Locally Finding and Testing Dense Bipartite-like Subgraphs

Pan Peng

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, P.R.China
and
Graduate University of Chinese Academy of Sciences, P.R.China.
and
University of Chinese Academy of Sciences, P.R.China.

Abstract

We study *local algorithms* for finding and testing dense bipartite-like subgraphs which characterize cyber-communities in the web [21]. We use the *bipartiteness (ratio)* of a set as the quality measure that was introduced by Trevisan [37].

Our first algorithm, denoted as $\texttt{FindDB}(v, K, \theta)$, is a *local exploration algorithm* that takes as input a starting vertex v, a volume target K and a bipartiteness parameter θ and outputs an induced subgraph of G. It is guaranteed to have the following approximation performance: for any subgraph S with bipartiteness θ, there exists a subset $S_\theta \subseteq S$ such that $\mathrm{vol}(S_\theta) \geq \mathrm{vol}(S)/9$ and that if the starting vertex $v \in S_\theta$ and $\mathrm{vol}(S) \leq K$, the algorithm $\texttt{FindDB}(v, K, \theta)$ outputs a subgraph with volume $O(K^2\Delta/\theta^2)$ and bipartiteness $O(\sqrt{\theta})$. The running time of the algorithm is $O(\frac{K^2}{\theta^2}(\log\frac{K}{\theta} + \Delta))$, where Δ is the maximum degree of G, independent of the size of G.

By using the analysis of the local exploration algorithm, we give a one-sided *testing algorithm* for testing if every small subgraph in the graph is *not* dense bipartite-like. A graph G is a (K, θ)-non-dense bipartite graph (abbreviated as (K, θ)-NDBGraph) if every subset of volume at most K has bipartiteness at least θ. A graph G is ϵ-far from any (K, θ)-NDBGraph if we have to modify at least ϵ fraction of edges to obtain a (K, θ)-NDBGraph. Our tester, called `TestNSDB`, takes as input a graph G, a volume target K,

Email address: `pengpan@ios.ac.cn` (Pan Peng)

a bipartiteness parameter θ and a distance parameter ϵ. The tester accepts G if it is a (K, θ)-NDBGraph and rejects it with high probability if G is ϵ-far from any (k, τ)-NDBGraph, where $\frac{\Delta k^6 \ln k}{\theta^4} = O(K)$ and $\tau = O(\theta^2)$. The running time of our tester is $O(\frac{K \log K}{\epsilon})$, also independent of the size of G.

Keywords: local exploration algorithms, graph property testing, dense bipartite subgraphs, spectral analysis

1. Introduction

In the era of large-scale network data, even a linear time algorithm that just scans the whole input is unbearable. It becomes increasingly important to design *local algorithms* that only needs to explore a small portion of the input graph as well as allows theoretical guarantees on the quality of the output. So far there have been several local algorithmic paradigms developed, such as local exploration algorithm [33], the property testing [14], the Jump and Crawl model [8], the decentralized searching [20] and so on, all with the (heavy) motivation of handling large-scale data efficiently. In this paper, we focus on the first two local algorithmic paradigms.

A local exploration algorithm for massive graphs is one that explores only portion of the given graph and finds a solution with good approximation guarantee. Given a graph G as an oracle, from which the algorithm can request the degree of a vertex or the adjacency list of a vertex, and a numerical property $\mathcal{P}$ of subgraphs (such as diameter, conductance), a local exploration algorithm is supposed to have the following form: it takes as input a starting vertex v (or a small set of vertices), only traverses the vertices that near v and outputs a subgraph S such that $\mathcal{P}(S)$ is close to $\mathcal{P}(S^*)$, where S^* is the subgraph containing v that has the optimal value for $\mathcal{P}$. However, in the design of local algorithms, an approximation guarantee result as above is too strong, if possible, to obtain and it is usually relaxed as follows: if S is a subgraph, then there exists a large subset $S' \subseteq S$ such that for any starting vertex $v \in S'$, the algorithm will output a subgraph for which the $\mathcal{P}$ value is close to $\mathcal{P}(S)$. This algorithmic paradigm was introduced by Spielman and Teng, who gave a local algorithm for finding subgraph with small conductance [35]. Building on their work, other local clustering algorithms with better approximation ratio and running time have been proposed by Anderson, Chung and Lang [5], Anderson and Yepes [6], Kwok and Lau [22]

and Oveis Gharan and Trevisan [29]. Local algorithm for finding dense subgraphs have been studied by Anderson [4]. These algorithms have important applications in graph sparsificasion, solving linear equations [34], Laplacian algorithmic paradigm [36] and have also been used to handle real networks data (e.g., [23, 27]). However, to our knowledge, only a few number of problems are shown to have such local exploration algorithms.

Another local algorithmic paradigm called graph property testing has a relatively longer history [31]. In the setting of testing some graph property $\mathcal{P}$, we are also given a graph G as an oracle to which we are able to perform queries, and we want to know whether G has $\mathcal{P}$ or is *ϵ-far* from having $\mathcal{P}$, which means that we have to change at least ϵ fraction of edges to obtain a graph satisfying $\mathcal{P}$. There are typically three different models depending on the sparsity (or density) of G and the types of queries we are allowed to perform. In the adjacency matrix model, typically for dense graphs, we are allowed to perform the *vertex-pair queries*, that is, we can query whether there is an edge between any vertex pair u, v. Strong and fruitful characterizations on efficiently testable properties are known in this model [2]. In the adjacency list model, typically for the bounded degree graphs, both the *degree queries* and *neighbor queries* are allowed, which means that we can query the degree of any vertex v and the i-the neighbor of v. It is still not well understood what kind of properties can be tested efficiently in this model. The model for the general graphs usually allows all the three queries. However, few testing algorithms are known in this model [19, 3, 24, 26], let alone the characterization of efficiently testable properties.

In this paper, we add one more problem to the list of problems that allow local exploration algorithm and one more problem to the list that allows efficiently testing algorithm for general graph. More precisely, we give a local exploration algorithm for extracting *dense bipartite-like* subgraphs and a one-sided property testing algorithm for testing if the input graph has *no small dense bipartite-like subgraph* for general graphs. Dense bipartite-like subgraphs serve as a good channel for us to understand the link structures of the web graph (that is, the nodes are the web pages and a directed edge (i, j) represents a hyperlink from i to j). We are interested in extracting useful information from this huge graph, one of particular interest are the cyber-communities, which gives insights into the intellectual evolution of the web and facilitates adverting at a more precise level [21]. As found by Kumar et al [21], the cyber-communities are characterized by dense bipartite subgraphs.

To measure the property of a group of web pages being cyber-community like, that is, whether the group is close to a dense bipartite subgraph or not, we will adopt a concept *bipartiteness (ratio)* introduced by Trevisan [37]. Given a graph $G = (V, E)$, a subgraph S and one of its partitions $S = (L, R)$, the bipartiteness $\beta(S_{L,R})$ of S under partition (L, R) is defined to be

$$\beta(S_{L,R}) = \frac{2e(L, L) + 2e(R, R) + e(S, V \backslash S)}{\text{vol}(S)}.$$

The bipartiteness $\beta(S)$ of the subgraph S is the minimum value of $\beta(S_{L,R})$ over all its possible partitions (L, R). Intuitively speaking, if $\beta(S)$ is small, then there must exist a partition (L, R) such that the number of edges from S to the outside as well as the number of edges that lie entirely in L or R is relatively small compared with all the edges involved with S. Thus, (L, R) can be seen close to a dense bipartite subgraph and S can be seen as a good web community. Using the bipartiteness as the measure of a set being dense and bipartite-like has the advantage that it unifies both properties in a natural way and admits theoretical analysis, which is difficult for many other measures.

Our first result is a local exploration algorithm for finding a subgraph with low bipartiteness around a starting vertex v. In particular, we show that for any subgraph S with bipartiteness ratio θ and volume at most K, there exists a subset $S_\theta \subseteq S$ such that $\text{vol}(S_\theta) \geq \text{vol}(S)/9$ and for any $v \in S_\theta$, our algorithm finds a subgraph of volume $O(K^2\Delta/\theta^2)$ and bipartiteness $O(\sqrt{\theta})$ and runs in time $O(\frac{K^2}{\theta^2}(\log \frac{K}{\theta} + \Delta))$, where Δ is the maximum degree of the graph.

We also give an algorithm to test if every small subgraph in the given graph G has high bipartiteness. More precisely, G is called a (K, θ)-non-dense bipartite graph (abbreviated as (K, θ)-NDBGraph) if every vertex subset of volume at most K has bipartiteness at least θ, and G is called ϵ-far from any (K, θ)-NDBGraph if we have to modify at least ϵ fraction of edges to obtain a (K, θ)-NDBGraph. We give a one-sided tester that distinguish between a (K, θ)-NDBGraph and a graph that is ϵ-far from any (k, τ)-NDBGraph, where $\frac{\Delta k^6 \ln k}{\theta^4} = O(K)$ and $\tau = O(\theta^2)$. Further, whenever the algorithm rejects a graph G, it provides a certificate that G is not a (K, θ)-NDBGraph in form of a subset of volume at most K and bipartiteness θ. The running time of our tester is $O(\frac{K \log K}{\epsilon})$.

1.1. Our techniques

Our local algorithms are based on Trevisan's spectral algorithm for bipartiteness [37], which shows that if the graph contains a subset with bipartiteness θ, then a *sweep process* over the largest eigenvector $\mathbf{v}$ of the Laplacian matrix $\mathcal{L}$ of the graph finds a subgraph with bipartiteness $O(\sqrt{\theta})$. On the other hand, the largest eigenvector $\mathbf{v}$ can be computed by the power method which starts from a randomly chosen vector $\mathbf{q}_0$ and then iteratively multiplies the vector $\mathbf{q}_t$ by $\mathcal{L}$ to obtain $\mathbf{q}_{t+1}$. It is known that once the number of iteration is sufficiently large, say T, $\mathbf{q}_T$ is a good approximation to $\mathbf{v}$.

