Detecting Sharp Drops in PageRank and a Simplified Local Partitioning Algorithm

Reid Andersen and Fan Chung

University of California at San Diego, La Jolla CA 92093, USA,
fan@ucsd.edu, http://www.math.ucsd.edu/~fan/,
randerse@ucsd.edu, http://www.math.ucsd.edu/~randerse/

Abstract. We show that whenever there is a sharp drop in the numerical rank defined by a personalized PageRank vector, the location of the drop reveals a cut with small conductance. We then show that for any cut in the graph, and for many starting vertices within that cut, an approximate personalized PageRank vector will have a sharp drop sufficient to produce a cut with conductance nearly as small as the original cut. Using this technique, we produce a nearly linear time local partitioning algorithm whose analysis is simpler than previous algorithms.

1 Introduction

When we are dealing with computational problems arising in complex networks with prohibitively large sizes, it is often desirable to perform computations whose cost can be bounded by a function of the size of their output, which may be quite small in comparison with the size of the whole graph. Such algorithms we call *local algorithms* (see [1]). For example, a local graph partitioning algorithm finds a cut near a specified starting vertex, with a running time that depends on the size of the small side of the cut, rather than the size of the input graph.

The first local graph partitioning algorithm was developed by Spielman and Teng [8], and produces a cut by computing a sequence of truncated random walk vectors. A more recent local partitioning algorithm achieves a better running time and approximation ratio by computing a single personalized PageRank vector [1]. Because a PageRank vector is defined recursively (as we will describe in the next section), a sweep over a single approximate PageRank vector can produce cuts with provably small conductance. Although this use of PageRank simplified the process of finding cuts, the analysis required to extend the basic cut-finding method into an efficient local partitioning algorithm remained complicated.

In this paper, we consider the following consequence of the personalized PageRank equation,

$$p = \alpha v + (1 - \alpha)pW,$$

where p is taken to be a row vector, v is the indicator vector for a single vertex v, and W is the probability transition matrix of a random walk on the graph (this will be defined in more detail later). When a random walk step is applied to the personalized PageRank vector p, every vertex in the graph has more probability from pW than it has from p, except for the seed vertex v. This implies something strong about the ordering of the vertices produced by the PageRank vector p: there cannot be many links between any set of vertices with high probability in p and any set of vertices with low probability in p. More precisely, whenever there is a *sharp drop* in probability, where the kth highest ranked vertex has much more probability than the $k(1+\delta)$th vertex, there must be few links between the k highest ranked vertices and the vertices not ranked in the top $k(1+\delta)$.

We will make this observation rigorous in Lemma 1, which provides an intuitive proof that personalized PageRank identifies a set with small conductance. In the section that follows, we will prove a series of lemmas that describe necessary conditions for a sharp drop to exist. In the final section, we will use these techniques to produce an efficient local partitioning algorithm, which finds cuts in nearly linear time by detecting sharp drops in approximate PageRank vectors.

2 Preliminaries

PageRank was introduced by Brin and Page [3, 7]. The PageRank vector $\mathrm{pr}(\alpha, s)$ is defined to be the unique solution of the equation

$$\mathrm{pr}(\alpha, s) = \alpha s + (1-\alpha)\mathrm{pr}(\alpha, s)W, \tag{1}$$

where α is a constant in $(0, 1]$ called the *teleportation constant*, s is a vector called the *starting vector*, and W is the random walk transition matrix $W = D^{-1}A$. Here D denotes the diagonal matrix whose diagonal entries are the degrees of the vertices, and A denotes the adjacency matrix of the graph.

The PageRank vector that is usually associated with search ranking has a starting vector equal to the uniform distribution $\frac{1}{n}\mathbf{1}$. PageRank vectors whose starting vectors are concentrated on a smaller set of vertices are often called *personalized PageRank vectors*. These were introduced by Haveliwala [5], and have been used to provide personalized search ranking and context-sensitive search [2, 4, 6]. We will consider PageRank vectors whose starting vectors are equal to the indicator function 1_v for a single vertex v. The vertex v will be called the *starting vertex*, and we will use the notation $\mathrm{pr}(\alpha, v) = \mathrm{pr}(\alpha, 1_v)$.

The *volume* of a subset $S \subseteq V$ of vertices is $\mathrm{Vol}(S) = \sum_{x \in S} d(x)$. We remark that $\mathrm{Vol}(V) = 2m$, and we will sometimes write $\mathrm{Vol}(G)$ in place of $\mathrm{Vol}(V)$. We write $e(S, T)$ to denote the number of edges between two disjoint sets of vertices S and T. The *conductance* of a set is

$$\Phi(S) = \frac{e(S, T)}{\min\left(\mathrm{Vol}(S), 2m - \mathrm{Vol}(S)\right)}.$$

The amount of probability from a vector p on a set S of vertices is written $p(S) = \sum_{x \in S} p(x)$. We will sometimes refer to the quantity $p(S)$ as an amount of probability even if $p(V)$ is not equal to 1. As an example of this notation, the amount of probability from the PageRank vector $\mathrm{pr}(\alpha, v)$ on a set S will be written $\mathrm{pr}(\alpha, \chi_v)(S)$. The support of a vector is the set of vertices on which it is nonzero, $\mathrm{Support}(p) = \{v \mid p(v) \neq 0\}$.

