

A Local Algorithm for Finding Well-Connected Clusters*

Zeyuan Allen Zhu
zeyuan@csail.mit.edu
MIT CSAIL

Silvio Lattanzi
silviol@google.com
Google Research

Vahab Mirrokni
mirrokni@google.com
Google Research

February 15, 2013

Abstract

Motivated by applications of large-scale graph clustering, we study random-walk-based *local* algorithms whose running times depend only on the size of the output cluster, rather than the entire graph. In particular, we develop a method with better theoretical guarantee compared to all previous work, both in terms of the clustering accuracy and the conductance of the output set. We also prove that our analysis is tight, and perform empirical evaluation to support our theory on both synthetic and real data.

More specifically, our method outperforms prior work when the cluster is *well-connected*. In fact, the better it is well-connected inside, the more significant improvement we can obtain. Our results shed light on why in practice some random-walk-based algorithms perform better than its previous theory, and help guide future research about local clustering.

1 Introduction

As a central problem in machine learning, clustering methods have been applied to data mining, computer vision, social network analysis. Although a huge number of results are known in this area, there is still need to explore methods that are robust and efficient on large data sets, and have good theoretical guarantees. In particular, several algorithms restrict the number of clusters, or impose constraints that make these algorithms impractical for large data sets.

To solve those issues, recently, local random-walk clustering algorithms [ST04, ACL06] have been introduced. The main idea behind those algorithms is to find a good cluster around a specific node. These techniques, thanks to their scalability, has had high impact in practical applications [LLDM09, GLMY11, GS12, AGM12, LLM10, WLS⁺12]. Nevertheless, the theoretical understanding of these techniques is still very limited. In this paper, we make an important contribution in this direction. First, we relate for the first time the performance of these local algorithms with the *internal connectivity* of a cluster instead of analyzing only its external connectivity. This change of perspective is relevant for practical applications where we are not only interested to find clusters that are loosely connected with the rest of the world, but also clusters that are well-connected internally. In particular, we show theoretically and empirically that this internal connectivity is a fundamental parameter for those algorithms and, by leveraging it, it is possible to improve their performances.

Formally, we study the clustering problem where the data set is given by a similarity matrix as a graph: given an undirected¹ graph $G = (V, E)$, we want to find a set S that minimizes the

*Part of this work was done when the authors are at Google Research New York City. An extended abstract of this paper has appeared in the proceedings of the 30th International Conference on Machine Learning (ICML 2013).

¹All our results can be easily generalized to weighted graphs.

relative number of edges going out of S with respect to the size of S (or the size of \bar{S} if S is larger than \bar{S}). To capture this concept rigorously, we consider the *cut conductance* of a set S as:²

$$\phi_c(S) \stackrel{\text{def}}{=} \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}} ,$$

where $\text{vol}(S) \stackrel{\text{def}}{=} \sum_{v \in S} \deg(v)$. Finding S with the smallest $\phi_c(S)$ is called the conductance minimization. This measure is a well-studied measure in different disciplines [SM00, ST04, ACL06, GLMY11, GS12], and has been identified as one of the most important cut-based measures in the literature [Sch07]. Many approximation algorithms have been developed for the problem, but most of them are global ones: their running time depends at least linearly on the size of the graph. A recent trend, initiated by Spielman and Teng [ST04], and then followed by [ST08, ACL06, AP09, GT12], attempts to solve this conductance minimization problem *locally*, with running time only dependent on the volume of the output set.

In particular, if there exists a set $A \subset V$ with $\phi_c(A) \leq \Psi$, these local algorithms guarantee the existence of some set $A^g \subseteq A$ with at least half the volume, such that for any “good” starting vertex $v \in A^g$, they output a set S with conductance $\phi_c(S) = \tilde{O}(\sqrt{\Psi})$.

Finding Well-Connectedness Clusters. All local clustering algorithms developed so far, both theoretical ones and empirical ones, only assume that $\phi_c(A)$ is small, i.e., A is poorly connected to \bar{A} . Notice that such set A , no matter how small $\phi_c(A)$ is, may be poorly connected or even disconnected inside. This cannot happen in reality if A is a “good” cluster, and in practice we are often interested in finding mostly good clusters. This motivates us to study an extra measure on A , that is the connectedness of A , denoted as $\text{Conn}(A)$ and we will define it formally in Section 2. We assume that, in addition to prior work, the cluster A satisfies the *gap assumption*

$$\text{Gap} = \text{Gap}(A) \stackrel{\text{def}}{=} \frac{\text{Conn}(A)}{\Psi} \geq \Omega(1) ,$$

which says that A is better connected inside than it is connected to \bar{A} . This assumption is particularly relevant when the edges of the graph represent pairwise similarity scores extracted from a machine learning algorithm: we would expect similar nodes to be well connected within themselves while dissimilar nodes to be loosely connected. As a result, it is not surprising that the notion of connectedness is not new. For instance [KVV04] studied a bicriteria optimization for this objective. However, local algorithms based on the above gap assumption is not well studied.³

Our Results. Under the gap assumption $\text{Gap} \geq \Omega(1)$, can we guarantee any better cut conductance than the previously shown $\tilde{O}(\sqrt{\Psi})$ ones? We prove that the answer is affirmative, along with some other desirable properties. In particular, we prove:

Theorem 1. *If there exists a non-empty set $A \subset V$ such that $\phi_c(A) \leq \Psi$ and $\text{Gap} \geq \Omega(1)$, then there exists some $A^g \subseteq A$ with $\text{vol}(A^g) \geq \frac{1}{2}\text{vol}(A)$ such that, when choosing a starting vertex $v \in A^g$, the PageRank-Nibble algorithm outputs a set S with*

1. $\text{vol}(S \setminus A) \leq O(\frac{1}{\text{Gap}})\text{vol}(A)$,
2. $\text{vol}(A \setminus S) \leq O(\frac{1}{\text{Gap}})\text{vol}(A)$,

²Others also study a related notion called expansion, i.e., $\frac{|E(S, \bar{S})|}{\min\{|S|, |\bar{S}|\}}$ or $\frac{|E(S, \bar{S})|}{|S| \cdot |\bar{S}|}$, but there exist well-known reductions between the approximation algorithms on them.

³One relevant paper using this assumption is [MMV12], who provided a *global* SDP-based algorithm to approximate the cut conductance.

3. $\phi_c(S) \leq O(\sqrt{\Psi/\text{Gap}})$, and

with running time $O(\frac{\text{vol}(A)}{\Psi \cdot \text{Gap}}) \leq O(\frac{\text{vol}(A)}{\Psi})$.

We interpret the above theorem as follows. The first two properties imply that under $\text{Gap} \geq \Omega(1)$, the volume for $\text{vol}(S \setminus A)$ and $\text{vol}(A \setminus S)$ are both small in comparison to $\text{vol}(A)$, and the larger the gap is, the more accurate S approximates A .⁴ For the third property on the cut conductance $\phi_c(S)$, we notice that our guarantee $O(\sqrt{\Psi/\text{Gap}}) \leq O(\sqrt{\Psi})$ outperforms all previous work on local clustering under this gap assumption. In addition, Gap might be very large in reality. For instance when A is a very-well-connected cluster it might satisfy $\text{Conn}(A) = \text{polylog}(n)$, and as a consequence Gap may be as large as $\tilde{\Omega}(1/\Psi)$. In this case our Theorem 1 guarantees a $\text{polylog}(n)$ approximation to the cut conductance.

Our proof of Theorem 1 uses almost the same PageRank algorithm as [ACL06], but with a very different analysis specifically designed for our gap assumption.⁵ This algorithm is simple and clean, and can be described in four steps: 1) compute the (approximate) PageRank vector starting from a vertex $v \in A^g$ with carefully chosen parameters, 2) sort all the vertices according to their (normalized) probabilities in this vector, 3) study all *sweep cuts* that are those separating high-value vertices from low-value ones, and 4) output the sweep cut with the best cut conductance. See Algorithm 1 on page 12 for details.

We also prove that our analysis is tight.

Theorem 2. *There exists a graph $G = (V, E)$ and a non-empty $A \subset V$ with Ψ and $\text{Gap} = \Omega(1)$, such that for all starting vertices $v \in A$, none of the sweep-cut based algorithm on the PageRank vector can output a set S with cut conductance better than $O(\sqrt{\Psi/\text{Gap}})$.*

We prove this tightness result by illustrating a hard instance, and proving upper and lower bounds on the probabilities of reaching specific vertices (up to a very high precision). Theorem 2 does not rule out existence of another local algorithm that can perform better than $O(\sqrt{\Psi/\text{Gap}})$. However, we conjecture that all existing (random-walk-based) local clustering algorithms share the same hard instance and do not outperform $O(\sqrt{\Psi/\text{Gap}})$, similar to the classical case where they all provide only $\tilde{O}(\sqrt{\Psi})$ guarantee due to Cheeger’s inequality. It is an interesting open question to design a flow-based local algorithm to overcome this barrier under our gap assumption.

1.1 Prior Work

Most relevant to our work are the ones on local algorithms for clustering. On the theoretical side, after the first such result [ST04, ST08], [ACL06] simply compute a PAGERANK random walk vector and then show that one of its sweep cuts satisfies cut conductance $O(\sqrt{\Psi \log n})$. The computation of this PAGERANK vector is deterministic and is essentially the algorithm we adopt in this paper. [AP09, GT12] use the theory of evolving set from [MP03]. They study a stochastic volume-biased evolving set process that is similar to a random walk. This leads to a better (but probabilistic) running time and but essentially with the same cut conductance guarantee.

The problem of conductance minimization is UGC-hard to approximate within any constant factor [CKK⁺06]. On the positive side, spectral partitioning algorithms output a solution with

⁴Very recently, [WLS⁺12] studied a variant of the PageRank random walk and their first experiment —although analyzed in a different perspective— essentially confirmed our first two properties in Theorem 1. However, they have not attempted to explain this in theory.

⁵Interestingly, their theorems do not imply any new result in our setting at least in any obvious way, and thus proofs different from the previous work are necessary in this paper. To the best of our knowledge, equation (3.1) is the only part that is a consequence of their result, and we will mention it without proof.

conductance $O(\sqrt{\Psi})$ where this idea traces back to [Alo86] and [SJ89]; Leighton and Rao [LR99] provide a first $O(\log n)$ approximation; and Arora, Rao and Vazirani [ARV09] provide a $O(\sqrt{\log n})$ approximation. Those results, along with recent improvements on the running time by for instance [AHK10, AK07, She09], are all *global* algorithms: their time complexities depend at least linearly on the size of G . There are also seminal work in machine learning to make such global algorithm practical, including the seminal work of [LC10] for spectral partitioning.

Less relevant to our work are supervised learning on finding clusters, and there exist algorithms that have a sub-linear running time in terms of the size of the training set [ZCZ⁺09, SSS08].

On the empirical side, random-walk-based graph clustering algorithms have been widely used in practice [GS12, GLMY11, ACE⁺13, AGM12] as they can be implemented in a distributed manner for very big graphs using map-reduce or similar distributed graph mining algorithms [LLDM09, GLMY11, GS12, AGM12]. Such local algorithms have been applied for (overlapping) clustering of big graphs for distributed computation [AGM12], or community detection on huge Youtube video graphs [GLMY11]. There also exist variants of the random walk, such as the multi-agent random walk, that are known to be local and perform well in practice [AvL10].

More recently, [WLS⁺12] studied a slight variant of the PageRank random walk and performed supportive experiments on it. Their experiments confirmed the first two properties in our Theorem 1, but their theoretical results are not strong enough to confirm it. This is because there is no well-connectedness assumption in their paper so they are forced to study random walks that start from a random vertex selected in A , rather than a fixed one like ours. In addition, they have not argued about the cut conductance (like our third property in Theorem 1) of the set they output.

Clustering is an important technique for community detections, and indeed local clustering algorithms have been widely applied there, see for instance [AL06]. Sometimes researchers care about finding all communities, i.e., clusters, in the entire graph and this can be done by repeatedly applying local clustering algorithms. However, if the ultimate goal is to find all clusters, global algorithms perform better in at least in terms of minimizing conductance [LLDM09, GLMY11, GS12, AGM12, LLM10].

1.2 Roadmap

We provide preliminaries in Section 2, and they are followed by the high level ideas of the proofs for Theorem 1 in Section 3 and Section 4. We then briefly describe how to prove our tightness result in Theorem 5, and end the main body of this paper with empirical studies in Section 6.

2 Preliminaries

2.1 Problem Formulation

Consider an undirected graph $G(V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. For any vertex $u \in V$ the degree of u is denoted by $\deg(u)$, and for any subset of the vertices $S \subseteq V$, *volume* of S is denoted by $\text{vol}(S) \stackrel{\text{def}}{=} \sum_{u \in S} \deg(u)$. Given two subsets $A, B \subset V$, let $E(A, B)$ be the set of edges between A and B .

For a vertex set $S \subseteq V$, we denote by $G[S]$ the induced subgraph of G on S with outgoing edges removed, by $\deg_S(u)$ the degree of vertex $u \in S$ in $G[S]$, and by $\text{vol}_S(T)$ the volume of $T \subseteq S$ in $G[S]$.

We respectively define the *cut conductance* and the *set conductance* of a non-empty set $S \subseteq V$

as follows:

$$\phi_c(S) \stackrel{\text{def}}{=} \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}} ,$$

$$\phi_s(S) \stackrel{\text{def}}{=} \min_{\emptyset \subset T \subset S} \frac{|E(T, S \setminus T)|}{\min\{\text{vol}_S(T), \text{vol}_S(S \setminus T)\}} .$$

Here $\phi_c(S)$ is classically known as the conductance of S , and $\phi_s(S)$ is classically known as the conductance of S on the induced subgraph $G[S]$.

We formalize our goal in this paper as a *promise problem*. Specifically, we assume the existence of a non-empty cluster of the vertices $A \subset V$ satisfying $\text{vol}(A) \leq \frac{1}{2}\text{vol}(V)$ as well as $\phi_s(A) \geq \Phi$ and $\phi_c(A) \leq \Psi$. This set A is *not* known to the algorithm. The goal is to find some set S that “reasonably” approximates A , and at the same time be *local*: running in time proportional to $\text{vol}(A)$ rather than n or m .

Our assumption. We assume that the following gap assumption:

$$\text{Gap} \stackrel{\text{def}}{=} \frac{\text{Conn}(A)}{\Psi} \stackrel{\text{def}}{=} \frac{\Phi^2 / \log \text{vol}(A)}{\Psi} \geq \Omega(1) \quad (\text{Gap Assumption})$$

holds throughout this paper. This assumption can be understood as the cluster A is more well-connected inside than it is connected to \bar{A} .