Therefore, we can find a set with low bipartiteness as follows: start from some properly chosen starting vector $\mathbf{q}_0$, which is a scaled indicator vector of some vertex v; then at each iteration $t \geq 1$, first compute $\mathbf{q}_{t+1} = \mathbf{q}_t\mathcal{L}$ and then sweep over $\mathbf{q}_t$ to see if there is a set with low bipartiteness. By combining Trevisan's spectral algorithm and the spectral analysis, we can show that if the graph contains a set of bipartiteness θ, then there exists a large subset of *useful* vertices, such that if the starting vertex v is useful, we will find a set of bipartiteness $\sqrt{\theta}$ as t is large enough.

The main problem about the above simple algorithm is that the naive computation of $\mathbf{q}_t\mathcal{L}$ by vector-matrix multiplication is global, and it can not be guaranteed to just explore a small portion of the graphs as required by the local algorithms. To design local algorithms, a simple but useful observation is that to compute $\mathbf{q}_{t+1} = \mathbf{q}_t\mathcal{L}$, we only need to keep track of the support $\mathrm{supp}(\mathbf{q}_t)$ of each $\mathbf{q}_t$, that is the set of vertices on which $\mathbf{q}_t$ is nonzero, and then the computation of $\mathbf{q}_{t+1}$ can be done in time proportional to $O(\mathrm{vol}(\mathrm{supp}(\mathbf{q}_t)))$ by the definition of the Laplacian $\mathcal{L}$. Therefore, we will try to use a different sequence of vectors instead of $\{\mathbf{q}_t\}$ such that each new vector is a good approximation of $\mathbf{q}_t$ as well as has a small support set.

In our local exploration algorithm, we introduce a *truncation operation* for any vector $\mathbf{x}$, which keeps a small fraction of non-zero elements of $\mathbf{x}$ by truncating elements of relatively small absolute values. More generally, we introduced a *truncation process* to simulate the power method for $\mathcal{L}$. Once we have find a good approximation vector $\tilde{\mathbf{q}}_t$ of small support to $\mathbf{q}_t$, we will first compute $\tilde{\mathbf{q}}_t\mathcal{L}$ and then truncate this vector to obtain $\tilde{\mathbf{q}}_{t+1}$, which guarantees that $\tilde{\mathbf{q}}_{t+1}$ is also a good approximation to $\mathbf{q}_{t+1}$ and has small support. By sweeping over $\tilde{\mathbf{q}}_t$, we can ensure both that a set with low bipartiteness can be found and the volume of the set vertices that has been traversed is small, which gives our local exploration algorithm. We remark that similar

truncation process has also been used in previous local exploration algorithms (eg.,[35, 4]).

To give a tester for (K, θ)-NDBGraph, we will perform the lazy random walks on the graph and use the information of the empirical probability distribution to approximate $\mathbf{q}_t$. Such an approximation is good as long as the number of lazy random walks is large enough. It can also be shown that the obtained approximation vector $\bar{\mathbf{q}}_t$ also has small support set. Our tester then sweeps over these vectors. It is easy to see that if the graph is a (K, θ)-NDBGraph, then no subgraph with volume K and bipartiteness θ will be found. If the input graph is ϵ-far from any (k, τ)-NDBGraph, we can show that there exists a large subset of useful vertices and we can find one such vertex by sampling a small fraction of edges. Finally, by the analysis similar to the local exploration algorithm, we know that the sweep process of $\bar{\mathbf{q}}_t$ starting from some useful vertex will find a set of volume at most K and bipartiteness at most θ.

By the above discussion, we can see that the two local algorithms are coherently connected to each other, which is very interesting since traditionally the local exploration algorithms and property testing algorithms are treated very differently and almost no connections are known, partly due to the reason that the former is on the property of subgraphs while the latter is on the property of the graph itself. Our local algorithms may hint a general framework of how to convert one local algorithm into another type for other problems.

1.2. Other related works

Previous work on extracting dense bipartite subgraphs from the web graph have used different measures and mainly focused on giving heuristic methods (e.g., [21, 1, 12, 11]). All of them did not give theoretical analysis on the performance of the corresponding algorithms on general graphs.

The definition of bipartiteness is closely related to the notion of conductance and dense subgraphs. The conductance of a vertex subset S is defined as $\frac{e(S,V\setminus S)}{\min\{\mathrm{vol}(S),\mathrm{vol}(V\setminus S)\}}$. A set of small conductance can be thought of a good community as the connections crossing the set are relatively smaller than the total number of edges involved with the set. In particular, Kannan, Vempala and Veta gave a bicriteria measure of the quality of clustering based on the concept of conductance and analyzed a corresponding spectral algorithm [17]. For the literature on dense subgraphs, Kannan and Vinay defined

$d(S,T) = \frac{e(S,T)}{\sqrt{|S||T|}}$ as the measure of the density of a subgraph induced on $S \cup T$ in a directed graph and gave a spectral algorithm for finding subgraphs with large density [18]. Other density measures are also extensively studied. For example, Goldberg [13] introduced the average degree as the density measure of a set S, that is, $d(S) = \frac{e(S,S)}{|S|}$. Though both conductance and the density provide us good measures to study the communities of the networks, they do not give us any information on the bipartiteness of these subgraphs, which is the main motivation of the paper.

The testing algorithm for conductance has been studied in bounded degree models [14, 10, 28, 16] and in the general graph models [24, 26]. In particular, the small set expansion tester in [26] also uses the techniques from the corresponding local exploration algorithm.

As mentioned above, the measure we are using here was introduced by Trevisan, who found its deep connections with the Max Cut problem, the Cheeger inequality and the Geomans-Williamson Relaxation [37]. Soto [32] and Kale and Seshadhri [15] gave further analysis on the quantity that is related to the bipartiteness. Both of their work are motivated by designing approximation algorithms for Max Cut.

This article extends the work of [30] by giving a tester for (K, θ)-NDBGraph as an application of the local exploration algorithm and in a follow-up paper [25], we give a bicriteria approximation algorithm for finding small subsets with low bipartiteness using the power method for the *quasi-Laplacian* of the graph and a different analysis, which allows us to give a better local exploration algorithm as well as a spectral characterization of small dense bipartite subgraphs. We remark that the techniques for the testing algorithm developed in this paper can also be combined with the improved local exploration algorithm to give a better tester.

1.3. Organization

In section 2, we give the basic definitions of the problem and introduce spectral tools and some processes that will be used in our algorithms. Then we give our local exploration algorithm and its analysis in Section 3. The design and analysis of the tester for testing if the graph has no small dense bipartite-like subgraph for general graphs is given in Section 4. Finally, we give some concluding remarks in Section 5.

2. Preliminaries

2.1. Notations and definitions

Let $G = (V, E)$ be an undirected graph. Let $n := |V|$ and $m := |E|$. We let d_v denote the degree of vertex v and let Δ denote the maximum degree of G. We assume that we can access to G by an oracle. In the setting of both the local exploration algorithms and testing algorithms, we are allowed to perform the following two queries to the oracle: the *degree queries* for which we can query the degree of v for any vertex v, and the *neighbor queries* for which we can query the i-th neighbor of any vertex v if $i \leq d_v$. We also assume that we can sample a vertex with probability proportional to its degree in unit time in our testing algorithm.

For a subset $S \subseteq V$, we define the volume of S to be the sum of degrees of the vertices in S, denoted by $\mathrm{vol}(S) := \sum_{v \in S} d_v$. In particular, $\mathrm{vol}(G) := \mathrm{vol}(V) = 2m$. For any two vertex subsets L and R, let $e(L, R)$ denote the number of edges between L and R. We will let S denote subgraphs induced on the vertex set S and also let $S = (L, R)$ denote the subgraphs induces on the $S = L \cup R$.

Definition 2.1. For any subgraph S and a partition (L, R) of S, that is, $L \cup R = S$ and $L \cap R = \emptyset$, the bipartiteness ratio $\beta(S_{L,R})$ of S under partition (L, R) is defined as

$$\beta(S_{L,R}) = \frac{2e(L, L) + 2e(R, R) + e(S, V \backslash S)}{\mathrm{vol}(S)}.$$

We define the bipartiteness $\beta(S)$ of the subgraph S to be the minimum value of $\beta(S_{L,R})$ over all its possible partitions (L, R), that is

$$\beta(S) = \min_{(L,R)\,\text{partion of}\,S} \beta(S_{L,R});$$

and define the bipartiteness of the graph $\beta(G)$ to be the minimum value of $\beta(S)$ over all induced subgraphs in G, that is

$$\beta(G) = \min_{S} \beta(S).$$

For two disjoint subsets L, R, we will call $S := (L, R)$ a *pair subgraph* and use $\beta(L, R)$ to denote $\beta(S_{L,R})$. We are interested in local exploration algorithm for finding set with low bipartiteness and property testing algorithm for testing if every small subgraph in the graph is *not* dense bipartite-like, which is formally defined as follows.

Definition 2.2. A graph G is a (K, θ)-non-dense bipartite graph (abbreviated as (K, θ)-NDBGraph) if every subgraph in G of volume at most K has bipartiteness at least θ.

We will use the following notion of being far from a (K, θ)-NDBGraph.

Definition 2.3. A graph G is ϵ-far from any (K, θ)-NDBGraph if one has to modify at least ϵm edges of G to obtain a (K, θ)-NDBGraph.

2.2. Laplacian and related quantities

For a graph G, let A denote its adjacency matrix and let D denote its diagonal matrix of vertex degrees. For a given matrix M, we let M_S denote the matrix of M restricted on S, that is, $M_S(u, v) = M(u, v)$ if $u, v \in S$ and $M_S(u, v) = 0$ otherwise.

The *(normalized) Laplacian* of the graph is defined to be $\mathcal{L}_G := I - D^{-1/2}AD^{-1/2}$. When it is clear, we will omit the subscript G. For any subset $S \subseteq V$, it is easy to see that $\mathcal{L}_S$ is the Laplacian of subgraph S for which the dimension has been extended to n by adding corresponding zero entries. We will call $\mathcal{L}_S$ the restricted Laplacian on S. Let $s = |S|$ and $2 = \lambda_{1,S} \geq \cdots \geq \lambda_{s,S} = \lambda_{s+1,S} = \cdots = \lambda_{n,S} = 0$ be the eigenvalues of $\mathcal{L}_S$ [9] and let $\mathbf{v}_{1,S}, \mathbf{v}_{2,S}, \cdots, \mathbf{v}_{s,S}, \mathbf{v}_{s+1,S}, \cdots, \mathbf{v}_{n,S}$ be the corresponding orthonormal left eigenvectors. Since $\mathcal{L}_S$ is a restriction of $\mathcal{L}$ on S, we can assume that for all i such that $i \leq s$, the support of $\mathbf{v}_{i,S}$'s are contained in S, and for all i such that $s < i \leq n$, the support of $\mathbf{v}_{i,S}$'s are contained in $V - S$. Furthermore, the vectors obtained by restricting the first s eigenvectors on S form an orthonormal basis of $\mathbb{R}^S$ and the vectors obtained by restricting the last $n - s$ eigenvectors on $V - S$ form an orthonormal basis of $\mathbb{R}^{V \setminus S}$.

