in the sorted list$\})$
6: Find k^* with the first local minimum $\Phi(S_{k^*})$
Output: Community $C = S_{k^*}$

4 Experiments

In this section, we compare LLSA with several state-of-the-art local community detection algorithms, and evaluate the performance by a popular F_1 metric.

4.1 Data Description

Seven synthetic datasets (parameters in Table 1) and five real-world datasets (Table 2) are considered for a comprehensive evaluation.

LFR Benchmark Graphs. Lancichinetti *et al.* [20,19] proposed a method for generating LFR[3] benchmark graphs with a built-in binary community structure, which simulates properties of real-world networks on heterogeneity of node degree and community size distributions. The LFR benchmark graphs are widely used for evaluating community detection algorithms, and Xie *et al.* [34] performed a thorough performance comparison of different community detection algorithms on the LFR benchmark datasets.

[3] http://santo.fortunato.googlepages.com/inthepress2

We adopt the same set of parameter settings used in [34] and generate seven LFR benchmark graphs. Table 1 summarizes the parameter settings, among which the mixing parameter μ has a big impact on the network topology. μ controls the average fraction of neighboring nodes that do not belong to any community for each node. μ is usually set to be 0.1 or 0.3 and the detection accuracy usually decays for a larger μ. 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 on the communities, leading to a harder community detection task.

Real-world Networks. We choose five real-world network datasets with labeled ground truth from the SNAP[4], namely Amazon, DBLP, LiveJ, YouTube and Orkut in the domains of product, collaboration and social contact [35]. Table 2 summarizes the statistics of the networks and the ground truth communities. We calculate the average and standard deviation of the community size, and the average conductance, where low conductance gives priority to communities with dense internal links and sparse external links.

4.2 Experimental Setup

We implement the proposed LLSA method in Matlab[5] through a C mex interface and conduct experiments on a computer with 2 Intel Xeon processors at 2.30GHz and 128GB memory. For the five SNAP datasets, we randomly locate 500 ground truth communities on each dataset, and randomly pick three exemplary seeds from each target community. For the seven LFR datasets, we deal with every ground truth community and randomly pick three exemplary seeds from each ground truth community. To make a fair comparison, we run all baseline algorithms using the same set of random seeds.

For the parameters, we fix $(t, \epsilon, N_1) = (3, 10^{-6}, 5000)$ for Algorithm 1 such that the resulting subgraph is large enough to cover almost all the members in the target community. We set $K = 4$ for Algorithm 2 to have a good trade-off on real-world datasets as well as the synthetic data.

Comparison Baselines. We select three representative local community detection algorithms as the baselines. All algorithms accept as inputs an adjacency matrix $\mathbf{A}$ and a seed set S, and run on their default parameter settings. They apply different techniques to compute diffusion ranks starting from the seed set, then perform a sweep cut on the resulting ranks.

- **pprpush (PR)** [2]: the popular PageRank Nibble method.
- **hk-relax (HK)** [15]: the current best-performing heat kernel diffusion method.
- **LOSP** [14]: the current best-performing local spectral subspace based method.

[4] http://snap.stanford.edu
[5] https://github.com/PanShi2016/LLSA

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
$[20, 100]$	range of the community size
$\tau_1 = 2$	node degree distribution exponent
$\tau_2 = 1$	community size distribution exponent
$om \in \{2, 3..., 8\}$	overlapping membership
$on = 500$	number of overlapping nodes

Table 1. Parameters for the LFR benchmarks.