(This assumption can be weakened by replacing the definition of $\text{Conn}(A)$ with $\text{Conn}(A) \stackrel{\text{def}}{=} \frac{1}{\tau_{\text{mix}}(A)}$, where $\tau_{\text{mix}}(A)$ is the mixing time for the relative pointwise distance in $G[A]$; or less weakly $\text{Conn}(A) \stackrel{\text{def}}{=} \frac{\lambda(A)}{\log \text{vol}(A)}$ where $\lambda(A)$ is the spectral gap, i.e., 1 minus the second largest eigenvalue of the random walk matrix on $G[A]$. We discuss them in Appendix A.)

Input parameters. Similar to prior work on local clustering, we assume the algorithm takes as input:

- Some “good” starting vertex $v \in A$, and an oracle to output the set of neighbors for any given vertex.

This requirement is essential because without such an oracle the algorithm may have to read all inputs and cannot be sublinear in time; and without a starting vertex the sublinear-time algorithm may be unable to even find an element in A .

We also need v to be “good”, as for instance the vertices on the boundary of A may not be helpful enough in finding good clusters. We call the set of good vertices $A^g \subseteq A$, and a local algorithm needs to ensure that A^g is large, i.e., $\text{vol}(A^g) \geq \frac{1}{2}\text{vol}(A)$.⁶

- The value of Φ .

In practice Φ can be viewed as a parameter and can be tuned for specific data. This is in contrast to the value of Ψ that is the target cut conductance and does not need to be known by the algorithm.⁷

- A value vol_0 satisfying $\text{vol}(A) \in [\text{vol}_0, 2\text{vol}_0]$.⁸

⁶This assumption is unavoidable in all local clustering work. One can replace this $\frac{1}{2}$ by any other constant at the expense of worsening the guarantees by a constant factor.

⁷In prior work when Ψ is the only quantity studied, Ψ plays both roles as a tuning parameter and as a target.

⁸This requirement is optional since otherwise the algorithm can try out different powers of 2 and pick the smallest one with a valid output. It blows up the running time only by a constant factor for local algorithms, since the running time of the last trial dominates.

2.2 PageRank Random Walk

We use the convention of writing vectors as row vectors in this paper. Let A be the adjacency matrix of G , and let D be the diagonal matrix with $D_{ii} = \deg(i)$, then the *lazy random walk matrix* $W \stackrel{\text{def}}{=} \frac{1}{2}(I + D^{-1}A)$. Accordingly, the PageRank vector $pr_{s,\alpha}$, is defined to be the unique solution of the following linear equation (cf. [ACL06]):

$$pr_{s,\alpha} = \alpha s + (1 - \alpha)pr_{s,\alpha}W \quad ,$$

where $\alpha \in (0, 1]$ is the *teleport probability* and s is a *starting vector*. Here s is usually a probability vector: its entries are in $[0, 1]$ and sum up to 1. For technical reasons we may use an arbitrary (and possibly negative) vector s inside the proof. When it is clear from the context, we drop α in the subscript for cleanness.

Given a vertex $u \in V$, let $\chi_u \in \{0, 1\}^V$ be the indicator vector that is 1 only at vertex u . Given non-empty subset $S \subseteq V$ we denote by π_S the degree-normalized uniform distribution on S , that is, $\pi_S(u) = \frac{\deg(u)}{\text{vol}(S)}$ when $u \in S$ and 0 otherwise. Very often we study a PageRank vector when $s = \chi_v$ is an indicator vector, and if so we abbreviate pr_{χ_v} by pr_v .

One equivalent way to study pr_s is to imagine the following random procedure: first pick a non-negative integer $t \in \mathbb{Z}_{\geq 0}$ with probability $\alpha(1 - \alpha)^t$, then perform a lazy random walk starting at vector s with exactly t steps, and at last define pr_s to be the vector describing the probability of reaching each vertex in this random procedure. In its mathematical formula we have (cf. [Hav02, ACL06]):

Proposition 2.1. $pr_s = \alpha s + \alpha \sum_{t=1}^{\infty} (1 - \alpha)^t (sW^t)$.

This implies that pr_s is linear: $a \cdot pr_s + b \cdot pr_t = pr_{as+bt}$.

2.3 Approximate PageRank Vector

In the seminal work of [ACL06], they defined approximate PageRank vectors and designed an algorithm to compute them efficiently.

Definition 2.2. An ε -approximate PageRank vector p for pr_s is a nonnegative PageRank vector $p = pr_{s-r}$ where the vector r is nonnegative and satisfies $r(u) \leq \varepsilon \deg(u)$ for all $u \in V$.

Proposition 2.3. For any starting vector s with $\|s\|_1 \leq 1$ and $\varepsilon \in (0, 1]$, one can compute an ε -approximate PageRank vector $p = pr_{s-r}$ for some r in time $O\left(\frac{1}{\varepsilon\alpha}\right)$, with $\text{vol}(\text{supp}(p)) \leq \frac{2}{(1-\alpha)\varepsilon}$.

For completeness we provide the algorithm and its proof in Appendix B. It can be verified that:

$$\forall u \in V, \quad pr_s(u) \geq p(u) \geq pr_s(u) - \varepsilon \deg(u) \quad . \quad (2.1)$$

2.4 Sweep Cuts

Given any approximate PageRank vector p , the *sweep cut (or threshold cut)* technique is the one to sort all vertices according to their degree-normalized probabilities $\frac{p(u)}{\deg(u)}$, and then study only those cuts that separate high-value vertices from low-value vertices. More specifically, let v_1, v_2, \dots, v_n be the decreasing order over all vertices with respect to $\frac{p(u)}{\deg(u)}$. Then, define *sweep sets* $S_j^p \stackrel{\text{def}}{=} \{v_1, \dots, v_j\}$ for each $j \in [n]$, and sweep cuts are the corresponding cuts $(S_j^p, \overline{S_j^p})$. Usually given a vector p , one looks for the best cut:

$$\min_{j \in [n-1]} \phi_c(S_j^p) \quad .$$

In almost all the cases, one only needs to enumerate j over $p(v_j) > 0$, so the above sweep cut procedure runs in time $O(\text{vol}(\text{supp}(p)) + |\text{supp}(p)| \cdot \log |\text{supp}(p)|)$. This running time is dominated by the time to compute p (see Proposition 2.3), so it is negligible.

2.5 Lovász-Simonovits Curve

Our proof requires the technique of *Lovász-Simonovits Curve* that has been more or less used in all local clustering algorithms so far. This technique was originally introduced by Lovász and Simonovits [LS90, LS93] to study the mixing rate of Markov chains. In our language, from a probability vector p on vertices, one can introduce a function $p[x]$ on real number $x \in [0, 2m]$. This function $p[x]$ is piecewise linear, and is characterized by all of its end points as follows (letting $p(S) \stackrel{\text{def}}{=} \sum_{a \in S} p(a)$):

$$p[0] \stackrel{\text{def}}{=} 0, \quad p[\text{vol}(S_j^p)] \stackrel{\text{def}}{=} p(S_j^p) \text{ for each } j \in [n] .$$

In other words, for any $x \in [\text{vol}(S_j^p), \text{vol}(S_{j+1}^p)]$,

$$p[x] \stackrel{\text{def}}{=} p(S_j^p) + \frac{x - \text{vol}(S_j^p)}{\text{deg}(v_{j+1})} p(v_{j+1}) .$$

Note that $p[x]$ is increasing and concave.

3 Guarantee Better Accuracy

In this section, we study PageRank random walks that start at a vertex $v \in A$ with teleport probability α . We claim the range of interesting α is $[\Omega(\Psi), O(\frac{\Phi^2}{\log n})]$. This is because, at a high level, when $\alpha \ll \Psi$ the random walk will leak too much to \bar{A} ; while when $\alpha \gg \frac{\Phi^2}{\log n}$ the random walk will not mix well inside A . In prior work, α is chosen to be $\Theta(\Psi)$, and we will instead choose $\alpha = \Theta(\frac{\Phi^2}{\log n}) = \Theta(\Psi \cdot \text{Gap})$. Intuitively, this choice of α ensures that under the condition the random walk mixes inside, it makes the walk leak as little as possible to \bar{A} . We prove the above intuition rigorously in this section. Specifically, we first show some properties on the exact PageRank vector in Section 3.1, and then move to the approximate vector in Section 3.2. This essentially proves the first two properties of Theorem 1.

3.1 Properties on the Exact Vector

We first introduce a new notation $\tilde{p}r_s$, that is the PageRank vector (with teleport probability α) starting at vector s but walking on the subgraph $G[A]$.

Next, we choose the set of “good” starting vertices A^g to satisfy two properties: (1) the total probability of leakage is upper bounded by $\frac{2\Psi}{\alpha}$, and (2) pr_v is close to $\tilde{p}r_v$ for vertices in A . Note that the latter implies that pr_v mixes well inside A as long as $\tilde{p}r_v$ does so.

Lemma 3.1. *There exists a set $A^g \subseteq A$ with volume $\text{vol}(A^g) \geq \frac{1}{2}\text{vol}(A)$ such that, for any vertex $v \in A^g$, in a PageRank vector with teleport probability α starting at v , we have:*

$$\sum_{u \notin A} pr_v(u) \leq \frac{2\Psi}{\alpha} . \tag{3.1}$$

In addition, there exists a non-negative leakage vector $l \in [0, 1]^V$ with norm $\|l\|_1 \leq \frac{2\Psi}{\alpha}$ satisfying

$$\forall u \in A, \quad pr_v(u) \geq \tilde{p}r_v(u) - \tilde{p}r_l(u) . \tag{3.2}$$

(Details of the proof are in Appendix C.1.)

Proof sketch. The proof for the first property (3.1) is classical and can be found in [ACL06]. The idea is to study an auxiliary PageRank random walk with teleport probability α starting at the degree-normalized uniform distribution π_A , and by simple computation, this random walk leaks to \bar{A} with probability no more than Ψ/α . Then, using Markov bound, there exists $A^g \subseteq A$ with $\text{vol}(A^g) \geq \frac{1}{2}\text{vol}(A)$ such that for each starting vertex $v \in A^g$, this leakage is no more than $\frac{2\Psi}{\alpha}$. This implies (3.1) immediately.

The interesting part is (3.2). Note that pr_v can be viewed as the probability vector from the following random procedure: start from vertex v , then at each step with probability α let the walk stop, and with probability $(1 - \alpha)$ follow the matrix W to go to one of its neighbors (or itself) and continue. Now, we divide this procedure into two rounds. In the first round, we run the same PageRank random walk but whenever the walk wants to use an outgoing edge from A to leak, we let it stop and temporarily “hold” this probability mass. We define l to be the non-negative vector where $l(u)$ denotes the amount of probability that we have “held” at vertex u . In the second round, we continue our random walk only from vector l . It is worth noting that l is non-zero only at boundary vertices in A .

Similarly, we divide the PageRank random walk for \tilde{pr}_v into two rounds. In the first round we hold exactly the same amount of probability $l(u)$ at boundary vertices u , and in the second round we start from l but continue this random walk only within $G[A]$. To bound the difference between pr_v and \tilde{pr}_v , we note that they share the same procedure in the first round; while for the second round, the random procedure for pr_v starts at l and walks towards $V \setminus A$ (so in the worst case it may never come back to A again), while that for \tilde{pr}_v starts at l and walks only inside $G[A]$ so induces a probability vector \tilde{pr}_l on A . This gives (3.2).

At last, to see $\|l\|_1 \leq \frac{2\Psi}{\alpha}$, one just needs to verify that $l(u)$ is essentially the probability that the original PageRank random walk leaks from vertex u . Then, $\|l\|_1 \leq \frac{2\Psi}{\alpha}$ follows from the fact that the total amount of leakage is upper bounded by $\frac{2\Psi}{\alpha}$. \square

As mentioned earlier, we want to use (3.2) to lower bound $pr_v(u)$ for vertices $u \in A$. We achieve this by first lower bounding \tilde{pr}_v which is the PageRank random walk on $G[A]$. Given a teleport probability α that is small compared to $\frac{\Phi^2}{\log \text{vol}(A)}$, this random walk should mix well. We formally state it as the following lemma, and provide its proof in the Appendix C.2.

Lemma 3.2. *When $\alpha \leq O(\Psi \cdot \text{Gap})$ we have that*

$$\forall u \in A, \quad \tilde{pr}_v(u) \geq \frac{4 \deg_A(u)}{5 \text{vol}(A)} .$$

Here $\deg_A(u)$ is the degree of u on $G[A]$, but $\text{vol}(A)$ is with respect to the original graph.

3.2 Properties of the Approximate Vector

From this section on we always use $\alpha \leq O(\Psi \cdot \text{Gap})$. We then fix a starting vertex $v \in A^g$ and study an ε -approximate Pagerank vector for pr_v . We choose

$$\varepsilon = \frac{1}{10 \cdot \text{vol}_0} \in \left[\frac{1}{20\text{vol}(A)}, \frac{1}{10\text{vol}(A)} \right] . \quad (3.3)$$

For notational simplicity, we denote by p this ε -approximation and recall from Section 2.3 that $p = pr_{\chi_v - r}$ where r is a non-negative vector with $0 \leq r(u) \leq \varepsilon \deg(u)$ for every $u \in V$. Recall from (2.1) that $pr_v(u) \geq p(u) \geq pr_v(u) - \varepsilon \cdot \deg(u)$ for all $u \in V$.

We now rewrite Lemma 3.1 in the language of approximate PageRank vectors using Lemma 3.2:

Corollary 3.3. For any $v \in A^g$ and $\alpha \leq O(\Psi \cdot \text{Gap})$, in an ε -approximate PageRank vector to pr_v denoted by $p = pr_{\chi_v - r}$, we have:

$$\sum_{u \notin A} p(u) \leq \frac{2\Psi}{\alpha} \text{ and } \sum_{u \notin A} r(u) \leq \frac{2\Psi}{\alpha} .$$

In addition, there exists a non-negative leakage vector $l \in [0, 1]^V$ with norm $\|l\|_1 \leq \frac{2\Psi}{\alpha}$ satisfying

$$\forall u \in A, \quad p(u) \geq \frac{4 \deg_A(u)}{5 \text{vol}(A)} - \frac{\deg(u)}{10 \text{vol}(A)} - \tilde{p}r_l(u) .$$

Proof. The only inequality that requires a proof is $\sum_{u \notin A} r(u) \leq \frac{2\Psi}{\alpha}$. In fact, if one takes a closer look at the algorithm to compute an approximate Pagerank vector (cf. Appendix B), the total probability mass that will be sent to r on vertices outside A , is upper bounded by the probability of leakage. However, the latter is upper bounded by $\frac{2\Psi}{\alpha}$ when we choose A^g . \square

We are now ready to state the main lemma of this section. We show that for all reasonable sweep sets S on this probability vector p , it satisfies that $\text{vol}(S \setminus A)$ and $\text{vol}(A \setminus S)$ are both at most $O(\frac{\Psi}{\alpha} \text{vol}(A))$.