All the vectors mentioned in the paper are row vectors. For a vector $\mathbf{x}$ on the vertex set V, let $\|\mathbf{x}\|_2$ denote its Euclid norm. For any subset S, let $\mathbf{x}(S) := \sum_{v \in S} \mathbf{x}(v)$ and let $\mathbf{x}_S$ denote the vector of $\mathbf{x}$ restricted on S, that is, $\mathbf{x}_S(v) = \mathbf{x}(v)$ if $v \in S$ and $\mathbf{x}_S(v) = 0$ otherwise. Any vector $\mathbf{x} \in \mathbb{R}^V$ can be expressed in terms of the eigenvectors of $\mathcal{L}_S$ so that $\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_{i,S}$. For any nonnegative number $\sigma \leq 2$, let $H_{\sigma,S} := \{i | \lambda_{i,S} \geq 2 - \sigma\}$. The following notion of $H_{\sigma,S}$-norm is very useful in the analysis of our algorithms.

Definition 2.4. For any vector $\mathbf{x}$ such that $\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{v}_{i,S}$, the $H_{\sigma,S}$-norm $\|\mathbf{x}\|_{H_{\sigma,S}}$ of x is defined to be the length of the projection of $\mathbf{x}$ onto the subspace spanned by the first $|H_{\sigma,S}|$ eigenvectors; that is, $\|\mathbf{x}\|_{H_{\sigma,S}} = \sqrt{\sum_{i \in H_{\sigma,S}} \alpha_i^2}$.

It is well known that $\|\mathbf{x}\|_{H_{\sigma,S}}$ is a semi-norm [7]. Note that $\|\mathbf{x}\|_2 = \|\mathbf{x}\|_{H_{2,S}}$ and that if $\sigma < 1$, then $|H_{\sigma,S}| \leq s$ and thus $\|\mathbf{x}\|_{H_{\sigma,S}} \leq \sqrt{\sum_{i \leq s} \alpha_i^2} = \sqrt{\sum_{i \leq s} \langle \mathbf{x}, \mathbf{v}_{i,S}^2 \rangle} = \|\mathbf{x}_S\|_2$.

Fact 2.1. *For any $\boldsymbol{x} \in \mathbb{R}^V$ and $\sigma < 1$, $\|\boldsymbol{x}\mathcal{L}\|_{H_{\sigma,S}} \geq (2-\sigma)\|\boldsymbol{x}\|_{H_{\sigma,S}}$.*

Proof. Let $\mathbf{x} = \sum_{i=1}^n \alpha_i \mathbf{v}_{i,S}$. Consider $i \in H_{\sigma,S}$. Since the support of $\mathbf{v}_{i,S}$ is contained in S and the corresponding elements in the matrix $\mathcal{L} - \mathcal{L}_S$ are all zero, then

$$\langle \mathbf{x}\mathcal{L}, \mathbf{v}_{i,S} \rangle = \langle \mathbf{x}(\mathcal{L} - \mathcal{L}_S) + \mathbf{x}\mathcal{L}_S, \mathbf{v}_{i,S} \rangle = \langle \mathbf{x}\mathcal{L}_S, \mathbf{v}_{i,S} \rangle.$$

Therefore,

$$\begin{aligned}
\|\mathbf{x}\mathcal{L}\|_{H_{\sigma,S}} &= \sqrt{\sum_{i \in H_{\sigma,S}} |\langle x\mathcal{L}, \mathbf{v}_{i,S} \rangle|^2} \\
&= \|x\mathcal{L}_S\|_{H_{\sigma,S}} \\
&= \sqrt{\sum_{i \in H_{\sigma,S}} \lambda_{i,S}^2 \alpha_i^2} \\
&\geq (2-\sigma) \sqrt{\sum_{i \in H_{\sigma,S}} \alpha_i^2} \\
&= (2-\sigma)\|x\|_{H_{\sigma,S}}.
\end{aligned}$$

□

From now on, when σ, S is clear from context, we will use $\|\mathbf{x}\|_H$ to denote $\|\mathbf{x}\|_{H_{\sigma,S}}$.

We will let $\text{supp}(\mathbf{x})$ denote the support of a vector $\mathbf{x}$ (the set of vertices on which $\mathbf{x}$ is non-zero). Let $\mathbf{1}_v$ denote the indicator vector on v.

2.3. Three processes

We will use the following truncation process to keep our algorithms local. Similar processes have also been used in the design of other local exploration algorithms ([33, 4]).

Definition 2.5. Given a vector $\mathbf{x}$ and a nonnegative real number ξ, the truncated vector of $\mathbf{x}$ with parameter ξ is defined as

$$[\mathbf{x}]_\xi(u) = \begin{cases} \mathbf{x}(u) & \text{if } |\mathbf{x}(u)| \geq \xi \|\mathbf{x}\|_2 \\ 0 & \text{otherwise} \end{cases}$$

Note that for a given vector $\mathbf{x}$, since the absolute value of $[\mathbf{x}]_\xi$ is at least $\xi\|\mathbf{x}\|_2$ whenever it is nonzero, and $\|\mathbf{x}\|_2^2 \geq \|[\mathbf{x}]_\xi\|_2^2$, we have that the number of nonzero entries in $[\mathbf{x}]_\xi$ is at most $1/\xi^2$. Therefore, we have the following fact.

Fact 2.2. *For any $\boldsymbol{x}$ and $\xi > 0$, $|supp([\boldsymbol{x}]_\xi)| \leq 1/\xi^2$.*

Definition 2.6. Given a vector $\mathbf{x} = \mathbf{x}_0$ and a set of real numbers $\xi_t \in (0, 1]$ for $t \leq T$, the *truncation process* with staring vector $\mathbf{x}_0$ and parameters $\{\xi_t\}$ is defined to be the process that generates a sequence of vectors $\mathbf{x}_0, \cdots, \mathbf{x}_T$ such that $\mathbf{x}_{t+1} = [\mathbf{x}_t\mathcal{L}]_{\xi_{t+1}}$.

Note that the truncation process is just an approximation to the power method for the Laplacian $\mathcal{L}$, which iteratively multiplies $\mathbf{x}_0$ by $\mathcal{L}$ without truncation and approximates well to the largest eigenvector of $\mathcal{L}$ as T large.

The following *sweep* process will be used to find a subgraph with low bipartiteness from a given vector.

Definition 2.7. Given a vector $\mathbf{x} \in \mathbb{R}^V$ and its support set such that $|\text{supp}(\mathbf{x})| = s$, the sweep process over vector $\mathbf{x}$ is defined as follows:

1. Order the vertices in the support of $\mathbf{x}$ so that
$$\frac{|\mathbf{x}(v_1)|}{\sqrt{d_{v_1}}} \geq \frac{|\mathbf{x}(v_2)|}{\sqrt{d_{v_2}}} \geq \cdots \geq \frac{|\mathbf{x}(v_s)|}{\sqrt{d_{v_s}}}.$$
2. For each $i \leq t$, define the sweep set of the first i vertices as $S_i := (L_i, R_i)$, where $L_i := \{v_j : \mathbf{x}(v_j) > 0 \text{ and } j \leq i\}$ and $R_i := \{v_j : \mathbf{x}(v_j) \leq 0 \text{ and } j \leq i\}$. Compute $\beta(S_i)$.
3. Output the set $S_m = (L_m, R_m)$ that achieves the minimum bipartiteness among all the s sweep sets. Let $\beta(\mathbf{x}) = \beta(L_m, R_m)$.

The *the lazy random walk* process will be used in the design of our testing algorithm. A lazy random walk simulates a stochastic process that at each step if we are at some vertex v, then in the next step we stay at v with

probability $1/2$ and otherwise moves to a random incident edge (v, w) with probability $1/2d_v$. The transition probability matrix of the *lazy random walk* on G is defined to be $W := (I + D^{-1}A)/2$.

Another useful way to view a lazy random walk of length t from some vertex v is as follows. We first flip an unbiased coin t times and if the number of Heads seen is h, then we perform a simple (non-lazy) random walk of length h. In a simple random walk step, we moves to a randomly chosen neighbor of the current vertex such that each neighbor is chosen with equal probability. We call h the *hop-length* of the walk.

3. A local exploration algorithm for finding subgraph with low bipartiteness

3.1. Description of the algorithm `FindDB`

Now we describe our local exploration algorithm `FindDB` (short for "find dense bipartite") as follows.

`FindDB`(v, K, θ)

Input: A vertex v, a target volume K and a target bipartiteness $\theta < 1/4$.
Output: A subgraph (X, Y).

1. Let $\tilde{\mathbf{q}}_0 = \frac{\mathbf{1}_v}{\sqrt{d_v}}$, $T = \log_{f(\theta)}(8K)$, and $\xi_t = \frac{\theta f(\theta)^{t/2}}{(1-3\theta)\sqrt{8K}}$, where $f(\theta) = \frac{(1-3\theta)^2}{1-8\theta} > 1$.
2. Compute $\tilde{\mathbf{q}}_1, \cdots, \tilde{\mathbf{q}}_T$ of the truncated process with starting vector $\tilde{\mathbf{q}}_0$ and parameters $\xi_1, \cdots, \xi_T$.
3. For each time $t \leq T$, sweep over $\tilde{\mathbf{q}}_t$ and find the subgraph (X_t, Y_t) such that $\beta(X_t, Y_t) = \beta(\tilde{\mathbf{q}}_t)$. Output the subgraph with the smallest bipartiteness among all such pairs.

Our main theorem about the algorithm is the following.