2.1 Approximate Personalized PageRank Vectors

Here are some useful properties of PageRank vectors (also see [5] and [6]).

Proposition 1. *For any starting vector s, and any constant α in $(0, 1]$, there is a unique vector $\mathrm{pr}(\alpha, s)$ satisfying $\mathrm{pr}(\alpha, s) = \alpha s + (1 - \alpha)\mathrm{pr}(\alpha, s)W$.*

Proposition 2. *For any fixed value of α in $(0, 1]$, there is a linear transformation R_α such that $\mathrm{pr}(\alpha, s) = sR_\alpha$. Furthermore, R_α is given by the matrix*

$$R_\alpha = \alpha I + \alpha \sum_{t=1}^{\infty} (1 - \alpha)^t W^t. \tag{2}$$

This implies that a PageRank vector $\mathrm{pr}(\alpha, s)$ is linear in its starting vector s.

Instead of computing the PageRank vector $\mathrm{pr}(\alpha, v)$ exactly, we will approximate it by another PageRank vector $\mathrm{pr}(\alpha, v - r)$ with a slightly different starting vector, where r is a vector with nonnegative entries. If $r(v) \leq \epsilon d(v)$ for every vertex in the graph, then we say $\mathrm{pr}(\alpha, v - r)$ is an ϵ-approximate PageRank vector for $\mathrm{pr}(\alpha, v)$.

Definition 1. *An* ϵ-approximate *PageRank vector for $\mathrm{pr}(\alpha, v)$ is a PageRank vector $\mathrm{pr}(\alpha, v - r)$ where the vector r is nonnegative and satisfies $r(u) \leq \epsilon d(u)$ for every vertex u in the graph.*

We will use the algorithm `ApproximatePR`(v, α, ϵ) described in the following theorem to compute ϵ-approximate PageRank vectors with small support. The running time of the algorithm depends on ϵ and α, but is independent of the size of the graph.

Theorem 1. *For any vertex v, any constant $\alpha \in (0, 1]$, and any constant $\epsilon \in (0, 1]$, The algorithm `ApproximatePR`(v, α, ϵ) computes an ϵ-approximate PageRank vector $p = \mathrm{pr}(\alpha, v - r)$ with support $\mathrm{Vol}(\mathrm{Support}(p)) \leq \frac{2}{(1-\alpha)\epsilon}$. The running time of the algorithm is $O(\frac{1}{\epsilon\alpha})$.*

The proof of this theorem, and the description of the algorithm, were given in [1]. We will use the algorithm as a black box.

3 A sharp drop in PageRank implies a good cut

We describe a normalized rank function derived from a PageRank vector, and the ordering of the vertices induced by that rank function.

Definition 2. *Given a PageRank vector p, we define the following.*

- *Define the rank function q to be $q(u) = p(u)/d(u)$.*
- *Let π be a permutation that places the vertices in nonincreasing order of rank, so that*
$$q(\pi(1)) \geq q(\pi(2)) \geq \cdots \geq q(\pi(n)).$$
This is the ordering induced by the PageRank vector. An integer $j \in [1, n]$ will be called an index, and we will say that $\pi(j)$ is the vertex at index j.
- *Let $S_j = \{\pi(1), \ldots, \pi(j)\}$ be the set containing the j vertices of highest rank. The set S_j is called the jth level set of the PageRank vector.*
- *Define the shorthand notation $q(j) = q(\pi(j))$ and $V(j) = \mathrm{Vol}(S_j)$.*

We now prove the main lemma. If there is a sharp drop in rank at S_j, then the set S_j has small conductance. We will prove the contrapositive instead, because that is how we will eventually apply the lemma. Namely, we will prove that either the set S_j has small conductance, or else there is an index $k > j$ where the volume $\mathrm{Vol}(S_k)$ is significantly larger than $\mathrm{Vol}(S_j)$, and the rank $q(k)$ is not much smaller than $q(j)$.

Lemma 1 (Sharp Drop Lemma). *Let $p = \mathrm{pr}(\alpha, v - r)$ be an approximate PageRank vector. Let $\phi \in (0, 1)$ be a real number, and let j be any index in $[1, n]$. Either the number of edges leaving S_j satisfies $e(S_j, \bar{S}_j) < 2\phi\mathrm{Vol}(S_j)$, or else there is some index $k > j$ such that*

$$\mathrm{Vol}(S_k) \geq \mathrm{Vol}(S_j)(1 + \phi) \qquad \textit{and} \qquad q(k) \geq q(j) - \alpha/\phi\mathrm{Vol}(S_j).$$