	Network			Ground truth communities	
Domain	**Name**	**# Nodes**	**# Edges**	**Avg. ± Std. Size**	**Avg. Cond.**
Product	Amazon	334,863	925,872	13 ± 18	0.073
Collaboration	DBLP	317,080	1,049,866	22 ± 201	0.414
Social	LiveJ	3,997,962	34,681,189	28 ± 58	0.388
Social	YouTube	1,134,890	2,987,624	21 ± 73	0.839
Social	Orkut	3,072,441	117,185,083	216 ± 321	0.731

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

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 defined by:

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

where the precision P and recall R are defined as:

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

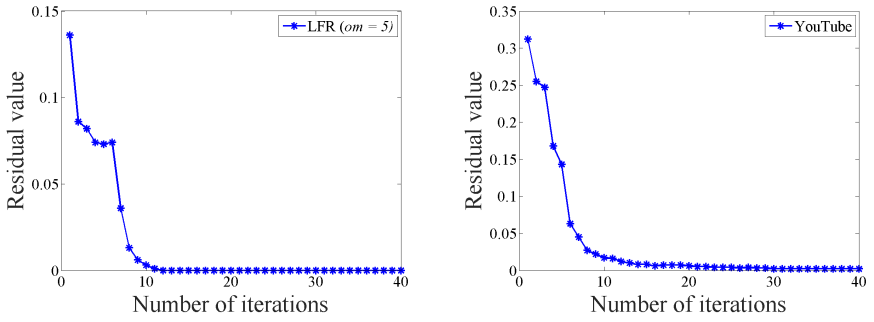
4.3 Experimental Results

Sampling. Table 3 shows the statistics for the heat kernel sampling on the real datasets. The sampled subgraphs are relatively small with 3200 nodes on average, only sampled about 0.3% of the nodes from the original graph. Nevertheless , there is a very high coverage ratio (ratio of ground truth nodes covered by the subgraph) of 96%, and the sampling procedure is very fast in less than 0.3 seconds. As for the LFR datasets, the sampling almost covers all the 5000 nodes as the synthetic networks are denser and much smaller.

Convergence Results. As LLSA involves the local Lanczos iteration, we experimentally investigate the convergence property of Algorithm 2 on two datasets: a synthetic network LFR for $om = 5$ and a real network YouTube. For the output of each iteration, we calculate the residual value $\|\mathbf{r}\|_\mathbf{2} = \|\mathbf{N_{rw}y} - \lambda\mathbf{y}\|_\mathbf{2}$, as shown in Fig. 1. One can see that the output $\mathbf{y}$ of Algorithm 2 converges very quickly, indicating that the spectra becomes global for more than 10 iterations. To gain a "local" eigenvector indicating the implicit topology structure of the local region around the seeds, we set the iteration step $K = 4$ for Algorithm 2.

Datasets	Coverage	n_s	n_s/n	Time(s)
Amazon	0.999	449	0.0013	0.016
DBLP	0.991	3034	0.0096	0.039
LiveJ	0.998	2639	0.0007	0.258
YouTube	0.919	4949	0.0044	0.437
Orkut	0.900	4990	0.0016	0.620
Average	0.961	3212	0.0035	0.274

Table 3. Statistics of the average values for the sampling.



(a) (b)

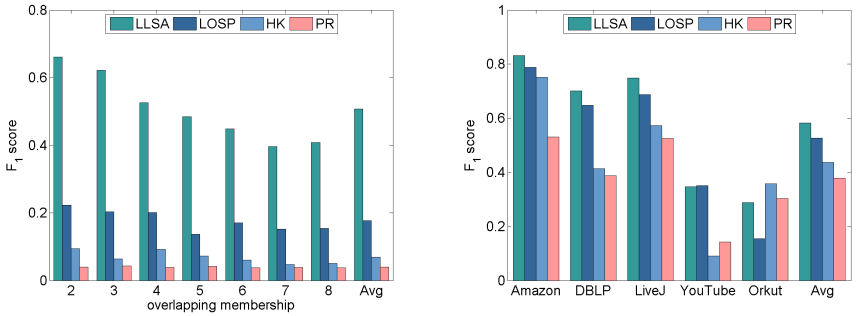
Fig. 1. Convergence analysis on LFR ($om = 5$) network and YouTube network.