Theorem 3.1. *If $S = (L, U)$ is a subgraph with volume $vol(S) \leq K$ and bipartiteness $\beta(S_{L,U}) \leq \theta$, then there exists a subset $S_\theta \subseteq S$ satisfying that $vol(S_\theta) \geq vol(S)/9$, and for any $v \in S_\theta$, the algorithm* `FindDB`(v, K, θ) *outputs a subset (X, Y) with volume $vol(X \cup Y) \leq O(K^2\Delta/\theta^2)$ and bipartiteness $\beta(X, Y) \leq 2\sqrt{2\theta}$. The running time of* `FindDB`(v, K, θ) *is $O(\frac{K^2}{\theta^2}(\log \frac{K}{\theta} + \Delta))$.*

Remark. we can make the bound condition on the bipartiteness $\theta < 1/4$ be $\theta < 1 - \delta$, for any constant δ smaller than 1, just with a different (constant fraction) bound on $\mathrm{vol}(S_\theta)$.

Let us first prove the running time of the algorithm. Note that given the support of any vector $\mathbf{x}$, the time to compute $\mathbf{x}\mathcal{L}$ is bounded by the volume of the degrees of the vertices in $\mathrm{supp}(x)$, which is $O(\mathrm{vol}(\mathrm{supp}(\mathbf{x}))) = O(s\Delta)$, where $s = |\mathrm{supp}(x)|$. On the other hand, the sweep process over vector $\mathbf{x}$ involves sorting the vertices in the support of $\mathbf{x}$, which takes time $O(s \log s)$, and computing the bipartiteness of each sweep set, which takes time $O(\mathrm{vol}(\mathrm{supp}(\mathbf{x}))) = O(\Delta s)$. Therefore, for each $t \leq T$, the time of computing $\tilde{\mathbf{q}}_t$ and sweeping over $\tilde{\mathbf{q}}_t$ is $O(|\mathrm{supp}(\tilde{\mathbf{q}}_t)|(\log |\mathrm{supp}(\tilde{\mathbf{q}}_t)|+\Delta))$, which is $O(\frac{1}{\xi_t^2}(\log \frac{1}{\xi_t^2} + \Delta)) = O(\frac{K^2}{\theta^2 f(\theta)^t}(\log \frac{K}{\theta} + \Delta))$ by Fact 2.2. By summing over all the T iterations and noting that $f(\theta) > 1$, we know that total running time of the algorithm is $O(\frac{K^2}{\theta^2}(\log \frac{K}{\theta} + \Delta))$.

3.2. Proof of the correctness of `FindDB`

In this section, we prove the correctness of the algorithm `FindDB` and thus finish the proof of the remaining part of Theorem 3.1.

The outline of the proof is as follows. We will first show Trevisan's spectral algorithm that if a vector $\mathbf{x}$ is has large *Rayleigh quotient*, which is defined as $\mathbf{x}\mathcal{L}\mathbf{x}^T/\|\mathbf{x}\|^2$, that is, $\mathbf{x}$ is close to the largest eigenvector of $\mathcal{L}$, then the sweep over $\mathbf{x}$ will produce a good subgraph with low bipartiteness (see Lemma 3.2). Then we will show that if there is subset S with low bipartiteness, then there are "many" *useful* vertices that can be used as a good starting vector of a truncated version of the power method (see Lemma 3.3), which will produce a vector that has large Rayleigh quotient.

3.2.1. A spectral algorithm

The following lemma relates a vector with large Rayleigh quotient to the bipartiteness of a given graph, which has been proved by Trevisan [37], and further analyzed by Soto [32] and Kale and Seshadhri [15]. Here, we give a self-contained proof that is somewhat different from the previous proofs. In fact, former proofs of the lemma all proceed by designing and analyzing a probabilistic algorithm. Instead, we prove the lemma by directly analyzing the deterministic version of the algorithm, which provides us more insight on the combinatorial property of the bipartiteness and may be of independent interest.

Lemma 3.2. *For any graph G and $\Theta < 1$, if there exists a vector $\boldsymbol{x} \in \mathbb{R}^V$ such that $\boldsymbol{x}\mathcal{L}\boldsymbol{x}^T \geq (2-\Theta)\|\boldsymbol{x}\|^2$, then the sweep over $\boldsymbol{x}$ produces a subgraph (X, Y) with bipartiteness $\beta(X, Y) \leq \sqrt{2\Theta}$.*

Proof. Let $\mathbf{z} = \mathbf{x}D^{-1/2}$. Let $u \sim v$ denote that $(u, v) \in E$ and let $\bar{S}$ denote $V \backslash S$. By the condition of the lemma, we have that $\mathbf{x}(2I - \mathcal{L})\mathbf{x}^T \leq \Theta\|\mathbf{x}\|^2$ and thus that

$$\begin{aligned}
\Theta &\geq \frac{\mathbf{x}(I + D^{-1/2}AD^{-1/2})\mathbf{x}^T}{\|\mathbf{x}\|^2} \\
&= \frac{\mathbf{z}(D+A)\mathbf{z}^T}{\langle \mathbf{z}, \mathbf{z}D\rangle} \\
&= \frac{\sum_{u\sim v}(\mathbf{z}(u)+\mathbf{z}(v))^2}{\sum_{v\in V}\mathbf{z}^2(v)d_v} \\
&= \frac{\sum_{u\sim v}(\mathbf{z}(u)+\mathbf{z}(v))^2 \sum_{u\sim v}(|\mathbf{z}(u)|+|\mathbf{z}(v)|)^2}{\sum_{v\in V}\mathbf{z}^2(v)d_v \sum_{u\sim v}(|\mathbf{z}(u)|+|\mathbf{z}(v)|)^2} \\
&\geq \frac{(\sum_{u\sim v}|\mathbf{z}(u)+\mathbf{z}(v)|(|\mathbf{z}(u)|+|\mathbf{z}(v)|))^2}{2(\sum_{v\in V}\mathbf{z}^2(v)d_v)^2}, \qquad (1)
\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality.

Assume the support of $\mathbf{x}$ has size s. We perform a sweep over $\mathbf{x}$ so that $\frac{|\mathbf{x}(v_1)|}{\sqrt{d_{v_1}}} \geq \frac{|\mathbf{x}(v_2)|}{\sqrt{d_{v_2}}} \geq \cdots \geq \frac{|\mathbf{x}(v_s)|}{\sqrt{d_{v_s}}}$. Equivalently, we have $|\mathbf{z}(v_1)| \geq |\mathbf{z}(v_2)| \geq \cdots \geq |\mathbf{z}(v_s)|$.

Let $L_i = \{v_j : j \leq i, \mathbf{z}(v_j) > 0\}, R_i = \{v_j : j \leq i, \mathbf{z}(v_j) \leq 0\}$ and $S_i = L_i \cup R_i$. Recall that $\beta(\mathbf{x}) = \min_i \beta(L_i, R_i)$. Then we have for any i, $\beta(\mathbf{x})\mathrm{vol}(S_i) \leq 2e(L_i, L_i) + 2e(R_i, R_i) + e(S_i, \bar{S}_i)$.

Now we consider the square root of the numerator of (1) to obtain

$$\begin{aligned}
& \sum_{u\sim v} |\mathbf{z}(u)+\mathbf{z}(v)|(|\mathbf{z}(u)|+|\mathbf{z}(v)|) \\
\geq & \sum_{u\sim v,\mathbf{z}(u)\mathbf{z}(v)<0} |\mathbf{z}^2(u)-\mathbf{z}^2(v)| + \sum_{u\sim v,\mathbf{z}(u)\mathbf{z}(v)\geq 0} (\mathbf{z}(u)+\mathbf{z}(v))^2 \\
\geq & \sum_{\substack{i<j,v_i\sim v_j,\\ \mathbf{z}(v_i)\mathbf{z}(v_j)<0}} (\mathbf{z}^2(v_i)-\mathbf{z}^2(v_j)) + \sum_{\substack{i<j,v_i\sim v_j,\\ \mathbf{z}(v_i)\mathbf{z}(v_j)\geq 0}} (\mathbf{z}^2(v_i)+\mathbf{z}^2(v_j)) \qquad (2) \\
= & \sum_{i=1}^{s} (\mathbf{z}^2(v_i)-\mathbf{z}^2(v_{i+1}))(2e(L_i,L_i)+2e(R_i,R_i)+e(S_i,\bar{S}_i)) \qquad (3) \\
\geq & \beta(\mathbf{x}) \sum_{i=1}^{s} (\mathbf{z}^2(v_i)-\mathbf{z}^2(v_{i+1}))\mathrm{vol}(S_i) \\
= & \beta(\mathbf{x}) \sum_{i=1}^{s} \mathbf{z}^2(v_i) d_{v_i}, \qquad (4)
\end{aligned}$$

where we define $\mathbf{z}(v_{n+1})$ to be 0 if $s = n$. The main difficulty lies in the third equation, which can be obtained by comparing the coefficient of $\mathbf{z}^2(v_k)$ on both sides for every $k \leq n$ and we defer the proof of it at the end. Now assuming that formula (2) is equivalent to formula (3) and thus inequality (4) holds, we have that

$$\Theta \geq \frac{\beta(\mathbf{x})^2(\sum_{v\in V}\mathbf{z}^2(v)d_v)^2}{2(\sum_{v\in V}\mathbf{z}^2(v)d_v)^2} = \frac{\beta(\mathbf{x})^2}{2},$$

and the lemma follows if we set $(X,Y) = (L_m, R_m)$ for which the bipartiteness achieves $\beta(\mathbf{x})$.

Now we show that formula (2) is equivalent to formula (3). Let $\mathrm{coef}_1(k)$ and $\mathrm{coef}_2(k)$ be the coefficient of $\mathbf{z}^2(v_k)$ in (2) and (3), respectively. We only need to show that for each $k \leq n$, $\mathrm{coef}_1(k) = \mathrm{coef}_2(k)$. Assume that $\mathbf{z}(v_k) \leq 0$. The case when $\mathbf{z}(v_k) > 0$ is similar.