Lemma 3.4. In the same definition of α and p from Corollary 3.3, let sweep set $S_c \stackrel{\text{def}}{=} \{u \in V : p(u) \geq c \frac{\deg(u)}{\text{vol}(A)}\}$ for any constant $c < \frac{3}{5}$, then we have the following guarantees on the size of $S_c \setminus A$ and $A \setminus S_c$:

1. $\text{vol}(S_c \setminus A) \leq \frac{2\Psi}{\alpha c} \text{vol}(A)$, and
2. $\text{vol}(A \setminus S_c) \leq \left(\frac{2\Psi}{\alpha(\frac{3}{5}-c)} + 8\Psi \right) \text{vol}(A)$.

Proof. First we notice that $p(S_c \setminus A) \leq p(V \setminus A) \leq \frac{2\Psi}{\alpha}$ owing to Corollary 3.3, and for each vertex $u \in S_c \setminus A$ it must satisfy $p(u) \geq c \frac{\deg(u)}{\text{vol}(A)}$. Those combined imply $\text{vol}(S_c \setminus A) \leq \frac{2\Psi}{\alpha c} \text{vol}(A)$ proving the first property.

We show the second property in two steps. First, let A_b be the set of vertices in A such that $\frac{4 \deg_A(u)}{5 \text{vol}(A)} - \frac{\deg(u)}{10 \text{vol}(A)} < \frac{3 \deg(u)}{5 \text{vol}(A)}$. Any such vertex $u \in A_b$ must have $\deg_A(u) < \frac{7}{8} \deg(u)$. This implies that u has to be on the boundary of A and $\text{vol}(A_b) \leq 8\Psi \text{vol}(A)$.

Next, for a vertex $u \in A \setminus A_b$ we have (using Corollary 3.3 again) $p(u) \geq \frac{3 \deg(u)}{5 \text{vol}(A)} - \tilde{p}r_l(u)$. If we further have $u \notin S_c$ so $p(u) < c \frac{\deg(u)}{\text{vol}(A)}$, it implies that $\tilde{p}r_l(u) \geq (\frac{3}{5} - c) \frac{\deg(u)}{\text{vol}(A)}$. As a consequence, the total volume for such vertices (i.e., $\text{vol}(A \setminus (A_b \cup S_c))$) cannot exceed $\frac{\|\tilde{p}r_l\|_1}{3/5-c} \text{vol}(A)$. At last, we notice that $\tilde{p}r_l$ is a non-negative probability vector coming from a random walk procedure, so $\|\tilde{p}r_l\|_1 = \|l\|_1 \leq \frac{2\Psi}{\alpha}$. This in sum provides that

$$\begin{aligned} \text{vol}(A \setminus S_c) &\leq \text{vol}(A \setminus (A_b \cup S_c)) + \text{vol}(A_b) \\ &\leq \left(\frac{2\Psi}{\alpha(\frac{3}{5}-c)} + 8\Psi \right) \text{vol}(A) . \end{aligned} \quad \square$$

Note that if one chooses $\alpha = \Theta(\Psi \cdot \text{Gap})$ in the above lemma, both those two volumes are at most $O(\text{vol}(A)/\text{Gap})$ satisfying the first two properties of Theorem 1.

4 Guarantee Better Cut Conductance

In the classical work of [ACL06], they have shown that when $\alpha = \Theta(\Psi)$, among all sweep cuts on vector p there exists one with cut conductance $O(\sqrt{\Psi \log n})$. In this section, we improve this result under our gap assumption $\text{Gap} \geq \Omega(1)$.

Lemma 4.1. *Letting $\alpha = \Theta(\Psi \cdot \text{Gap})$, among all sweep sets $S_c = \{u \in V : p(u) \geq c \frac{\deg(u)}{\text{vol}(A)}\}$ for $c \in [\frac{1}{8}, \frac{1}{4}]$, there exists one, denoted by S_{c^*} , with cut conductance $\phi_c(S_{c^*}) = O(\sqrt{\Psi/\text{Gap}})$.*

Proof sketch. To convey the idea of the proof, we only consider the case when $p = pr_v$ is the exact PageRank vector, and the proof for the approximate case is a bit more involved and deferred to Appendix D.1.

Suppose that all sweep sets S_c for $c \in [\frac{1}{8}, \frac{1}{4}]$ satisfy $|E(S_c, V \setminus S_c)| \geq E_0$ for some value E_0 , then it suffices to prove $E_0 \leq O(\frac{\Psi}{\sqrt{\alpha}})\text{vol}(A)$. This is because, if so, then there exists some S_{c^*} with $|E(S_{c^*}, V \setminus S_{c^*})| \leq E_0$ and this combined with the result in Lemma 3.4 (i.e., $\text{vol}(S_{c^*}) = (1 \pm O(1/\text{Gap}))\text{vol}(A)$) gives

$$\phi_c(S_{c^*}) \leq O\left(\frac{E_0}{\text{vol}(S_{c^*})}\right) = O(\Psi/\sqrt{\alpha}) = O(\sqrt{\Psi/\text{Gap}}) .$$

We introduce some classical notations before we proceed in the proof. For any vector q we denote by $q(S) \stackrel{\text{def}}{=} \sum_{u \in S} q(u)$. Also, given a directed edge⁹, $e = (a, b) \in E$ we let $p(e) = p(a, b) \stackrel{\text{def}}{=} \frac{p(a)}{\deg(a)}$, and for a set of directed edges E' we let $p(E') \stackrel{\text{def}}{=} \sum_{e \in E'} p(e)$. We also let $E(A, B) \stackrel{\text{def}}{=} \{(a, b) \in E \mid a \in A \wedge b \in B\}$ be the set of directed edges from A to B .

Now for any set $S_{1/4} \subseteq S \subseteq S_{1/8}$, we compute that

$$\begin{aligned} p(S) &= pr_v(S) = \alpha \chi_v(S) + (1 - \alpha)(pW)(S) \\ &\leq \alpha + (1 - \alpha)(pW)(S) \\ \implies (1 - \alpha)p(S) &\leq \alpha(1 - p(S)) + (1 - \alpha)(pW)(S) \\ \implies (1 - \alpha)p(S) &\leq 2\Psi + (1 - \alpha)(pW)(S) \\ \implies p(S) &< O(\Psi) + (pW)(S) . \end{aligned} \tag{4.1}$$

Here we have used the fact that when $p = pr_v$ is exact, it satisfies $1 - p(S) = p(V - S) \leq 2\Psi/\alpha$ according to Corollary 3.3. In the next step, we use the definition of the lazy random walk matrix W to compute that

$$\begin{aligned} &(pW)(S) \\ &= \left(\sum_{(a,b) \in E(S,S)} p(a,b) + \sum_{(a,b) \in E(S,\bar{S})} \frac{p(a,b) + p(b,a)}{2} \right) \\ &= \left(\frac{1}{2}p\left(E(S,S)\right) + \frac{1}{2}p\left(E(S,S) \cup E(S,\bar{S}) \cup E(\bar{S},S)\right) \right) \\ &\leq \left(\frac{1}{2}p\left[|E(S,S)|\right] + \frac{1}{2}p\left[|E(S,S) \cup E(S,\bar{S}) \cup E(\bar{S},S)|\right] \right) \\ &= \left(\frac{1}{2}p\left[\text{vol}(S) - |E(S,\bar{S})|\right] + \frac{1}{2}p\left[\text{vol}(S) + |E(S,\bar{S})|\right] \right) \\ &\leq \left(\frac{1}{2}p\left[\text{vol}(S) - E_0\right] + \frac{1}{2}p\left[\text{vol}(S) + E_0\right] \right) . \end{aligned} \tag{4.2}$$

⁹ G is an undirected graph, but we study undirected edges with specific directions for analysis purpose only.

Digit	0	1	2	3	4	5	6	7	8	9
$\Psi = \phi_c(A)$	0.00294	0.00304	0.08518	0.03316	0.22536	0.08580	0.01153	0.03258	0.09761	0.05139
$\phi_c(S)$	0.00272	0.00067	0.03617	0.02220	0.00443	0.01351	0.00276	0.00456	0.03849	0.00448
Precision	0.993	0.995	0.839	0.993	0.988	0.933	0.946	0.985	0.941	0.994
Recall	0.988	0.988	0.995	0.773	0.732	0.896	0.997	0.805	0.819	0.705

Table 1: Clustering results on the USPS zipcode data set. We report precision $|A \cap S|/|S|$ and recall $|A \cap S|/|A|$.

Here the first inequality is due to the definition of the Lovász-Simonovits curve $p[x]$, and the second inequality is because $p[x]$ is concave. Next, suppose that in addition to $S_{1/4} \subseteq S \subseteq S_{1/8}$, we also know that S is a sweep set, i.e., $\forall a \in S, b \notin S$ we have $\frac{p(a)}{\deg(a)} \geq \frac{p(b)}{\deg(b)}$. This implies $p(S) = p[\text{vol}(S)]$ and combining (4.1) and (4.2) we obtain that

$$(p[\text{vol}(S)] - p[\text{vol}(S) - E_0]) \leq O(\Psi) + (p[\text{vol}(S) + E_0] - p[\text{vol}(S)]) .$$

Since we can choose S to be an arbitrary sweep set between $S_{1/4}$ and $S_{1/8}$, we have that the inequality $p[x] - p[x - E_0] \leq O(\Psi) + p[x + E_0] - p[x]$ holds for all end points $x \in [\text{vol}(S_{1/4}), \text{vol}(S_{1/8})]$ on the piecewise linear curve $p[x]$. This implies that the same inequality holds for any real number $x \in [\text{vol}(S_{1/4}), \text{vol}(S_{1/8})]$ as well. We are now ready to draw our conclusion by repeatedly applying this inequality. Letting $x_1 := \text{vol}(S_{1/4})$ and $x_2 := \text{vol}(S_{1/8})$, we have

$$\begin{aligned} \frac{E_0}{4\text{vol}(A)} &\leq p[x_1] - p[x_1 - E_0] \\ &\leq O(\Psi) + (p[x_1 + E_0] - p[x_1]) \\ &\leq 2 \cdot O(\Psi) + (p[x_1 + 2E_0] - p[x_1 + E_0]) \leq \dots \\ &\leq \left\lfloor \frac{x_2 - x_1}{E_0} + 1 \right\rfloor O(\Psi) + (p[x_2 + E_0] - p[x_2]) \\ &\leq \frac{\text{vol}(S_{1/8} \setminus S_{1/4})}{E_0} O(\Psi) + \frac{E_0}{8\text{vol}(A)} \\ &\leq \frac{\text{vol}(S_{1/8} \setminus A) + \text{vol}(A \setminus S_{1/4})}{E_0} O(\Psi) + \frac{E_0}{8\text{vol}(A)} \\ &\leq \frac{O(\Psi/\alpha) \cdot \text{vol}(A)}{E_0} O(\Psi) + \frac{E_0}{8\text{vol}(A)} , \end{aligned}$$

where the first inequality uses the definition of $S_{1/4}$, the fifth inequality uses the definition of $S_{1/8}$, and last inequality uses Lemma 3.4 again. After re-arranging the above inequality we conclude that $E_0 \leq O\left(\frac{\Psi}{\alpha}\right)\text{vol}(A)$ and finish the proof. \square

The lemma above essentially shows the third property of Theorem 1 and finishes the proof of Theorem 1. For completeness of the paper, we still provide the formal proof for Theorem 1 in Appendix D.2, and summarize our final algorithm in Algorithm 1.

5 Tightness of Our Analysis

It is a natural question to ask under our newly introduced assumption $\text{Gap} \geq \Omega(1)$: is $O(\sqrt{\Psi/\text{Gap}})$ the best cut conductance we can obtain from a local algorithm? We show that this is true if one sticks to a sweep-cut algorithm using PageRank vectors.

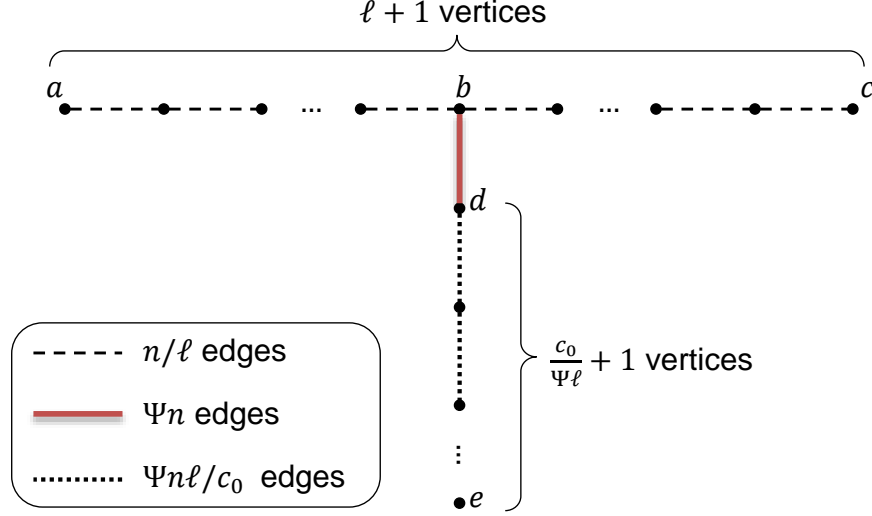


Figure 1: Our hard instance for proving tightness. One can pick for instance $\ell \approx n^{0.4}$ and $\Psi \approx \frac{1}{n^{0.9}}$, so that $n/\ell \approx n^{0.6}$, $\Psi n \approx n^{0.1}$ and $\Psi n \ell \approx n^{0.5}$.

Algorithm 1 PageRank-Nibble

Input: v, Φ and $\text{vol}_0 \in [\frac{\text{vol}(A)}{2}, \text{vol}(A)]$.

Output: set S .

- 1: $\alpha \leftarrow \Theta(\frac{\Phi^2}{\log \text{vol}(A)}) = \Theta(\Psi \cdot \text{Gap})$.
 - 2: $p \leftarrow$ a $\frac{1}{10 \cdot \text{vol}_0}$ -approximate PageRank vector with starting vertex v and teleport probability α .
 - 3: Sort all vertices in $\text{supp}(p)$ according to $\frac{p(u)}{\deg(u)}$.
 - 4: Consider all sweep sets $S'_c \stackrel{\text{def}}{=} \{u \in \text{supp}(p) : p(u) \geq \frac{c \deg(u)}{\text{vol}_0}\}$ for $c \in [\frac{1}{8}, \frac{1}{2}]$, and let S be the one among them with the best $\phi_c(S)$.
-

More specifically, we show that our analysis in Section 4 is tight by constructing the following hard instance. Consider a (multi-)graph with two chains (see Figure 1) of vertices, and there are multi-edges connecting them.¹⁰ In particular:

- the top chain (ended with vertex a and c and with midpoint b) consists of $\ell + 1$ vertices where ℓ is even with $\frac{n}{\ell}$ edges between each consecutive pair;
- the bottom chain (ended with vertex d and e) consists of $\frac{c_0}{\Psi \ell} + 1$ vertices with $\frac{\Psi n \ell}{c_0}$ edges between each consecutive pair, where the constant c_0 is to be determined later; and
- vertex b and d are connected with Ψn edges.