Accuracy Comparison. Fig. 2(a) illustrates the average detection accuracy of LLSA and the baselines on the LFR networks. LLSA significantly outperforms all baseline methods on all the seven synthetic networks. As $on = 500$ overlapping nodes are assigned to $om = 2, 3$, or 8 communities, a larger om makes the detection more difficult, leading to a lower accuracy.

Fig. 2(b) illustrates the average detection accuracy on real-world networks. LLSA outperforms all baseline methods on Amazon, DBLP and LiveJ. LOSP performs the best on Youtube but is in the last place on Orkut; HK and PR show better performance on Orkut but behave poorly on Youtube. Though LLSA does not outperform all other methods on YouTube and Orkut, it is the most robust method and very competitive on average. As a whole, LLSA performs the best on the five real-world datasets.

	Conductance				Size				Time(s)			
	LLSA	**LOSP**	**HK**	**PR**	**LLSA**	**LOSP**	**HK**	**PR**	**LLSA**	**LOSP**	**HK**	**PR**
Amazon	0.227	0.297	0.042	0.030	9	8	48	4485	0.045	0.040	0.008	0.015
DBLP	0.309	0.414	0.110	0.114	12	22	87	9077	1.038	0.546	0.025	0.075
LiveJ	0.243	0.419	0.083	0.086	43	29	119	512	1.191	1.132	0.029	0.264
YouTube	0.618	0.800	0.175	0.302	120	10	122	13840	1.572	2.896	0.038	0.955
Orkut	0.659	0.930	0.513	0.546	920	17	341	1648	4.352	2.662	0.027	1.392
Average	0.411	0.572	0.185	0.216	221	17	143	5912	1.640	1.455	0.025	0.540

Table 4. Average conductance and size of the identified communities and average running time of the algorithms on real-world networks.



(a) LFR networks (b) Real-world networks

Fig. 2. Accuracy comparison on all datasets.

Table 4 shows more comparisons on real-world datasets. Compared with LLSA and LOSP that finds a local minimal conductance, HK and PR seek for a global minimum conductance, and often find larger communities with lower conductance. On the other hand, as shown in Table 2, the ground truth communities are small with lower conductance for the first three datasets. This may explain why the four algorithms provide favorable results for the first three datasets but are adverse to YouTube and Orkut which have higher conductance, indicating many links to external nodes, hence lower conductance alone is not suited in finding the local, small communities. This may explain why HK and PR show better performance on Orkut which contains communities with hundreds of nodes but behave poorly on Youtube with small community size. Table 4 shows that LOSP finds small communities with higher conductance, this may explain why LOSP performs the best on Youtube but is in the last place on Orkut.

For the running time, all algorithms are very fast and run in seconds. On average, HK is the fastest in 0.025 seconds, and the other three are similar in 0.5 to 1.6 seconds. LLSA and LOSP costs one more second as compared with PR, as they involve finding a community with the local minimal conductance. Also, different methods are implemented in different languages (LLSA and LOSP use Matlab, HK and PR use C++), so the running times could give an indication of the overall trend, and it can not be compared directly.

5 Conclusion

In this paper, we propose a novel Local Lanczos Spectral Approximation (LLSA) approach for local community detection, which is, to the best of our knowledge, the first time to apply Lanczos for local community detection. The favorable results on the synthetic LFR datasets and the real-world SNAP datasets suggest that the Lanczos method could be a new and effective way to detect local communities in large graphs. Based on Rayleigh quotient and conductance, we provide theoretical base for the proposed method. In addition, we also utilize the very fast heat kernel diffusion to get a local sampled subgraph that largely reduces the complexity of the subsequent computation. We wish our work in-

spire more researches based on the Lanczos method for network analysis and community detection.

Acknowledgments. The work is supported by NSFC (61472147), US Army Research Office (W911NF-14-1-0477), and MSRA Collaborative Research (97354136).