By definition and our assumption that $\mathbf{z}(v_k) \leq 0$, we know that $L_{k-1} = L_k$ and $R_k = R_{k-1} \cup \{v_k\}$ (see Figure 1). It is easy to see that only edges incident to vertex v_k can contribute to $\mathrm{coef}_1(k)$. More specifically, for each edge $u \sim v_k$, if $u \in R_{k-1} \cup \bar{S}_k$, it contributes 1 to $\mathrm{coef}_1(k)$ and if $u \in L_{k-1}$, it contributes -1 to $\mathrm{coef}_1(k)$. Totally, we have $\mathrm{coef}_1(k) = e(\{v_k\}, R_{k-1}) + e(\{v_k\}, \bar{S}_k) - e(\{v_k\}, L_{k-1})$.

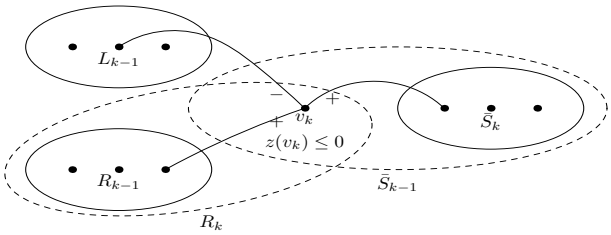

Figure 1: The case when $\mathbf{z}(v_k) \le 0$. The sign on an edge denotes whether it contributes 1 or -1 to the coefficient $\mathrm{coef}_1(k)$ of $\mathbf{z}^2(v_k)$ in (2)

On the other hand, from (3), we can get that

$$\begin{aligned}
\mathrm{coef}_2(k) &= (2e(L_k, L_k) + 2e(R_k, R_k) + e(S_k, \bar{S}_k)) \\
&\quad -(2e(L_{k-1}, L_{k-1}) + 2e(R_{k-1}, R_{k-1}) + e(S_{k-1}, \bar{S}_{k-1})) \\
&= 2e(\{v_k\}, R_{k-1}) + e(S_{k-1}, \bar{S}_k) + e(\{v_k\}, \bar{S}_k) \\
&\quad -e(S_{k-1}, \{v_k\}) - e(S_{k-1}, \bar{S}_k) \\
&= 2e(\{v_k\}, R_{k-1}) + e(\{v_k\}, \bar{S}_k) - e(S_{k-1}, \{v_k\}) \\
&= e(\{v_k\}, R_{k-1}) + e(\{v_k\}, \bar{S}_k) - e(\{v_k\}, L_{k-1}) \\
&= \mathrm{coef}_1(k).
\end{aligned}$$

This completes the proof. □

3.2.2. The prevalence of useful starting vertices

We now show that for any given subgraph S with bipartiteness θ, there exists a large subset $S_\theta \subseteq S$ of *useful* vertices which are defined as follows.

Definition 3.1. A vertex v is called σ-useful with respect to S if

$$\|\mathbf{1}_v/\sqrt{d_v}\|_{H_{\sigma,S}} \ge 1/\sqrt{8\mathrm{vol}(S)}.$$

We will show in the next section that the truncation process with starting vector $\mathbf{1}_v/\sqrt{d_v}$ with appropriate parameters $\{\xi_t\}_{t=1}^T$ for a useful vertex v is guaranteed to produce a vector satisfying the condition of Lemma 3.2 for properly chosen T. In the following, we show there is a large subset S_θ of 4θ-useful vertices within a set S of bipartiteness θ.

Lemma 3.3. *If $S = (L, R)$ is a subgraph with bipartiteness $\beta(L, R) \le \theta$, then there exists a subset $S_\theta \subseteq S$ satisfying that $vol(S_\theta) \ge vol(S)/9$, and for any $v \in S_\theta$, v is 4θ-useful with respect to S.*

Proof. By the definition of useful vertices, we need to show that

$$\|\mathbf{1}_v/\sqrt{d_v}\|_{H_{4\theta,S}} \geq 1/\sqrt{8\mathrm{vol}(S)}.$$

We will omit the subscripts of $H_{4\theta,S}$ in the proof. Define a vector ψ as follows:

$$\psi(v) = \begin{cases} \sqrt{d_v}/\mathrm{vol}(S) & \text{if } v \in L, \\ -\sqrt{d_v}/\mathrm{vol}(S) & \text{if } v \in R, \\ 0 & \text{otherwise.} \end{cases}$$

Then, we have

$$\begin{aligned} \psi(2I - \mathcal{L})\psi^T &= \psi(I + D^{-1/2}AD^{-1/2})\psi^T \\ &= \sum_{u\sim v}(\psi(u)/\sqrt{d_u} + \psi(v)/\sqrt{d_v})^2 \\ &= \frac{4e(L,L) + 4e(R,R) + e(S, V\backslash S)}{\mathrm{vol}(S)^2} \\ &\leq 2\theta/\mathrm{vol}(S) \end{aligned}$$

Now let $\psi = \sum_{i=1}^n \alpha_i \mathbf{v}_{i,S}$, then $\|\psi\|_2^2 = 1/\mathrm{vol}(S) = \sum_{i=1}^n \alpha_i^2$, and

$$\begin{aligned} \psi(2I - \mathcal{L})\psi^T &= \frac{2}{\mathrm{vol}(S)} - \sum_{i=1}^n \lambda_{i,S}\alpha_i^2 \\ &\geq \frac{2}{\mathrm{vol}(S)} - 2\sum_{i\in H}\alpha_i^2 - (2-4\theta)\sum_{i\notin H}\alpha_i^2 \\ &= \frac{2}{\mathrm{vol}(S)} - 2\|\psi\|_H^2 - (2-4\theta)(\frac{1}{\mathrm{vol}(S)} - \|\psi\|_H^2). \end{aligned}$$

From the above bounds, we can get

$$\|\psi\|_H^2 \geq \frac{4\theta - 2\theta}{4\theta\mathrm{vol}(S)} = \frac{1}{2\mathrm{vol}(S)}.$$

We now define $T = \{v \in S : \|\mathbf{1}_v/\sqrt{d_v}\|_2^2 < \frac{1}{8\mathrm{vol}(S)}\}$. Assume that $\mathrm{vol}(T) \geq 8\mathrm{vol}(S)/9$, we will derive a contradiction.

Define a vector η as follows:

$$\eta(v) = \begin{cases} \sqrt{d_v}/\mathrm{vol}(T) & \text{if } v \in L \cap T, \\ -\sqrt{d_v}/\mathrm{vol}(T) & \text{if } v \in R \cap T, \\ 0 & \text{otherwise.} \end{cases}$$

Then, we have

$$\|\eta\|_H^2 \;=\; \|\sum_{v\in T}\frac{d_v}{\mathrm{vol}(T)}\cdot\frac{\mathbf{1}_v}{\sqrt{d_v}}\|_H^2 \le \sum_{v\in T}\frac{d_v}{\mathrm{vol}(T)}\cdot\|\frac{\mathbf{1}_v}{\sqrt{d_v}}\|_H^2 < \frac{1}{8\mathrm{vol}(S)},$$

where the second inequality follows by the Jensen's inequality.

To get a lower bound of $\|\eta\|_H$, we first get an upper bound of $\|\eta-\psi\|_H$.

$$\begin{aligned}\|\psi-\eta\|_H^2 &\le \|\psi-\eta\|_2^2\\ &= \sum_{v\in T}(\frac{\sqrt{d_v}}{\mathrm{vol}(T)}-\frac{\sqrt{d_v}}{\mathrm{vol}(S)})^2+\sum_{v\in S\setminus T}(\frac{\sqrt{d_v}}{\mathrm{vol}(S)})^2\\ &= \frac{1}{\mathrm{vol}(T)}-\frac{1}{\mathrm{vol}(S)}\\ &\le \frac{1}{8\mathrm{vol}(S)},\end{aligned}$$

where the last inequality follows from our assumption on $\mathrm{vol}(T)$. By the triangle inequality, we have

$$\|\eta\|_H \;\ge\; \|\psi\|_H-\|\psi-\eta\|_H \ge \sqrt{\frac{1}{2\mathrm{vol}(S)}}-\sqrt{\frac{1}{8\mathrm{vol}(S)}}=\sqrt{\frac{1}{8\mathrm{vol}(S)}},$$

which contradicts the upper bound we obtained for $\|\eta\|_H$. Therefore, we must have $\mathrm{vol}(T)\le 8\mathrm{vol}(S)/9$. Let $S_\theta = S\setminus T$. We have $\mathrm{vol}(S_\theta)\ge \mathrm{vol}(S)/9$ and for any $v\in S_\theta$, $\|\mathbf{1}_v/\sqrt{d_v}\|_H^2 \ge \frac{1}{8\mathrm{vol}(S)}$.

This complete the proof of the lemma. □

3.2.3. Putting it together

Proof of Theorem 3.1. For any $S=(L,R)$ with bipartiteness $\beta(L,R)\le\theta<1/4$, we will consider the restricted Laplacian matrix $\mathcal{L}_S$ on S and the corresponding $H_{4\theta,S}$-norm (abbreviated as H-norm) of vectors. Let S_θ be the subset of 4θ-useful vertices with respect to S as described in Lemma 3.3. We show that for any vertex $v\in S_\theta$ and $K\ge\mathrm{vol}(S)$, the algorithm $\texttt{FindDB}(v,K,\theta)$ produces a subgraph (X,Y) such that $\mathrm{vol}(X\cup Y)\le O(K^2\Delta/\theta^2)$ and $\beta(X,Y)\le 4\sqrt{2\theta}$.

Let $\tilde{\mathbf{q}}_1,\cdots,\tilde{\mathbf{q}}_T$ be the vectors produced by the pruned process with starting vector $\mathbf{1}_v/\sqrt{d_v}$ and parameters $\xi_1,\cdots,\xi_T$. If there exists a vector $\tilde{\mathbf{q}}_t$ such that $\tilde{\mathbf{q}}_t\mathcal{L}\tilde{\mathbf{q}}_t^T\ge(2-16\theta)\|\tilde{\mathbf{q}}_t\|_2^2$, then the volume of the output set is at most

$O(|\mathrm{supp}(\tilde{\mathbf{q}}_t)|\Delta) \leq O(K^2\Delta/\theta^2)$ and by Lemma 3.2, we are done. Now assume that there is no such vector, that is, for all $t \leq T$, $\tilde{\mathbf{q}}_t\mathcal{L}\tilde{\mathbf{q}}_t^T < (2-16\theta)\|\tilde{\mathbf{q}}_t\|_2^2$. We will derive a contradiction.