We let the top chain to be our promised cluster A . The total volume of A is $2n + \Psi n$, while the total volume of the entire graph is $4n + 2\Psi n$. The mixing time for A is $\tau_{\text{mix}}(A) = \Theta(\ell^2)$, and the cut conductance $\phi_c(A) = \frac{\Psi n}{\text{vol}(A)} \approx \frac{\Psi}{2}$. Suppose that the gap assumption¹¹ $\text{Gap} \stackrel{\text{def}}{=} \frac{1}{\tau_{\text{mix}}(A) \cdot \phi_c(A)} \approx \frac{1}{\Psi \ell^2} \gg 1$ is satisfied, i.e., $\Psi \ell^2 = o(1)$. (For instance one can let $\ell \approx n^{0.4}$ and $\Psi \approx \frac{1}{n^{0.9}}$ to achieve this requirement.)

¹⁰One can transform this example into a graph without parallel edges by splitting vertices into expanders, but that goes out of the purpose of this section.

¹¹We are using Theorem 1 in the language of gap assumption on τ_{mix} . See Section 2.1 and Appendix A for details.

We then consider a PageRank random walk that starts at vertex $v = a$ and with teleport probability $\alpha = \frac{\gamma}{\ell^2}$ for some arbitrarily small constant $\gamma > 0$.¹² Let pr_a be this PageRank vector, and we prove in Appendix E the following lemma:

Lemma 5.1. *For any $\gamma \in (0, 4]$ and letting $\alpha = \gamma/\ell^2$, there exists some constant c_0 such that when studying the PageRank vector pr_a starting from vertex a in Figure 1, the following holds $\frac{pr_a(d)}{\deg(d)} > \frac{pr_a(c)}{\deg(c)}$.*

This lemma implies that, for any sweep-cut algorithm based on this vector pr_a , even if it computes pr_a exactly and looks for all possible sweep cuts, then none of them gives a better cut conductance than $O(\sqrt{\Psi/\text{Gap}})$. More specifically, for any sweep set S :

- if $c \notin S$, then $|E(S, V \setminus S)|$ is at least $\frac{n}{\ell}$ because it has to contain a (multi-)edge in the top chain. Therefore, the cut conductance $\phi_c(S) \geq \Omega(\frac{n}{\ell \text{vol}(S)}) \geq \Omega(\frac{1}{\ell}) \geq \Omega(\sqrt{\Psi/\text{Gap}})$; or
- if $c \in S$, then d must be also in S because it has a higher normalized probability than c using Lemma 5.1. In this case, $|E(S, V \setminus S)|$ is at least $\frac{\Psi n \ell}{c_0}$ because it has to contain a (multi-)edge in the bottom chain. Therefore, the cut conductance $\phi_c(S) \geq \Omega(\frac{\Psi n \ell}{\text{vol}(S)}) \geq \Omega(\Psi \ell) = \Omega(\sqrt{\Psi/\text{Gap}})$.

This ends the proof of Theorem 2. □

6 Empirical Evaluation

The PageRank local clustering method has been studied empirically in various previous work. For instance, Gleich and Seshadhri [GS12] performed experiments on 15 datasets and confirmed that PageRank outperformed many others in terms of cut conductance, including the famous METIS algorithm. Moreover, [LLDM09] studied PageRank against METIS+MQI which is the METIS algorithm plus a flow-based post-processing. Their experiments confirmed that although METIS+MQI outperforms PageRank in terms of cut conductance, however, the PageRank algorithm’s outputs are more “community-like”, and they enjoy other desirable properties.

Since our **PageRank-Nibble** is essentially the same PageRank method as before with only theoretical changes in the parameters, it certainly embraces the same empirical behavior as those literatures above. Therefore, in this section we perform experiments *only for the sake of* demonstrating our theoretical discoveries in Theorem 1, without comparisons to other methods. We run our algorithm against both synthetic and real datasets.

Recall that Theorem 1 has three properties. The first two properties are *accuracy guarantees* that ensure the output set S well approximates A in terms of volume; and the third property is a *cut-conductance guarantee* that ensures the output set S has small $\phi_c(S)$. We now provide experimental results to support them.

Experiment 1. In the first experiment, we study a synthetic graph of 870 vertices. We carefully choose the parameters as follows in order to confuse the **PageRank-Nibble** algorithm so that it cannot identify A up to a very high accuracy. We let the vertices be divided into three disjoint subsets: subset A (which is the desired set) of 300 vertices, subset B of 20 vertices and subset C of 550 vertices. We assume that A is constructed from the Watts-Strogatz model¹³ with mean degree $K = 60$ and a parameter $\beta \in [0, 1]$ to control the connectivity of $G[A]$: varying β makes it possible

¹²Although we promised in Theorem 2 to study all starting vertices $v \in A$, in this version of the paper we only concentrate on $v = a$ because other choices of v are only easier and can be analyzed similarly. In addition, this choice of $\alpha = \frac{\gamma}{\ell^2}$ is consistent with the one used Theorem 1.

¹³See http://en.wikipedia.org/wiki/Watts_and_Strogatz_model.

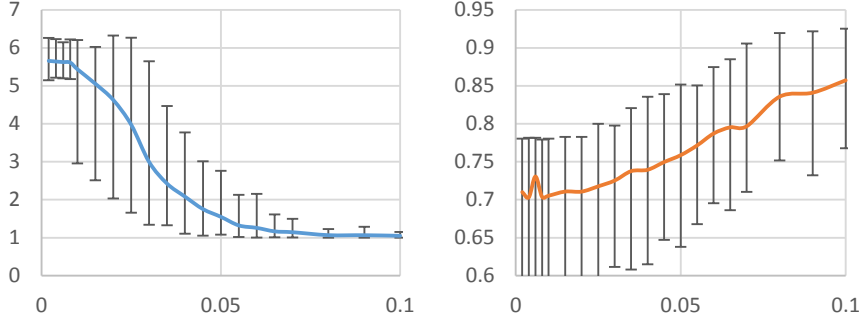


Figure 2: Experimental result on the synthetic data. The horizontal axis represents the value of β for constructing our graph, the blue curve (left) represents the ratio $\frac{\phi_c(S)}{\Psi}$, and the red curve (right) represents the clustering accuracy. The vertical bars are 94% confidence intervals for 100 runs.

to interpolate between a regular lattice ($\beta = 0$) that is not-well-connected and a random graph ($\beta = 1$) that is well-connected. We then construct the rest of the graph by throwing in random edges, or more specifically, we add an edge

- with probability 0.3 between each pair of vertices in B and B ;
- with probability 0.02 between each pair of vertices in C and C ;
- with probability 0.001 between each pair of vertices in A and B ;
- with probability 0.002 between each pair of vertices in A and C ; and
- with probability 0.002 between each pair of vertices in B and C .

It is not hard to verify that in this randomly generated graph, the (expected) cut conductance $\Psi = \phi_c(A)$ is independent of β . As a result, the larger β is, we should expect the larger the well-connectedness A enjoys, and therefore the larger the gap **Gap** is in Theorem 1. This should lead to a better performance both in terms of accuracy and conductance when β goes larger.

To confirm this, we perform an experiment on this randomly generated graph with various choices of β . For each choice of β , we run our **PageRank-Nibble** algorithm with teleport probability α chosen to be the best one in the range of $[0.001, 0.3]$, starting vertex v chosen to be a random one in A , and ε to be sufficiently small. We then run our algorithm 100 times each time against a different random graph instance. We then plot in Figure 2 two curves (along with their 94% confidence intervals) as a function of β : the average cut conductance over Ψ ratio, i.e., $\frac{\phi(S)}{\Psi}$, and the average clustering accuracy, i.e., $1 - \frac{|A \Delta S|}{|V|}$. Our experiment confirms our result in Theorem 1: **PageRank-Nibble** performs better both in accuracy and cut conductance as **Gap** goes larger.

Experiment 2. In the second experiment, we use the USPS zipcode data set¹⁴ that was also used in the work from [WLS⁺12]. Following their experiment, we construct a weighted k -NN graph with $k = 20$ out of this data set. The similarity between vertex i and j is computed as $w_{ij} = \exp(-d_{ij}^2/\sigma)$ if i is within j 's k nearest neighbors or vice versa, and $w_{ij} = 0$ otherwise, where $\sigma = 0.2 \times r$ and r denotes the average square distance between each point to its 20th nearest neighbor.

This is a dataset with 9298 images of handwritten digits between 0 to 9, and we treat it as 10 separate binary-classification problems. For each of them, we pick an arbitrary starting vertex in it, let $\alpha = 0.003$ and $\varepsilon = 0.00005$, and then run our **PageRank-Nibble** algorithm. We report our results in Table 1. For each of the 10 binary-classifications, we have a ground-truth set A that contains all data points associated with the given digit. We then compare the cut conductance of our output set $\phi_c(S)$ against the desired cut conductance $\Psi = \phi_c(A)$, and our algorithm consistently

¹⁴<http://www-stat.stanford.edu/~tibs/ElemStatLearn/data.html>.

outperforms the desired one on all 10 clusters. (Notice that it is possible to see an output set S to have smaller conductance than A , because A is not necessarily the sparsest cut in the graph.) In addition, one can also confirm from Table 1 that our algorithm enjoys high precision and recall.

Acknowledgements

We thank Lorenzo Orecchia, Jon Kelner, Aditya Bhaskara for helpful conversations. This work is partly supported by Google and a Simons award (grant no. 284059).

Appendix

A Two Weaker Gap Assumptions

As mentioned in Section 2, we can relax our gap assumption to

$$\begin{aligned} \text{Gap} &= \frac{\text{Conn}(A)}{\Psi} \stackrel{\text{def}}{=} \frac{\lambda(A)}{\Psi \log \text{vol}(A)} \geq \Omega(1) \quad , \quad \text{or} && \text{(Gap Assumption')} \\ \text{Gap} &= \frac{\text{Conn}(A)}{\Psi} \stackrel{\text{def}}{=} \frac{1}{\Psi \cdot \tau_{\text{mix}}(A)} \geq \Omega(1) && \text{(Gap Assumption'')} \end{aligned}$$

- Here $\lambda(A)$ is the *spectral gap*, that is the difference between the first and second largest eigenvalues of the lazy random walk matrix on $G[A]$. (Notice that the largest eigenvalue of any random walk matrix is always 1.) Equivalently, $\lambda(A)$ can be defined as the second smallest eigenvalue of the Laplacian matrix of $G[A]$.
- Here τ_{mix} is the *mixing time for the relative pointwise distance* in $G[A]$ (cf. Definition 6.14 in [MR95]), that is, the minimum time required for a lazy random walk to mix *relatively* on all vertices regardless of the starting distribution. Formally, let W_A be the lazy random walk matrix on $G[A]$, and π be the stationary distribution on $G[A]$ that is $\pi(u) = \deg_A(u)/\text{vol}_A(A)$, then

$$\tau_{\text{mix}} = \min \left\{ t \in \mathbb{Z}_{\geq 0} : \max_{u,v} \left| \frac{(\chi_v W_A^t)(u) - \pi(u)}{\pi(u)} \right| \leq \frac{1}{2} \right\} .$$

Notice that using Cheeger's inequality, we always have $\frac{\phi_s(A)^2}{\log \text{vol}(A)} \leq O\left(\frac{\lambda(A)}{\log \text{vol}(A)}\right) \leq O\left(\frac{1}{\tau_{\text{mix}}}\right)$. This is why (Gap Assumption'') is weaker than (Gap Assumption') which is then weaker than (Gap Assumption).

We emphasize that the exact statement of our Theorem 1 is still true under those two weaker assumptions, leading to two strictly stronger results. To see this, one only needs to restudy Lemma 3.2 under the two new assumptions, and we provide such analysis in Appendix C.2.

B Algorithm for Computing Approximate PageRank Vector

In this section we briefly summarize the algorithm **Approximate-PR** (see Algorithm 2) proposed by Andersen, Chung and Lang [ACL06] to compute an approximate PageRank vector. At high level, **Approximate-PR** is an iterative algorithm, and maintains an invariant that p is always equal to pr_{s-r} at each iteration.

Initially it lets $p = \vec{0}$ and $r = s$ so that $p = \vec{0} = pr_{\vec{0}}$ satisfies this invariant. Notice that r does not necessarily satisfy $r(u) \leq \varepsilon \deg(u)$ for all vertices u , and thus this p is often not an ε -approximate PageRank vector according to Definition 2.2 at this initial step.

In each following iteration, **Approximate-PR** considers a vertex u that violates the ε -approximation of p , i.e., $r(u) \geq \varepsilon \deg(u)$, and *pushes* this $r(u)$ amount of probability mass elsewhere:

- $\alpha \cdot r(u)$ amount of them is pushed to $p(u)$;
- $\frac{1-\alpha}{2 \deg(u)} r(u)$ amount of them is pushed to $r(v)$ for each neighbor v of u ; and
- $\frac{1-\alpha}{2} r(u)$ amount of them remains at $r(u)$.

One can verify that after any push step the newly computed p and r will still satisfy $p = pr_{s-r}$. This indicates that the invariant is satisfied at all iterations. When `Approximate-PR` terminates, it satisfies both $p = pr_{s-r}$ and $r(u) \leq \varepsilon \deg(u)$ for all vertices u , so p must be an ε -approximate PageRank vector.

We are left to show that `Approximate-PR` terminates quickly, and the support volume of p is small:

Proposition 2.3. *For any starting vector s with $\|s\|_1 \leq 1$ and $\varepsilon \in (0, 1]$, `Approximate-PR` computes an ε -approximate PageRank vector $p = pr_{s-r}$ for some r in time $O\left(\frac{1}{\varepsilon\alpha}\right)$, with $\text{vol}(\text{supp}(p)) \leq \frac{2}{(1-\alpha)\varepsilon}$.*

Proof sketch. To show that this algorithm converges fast, one just needs to notice that at each iteration $\alpha r(u) \geq \alpha\varepsilon \deg(u)$ amount of probability mass is pushed from vector r to vector p , so the total amount of them cannot exceed 1 (because $\|s\|_1 \leq 1$). This gives $\sum_{i=1}^T \deg(u_i) \leq \frac{1}{\varepsilon\alpha}$ where u_i is the vertex chosen at the i -th iteration and T is the number of iterations. However, it is not hard to verify that the total running time of `Approximate-PR` is exactly $O\left(\sum_{i=1}^T \deg(u_i)\right)$, and thus `Approximate-PR` runs in time $O\left(\frac{1}{\varepsilon\alpha}\right)$.