References

1. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature 466(7307), 761–764 (2010)
2. Andersen, R., Chung, F., Lang, K.: Local graph partitioning using pagerank vectors. In: FOCS. pp. 475–486 (2006)
3. Andersen, R., Lang, K.J.: Communities from seed sets. In: WWW. pp. 223–232 (2006)
4. Andersen, R., Lang, K.J.: An algorithm for improving graph partitions. In: SODA. pp. 651–660 (2008)
5. Andersen, R., Peres, Y.: Finding sparse cuts locally using evolving sets. In: STOC. pp. 235–244 (2009)
6. Barnes, E.R.: An algorithm for partitioning the nodes of a graph. Siam Journal on Algebraic and Discrete Methods 3(4), 303–304 (1982)
7. Bentbib, A.H., El Guide, M., Jbilou, K., Reichel, L.: A global lanczos method for image restoration. Journal of Computational and Applied Mathematics 300, 233–244 (2016)
8. Chung, F.: The heat kernel as the PageRank of a graph. PNAS 104(50), 19735–19740 (2007)
9. Chung, F., Simpson, O.: Solving linear systems with boundary conditions using heat kernel pagerank. In: Algorithms and Models for the Web Graph(WAW). pp. 203–219 (2013)
10. Chung, F.: A local graph partitioning algorithm using heat kernel pagerank. Internet Mathematics 6(3), 315–330 (2009)
11. Chung, F.: Spectral graph theory, vol. 92. American Mathematical Soc. (1997)
12. Coscia, M., Giannotti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. Stastical Analysis and Data Mining 4(5), 512–546 (2011)
13. Coscia, M., Rossetti, G., Giannotti, F., Pedreschi, D.: DEMON: a local-first discovery method for overlapping communities. In: KDD. pp. 615–623 (2012)
14. He, K., Sun, Y., Bindel, D., Hopcroft, J., Li, Y.: Detecting overlapping communities from local spectral subspaces. In: ICDM. pp. 769–774 (2015)
15. Kloster, K., Gleich, D.F.: Heat kernel based community detection. In: KDD. pp. 1386–1395 (2014)
16. Kloumann, I.M., Kleinberg, J.M.: Community membership identification from small seed sets. In: KDD. pp. 1366–1375 (2014)
17. Knight, P.A.: Fast rectangular matrix multiplication and qr decomposition. Linear algebra and its applications 221, 69–81 (1995)
18. Komzsik, L.: The Lanczos method: evolution and application, vol. 15. SIAM (2003)
19. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E 80(1), 016118 (2009)

20. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Physical Review E 78(4), 046110 (2008)
21. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the National Bureau of Standards 45, 255–282 (1950)
22. Li, Y., He, K., Bindel, D., Hopcroft, J.: Uncovering the small community structure in large networks. In: WWW. pp. 658–668 (2015)
23. Lütkepohl, H.: Handbook of matrices, vol. 2 (1997)
24. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
25. Mahoney, M.W., Orecchia, L., Vishnoi, N.K.: 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), 2339–2365 (2012)
26. Orecchia, L., Zhu, Z.A.: Flow-based algorithms for local graph clustering. In: SODA. pp. 1267–1286 (2014)
27. Paige, C.: The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph.D. thesis, University of London (1971)
28. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818 (2005)
29. Parlett, B.N., Poole, Jr, W.G.: A geometric theory for the qr, lu and power iterations. SIAM Journal on Numerical Analysis 10(2), 389–412 (1973)
30. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
31. Spielman, D.A., Teng, S.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: STOC. pp. 81–90 (2004)
32. Stewart, G.W.: On the early history of the singular value decomposition. SIAM Review 35(4), 551–566 (1993)
33. Wu, G., Xu, W., Leng, H.: Inexact and incremental bilinear Lanczos components algorithms for high dimensionality reduction and image reconstruction. Pattern Recognition 48(1), 244–263 (2015)
34. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Computing Surveys (CSUR) 45(4), 43 (2013)
35. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. In: ICDM. pp. 745–754 (2012)