By our assumption, for any $t \leq T$, we have

$$\begin{aligned}
\|\tilde{\mathbf{q}}_t\|_2^2 &\leq \|\tilde{\mathbf{q}}_{t-1}\mathcal{L}\|_2^2 \\
&= \tilde{\mathbf{q}}_{t-1}\mathcal{L}^2\tilde{\mathbf{q}}_{t-1}^T \\
&\leq 2\tilde{\mathbf{q}}_{t-1}\mathcal{L}\tilde{\mathbf{q}}_{t-1}^T \\
&\leq 2(2-16\theta)\|\tilde{\mathbf{q}}_{t-1}\|_2^2 \\
&\leq 2^t(2-16\theta)^t\|\tilde{\mathbf{q}}_0\|_2^2 \\
&\leq 2^t(2-16\theta)^t. \qquad (5)
\end{aligned}$$

Now we show that $\|\tilde{\mathbf{q}}_t\|_H$ increase exponentially with t such that

$$\|\tilde{\mathbf{q}}_t\|_H \geq \frac{1}{\sqrt{8\mathrm{vol}(S)}}(2-6\theta)^t. \qquad (6)$$

By the fact that $\|\tilde{\mathbf{q}}_t\|_2 \geq \|\tilde{\mathbf{q}}_t\|_H$, this contradicts equation (5) when $t = \log_{f(\theta)}(8\mathrm{vol}(S)) \leq T$, where $f(\theta) = (1-3\theta)^2/(1-8\theta) > 1$, and this completes the proof.

We will prove equation (6) by induction. When $t = 0$, it is true by the choice of v. Now let $\mathbf{r}_t = \tilde{\mathbf{q}}_{t-1}\mathcal{L} - \tilde{\mathbf{q}}_t = \tilde{\mathbf{q}}_{t-1}\mathcal{L} - [\tilde{\mathbf{q}}_{t-1}\mathcal{L}]_{\xi_t}$ be the vector that is removed from the pruned process for any t. Recall that $(\mathbf{r}_t)_S$ is the vector $\mathbf{r}_t$ restricted on S. We have

$$\|\mathbf{r}_t\|_H \leq \|(\mathbf{r}_t)_S\|_2 \leq \xi_t\|\tilde{\mathbf{q}}_{t-1}\mathcal{L}\|_2\sqrt{|S|} \leq \xi_t\sqrt{\mathrm{vol}(S)}2^t(1-8\theta)^{t/2},$$

where the last inequality follows from inequality (5) and that $|S| \leq \mathrm{vol}(S)$.

Now assume that the induction hypothesis holds for $t-1$, that is, $\|\tilde{\mathbf{q}}_{t-1}\|_H \geq$

$\frac{1}{\sqrt{8\mathrm{vol}(S)}}(2-6\theta)^{t-1}$. Then by the properties of H-norm,

$$\begin{aligned}
\|\tilde{\mathbf{q}}_t\|_H &= \|\tilde{\mathbf{q}}_{t-1}\mathcal{L} - r_t\|_H \\
&\geq \|\tilde{\mathbf{q}}_{t-1}\mathcal{L}\|_H - \|r_t\|_H \\
&\geq (2-4\theta)\|\tilde{\mathbf{q}}_{t-1}\|_H - \xi_t\sqrt{\mathrm{vol}(S)}2^t(1-8\theta)^{t/2} \\
&\geq (2-4\theta)\frac{1}{\sqrt{8\mathrm{vol}(S)}}(2-6\theta)^{t-1} - \xi_t\sqrt{\mathrm{vol}(S)}2^t(1-8\theta)^{t/2} \\
&\geq \frac{1}{\sqrt{8\mathrm{vol}(S)}}(2-6\theta)^t[\frac{2-4\theta}{2-6\theta} - \xi_t\sqrt{8}\mathrm{vol}(S)\frac{(1-8\theta)^{t/2}}{(1-3\theta)^t}] \\
&\geq \frac{1}{\sqrt{8\mathrm{vol}(S)}}(2-3\theta)^t,
\end{aligned}$$

where the last inequality follows because $\xi_t = \frac{\theta f(\theta)^{t/2}}{(1-3\theta)\sqrt{8K}}$ and $K \geq \mathrm{vol}(S)$. This completes the proof. □

4. A testing algorithm for (K, θ)-non-dense bipartite graph

4.1. Description of the tester `TestNSDB`

Our tester `TestNSDB` (short for "test non small dense bipartite") is defined as follows.

`TestNSDB`(G, K, θ, ϵ)

1. Repeat $O(1/\epsilon)$ times:
   (a) Pick a random vertex v with probability proportional to its degree.
   (b) If `DetectDB`(G, v, K, θ) returns **found** then abort and output **reject**.
2. In case no call to `DetectDB` returned **found** then output **accept**.

The subroutine `DetectDB` is defined as follows.

Theorem 4.1. *The algorithm* `TestNSDB`(G, K, θ, ϵ) *accepts the graph G if it is a (K, θ)-NDBGraph and rejects G with high probability if it is ϵ-far from any (k, τ)-NDBGraph, where k satisfies that $\frac{\Delta k^6 \ln k}{\theta^4} = O(K)$, and $\tau = O(\theta^2)$. Furthermore, whenever the algorithm rejects a graph G, it provides a*

`DetectDB`(G, v, K, θ)

1. Let k be the largest integer such that $\frac{4800000\Delta k^6 \ln k}{\theta^4} \leq K$. Let $T := \log_{g(\theta)}(8k)$ and $N := \frac{4800000k^6 \ln k}{\theta^4}$, where $g(\theta) := \frac{(1-\theta^2/20)^2}{1-\theta^2/4} > 1$.
2. Perform N lazy random walk of length T from v.
3. For each length $t = 0, 1, 2, ..., T$:
   (a) Let $N_o(u)$ (resp. $N_e(u)$) denote the number of walks of length t with odd (resp. even) hop-length that ends at vertex u. Let $\bar{\mathbf{x}}_t(u) = (N_e(u) - N_o(u))/N$.
   (b) Sweep over $\bar{\mathbf{x}}_t$ and if any sweep set has bipartiteness less than θ, then return **found**.
4. Return **not-found**.

certificate that G is not a (K, θ)-NDBGraph in form of a subset of volume at most K and bipartiteness θ. The running time of the algorithm is $O(\frac{K \log K}{\epsilon})$.

Let us first prove the running time of the algorithm. In each call to the subroutine `DetectDB`, we need to perform N random walks of length T, which takes time NT, and to sweep over all possible $\bar{\mathbf{x}}_t$ for each $t \leq T$, which takes time $O(\sum_{t \leq T} |\text{supp}(\bar{\mathbf{x}}_t)|(\log |\text{supp}(\bar{\mathbf{x}}_t)| + \Delta)) = O(TN(\log N + \Delta))$, since the support of each $\bar{\mathbf{x}}_t$ has size at most N. Therefore, the running time of `DetectDB` is $O(TN(\log N + \Delta)) = O(\frac{\Delta k^6 \ln^2 k}{\theta^4}(\ln \frac{k}{\theta} + \Delta)) = O(K \log K)$. Since there are $O(1/\epsilon)$ calls to `DetectDB`, the running time of our tester `TestNSDB` is $O(\frac{K \log K}{\epsilon})$.

Note that if the input graph G is a (K, θ)-NDBGraph, then by the fact that every sweep set computed in the algorithm has volume at most $N\Delta \leq K$, we know that `TestNSDB`(G, K, θ, ϵ) will always accept the graph.

4.2. Proof of the correctness of `TestNSDB`

By the analysis in the last section, to prove Theorem 4.1, it suffices to show that if G is ϵ-far from any (k, τ)-NDBGraph, then `TestNSDB` rejects G with high probability, where $\tau := c\theta^2$ for some small constant c to be chosen later. Therefore, from now on, we will assume that G is ϵ-far from any (k, τ)-NDBGraph.

We will first show that the power method of $\mathcal{L}$ can be approximated by using the empirical probability distribution of the lazy random walk. Then

we show that if G is ϵ-far from any (k,τ)-NDBGraph, then there is a large subset of useful vertices which may be used as a good starting vector for the approximated power method such that sweeping over these vectors will produce a set of low bipartiteness.

4.2.1. Approximation vectors from the lazy random walk

Assume that the random walks in `DetectDB` starts from some vertex v. Let $\mathbf{q}_t = \frac{\mathbf{1}_v \mathcal{L}^t}{\sqrt{d_v}}$. Kale and Seshadhri [15] found that $\mathbf{q}_t$ can be fully characterized by the probability distributions of the lazy random walk with even/odd hop-lengths. More precisely, consider a lazy random walk of length t starts vertex v. Let $\mathbf{p}_o(u)$ (resp. $\mathbf{p}_e(u)$) be the probability that u is reached by the walk with odd (resp. even) hop-length. Let $\mathbf{x}_t(u) := \mathbf{p}_e(u) - \mathbf{p}_o(u)$. The following lemma is shown in [15].

Lemma 4.2 (Claim 3.8 in [15])**.** *For any $t \geq 1$ and vertex u, $\boldsymbol{q}_t(u) = 2^t \boldsymbol{x}_t(u)$.*

On the other hand, we note that $\mathbf{p}_o(u)$ (resp. $\mathbf{p}_e(u)$) can estimated by first performing lazy random walks of length t and then computing the fraction of walks that reach u with odd (resp. even) hop-length. The step (3a) in the algorithm `TestNSDB` does exactly this job by using $N_o(u)/N$ (resp. $N_e(u)/N$) to approximate $\mathbf{p}_o(u)$ (resp. $\mathbf{p}_e(u)$) and thus using $\bar{\mathbf{x}}_t$ to approximate $\mathbf{x}_t$. Now define $\bar{\mathbf{q}}_t := 2^t \bar{\mathbf{x}}_t(u)$. The following lemma shows that by choosing proper parameters, $\bar{\mathbf{x}}_t$ is a good approximation of $\mathbf{x}_t$, and thus $\bar{\mathbf{q}}_t$ is a good approximation of $\mathbf{q}_t$.