To bound the support volume, we consider an arbitrary vertex $u \in V$ with $p(u) > 0$. This $p(u)$ amount of probability mass must come from $r(u)$ during the algorithm, and thus vertex u must be pushed at least once. Notice that when u is last pushed, it satisfies $r(u) \geq \frac{1-\alpha}{2}\varepsilon \deg(u)$ after the push, and this value $r(u)$ cannot decrease in the remaining iterations of the algorithm. This implies that for all $u \in V$ with $p(u) > 0$, it must be true that $r(u) \geq \frac{1-\alpha}{2}\varepsilon \deg(u)$. However, we must have $\|r\|_1 \leq 1$ because $\|s\|_1 \leq 1$, so the total volume for such vertices cannot exceed $\frac{2}{(1-\alpha)\varepsilon}$. \square

Algorithm 2 `Approximate-PR` (from [ACL06])

Input: starting vector s , teleport probability α , and approximate ratio ε .

Output: the ε -approximate PageRank vector $p = pr_{s-r}$.

- 1: $p \leftarrow \vec{0}$ and $r \leftarrow s$.
 - 2: **while** $r(u) \geq \varepsilon \deg(u)$ for some vertex $u \in V$ **do**
 - 3: Pick an arbitrary u satisfying $r(u) \geq \varepsilon \deg(u)$.
 - 4: $p(u) \leftarrow p(u) + \alpha r(u)$.
 - 5: For each vertex v such that $(u, v) \in E$:
 $\quad r(v) \leftarrow r(v) + \frac{1-\alpha}{2\deg(u)} r(u)$.
 - 6: $r(u) \leftarrow \frac{1-\alpha}{2} r(u)$.
 - 7: **end while**
 - 8: **return** p .
-

C Missing Proofs in Section 3

C.1 Proof of Lemma 3.1

Lemma 3.1. *There exists a set $A^g \subseteq A$ with volume $\text{vol}(A^g) \geq \frac{1}{2}\text{vol}(A)$ such that, for any vertex $v \in A^g$, in a PageRank vector with teleport probability α starting at v , we have:*

$$\sum_{u \notin A} pr_v(u) \leq \frac{2\Psi}{\alpha} . \quad (3.1)$$

In addition, there exists a non-negative leakage vector $l \in [0, 1]^V$ with norm $\|l\|_1 \leq \frac{2\Psi}{\alpha}$ satisfying

$$\forall u \in A, \quad pr_v(u) \geq \tilde{p}r_v(u) - \tilde{p}r_l(u) . \quad (3.2)$$

Leakage event. We begin our proof by defining the *leaking event* in a random walk procedure. We start the definition of a lazy random walk and then move to a PageRank random walk. At high level, we say that a lazy random walk of length t starting at a vertex $u \in A$ does not *leak* from A if it never goes out of A , and let $\text{Leak}(u, t)$ denote the probability that such a random walk leaks.

More formally, for each vertex $u \in V$ in the graph with degree $\deg(u)$, recall that in its random walk graph it actually has degree $2\deg(u)$, with $\deg(u)$ edges going to each of its neighbors, and $\deg(u)$ self-loops. For a vertex $u \in A$, let us call its neighboring edge $(u, v) \in E$ a *bad edge* if $v \notin A$. In addition, if u has k bad edges, we also distinguish k self-loops at u in the lazy random walk graph, and call them *bad self-loops*. Now, we say that a random walk does not leak from A , *if it never uses any of those bad edges of self-loops*. The purpose of this definition is to make sure that if a random walk chooses only good edges at each step, it is equivalent to a lazy random walk on the induced subgraph $G[A]$ with outgoing edges removed.

For a PageRank random walk with teleport probability α starting at a vertex u , recall that it is also a random procedure and can be viewed as first picking a length $t \in \{0, 1, \dots\}$ with probability $\alpha(1 - \alpha)^t$, and then performing a lazy random walk of length t starting from u . By the linearity of random walk vectors, the probability of leakage for this PAGERANK random walk is exactly $\sum_{t=0}^{\infty} \alpha(1 - \alpha)^t \text{Leak}(u, t)$.

Upper bounding leakage. We now give an upper bound on the probability of leakage. We start with an auxiliary lazy random walk of length t starting from a “uniform” distribution $\pi_A(u)$. Recall that $\pi_A(u) = \frac{\deg(u)}{\text{vol}(A)}$ for $u \in A$ and 0 elsewhere. We now want to show that this random walk leaks with probability at most $1 - t\Psi$.¹⁵ This is because, one can verify that: (1) in the first step of this random walk, the probability of leakage is upper bounded by Ψ by the definition of cut conductance; and (2) in the i -th step in general, this random walk satisfies $(\pi_A W^{i-1})(u) \leq \pi_A(u)$ for any vertex $u \in A$, and therefore the probability of leakage in the i -th step is upper bounded by that in the first step. In sum, the total leakage is at most $t\Psi$, or equivalently, $\sum_{u \in A} \pi_A(u) \text{Leak}(u, t) \leq t\Psi$.

We now sum this up over the distribution of t in a PageRank random walk:

$$\begin{aligned} \sum_{u \in A} \pi_A(u) \left(\sum_{t=0}^{\infty} \alpha(1 - \alpha)^t \text{Leak}(u, t) \right) &= \sum_{t=0}^{\infty} \alpha(1 - \alpha)^t \left(\sum_{u \in A} \pi_A(u) \text{Leak}(u, t) \right) \\ &\leq \sum_{t=0}^{\infty} \alpha(1 - \alpha)^t t\Psi = \frac{\Psi(1 - \alpha)}{\alpha} . \end{aligned}$$

This implies, using Markov bound, there exists a set $A^g \subseteq A$ with volume $\text{vol}(A^g) \geq \frac{1}{2}\text{vol}(A)$ satisfying

$$\forall v \in A^g, \quad \sum_{t=0}^{\infty} \alpha(1 - \alpha)^t \text{Leak}(v, t) \leq \frac{2\Psi(1 - \alpha)}{\alpha} < \frac{2\Psi}{\alpha} , \quad (\text{C.1})$$

or in words: the probability of leakage is at most $\frac{2\Psi(1 - \alpha)}{\alpha}$ in a PAGERANK random walk that starts at vertex $v \in A^g$. This inequality immediately implies (3.1), so for the rest of the proof, we concentrate on (3.2).

Lower bounding pr . Now we pick some $v \in A^g$, and try to lower bound pr_v . To begin with, we define two $|A| \times |A|$ lazy random walk matrices on the induced subgraph $G[A]$ (recall that $\deg(u)$ is the degree of a vertex and for $u \in A$ we denote by $\deg_A(u)$ the number of neighbors of u inside A):

¹⁵Note that this step of the proof coincides with that of Proposition 2.5 from [ST08]. Our $t\Psi$ is off by a factor of 2 from theirs because we also regard bad self-loops as edges that leak.

1. Matrix \widehat{W} . This is a random walk matrix assuming that all outgoing edges from A being “phantom”, that is, at each vertex $u \in A$:

- it picks each neighbor in A with probability $\frac{1}{2\deg(u)}$, and
- it stays where it is with probability $\frac{\deg_A(u)}{2\deg(u)}$.

For instance, let u be a vertex in A with four neighbors w_1, w_2, w_3, w_4 such that $w_1, w_2, w_3 \in A$ but $w_4 \notin A$. Then, for a lazy random walk using matrix \widehat{W} , if it starts from u then in the next step it stays at u with probability $3/8$, and goes to w_1, w_2 and w_3 each with probability $1/8$. Note that, for the rest $1/4$ probability (which corresponds to w_4) it goes *nowhere* and this random walk “disappears”! This can be viewed as that the random walk leaks A .

2. Matrix \widetilde{W} . This is a random walk matrix assuming that all outgoing edges from A are removed, that is, at each vertex $u \in A$:

- it picks each neighbor in A with probability $\frac{1}{2\deg_A(u)}$, and
- it stays where it is with probability $\frac{1}{2}$.

The major difference between \widetilde{W} and \widehat{W} is that they are normalized by different degrees in the rows, and the rows of \widetilde{W} sum up to 1 but those of \widehat{W} do not necessarily. More specifically, if we denote by D the diagonal matrix with $\deg(u)$ on the diagonal for each vertex $u \in A$, and D_A the diagonal matrix with $\deg_A(u)$ on the diagonal, then $\widehat{W} = D^{-1}D_A\widetilde{W}$. It is worth noting that, if one sums up all entries of the nonnegative vector $\chi_v\widehat{W}^t$, the summation is exactly $1 - \text{Leak}(v, t)$ by our definition of Leak .

We now precisely study the difference between \widetilde{W} and \widehat{W} using the following claim.

Claim C.1. *There exists non-negative vectors l_t for all $t \in \{1, 2, \dots\}$ satisfying:*

$$\|l_t\|_1 = \text{Leak}(v, t) - \text{Leak}(v, t-1) ,$$

and

$$\chi_v\widehat{W}^t = \left(\chi_v\widehat{W}^{t-1} - l_t\right)\widetilde{W} .$$

Proof. To obtain the result of this claim, we write

$$\begin{aligned} \chi_v\widehat{W}^t &= \left(\chi_v\widehat{W}^{t-1}\right)D^{-1}D_A\widetilde{W} \\ &= \left(\chi_v\widehat{W}^{t-1}\right)\widetilde{W} - \left(\chi_v\widehat{W}^{t-1}\right)(I - D^{-1}D_A)\widetilde{W} \end{aligned}$$

Now, we simply let $l_t \stackrel{\text{def}}{=} \left(\chi_v\widehat{W}^{t-1}\right)(I - D^{-1}D_A)$. It is a non-negative vector because $\deg_A(u)$ is no larger than $\deg(u)$ for all $u \in A$. Furthermore, recall that in the lazy random walk characterized by \widehat{W} , the amount of probability to disappear at a vertex u in the t -th step, is exactly its probability after a $(t-1)$ -th step random walk, i.e., $(\chi_v\widehat{W}^{t-1})(u)$, multiplied by the probability to leak in this step, i.e., $1 - \frac{\deg_A(u)}{\deg(u)}$. Therefore, $l_t(u)$ exactly equals to the amount of probability to disappear in the t -th step; or equivalently, $\|l_t\|_1 = \text{Leak}(v, t) - \text{Leak}(v, t-1)$. \square

Now we use the above definition of l_t and deduce that:

Claim C.2. Letting $l \stackrel{\text{def}}{=} \sum_{j=1}^{\infty} (1-\alpha)^{j-1} l_j$, we have $\|l\|_1 \leq \frac{2\Psi}{\alpha}$ and the following inequality on vector holds coordinate-wisely on all vertices in A :

$$pr_v|_A \geq \sum_{t=0}^{\infty} \alpha(1-\alpha)^t (\chi_v - l) \widetilde{W}^t = \widetilde{pr}_v - \widetilde{pr}_l .$$

Proof. We begin the proof with a simple observation. The following inequality on vector holds coordinate-wisely on all vertices in A according to the definition of \widetilde{W} :

$$pr_v|_A = \sum_{t=0}^{\infty} \alpha(1-\alpha)^t (\chi_v W^t)|_A \geq \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \chi_v \widehat{W}^t .$$

Therefore, to lower bound $pr_v|_A$ it suffices to lower bound the right hand side. Now owing to Claim C.1 we further reduce the computation on matrix \widehat{W} to that on matrix \widetilde{W} :

$$\chi_v \widehat{W}^t = (\chi_v \widehat{W}^{t-1} - l_t) \widetilde{W} = \left((\chi_v \widehat{W}^{t-2} - l_{t-1}) \widetilde{W} - l_t \right) \widetilde{W} = \dots = \chi_v \widetilde{W}^t - \sum_{j=1}^t l_j \widetilde{W}^{t-j+1} .$$

We next combine the above two inequalities and compute

$$\begin{aligned} pr_v|_A &\geq \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \chi_v \widehat{W}^t = \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \left(\chi_v \widetilde{W}^t - \sum_{j=1}^t l_j \widetilde{W}^{t-j+1} \right) \\ &= \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \chi_v \widetilde{W}^t - \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \sum_{j=1}^t l_j \widetilde{W}^{t-j+1} \\ &= \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \chi_v \widetilde{W}^t - \sum_{j=1}^{\infty} (1-\alpha)^{j-1} l_j \sum_{t=1}^{\infty} \alpha(1-\alpha)^t \widetilde{W}^t \\ &\geq \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \chi_v \widetilde{W}^t - \sum_{j=1}^{\infty} (1-\alpha)^{j-1} l_j \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \widetilde{W}^t \\ &= \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \left(\chi_v - \sum_{j=1}^{\infty} (1-\alpha)^{j-1} l_j \right) \widetilde{W}^t = \sum_{t=0}^{\infty} \alpha(1-\alpha)^t (\chi_v - l) \widetilde{W}^t . \end{aligned}$$

At last, we upper bound the one norm of l using Claim C.1 again:

$$\begin{aligned} \|l\|_1 &= \sum_{j=1}^{\infty} (1-\alpha)^{j-1} \|l_j\|_1 = \sum_{j=1}^{\infty} (1-\alpha)^{j-1} (\text{Leak}(v, j) - \text{Leak}(v, j-1)) \\ &= \sum_{j=1}^{\infty} \alpha(1-\alpha)^{j-1} \text{Leak}(v, j) \leq \frac{2\Psi(1-\alpha)}{\alpha(1-\alpha)} = \frac{2\Psi}{\alpha} , \end{aligned}$$

where the last inequality uses (C.1). □

So far we have also shown (3.2) and this ends the proof of Lemma 3.1. ■

C.2 Proof of Lemma 3.2

Lemma 3.2 (restated). *When the teleport probability $\alpha \leq \frac{\Phi^2}{72(3+\log \text{vol}(A))}$ (or more weakly when $\alpha \leq \frac{\lambda(A)}{9(3+\log \text{vol}(A))}$, or $\alpha \leq O(\frac{1}{\tau_{\text{mix}}})$), we have that*

$$\forall u \in A, \quad \tilde{p}r_v(u) = \sum_{t=0}^{\infty} \alpha(1-\alpha)^t (\chi_v \widetilde{W}^t)(u) > \frac{4}{5} \frac{\deg_A(u)}{\text{vol}(A)}.$$

Proof. We first prove this lemma in the case when $\alpha \leq \frac{\Phi^2}{72(3+\log \text{vol}(A))}$ or $\alpha \leq \frac{\lambda(A)}{9(3+\log \text{vol}(A))}$. We will then extend it to the weakest assumption $\alpha \leq O(\frac{1}{\tau_{\text{mix}}})$. For a discussion on the comparisons between those three assumptions, see Appendix A.

Recall that we defined \widetilde{W} to be the lazy random walk matrix on A with outgoing edges removed, and denoted by $\lambda = \lambda(A)$ the spectral gap on the lazy random walk matrix of $G[A]$ (cf. Appendix A). Then, by the theory of infinity-norm mixing time of a Markov chain, the length- t random walk starting at any vertex $v \in A$ will land in a vertex $u \in A$ with probability:

$$\begin{aligned} (\chi_v \widetilde{W}^t)(u) &\geq \frac{\deg_A(u)}{\sum_{w \in A} \deg_A(w)} - (1-\lambda)^t \sqrt{\frac{\deg_A(v)}{\min_y \deg_A(y)}} \\ &\geq \frac{\deg_A(u)}{\sum_{w \in A} \deg_A(w)} - (1-\lambda)^t \deg_A(v).^{16} \end{aligned}$$

Now if we choose $T_0 = \frac{3+\log \text{vol}(A)}{\lambda}$ then for any $t \geq T_0$:

$$(\chi_v \widetilde{W}^t)(u) \geq \frac{9}{10} \frac{\deg_A(u)}{\sum_{w \in V} \deg_A(w)} \geq \frac{9}{10} \frac{\deg_A(u)}{\text{vol}(A)}. \quad (\text{C.2})$$

We then convert this into the language of PageRank vectors:

$$\begin{aligned} \sum_{t=0}^{\infty} \alpha(1-\alpha)^t (\chi_v \widetilde{W}^t)(u) &\geq (1-\alpha)^{T_0} \alpha \sum_{t=0}^{\infty} (1-\alpha)^t (\chi_v \widetilde{W}^{t+T_0})(u) \\ &\geq (1-\alpha)^{T_0} \alpha \sum_{t=0}^{\infty} (1-\alpha)^t \left(\frac{9}{10} \frac{\deg_A(u)}{\text{vol}(A)} \right) = (1-\alpha)^{T_0} \left(\frac{9}{10} \frac{\deg_A(u)}{\text{vol}(A)} \right). \end{aligned}$$

At last, we notice that $\alpha \leq \frac{1}{9T_0}$ holds: this is either because we have chosen $\alpha \leq \frac{\lambda(A)}{9(3+\log \text{vol}(A))}$, or because we have chosen $\alpha \leq \frac{\Phi^2}{72(3+\log \text{vol}(A))}$ and Cheeger's inequality $\lambda \geq \Phi^2/8$ holds. As a consequence, it satisfies that $(1-\alpha)^{T_0} \geq 1 - \alpha T_0 \geq \frac{8}{9}$ and thus $(1-\alpha)^{T_0} \left(\frac{9}{10} \frac{\deg_A(u)}{\text{vol}(A)} \right) \geq \frac{4}{5} \frac{\deg_A(u)}{\text{vol}(A)}$.

We can also show our lemma under the assumption that $\alpha \leq O(1/\tau_{\text{mix}})$. In such a case, one can choose $T_0 = \Theta(\tau_{\text{mix}})$ so that (C.2) and the rest of the proof still hold. It is worth emphasizing that since we always have $\frac{\phi_s(A)^2}{\log \text{vol}(A)} \leq O(\frac{\lambda(A)}{\log \text{vol}(A)}) \leq O(\frac{1}{\tau_{\text{mix}}})$, this last assumption is the weakest one among all three. \square

¹⁶Here we have used the fact that $\min_y \deg_A(y) \geq 1$. This is because otherwise $G[A]$ will be disconnected so that $\Phi = \phi_s(A) = 0$, $\lambda(A) = 0$ and $\tau_{\text{mix}}(A) = \infty$, but none of the three can happen under our gap assumption $\text{Gap} \geq \Omega(1)$.

D Missing Proofs in Section 4

D.1 Proof of Lemma 4.1

Lemma 4.1. *Letting $\alpha = \Theta(\Psi \cdot \text{Gap})$, among all sweep sets $S_c = \{u \in V : p(u) \geq c \frac{\deg(u)}{\text{vol}(A)}\}$ for $c \in [\frac{1}{8}, \frac{1}{4}]$, there exists one, denoted by S_{c^*} , with cut conductance $\phi_c(S_{c^*}) = O(\sqrt{\Psi/\text{Gap}})$.*

Proof. We only point out how to extend our proof in the exact case (see Section 4) to the case when p is an ε -approximate PageRank vector. For any set $S_{1/4} \subseteq S \subseteq S_{1/8}$, we compute that

$$\begin{aligned} p(S) &= pr_{\chi_v - r}(S) = \alpha(\chi_v - r)(S) + (1 - \alpha)(pW)(S) \\ &= \alpha(\chi_v - r)(V) + \alpha r(V \setminus S) + (1 - \alpha)(pW)(S) \\ &\leq \alpha(\chi_v - r)(V) + \alpha(r(V \setminus A) + r(A \setminus S)) + (1 - \alpha)(pW)(S) \\ &= \alpha p(V) + \alpha(r(V \setminus A) + r(A \setminus S)) + (1 - \alpha)(pW)(S) \end{aligned}$$

where in the last equality we have used $(\chi_v - r)(V) = p(V)$, owing to the fact that $p = (\chi_v - r) \sum_{t=0}^{\infty} \alpha(1 - \alpha)^t W^t$, but W is a random walk matrix that preserves the total probability mass.

We next notice that $r(V \setminus A) \leq \frac{2\Psi}{\alpha}$ according to Corollary 3.3, as well as

$$\begin{aligned} r(A \setminus S) &\leq \varepsilon \text{vol}(A \setminus S) && \text{(according to Definition 2.2)} \\ &\leq \varepsilon \left(\frac{2\Psi}{\alpha(\frac{3}{5} - \frac{1}{4})} + 8\Psi \right) \text{vol}(A) && \text{(according to Lemma 3.4 and } S \supseteq S_{1/4}) \\ &< \frac{7\Psi}{\alpha} \varepsilon \text{vol}(A) && \text{(using } \alpha \leq \frac{1}{9} \text{ from our choice in Appendix C.2)} \\ &\leq \frac{0.7\Psi}{\alpha} . && \text{(using our choice of } \varepsilon \leq \frac{1}{10} \text{vol}(A) \text{ in Section 3.2)} \end{aligned}$$

Therefore, we have

$$\begin{aligned} p(S) &\leq \alpha p(V) + \alpha \left(\frac{2\Psi}{\alpha} + \frac{0.7\Psi}{\alpha} \right) + (1 - \alpha)(pW)(S) \\ &= \alpha p(V) + 2.7\Psi + (1 - \alpha)(pW)(S) \\ \implies (1 - \alpha)p(S) &\leq \alpha \cdot p(V \setminus S) + 2.7\Psi + (1 - \alpha)(pW)(S) \\ \implies (1 - \alpha)p(S) &\leq 4.7\Psi + (1 - \alpha)(pW)(S) && \text{(using Corollary 3.3)} \\ \implies p(S) &\leq 5.3\Psi + (pW)(S) && \text{(using } \alpha \leq \frac{1}{9} \text{ again)} \end{aligned}$$

In sum, we have arrived at the same conclusion as (4.1) in the case when p is only approximate, and the rest of the proof follows in the same way as in the exact case. \square

D.2 Proof of Theorem 1

We are ready to put together all previous lemmas to show the main theorem of this paper.

Theorem 1. *If there exists a non-empty set $A \subset V$ such that $\phi_c(A) \leq \Psi$ and $\text{Gap} \geq \Omega(1)$, then there exists some $A^g \subseteq A$ with $\text{vol}(A^g) \geq \frac{1}{2} \text{vol}(A)$ such that, when choosing a starting vertex $v \in A^g$, the PageRank-Nibble algorithm outputs a set S with*

1. $\text{vol}(S \setminus A) \leq O(\frac{1}{\text{Gap}}) \text{vol}(A)$,
2. $\text{vol}(A \setminus S) \leq O(\frac{1}{\text{Gap}}) \text{vol}(A)$,

3. $\phi_c(S) \leq O(\sqrt{\Psi/\mathbf{Gap}})$, and

with running time $O(\frac{\text{vol}(A)}{\Psi \cdot \mathbf{Gap}}) \leq O(\frac{\text{vol}(A)}{\Psi})$.

Proof. As in Algorithm 1, we choose $\alpha = \Theta(\Psi \cdot \mathbf{Gap})$ to satisfy the requirements of all previous lemmas. We define A^g according to Lemma 3.1 and compute an ε -approximate PageRank vector starting from v where $\varepsilon = \frac{1}{10\text{vol}_0}$ satisfies (3.3).

Next we study all sweep sets $S'_c \stackrel{\text{def}}{=} \{u \in \text{supp}(p) : p(u) \geq \frac{c \deg(u)}{\text{vol}_0}\}$ for $c \in [\frac{1}{16}, \frac{1}{4}]$. Notice that since $\text{vol}_0 \in [\frac{\text{vol}(A)}{2}, \text{vol}(A)]$, all such sweep sets correspond to $S_d = \{u \in \text{supp}(p) : p(u) \geq \frac{d \deg(u)}{\text{vol}(A)}\}$ for some $d \in [\frac{1}{16}, \frac{1}{2}]$. Therefore, the output S is also some S_d sweep set with $d \in [\frac{1}{16}, \frac{1}{2}]$ and Lemma 3.4 guarantees the first two properties of the theorem.

On the other hand, Lemma 4.1 guarantees the existence of some sweep set S_{d^*} satisfying $\phi_c(S_{d^*}) = O(\sqrt{\Psi \cdot \mathbf{Gap}})$. Since $d^* \in [\frac{1}{8}, \frac{1}{4}]$, this S_{d^*} is also a sweep set S'_c with $c \in [\frac{1}{16}, \frac{1}{4}]$, and must be considered as sweep set candidate in our Algorithm 1. This immediately implies that the output S of Algorithm 1 must have a cut conductance $\phi_c(S)$ that is at least as good as $\phi_c(S_{d^*}) = O(\sqrt{\Psi \cdot \mathbf{Gap}})$, finishing the proof for the third property of the theorem.

At last, as a direct consequence of Proposition 2.3 and the fact that the computation of the approximate PageRank vector is the bottleneck for the running time, we conclude that Algorithm 1 runs in time $O(\frac{\text{vol}(A)}{\alpha}) = O(\frac{\text{vol}(A)}{\Psi \cdot \mathbf{Gap}})$. \square

E Missing Proofs in Section 5

In this section we show that our cut conductance analysis for Theorem 1 is tight. We emphasize here that such a tightness proof is very non-trivial, because one has to provide a graph hard instance and start to upper and lower bound the probabilities of reaching specific vertices up to a very high precision. This is different from the mixing time theory on Markov chains, as for instance, on a chain of ℓ vertices it is known that a random walk of $O(\ell^2)$ steps mixes, but in addition we need to compute how faster it mixes on one vertex than another vertex.

In Appendix E.1 we begin with some warm-up lemmas for the PageRank vector on a single chain, and then in Appendix E.2 we formally prove Lemma 5.1 with the help from those lemmas.

E.1 Useful Lemmas for a PageRank Random Walk on a Chain

In this subsection we provide four useful lemmas about a PageRank random walk on a single chain. For instance, in the first of them we study a chain of length ℓ and compute an upper bound on the probability to reach the rightmost vertex from the leftmost one. The other three lemmas are similar in this format. Those lemmas require the study of the eigensystem of a lazy random walk matrix on this chain, followed by very careful but problem-specific analyses.

Lemma E.1. *Let ℓ be an even integer, and consider a chain of $\ell + 1$ vertices with the leftmost vertex indexed by 0 and the rightmost vertex indexed by ℓ . Let pr_{x_0} be the PageRank vector for a random walk starting at vertex 0 with teleport probability $\alpha = \frac{\gamma}{\ell^2}$ for some constant γ . Then,*

$$pr_{x_0}(\ell) \leq \frac{1}{2\ell} \left(1 - \frac{2\gamma}{\pi^2/4 + \gamma} + \frac{2\gamma}{\pi^2 + \gamma} + O\left(\frac{1}{\ell^2}\right) \right).$$

Proof. Let us define

$$W = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & & \\ & \frac{1}{4} & \frac{1}{2} & \ddots & \\ & & \ddots & \ddots & \frac{1}{4} \\ & & & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

to be the $(\ell + 1) \times (\ell + 1)$ lazy random walk matrix of our chain. For $k = 0, 1, \dots, \ell$, define:

$$\begin{aligned} \lambda_k &\stackrel{\text{def}}{=} \frac{1 + \cos\left(\frac{\pi k}{\ell}\right)}{2} = \cos^2\left(\frac{\pi k}{2\ell}\right) \\ v_k(u) &\stackrel{\text{def}}{=} \deg(u) \cdot \cos\left(\frac{\pi k u}{\ell}\right) \quad (u = 0, 1, \dots, \ell) , \end{aligned} \quad (\text{E.1})$$

where $\deg(u)$ is the degree for the u -th vertex, that is, $\deg(0) = \deg(\ell) = 1$ while $\deg(i) = 2$ for $i \in \{1, 2, \dots, \ell - 1\}$. Then it is routinary to verify that $v_k \cdot W = \lambda_k \cdot v_k$ and thus

v_k is the k -th (left-)eigenvector and λ_k is the k -th eigenvalue for matrix W .

We remark here that since W is not symmetric, those eigenvectors are not orthogonal to each other in the standard basis. However, under the notion of inner product $\langle x, y \rangle \stackrel{\text{def}}{=} \sum_{i=0}^{\ell} x(i)y(i) \deg(i)^{-1}$, they form an orthonormal basis.