Lemma 4.3. *Assume that v is a σ-useful vertex with respect to some subset S of volume at most k for any $0 \leq \sigma < 1$. Let $\boldsymbol{q}_t, \bar{\boldsymbol{q}}_t$ defined as above. For $t \leq T$, with probability at least $1 - k^{-4}$,*

$$|\|\bar{\boldsymbol{q}}_t\|_{H_{\sigma,S}} - \|\boldsymbol{q}_t\|_{H_{\sigma,S}}| \leq 2^t c_t, \tag{7}$$

where $c_t := \frac{\theta^2(1-\theta^2/20)^t}{400\sqrt{k}}$.

Proof. We omit the subscripts of $H_{\sigma,S}$ in the proof. Since $\sigma < 1$, then $|H| \leq |S| \leq k$, and for each $i \in H$, $|\mathrm{supp}(\mathbf{v}_{i,S})| \leq k$. We also have that

$$|\|\bar{\mathbf{q}}_t\|_H - \|\mathbf{q}_t\|_H| \leq \|\bar{\mathbf{q}}_t - \mathbf{q}_t\|_H = \sqrt{\sum_{i\in H} \langle \bar{\mathbf{q}}_t - \mathbf{q}_t, \mathbf{v}_{i,S}\rangle^2} \leq \sum_{i\in H} |\langle \bar{\mathbf{q}}_t - \mathbf{q}_t, \mathbf{v}_{i,S}\rangle|.$$

Thus, it suffices to show that for any unit vector $\mathbf{u}$ with support size at most k, the inequality

$$|\langle \bar{\mathbf{q}}_t - \mathbf{q}_t, \mathbf{u}\rangle| \leq 2^t c_t/k, \quad \text{or equivalently} \quad |\langle \bar{\mathbf{x}}_t - \mathbf{x}_t, \mathbf{u}\rangle| \leq c_t/k$$

holds with probability $1 - k^{-5}$. Also note that

$$\begin{aligned} |\langle \bar{\mathbf{x}}_t - \mathbf{x}_t, \mathbf{u}\rangle| &= |\sum_{w \in \mathrm{supp}(\mathbf{u})} (\bar{\mathbf{x}}_t(w)\mathbf{u}(w) - \mathbf{x}_t(w)\mathbf{u}(w))| \\ &\leq \sum_{w \in \mathrm{supp}(\mathbf{u})} |(\bar{\mathbf{x}}_t(w)\mathbf{u}(w) - \mathbf{x}_t(w)\mathbf{u}(w))|, \end{aligned}$$

and that $|\mathbf{u}(w)| \leq 1$ for any w. It suffices to show that for any for any γ such that $|\gamma| \leq 1$, the inequality

$$|\bar{\mathbf{x}}_t(w)\gamma - \mathbf{x}_t(w)\gamma| \leq c_t/k^2, \tag{8}$$

holds with probability $1 - k^{-6}$.

To show the above statement, we define

$$Y_i(w) = \begin{cases} \gamma & \text{if the } i\text{-th walk ends at } u \text{ with even hop-length,} \\ -\gamma & \text{if the } i\text{-th walk ends at } u \text{ with even hop-length,} \\ 0 & \text{if the } i\text{-th walk does not end at } u. \end{cases}$$

Then $\gamma\bar{\mathbf{x}}_t(w) = \frac{1}{N}\sum_{i=1}^{N} Y_i(w)$, $\mathrm{E}[Y_i(w)] = \gamma\mathbf{x}_t(w)$ and $\mathrm{E}[Y_i(w)^2] \leq \gamma^2$. By the Bernstein's inequality,

$$\begin{aligned} \Pr[|\frac{1}{N}\sum_{i=1}^{N} Y_i(w) - \gamma\mathbf{x}_t(w)|| \geq c_t/k^2] &\leq 2\exp(-\frac{c_t^2 N}{2k^4(\gamma^2 + c_t\gamma/3k^2))}) \\ &\leq k^{-9}, \end{aligned}$$

where the last inequality follows from the fact that $N = \frac{4800000k^6 \ln k}{\theta^4} \geq 30k^4 \ln k/c_T^2 \geq 30k^4 \ln k/c_t^2$. □

Lemma 4.4. *If v is a $\theta^2/16$-useful vertex with respect to some subset S of volume at most k, then* `DetectDB(`G, v, K, θ`)` *will return* ***found****.*

Proof. Assume on the contrary that the subroutine `DetectDB(`G, v, K, θ`)` does not find a set with bipartiteness less than θ. Then, by Lemma 3.2, for

any t, $\bar{\mathbf{x}}_t \mathcal{L} \bar{\mathbf{x}}_t < (2-\theta^2/2)\|\bar{\mathbf{x}}_t\|_2^2$ and thus $\bar{\mathbf{q}}_t \mathcal{L} \bar{\mathbf{q}}_t < (2-\theta^2/2)\|\bar{\mathbf{q}}_t\|_2^2$. By the arguments similar to the proof of inequality (5),

$$\|\bar{\mathbf{q}}_t\|_2 \leq 2^t (1-\theta^2/4)^{t/2}. \tag{9}$$

Now we show that

$$\|\bar{\mathbf{q}}_t\|_{H_{\theta^2/16,S}} \geq \frac{1}{\sqrt{8k}}(2-\theta^2/10)^t. \tag{10}$$

We omit the subscripts of $H_{\theta^2/16,S}$ in the proof. By the fact that $\|\bar{\mathbf{q}}_t\|_H \leq \|\bar{\mathbf{q}}_t\|_2$, this contradicts inequality (9) when $t = T = \log_{g(\theta)}(8k^*)$, where $g(\theta) := (1-\theta^2/20)^2/(1-\theta^2/4)$.

We prove inequality (10) by induction. This is true for $t = 0$ since v is a $\theta^2/16$-useful vertex with respect to S and $\mathrm{vol}(S) \leq k$. Assume that the inequality holds for $t-1$. Then

$$\begin{aligned}
\|\bar{\mathbf{q}}_t\|_H &\geq \|\mathbf{q}_t\|_H - 2^t c_t \\
&\geq \frac{1}{\sqrt{8k}}(2-\theta^2/16)\|\mathbf{q}_{t-1}\|_H - 2^t c_t \\
&\geq \frac{1}{\sqrt{8k}}(2-\theta^2/16)(\|\bar{\mathbf{q}}_{t-1}\|_H - 2^{t-1}c_{t-1}) - 2^t c_t \\
&= \frac{1}{\sqrt{8k}}(2-\theta^2/16)(2-\theta^2/10)^{t-1} - (1-\theta^2/16)c_{t-1}2^t - 2^t c_t \\
&\geq \frac{1}{\sqrt{8k}}(2-\theta^2/10)^t,
\end{aligned}$$

where the last inequality follows from the choice of c_t. □

4.2.2. Fertility of useful vertices

In this section, we show that if G is ϵ-far from any (k,τ)-NDBGraph, then there are many useful vertices, which will guarantee that with high probability, we will sample out at least one useful vertex from which the subroutine `DetectDB` will find a subset of volume at most K and bipartiteness at most θ and the algorithm will rejects G. In the following we argue the contrapositive and will show that if there are few useful vertices, then there is a partition of vertex set of G that will enable us to modify at most ϵm edges of G to obtain a (k,τ)-NDBGraph.

Lemma 4.5. *Given a graph G, let U denote the set of all the $\theta^2/16$-useful vertices with respect to some subgraph of volume at most k. If $vol(U) \leq \epsilon m/18$, then there exists a partition of V, written as $(A_1, \cdots, A_r, B)$, such that $vol(A_i) \leq k$ and $\beta(A_i) \leq \theta^2/16$ for any $1 \leq i \leq r$, $vol(A_1 \cup \cdots \cup A_r) \leq \epsilon m/2$ and any subset of volume at most k in B has bipartiteness $\Omega(\theta^2)$, where both the volume and bipartiteness are measured in the original graph G.*

Proof. Let $V_0 := V$, $A_0 := \emptyset$. The partition can be constructed recursively as follows. For $i \geq 1$, if there exists some pair subgraph (L_i, R_i) in V_{i-1} of volume at most k and bipartiteness at most $\theta^2/16$, then let $A_i = L_i \cup R_i, V_i = V_{i-1} \backslash (A_1 \cup \cdots \cup A_i)$; otherwise, let $r = i - 1, B = V_{i-1}$ and stop.

By the construction, we know that for any vertex subset in B of volume at most k has bipartiteness $\theta^2/16$.

Now we assume that $\text{vol}(A_1 \cup \cdots \cup A_r) > \epsilon m/2$. We know that for each $i \leq r$, $\text{vol}(A_i) \leq k$ and $\beta(A_i) \leq \theta^2/16$. By Lemma 3.3, there exists subset $A_i' \subseteq A_i$ such that $\text{vol}(A_i') \geq \text{vol}(A_i)/9$ and for any $v \in A_i'$,

$$\|\mathbf{1}_v/\sqrt{d_v}\|_{H_{\theta^2/16, A_i}} \geq 1/\sqrt{8\text{vol}(A_i)}$$

which means that such a vertex v is a $\theta^2/16$-useful vertex with some subgraph with volume at most k and thus $v \in U$. Therefore, $\text{vol}(U) \geq \text{vol}(A_1' \cup \cdots \cup A_r') \geq \text{vol}(A_1 \cup \cdots \cup A_r)/9 > \epsilon m/18$, which is a contradiction. □

Lemma 4.6. *If a graph G has a vertex partition as guaranteed by Lemma 4.5, then G can be modified to be a (k, τ)-NDBGraph by changing at most ϵm edges.*

Proof. Let $A := A_1 \cup \cdots \cup A_r$. Modify the edges of G as follows. First we delete all the edges that lie entirely in A. Then for any edge (u, v) such that $u \in A$ and $v \in B$, add a new edge (w, v), where $w \in B$ is chosen randomly with probability proportional to d_w. Denote the resulting graph as G'.

It is easy to see that the number of edges that are deleted is at most $\text{vol}(A)$ and the number of newly added edges is $\text{vol}(A)$. Therefore, the number of edges changed is at most $2\text{vol}(A) \leq \epsilon m$.