It now expand our starting probability vector χ_0 under this orthonormal basis:

$$\chi_0 = (1, 0, 0, \dots, 0) = \frac{1}{2\ell} \left(v_0 + 2 \sum_{k=1}^{\ell-1} v_k + v_\ell \right) .$$

As a consequence when $t > 0$, using $\lambda_\ell = 0$:

$$\chi_0 W^t = \frac{1}{2\ell} \left(v_0 + 2 \sum_{k=1}^{\ell-1} (\lambda_k)^t v_k \right) .$$

Now it is easy to compute the exact probability of reaching the right-most vertex ℓ :

$$\begin{aligned} \chi_0 W^t(\ell) &= \frac{1}{2\ell} \left(v_0(\ell) + 2 \sum_{k=1}^{\ell-1} (\lambda_k)^t v_k(\ell) \right) = \frac{1}{2\ell} \left(1 + 2 \sum_{k=1}^{\ell-1} \cos^{2t} \left(\frac{\pi k}{2\ell} \right) \cos(\pi k) \right) \\ &= \frac{1}{2\ell} \left(1 + 2 \sum_{k=1}^{\ell-1} \cos^{2t} \left(\frac{\pi k}{2\ell} \right) (-1)^k \right) \leq \frac{1}{2\ell} \left(1 - 2 \cos^{2t} \left(\frac{\pi}{2\ell} \right) + 2 \cos^{2t} \left(\frac{\pi}{\ell} \right) \right) . \end{aligned}$$

At last, we translate this language into the PageRank vector pr_{χ_0} and obtain

$$\begin{aligned} pr_{\chi_0}(\ell) &= \sum_{t=0}^{\infty} \alpha(1 - \alpha)^t \chi_0 W^t(\ell) \leq \frac{1}{2\ell} \left(\alpha v_\ell(\ell) + \sum_{t=0}^{\infty} \alpha(1 - \alpha)^t \left(1 - 2 \cos^{2t} \left(\frac{\pi}{2\ell} \right) + 2 \cos^{2t} \left(\frac{\pi}{\ell} \right) \right) \right) \\ &= \frac{1}{2\ell} \left(\alpha + 1 - \frac{2\alpha}{1 - (1 - \alpha) \cos^2 \left(\frac{\pi}{2\ell} \right)} + \frac{2\alpha}{1 - (1 - \alpha) \cos^2 \left(\frac{\pi}{\ell} \right)} \right) \\ &\leq \frac{1}{2\ell} \left(1 - \frac{2\gamma}{\pi^2/4 + \gamma} + \frac{2\gamma}{\pi^2 + \gamma} + O\left(\frac{1}{\ell^2}\right) \right) . \end{aligned}$$

We remark here that the last inequality is obtained using Taylor approximation. \square

Lemma E.2. *Let ℓ be an even integer, and consider a chain of $\ell + 1$ vertices with the leftmost vertex indexed by 0 and the rightmost vertex indexed by ℓ . Let pr_{χ_0} be the PageRank vector for a random walk starting at vertex 0 with teleport probability $\alpha = \frac{\gamma}{\ell^2}$ for some constant γ . Then,*

$$pr_{\chi_0}\left(\frac{\ell}{2}\right) \geq \frac{1}{\ell} \left(1 - \frac{2\gamma}{\pi^2 + \gamma} - O\left(\frac{1}{\ell^2}\right)\right) .$$

Proof. Recall from the proof of Lemma E.1 that for $t > 0$ we have

$$\chi_0 W^t = \frac{1}{2\ell} \left(v_0 + 2 \sum_{k=1}^{\ell-1} (\lambda_k)^t v_k \right) .$$

Now it is easy to compute the exact probability of reaching the middle vertex $\frac{\ell}{2}$:

$$\begin{aligned} \chi_0 W^t\left(\frac{\ell}{2}\right) &= \frac{1}{2\ell} \left(v_0\left(\frac{\ell}{2}\right) + 2 \sum_{k=1}^{\ell-1} (\lambda_k)^t v_k\left(\frac{\ell}{2}\right) \right) = \frac{1}{\ell} \left(1 + 2 \sum_{k=1}^{\ell-1} \cos^{2t} \left(\frac{\pi k}{2\ell} \right) \cos \left(\frac{\pi k}{2} \right) \right) \\ &= \frac{1}{\ell} \left(1 + 2 \sum_{q=1}^{\ell/2-1} \cos^{2t} \left(\frac{2\pi q}{2\ell} \right) (-1)^q \right) \geq \frac{1}{\ell} \left(1 - 2 \cos^{2t} \left(\frac{\pi}{\ell} \right) \right) . \end{aligned}$$

At last, we translate this language into the PageRank vector pr_{χ_0} and obtain

$$\begin{aligned} pr_{\chi_0}\left(\frac{\ell}{2}\right) &= \sum_{t=0}^{\infty} \alpha (1 - \alpha)^t \chi_0 W^t\left(\frac{\ell}{2}\right) \geq \frac{1}{\ell} \left(\alpha v_{\ell}\left(\frac{\ell}{2}\right) + \sum_{t=0}^{\infty} \alpha (1 - \alpha)^t \left(1 - 2 \cos^{2t} \left(\frac{\pi}{\ell} \right) \right) \right) \\ &= \frac{1}{\ell} \left(\alpha v_{\ell}\left(\frac{\ell}{2}\right) + 1 - \frac{2\alpha}{1 - (1 - \alpha) \cos^2 \left(\frac{\pi}{\ell} \right)} \right) \\ &\geq \frac{1}{\ell} \left(1 - \frac{2\gamma}{\pi^2 + \gamma} - O\left(\frac{1}{\ell^2}\right) \right) . \end{aligned}$$

We remark here that the last inequality is obtained using Taylor approximation. \square

Lemma E.3. *Let ℓ be an even integer, and consider a chain of $\ell + 1$ vertices with the leftmost vertex indexed by 0 and the rightmost vertex indexed by ℓ . Let $pr_{\chi_{\ell/2}}$ be the PageRank vector for a random walk starting at the middle vertex $\ell/2$ with teleport probability $\alpha = \frac{\gamma}{\ell^2}$ for some constant γ . Then,*

$$pr_{\chi_{\ell/2}}\left(\frac{\ell}{2}\right) \leq \frac{1}{\ell} \left(1 + \sqrt{\gamma} + O\left(\frac{1}{\ell}\right) \right) .$$

Proof. Following the notion of λ_k and v_k in (E.1), we expand our starting probability vector $\chi_{\ell/2}$ under this orthonormal basis:

$$\chi_{\ell/2} = (0, \dots, 0, 1, 0, \dots, 0) = \frac{1}{2\ell} \left(v_0 + 2 \sum_{q=1}^{\ell/2-1} (-1)^q v_{2q} + (-1)^{\ell/2} v_{\ell} \right) .$$

Then similar to the proof of Lemma E.1 we have that for all $t > 0$

$$\chi_{\ell/2} W^t = \frac{1}{2\ell} \left(v_0 + 2 \sum_{q=1}^{\ell/2-1} (-1)^q (\lambda_{2q})^t v_{2q} \right) .$$

Now it is easy to compute the exact probability of reaching the middle vertex $\frac{\ell}{2}$:

$$\begin{aligned} \chi_{\ell/2} W^t \left(\frac{\ell}{2} \right) &= \frac{1}{2\ell} \left(v_0 \binom{\ell}{2} + 2 \sum_{q=1}^{\ell/2-1} (-1)^q (\lambda_{2q})^t v_{2q} \binom{\ell}{2} \right) = \frac{1}{\ell} \left(1 + 2 \sum_{q=1}^{\ell/2-1} (-1)^q \cos^{2t} \left(\frac{2\pi q}{2\ell} \right) \cos \left(\frac{2\pi q}{2} \right) \right) \\ &= \frac{1}{\ell} \left(1 + 2 \sum_{q=1}^{\ell/2-1} \cos^{2t} \left(\frac{2\pi q}{2\ell} \right) \right) = \frac{1}{\ell} \left(\frac{\ell}{2^{2t}} \sum_{k=-\lfloor t/\ell \rfloor}^{\lfloor t/\ell \rfloor} \binom{2t}{t+k\ell} \right). \end{aligned}$$

Notice that in the last equality we have used a recent result on power sum of cosines that can be found in Theorem 1 of [Mer12]. Next we perform some classical tricks on binomial coefficients:

$$\begin{aligned} \sum_{k=-\lfloor t/\ell \rfloor}^{\lfloor t/\ell \rfloor} \binom{2t}{t+k\ell} &= \binom{2t}{t} + 2 \sum_{k=1}^{\lfloor t/\ell \rfloor} \binom{2t}{t+k\ell} \\ &\leq \binom{2t}{t} + 2 \sum_{k=1}^{\lfloor t/\ell \rfloor} \frac{1}{\ell} \left(\binom{2t}{t+(k-1)\ell+1} + \binom{2t}{t+(k-1)\ell+2} + \dots + \binom{2t}{t+k\ell} \right) \\ &\leq \binom{2t}{t} + \frac{1}{\ell} \sum_{q=0}^{2t} \binom{2t}{q} \leq \frac{2^{2t}}{\sqrt{\pi t}} + \frac{2^{2t}}{\ell}, \end{aligned}$$

and in the last inequality we have used a famous upper bound on the central binomial coefficient that says $\binom{2t}{t} \leq \frac{2^{2t}}{\sqrt{\pi t}}$ for any integer $t \geq 1$ and $p \in \{0, 1, \dots, 2t\}$.

At last, we translate this language into the PageRank vector $pr_{\chi_{\ell/2}}$ and obtain

$$\begin{aligned} pr_{\chi_{\ell/2}} \left(\frac{\ell}{2} \right) &= \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \chi_{\ell/2} W^t \left(\frac{\ell}{2} \right) \leq \alpha + \frac{1}{\ell} \left(\sum_{t=1}^{\infty} \alpha(1-\alpha)^t \frac{\ell}{2^{2t}} \left(\frac{2^{2t}}{\sqrt{\pi t}} + \frac{2^{2t}}{\ell} \right) \right) \\ &= \alpha + \frac{1}{\ell} \left(1 + \sum_{t=1}^{\infty} \alpha(1-\alpha)^t \frac{\ell}{\sqrt{\pi t}} \right) \leq \alpha + \frac{1}{\ell} \left(1 + \int_{t=0}^{\infty} \alpha(1-\alpha)^t \frac{\ell}{\sqrt{\pi t}} dt \right) \\ &= \alpha + \frac{1}{\ell} \left(1 + \frac{\alpha\ell}{\sqrt{-\log(1-\alpha)}} \right) \leq \frac{1}{\ell} \left(1 + \sqrt{\gamma} + O\left(\frac{1}{\ell}\right) \right). \end{aligned}$$

We remark here that the last inequality is obtained using Taylor approximation. \square

Lemma E.4. *Consider an infinite chain with one special vertex called the origin. Note that the chain is infinite both to the left and to the right of the origin. Now we study the PageRank random walk on this infinite chain that starts from the origin with teleport probability $\alpha = \frac{\gamma}{\ell^2}$, and denote by $pr_{\chi_0}(0)$ be the probability of reaching the origin. Then,*

$$pr_{\chi_0}(0) \geq \frac{\sqrt{\pi\gamma}}{2\ell} - O\left(\frac{1}{\ell^2}\right).$$

Proof. As before we begin with the analysis of a lazy random walk of a fixed length t , and will translate it into the language of a PageRank random walk in the end. Suppose in the t actual number of steps, there are $t_1 \leq t$ number of them in which the random walk moves either to the left or to the right, while in the remaining $t - t_1$ of them the random walk stays. This happens with probability $\binom{t}{t_1} 2^{-t}$. When t_1 is fixed, to reach the origin it must be the case that among t_1 left-or-right moves, exactly $t_1/2$ of them are left moves, and the other half are right moves. This

happens with probability $\binom{t_1}{t_1/2}2^{-t_1}$. In sum, the probability to reach the origin in a t -step lazy random walk is:

$$\sum_{t_1=0}^t \binom{t}{t_1} 2^{-t} \binom{t_1}{t_1/2} 2^{-t_1} = \sum_{y=0}^{t/2} \binom{2y}{y} \binom{t}{2y} 2^{-2y-t} = \frac{1}{(t)!} \frac{(2t-1)!!}{2^t} = \frac{1}{(t)!} \cdot \frac{(2t)!}{t!2^{2t}} = \binom{2t}{t} 2^{-2t} \geq \frac{1}{\sqrt{4t}}.$$

Here in the last inequality we have used the famous lower bound on the central binomial coefficient that says $\binom{2t}{t} \geq \frac{2^{2t}}{\sqrt{4t}}$ for $t \geq 1$. At last, we translate this into the language of a PageRank random walk:

$$\begin{aligned} pr_{x_0}(0) &\geq \alpha + \sum_{t=1}^{\infty} \alpha(1-\alpha)^t \frac{1}{\sqrt{4t}} \geq \alpha + \int_{t=1}^{\infty} \alpha(1-\alpha)^t \frac{1}{\sqrt{4t}} dt = \alpha + \frac{\alpha\sqrt{\pi} \left(1 - \operatorname{erf}(\sqrt{-\log(1-\alpha)})\right)}{2\sqrt{-\log(1-\alpha)}} \\ &\geq \frac{\sqrt{\pi\gamma}}{2\ell} - O\left(\frac{1}{\ell^2}\right). \end{aligned}$$

Here in the last inequality we have used the Taylor approximation for the Gaussian error function erf . \square

E.2 Proof of Lemma 5.1

We are now ready to show the proof for Lemma 5.1.

Lemma 5.1. *For any $\gamma \in (0, 4]$ and letting $\alpha = \gamma/\ell^2$, there exists some constant c_0 such that when studying the PageRank vector pr_a starting from vertex a in Figure 1, it satisfies that $\frac{pr_a(d)}{\deg(d)} > \frac{pr_a(c)}{\deg(c)}$.*

We divide the proof into four steps. In the first step we provide an upper bound on $\frac{pr_a(c)}{\deg(c)}$ for vertex c , and in the second step we provide a lower bound on $\frac{pr_a(b)}{\deg(b)}$ for vertex b . Both these steps require a careful study on a finite chain (and in fact the top chain in Figure 1) which we have already done in Appendix E.1. They together will imply that

$$\frac{pr_a(b)}{\deg(b)} > (1 + \Omega(1)) \frac{pr_a(c)}{\deg(c)}. \quad (\text{E.2})$$

In the third step, we show that

$$\frac{pr_a(d)}{\deg(d)} > (1 - O(1)) \frac{pr_a(b)}{\deg(b)}, \quad (\text{E.3})$$

that is, the (normalized) probability for reaching d must be roughly as large as b . This is a result of the fact that, suppose towards contradiction that $\frac{pr_a(d)}{\deg(d)}$ is much smaller than $\frac{pr_a(b)}{\deg(b)}$, then there must be a large amount of probability mass moving from b to d due to the nature of PageRank random walk, while a large fraction of them should remain at vertex d due to the chain at the bottom, giving a contradiction to $\frac{pr_a(d)}{\deg(d)}$ being small.

And in the last step, we choose the constants very carefully to deduce $\frac{pr_a(d)}{\deg(d)} > \frac{pr_a(c)}{\deg(c)}$ out of (E.2) and (E.3).

Step 1: upper bounding $pr_a(c)/\deg(c)$. In the first step we upper bound the probability of reaching vertex c . Since removing the edges between b and d will disconnect the graph and thus only increase such probability, it suffices for us to consider just the top chain, which is equivalent

to the PageRank random walk on a finite chain of length $\ell + 1$ studied in Lemma E.1. In our language, taking into account the multi-edges, we have that

$$\begin{aligned} \frac{pr_a(c)}{\deg(c)} &\leq \frac{1}{n/\ell} \frac{1}{2\ell} \left(1 - \frac{2\gamma}{\pi^2/4 + \gamma} + \frac{2\gamma}{\pi^2 + \gamma} + O\left(\frac{1}{\ell^2}\right) \right) \\ &= \frac{1}{2n} \left(1 - \frac{2\gamma}{\pi^2/4 + \gamma} + \frac{2\gamma}{\pi^2 + \gamma} + O\left(\frac{1}{\ell^2}\right) \right). \end{aligned} \quad (\text{E.4})$$

Step 2: lower bounding $pr_a(b)/\deg(b)$. In this step we ask for help from a variant of Lemma 3.1. Letting \tilde{pr}_s be the PageRank vector on the induced subgraph $G[A]$ starting from s with teleport probability α , then Lemma 3.1 (and its actual proof) implies that $pr_a(b) \geq \tilde{pr}_a(b) - \tilde{pr}_l(b)$ where l is a vector that is only non-zero at the boundary vertex b , and in addition, $\|l\|_1 = l(b) \leq \frac{2\Psi}{\alpha}$ since $a \in A^g$ is a good starting vertex. We can rewrite this as

$$pr_a(b) \geq \tilde{pr}_a(b) - \frac{2\Psi}{\alpha} \tilde{pr}_b(b).$$

Next we use Lemma E.2 and Lemma E.3 to deduce that:

$$pr_a(b) \geq \frac{1}{\ell} \left(1 - \frac{2\gamma}{\pi^2 + \gamma} - O\left(\frac{1}{\ell^2}\right) \right) - \frac{2\Psi}{\alpha} \frac{1}{\ell} \left(1 + \sqrt{\gamma} + O\left(\frac{1}{\ell}\right) \right).$$

At last, we normalize this probability by its degree $\deg(b) = 2n/\ell + \Psi n$ and get:

$$\begin{aligned} \frac{pr_a(b)}{\deg(b)} &\geq \frac{1}{2n + \Psi n\ell} \left(1 - \frac{2\gamma}{\pi^2 + \gamma} - O\left(\frac{1}{\ell^2}\right) - \frac{2\Psi}{\alpha} \left(1 + \sqrt{\gamma} + O\left(\frac{1}{\ell}\right) \right) \right) \\ &\geq \frac{1}{2n} \left(1 - \frac{2\gamma}{\pi^2 + \gamma} - O(\Psi\ell^2) \right). \end{aligned} \quad (\text{E.5})$$

Step 3: lower bounding $pr_a(d)/\deg(d)$. Since we have already shown a good lower bound on $pr_a(b)/\deg(b)$ in the previous step, one may naturally guess that a similar lower bound should apply to vertex d as well because b and d are neighbors. This is not true in general, for instance if d were connected to a very large complete graph then all probability mass that reached d would be badly diluted. However, with our careful choice of the bottom chain, we will show that this is true in our case.

Lemma E.5. *Let $p^* \stackrel{\text{def}}{=} \frac{pr_a(b)}{\deg(b)}$, then either $\frac{pr_a(d)}{\deg(d)} \geq (1 - c_1)p^*$ or $\frac{pr_a(d)}{\deg(d)} \geq \frac{c_1 c_0}{2} p^* (1 - O(\frac{1}{\ell}))$.*

Proof. Throughout the proof we assume that $\frac{pr_a(d)}{\deg(d)} < (1 - c_1)p^*$ because otherwise we are done.

Therefore, we only need to show that $\frac{pr_a(d)}{\deg(d)} \geq \frac{c_1 c_0}{2} p^* (1 - O(\frac{1}{\ell}))$ is true under this assumption.

We first show a lower bound on the amount of *net* probability that will leak from A during the given PageRank random walk, i.e., $\text{NetLeakage} \stackrel{\text{def}}{=} \sum_{u \notin A} pr_a(u)$. Loosely speaking, this net probability is the amount of probability that will leak from A , subtracted by the amount of probability that will come back to A .

We introduce some notation first. Let $p^{(t)} \stackrel{\text{def}}{=} \chi_a W^t$ be the lazy random walk vector after t steps, and using the similar notation as Lemma 4.1, we let $p^{(t)}(b, d) \stackrel{\text{def}}{=} \frac{p^{(t)}(b)}{\deg(b)}$ be the amount of probability mass sent from b to d per edge at time step t to $t+1$, and similarly $p^{(t)}(d, b) \stackrel{\text{def}}{=} \frac{p^{(t)}(d)}{\deg(b)}$. If the PageRank random walk runs for a total of t steps (which happens with probability $\alpha(1 - \alpha)^t$),

then the total amount of net leakage becomes $\sum_{i=0}^{t-1} (p^{(i)}(b, d) - p^{(i)}(d, b)) \cdot \Psi n$. This gives another way to compute the total amount of net leakage of a PageRank random walk:

$$\begin{aligned}
\text{NetLeakage} &= \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \sum_{i=0}^{t-1} (p^{(i)}(b, d) - p^{(i)}(d, b)) \cdot \Psi n = \sum_{i=0}^{\infty} (p^{(i)}(b, d) - p^{(i)}(d, b)) \cdot \Psi n \sum_{t=i+1}^{\infty} \alpha(1-\alpha)^t \\
&= \sum_{i=0}^{\infty} (p^{(i)}(b, d) - p^{(i)}(d, b)) \cdot \Psi n \cdot (1-\alpha)^{i+1} = \frac{1-\alpha}{\alpha} \sum_{i=0}^{\infty} \alpha(1-\alpha)^i (p^{(i)}(b, d) - p^{(i)}(d, b)) \cdot \Psi n \\
&= \frac{1-\alpha}{\alpha} \left(\frac{pr_a(b)}{\deg(b)} - \frac{pr_a(d)}{\deg(d)} \right) \cdot \Psi n \geq \frac{1-\alpha}{\alpha} c_1 p^* \Psi n . \tag{E.6}
\end{aligned}$$

Now we have a decent lower bound on the amount of net leakage, and we want to further lower bound $pr_a(d)$ using this **NetLeakage** quantity. We achieve so by studying an auxiliary ‘‘random walk’’ procedure $q^{(t)}$, where $q^{(0)} = p^{(0)} = \chi_a$, but

$$q^{(t+1)} \stackrel{\text{def}}{=} q^{(t)}W + \delta^{(t)}, \text{ where } \delta^{(t)}(u) \begin{cases} 0, & \text{if } u \neq b, u \neq d; \\ p^{(t)}(d, b) \cdot \Psi n, & \text{if } u = b; \\ -p^{(t)}(b, d) \cdot \Psi n, & \text{if } u = d. \end{cases}$$

It is not hard to prove by induction that for all $t \geq 0$, it satisfies $q^{(t)}(u) = p^{(t)}(u)$ for $u \in A$ and $q^{(t)}(u) = 0$ for $u \notin A$.¹⁷ Then we have that:

$$\Delta \stackrel{\text{def}}{=} \sum_{t=0}^{\infty} \alpha(1-\alpha)^t q^{(t)} - pr_a$$

is precisely the vector that is zero everywhere in A and equal to pr_a everywhere in $V \setminus A$. We further notice that

$$\begin{aligned}
\Delta &= \sum_{t=0}^{\infty} \alpha(1-\alpha)^t (q^{(t)} - p^{(t)}) = \sum_{t=0}^{\infty} \alpha(1-\alpha)^t \left(\sum_{i=0}^{t-1} \delta^{(i)} W^{t-i-1} \right) \\
&= \sum_{k=0}^{\infty} \sum_{i=0}^{\infty} \alpha(1-\alpha)^{k+i+1} \delta^{(i)} W^k = \sum_{k=0}^{\infty} \alpha(1-\alpha)^k \left(\sum_{i=0}^{\infty} (1-\alpha)^{i+1} \delta^{(i)} \right) W^k .
\end{aligned}$$

Therefore, as long as we define $\delta \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} (1-\alpha)^{i+1} \delta^{(i)} = \frac{1-\alpha}{\alpha} \sum_{i=0}^{\infty} \alpha(1-\alpha)^i \delta^{(i)}$, we can write $\Delta = pr_{\delta}$ also as a PageRank vector. We highlight here that δ is a vector that is non-zero only at vertex b and d (and in fact $\delta(d) \geq 0$ and $\delta(b) \leq 0$), such that $\delta(d) + \delta(b) = \text{NetLeakage}$ according to the first equality in (E.6).

Now we are ready to lower bound $pr_a(d)$. Using the linearity of PageRank vectors we have

$$pr_a(d) = \Delta(d) = pr_{\delta}(d) = pr_{(\delta(d)\chi_d + \delta(b)\chi_b)}(d) = \delta(d) \cdot pr_d(d) + \delta(b) \cdot pr_b(d) \geq (\delta(d) + \delta(b)) \cdot pr_d(d)$$

where in the last inequality we have used $pr_b(d) \leq pr_d(d)$ which is true by monotonicity. Then we continue

$$pr_a(d) \geq (\text{NetLeakage}) \cdot pr_d(d) \geq \left(\frac{1-\alpha}{\alpha} c_1 p^* \Psi n \right) \cdot pr_d(d) \geq \left(\frac{1-\alpha}{\alpha} c_1 p^* \Psi n \right) \cdot \left(\frac{\pi\gamma}{2\ell} - O\left(\frac{1}{\ell^2}\right) \right)$$

¹⁷This is obvious when $t = 0$. For $q^{(t+1)}$, we compute $p^{(t+1)} = p^{(t)}W$ and $q^{(t+1)} = q^{(t)}W + \delta^{(t)}$. Based on the inductive assumption that the claim holds for $q^{(t)}$, it is automatically true that for $u \in A \setminus \{b\}$, $p^{(t+1)}(u) = q^{(t+1)}(u)$, and $u \in V \setminus (A \cup \{d\})$ we have $q^{(t+1)}(u) = 0$. For $u = b$ or $u = d$, one can carefully check that $\delta^{(t)}$ is introduced to precisely make $q^{(t+1)}(b) = p^{(t+1)}(b)$ and $q^{(t+1)}(d) = 0$, so the claim holds.

using (E.6) in the second inequality and Lemma E.4 in the last inequality, so we conclude that $pr_a(d) \geq \frac{c_1}{2} p^* \Psi n(\ell - O(1))$ and then $\frac{pr_a(d)}{\deg(d)} \geq \frac{c_1 c_0}{2} p^* (1 - O(\frac{1}{\ell}))$. \square

Step 4: putting it all together. We now define (using the fact that $\gamma > 0$ and $\gamma < 4$) constant c_2 to satisfy

$$1 - c_2 \stackrel{\text{def}}{=} \frac{1 - \frac{2\gamma}{\pi^2/4 + \gamma} + \frac{2\gamma}{\pi^2 + \gamma}}{1 - \frac{2\gamma}{\pi^2 + \gamma}} < 1 .$$

This constant is asymptotically the ratio between (E.4) and (E.5), so once we let $p^* \stackrel{\text{def}}{=} \frac{pr_a(b)}{\deg(b)}$ it satisfies that (using the fact that $\Psi \ell^2 = o(1)$)

$$\frac{pr_a(c)}{\deg(c)} \leq (1 - c_2) p^* (1 + o(1)) .$$

Next, if we choose $c_1 = \frac{c_2}{2}$ and $c_0 = \frac{2}{c_1}$ in Lemma E.5, this gives

$$\frac{pr_a(d)}{\deg(d)} \geq \min \left\{ 1 - \frac{c_2}{2}, 1 - O\left(\frac{1}{\ell}\right) \right\} p^* .$$

It is now clear from the above two inequalities that in the asymptotic case, i.e., when n, ℓ are sufficiently large, we always have $\frac{pr_a(d)}{\deg(d)} > \frac{pr_a(c)}{\deg(c)}$. This finishes the proof of Lemma 5.1. \blacksquare

References

- [ACE⁺13] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi. The evolution of sybil defense via social networks. In *IEEE Symposium on Security and Privacy*, 2013.
- [ACL06] Reid Andersen, Fan Chung, and Kevin Lang. Using pagerank to locally partition a graph. 2006. An extended abstract appeared in FOCS '2006.
- [AGM12] Reid Andersen, David F. Gleich, and Vahab Mirrokni. Overlapping clusters for distributed computation. WSDM '12, pages 273–282, 2012.
- [AHK10] Sanjeev Arora, Elad Hazan, and Satyen Kale. $O(\sqrt{\log(n)})$ approximation to sparsest cut in $\tilde{O}(n^2)$ time. *SIAM Journal on Computing*, 39(5):1748–1771, 2010.
- [AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. STOC '07, pages 227–236, 2007.
- [AL06] Reid Andersen and Kevin J. Lang. Communities from seed sets. WWW '06, pages 223–232, 2006.
- [Alo86] Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.
- [AP09] Reid Andersen and Yuval Peres. Finding sparse cuts locally using evolving sets. STOC, 2009.
- [ARV09] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM*, 56(2), 2009.
- [AvL10] Morteza Alamgir and Ulrike von Luxburg. Multi-agent random walks for local clustering on graphs. ICDM '10, pages 18–27, 2010.
- [CKK⁺06] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, June 2006.
- [GLMY11] Ullas Gargi, Wenjun Lu, Vahab S. Mirrokni, and Sangho Yoon. Large-scale community detection on youtube for topic discovery and exploration. In *AAAI Conference on Weblogs and Social Media*, 2011.
- [GS12] David F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *KDD '2012*, 2012.
- [GT12] Shayan Oveis Gharan and Luca Trevisan. Approximating the expansion profile and almost optimal local graph clustering. FOCS, pages 187–196, 2012.
- [Hav02] Taher H. Haveliwala. Topic-sensitive pagerank. In *WWW '02*, pages 517–526, 2002.
- [KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.
- [LC10] Frank Lin and William W. Cohen. Power iteration clustering. In *ICML '10*, pages 655–662, 2010.

- [LLDM09] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.
- [LLM10] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. WWW, 2010.
- [LR99] Frank Thomson Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.
- [LS90] László Lovász and Miklós Simonovits. The mixing rate of markov chains, an isoperimetric inequality, and computing the volume. FOCS, 1990.
- [LS93] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Struct. Algorithms*, 4(4):359–412, 1993.
- [Mer12] Mircea Merca. A note on cosine power sums. *Journal of Integer Sequences*, 15:12.5.3, May 2012.
- [MMV12] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *STOC '12*, pages 367–384, 2012.
- [MP03] Ben Morris and Yuval Peres. Evolving sets and mixing. STOC '03, pages 279–286. ACM, 2003.
- [MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.
- [Sch07] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.
- [She09] Jonah Sherman. Breaking the multicommodity flow barrier for $o(\sqrt{\log n})$ -approximations to sparsest cut. FOCS '09, pages 363–372, 2009.
- [SJ89] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82(1):93–133, 1989.
- [SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [SSS08] Shai Shalev-Shwartz and Nathan Srebro. SVM optimization: inverse dependence on training set size. In *ICML*, 2008.
- [ST04] Daniel Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. STOC, 2004.
- [ST08] Daniel Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. *CoRR*, abs/0809.3232, 2008.
- [WLS⁺12] Xiao-Ming Wu, Zhenguo Li, Anthony Man-Cho So, John Wright, and Shih-Fu Chang. Learning with partially absorbing random walks. In *NIPS*, 2012.

- [ZCZ⁺09] Zeyuan Allen Zhu, Weizhu Chen, Chenguang Zhu, Gang Wang, Haixun Wang, and Zheng Chen. Inverse time dependency in convex regularized learning. ICDM, 2009.