Now consider an arbitrary pair subgraph $S = (L, R)$ in G', we will use $\text{vol}'(S)$, $\beta'(S)$ and $e'(L, R)$ to denote the volume, the bipartiteness of S and the number of edges between L, R in G', respectively. Let S_A, L_A, R_A denote $S \cap A, L \cap A, R \cap A$, respectively. The notion of S_B, L_B, R_B can be defined similarly. Assume that $\text{vol}'(S) = r \leq k \ll m$ for some small constant c. We

show that with high probability over the randomness of the modification process, S has high bipartiteness. Then we show that with positive probability, all the subsets in G' of volume at most k has high bipartiteness by taking a union bound over all possible bipartite subgraphs. By the probabilistic method, this completes the proof.

1. If $\mathrm{vol}'(S_A) \geq 2r/3$, then by the fact that all the edges incident to A are connected to some vertex in B, $e'(S_A, B\backslash S_B) \geq \mathrm{vol}'(S_A) - \mathrm{vol}'(S_B) \geq r/3$. Therefore,

$$\beta'(L, R) \geq \frac{2e'(S_A, B\backslash S_B)}{r} \geq 2/3 = \Omega(\theta^2).$$

2. If $\mathrm{vol}'(S_A) < 2r/3$, then $\mathrm{vol}'(S_B) \geq r/3$. Note that by construction, $\mathrm{vol}(S_B) \geq \mathrm{vol}'(S_B)/2 \geq r/6$. In the original graph G, we have that $\beta(L_B, R_B) \geq \theta^2$, that is,

   $$2e(L_B) + 2e(R_B) + e(S_B, B\backslash S_B) + e(S_B, A) \geq \theta^2 \mathrm{vol}(S_B) \geq \theta^2 r/6.$$

   In the modified graph G', we have that

   $$2e'(L) + 2e'(R) + e'(S, \bar{S}) \geq 2e(L_B) + 2e(R_B) + e(S_B, B\backslash S_B) + X,$$

   where X denotes the number of newly added edges between S_B and $B\backslash S_B$.
   Now consider the following two cases.
   (a) If $e(S_B, A) < \theta^2 r/12$, then $2e'(L) + 2e'(R) + e'(S, \bar{S}) \geq 2e(L_B) + 2e(R_B) + e(S_B, B\backslash S_B) \geq \theta^2 r/12$, which means that $\beta'(L, R) \geq \theta^2/12$.
   (b) If $e(S_B, A) \geq \theta^2 r/12$. Since for each an edge (u, v) between S_B and A, the chosen endpoint of the newly added edge lie in $B\backslash S_B$ is $\mathrm{vol}(B\backslash S_B)/2m \geq (2m - \epsilon m/2 - r)/2m \geq 1 - 5r/m$. Therefore, X can be seen as a sum of at least $\theta^2 r/12$ independent $0-1$ random variables each of which takes value 1 with probability at least $1 - 5r/m$. Then by Chernoff bound, we know that the probability that X is smaller than $\theta^2 r/24$ is $O((m/r)^{-\theta^2 r/24})$, which is also an upper bound of the probability of the event $\beta'(L, R) \leq c\theta^2$ for some small constant c. Taking a union bound of all possible pair subgraphs with volume at most $k \ll m$, we know that the probability that G' is a (k, τ)-NDBGraph is non-zero.

□

Now we are ready to prove the remaining part of Theorem 4.1.

Proof of Theorem 4.1. As mentioned before, we only need to prove the case when the input graph G is ϵ-far from any (k, τ)-NDBGraph. By Lemma 4.5 and 4.6, we know that the volume of the set of $\theta^2/16$-useful vertices with respect to some subset of volume at most k is $\Omega(\epsilon m)$. Since the size of sample vertex set is $\Theta(1/\epsilon)$, with probability at least $5/6$, we will sample out at least one such useful vertex v. Then by Lemma 4.4, with probability at least 0.99, the subroutine $\texttt{DetectDB}(G, v, K, \theta)$ will find a set of volume at most K and bipartiteness at most θ. Therefore, with high probability, our tester will reject the graph G. This completes the proof. □

5. Conclusion

We give a local exploration algorithm for finding subgraphs with small bipartiteness and a property testing algorithm for testing whether every subgraph of small volume has high bipartiteness. Both algorithms use a sequence of vectors with small support to approximate the sequence of vectors from standard power method. Though the approximation vectors are different in the two settings, our results show strong connections between the two local algorithmic paradigms such that the analysis of local exploration algorithm can be used to design one-sided graph property tester. This framework can also be generalized to other problems such as the small set expansion [26]. It would be interesting to see whether the converse is true, that is, is there a general framework that one can use the analysis of property testing algorithm to design better local exploration algorithms?

Acknowledgements

The author is partially supported by the Grand Project "Network Algorithms and Digital Information" of the Institute of Software, Chinese Academy of Sciences.

References

[1] J. Abello, M. Resende, S. Sudarsky, Massive quasi-clique detection, LATIN 2002: Theoretical Informatics (2002) 598–612.

[2] N. Alon, E. Fischer, I. Newman, A. Shapira, A combinatorial characterization of the testable graph properties: it's all about regularity, SIAM Journal on Computing 39 (2009) 143–167.

[3] N. Alon, T. Kaufman, M. Krivelevich, D. Ron, Testing triangle-freeness in general graphs, in: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm, ACM, pp. 279–288.

[4] R. Andersen, A local algorithm for finding dense subgraphs, ACM Trans. Algorithms 6 (2010) 60:1–60:12.

[5] R. Andersen, F. Chung, K. Lang, Local graph partitioning using pagerank vectors, in: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, 2006, pp. 475–486.

[6] R. Andersen, Y. Peres, Finding sparse cuts locally using evolving sets, in: Proceedings of the 41st annual ACM symposium on Theory of computing, STOC '09, ACM, New York, NY, USA, 2009, pp. 235–244.

[7] R. Bhatia, Matrix Analysis, Springer-Verlag, New York, 1997.

[8] M. Brautbar, M. Kearns, Local algorithms for finding interesting individuals in large networks., in: A.C.C. Yao (Ed.), ICS, Tsinghua University Press, 2010, pp. 188–199.

[9] F.R.K. Chung, Spectral graph theory, Regional Conference Series in Mathematics, American Mathematical Society 92 (1997) 1–212.

[10] A. Czumaj, C. Sohler, Testing expansion in bounded-degree graphs, in: Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on, IEEE, pp. 570–578.

[11] Y. Dourisboure, F. Geraci, M. Pellegrini, Extraction and classification of dense implicit communities in the web graph, ACM Transactions on the Web (TWEB) 3 (2009) 7.

[12] D. Gibson, R. Kumar, A. Tomkins, Discovering large dense subgraphs in massive graphs, in: Proceedings of the 31st international conference on Very large data bases, VLDB Endowment, pp. 721–732.

[13] A.V. Goldberg, Finding a Maximum Density Subgraph, Technical Report, Berkeley, CA, USA, 1984.

[14] O. Goldreich, D. Ron, Property testing in bounded degree graphs, Algorithmica 32 (2002) 302–343.

[15] S. Kale, C. Seshadhri, Combinatorial approximation algorithms for maxcut using random walks, In 2nd Symposium on Innovations in Computer Science (ICS 2011).

[16] S. Kale, C. Seshadhri, An expansion tester for bounded degree graphs, SIAM J. Comput. 40 (2011) 709–720.

[17] R. Kannan, S. Vempala, A. Vetta, On clusterings: Good, bad and spectral, J. ACM 51 (2004) 497–515.

[18] R. Kannan, V. Vinay, Analyzing the structure of large graphs (1999).

[19] T. Kaufman, M. Krivelevich, D. Ron, Tight bounds for testing bipartiteness in general graphs, SIAM Journal on computing 33 (2004) 1441–1483.

[20] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, in: Proceedings of the 32nd ACM Symposium on the Theory of Computing, 2000.

[21] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the web for emerging cyber-communities, in: Proceedings of the eighth international conference on World Wide Web, WWW '99, Elsevier North-Holland, Inc., New York, NY, USA, 1999, pp. 1481–1493.

[22] T. Kwok, L. Lau, Finding small sparse cuts by random walk, in: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, volume 7408 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2012, pp. 615–626.

[23] J. Leskovec, K.J. Lang, A. Dasgupta, M.W. Mahoney, Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters, CoRR abs/0810.1355 (2008). Informal publication.

[24] A. Li, Y. Pan, P. Peng, Testing conductance in general graphs, in: Electronic Colloquium on Computational Complexity (ECCC), volume 18, p. 101.

[25] A. Li, P. Peng, Detecting and characterizing small dense bipartite-like subgraphs by the bipartiteness ratio measure, ArXiv preprint arXiv:1209.5045 (2012).

[26] A. Li, P. Peng, Testing small set expansion in general graphs, Arxiv preprint arXiv:1209.5052 (2012).

[27] C.S. Liao, K. Lu, M. Baym, R. Singh, B. Berger, IsoRankN: spectral methods for global alignment of multiple protein networks, Bioinformatics 25 (2009) i253–i258.

[28] A. Nachmias, A. Shapira, Testing the expansion of a graph, Information and Computation 208 (2010) 309–314.

[29] S. Oveis Gharan, L. Trevisan, Approximating the expansion profile and almost optimal local graph clustering, in: 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS '12.

[30] P. Peng, A local algorithm for finding dense bipartite-like subgraphs, in: J. Gudmundsson, J. Mestre, T. Viglas (Eds.), Computing and Combinatorics, volume 7434 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2012, pp. 145–156.

[31] D. Ron, Algorithmic and analysis techniques in property testing, Found. Trends Theor. Comput. Sci. 5 (2010) 73–205.

[32] J. Soto, Improved analysis of a max cut algorithm based on spectral partitioning, CoRR abs/0910.0504 (2009).

[33] D. Spielman, S. Teng, A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning, Arxiv preprint arXiv:0809.3232 (2008).

[34] D.A. Spielman, Algorithms, graph theory, and linear equations IV (2010) 2698–2722.

[35] D.A. Spielman, S.H. Teng, Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems, in: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, STOC '04, ACM, New York, NY, USA, 2004, pp. 81–90.

[36] S.H. Teng, The laplacian paradigm: Emerging algorithms for massive graphs, in: Theory and Applications of Models of Computation, volume 6108 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2010, pp. 2–14.

[37] L. Trevisan, Max cut and the smallest eigenvalue, in: Proceedings of the 41st annual ACM symposium on Theory of computing, STOC '09, ACM, New York, NY, USA, 2009, pp. 263–272.