Proof. For any set S of vertices,

$$pW(S) = p(S) - \sum_{(u,v) \in e(S, \bar{S})} q(u) - q(v).$$

Since $p = \mathrm{pr}(\alpha, v - r)$ and the vector r is nonnegative,

$$pW = (1 - \alpha)^{-1}(p - \alpha(v - r)) \geq p - \alpha v.$$

The two equations above imply that

$$\sum_{(u,v) \in e(S, \bar{S})} q(u) - q(v) \leq \alpha. \tag{3}$$

Now consider the level set S_j. If $\text{Vol}(S_j)(1+\phi) > \text{Vol}(G)$, then

$$e(S_j\bar{S}_j) \leq \text{Vol}(G)(1 - \frac{1}{1+\phi}) \leq \phi\text{Vol}(S_j),$$

and the theorem holds trivially. If $\text{Vol}(S_j)(1+\phi) \leq \text{Vol}(G)$, then there is a unique index k such that

$$\text{Vol}(S_{k-1}) \leq \text{Vol}(S_j)(1+\phi) \leq \text{Vol}(S_k).$$

If $e(S_j, \bar{S}_j) < 2\phi\text{Vol}(S_j)$, we are done. If $e(S_j, \bar{S}_j) \geq 2\phi\text{Vol}(S_j)$, then $e(S_j, \bar{S}_{k-1})$ is also large,

$$e(S_j, \bar{S}_{k-1}) \geq \partial(S_j) - \text{Vol}(S_{k-1} \setminus S_j) \geq 2\phi\text{Vol}(S_j) - \phi\text{Vol}(S_j) = \phi\text{Vol}(S_j).$$

Using equation (3),

$$\begin{aligned}\alpha \geq \sum_{(u,v)\in e(S_j,\bar{S}_j)} q(u) - q(v) &\geq \sum_{(u,v)\in e(S_j,\bar{S}_{k-1})} q(u) - q(v) \\ &\geq e(S_j, \bar{S}_{k-1})(q(j) - q(k)) \\ &\geq \phi\text{Vol}(S_j) \cdot (q(j) - q(k)).\end{aligned}$$

This shows that $q(j) - q(k) \leq \alpha/\phi\text{Vol}(S_j)$, completing the proof.

4 Ensuring that a sharp drop exists

In this section, we will introduce several tools that will allow us to show that a sharp drop in rank exists for many personalized PageRank vectors. When we present the local partitioning algorithm in the next section, these tools will be used to prove its approximation guarantee.

Throughout this section and the next, we have two PageRank vectors to consider, the PageRank vector $p = \text{pr}(\alpha, v)$, and the approximate PageRank vector $\tilde{p} = \text{pr}(\alpha, v - r)$ that will be computed by the local partitioning algorithm. These two PageRank vectors induce two different orderings π and $\tilde{\pi}$, which lead to two different rank functions $q(k)$ and $\tilde{q}(k)$, which produce two collections of level sets S_k and $\tilde{S}_k$, which have different volumes $V(k) = \text{Vol}(S_k)$ and $\tilde{V}(k) = \text{Vol}(\tilde{S}_k)$.

We start by showing there is some index i where the rank $q(i)$ is not much smaller than $1/V(i)$. This lemma doesn't use any special properties of PageRank vectors, and is true for any nonnegative vector whose entries sum to 1.

Lemma 2 (Integration Lemma). *Let q be the rank function of any vector p for which $\|p\|_1 = 1$. Then, there exists an index i such that $q(i) \geq \frac{1}{H(2m)V(i)}$, where $H(2m) = \sum_{k=1}^{2m} 1/k = O(\log m)$.*

Proof. If we assume that $q(i) < c/V(i)$ for all $i \in [1, n]$, then

$$\begin{aligned}\sum_{i=1}^{n} q(i)d(i) &< c \sum_{i=1}^{n} \frac{d(i)}{V(i)} \\ &\leq c \sum_{k=1}^{2m} \frac{1}{k} \\ &= cH(2m).\end{aligned}$$

If $c = 1/H(2m)$, this would imply $\|p\|_1 = \sum_{i=1}^{n} q(i)d(i) < 1$, so we must have $q(i) \geq \frac{1}{H(2m)V(i)}$ for some index i.

We now give a lower bound on the rank function of an ϵ-approximate PageRank vector $\tilde{p} = \mathrm{pr}(\alpha, v - r)$ that depends on the rank function of the PageRank vector $p = \mathrm{pr}(\alpha, v)$ that is being approximated, and on the error parameter ϵ.

Lemma 3 (Approximation Error Lemma). *Let q be the rank function for a PageRank vector $p = \mathrm{pr}(\alpha, v)$, and let $\tilde{q}$ be the rank for an ϵ-approximate PageRank vector $\tilde{p} = \mathrm{pr}(\alpha, v - r)$. For any index i, there is an index j such that*

$$\tilde{q}(j) \geq q(i) - \epsilon \qquad \textit{and} \qquad \mathrm{Vol}(\tilde{S}_j) \geq \mathrm{Vol}(S_i).$$

Proof. If $v \in S_i$, then $p(v)/d(v) \geq q(i)$. Since $\tilde{p}$ is an ϵ-approximation of p,

$$\tilde{p}(v)/d(v) \geq p(v)/d(v) - \epsilon \geq q(i) - \epsilon.$$

Therefore, the set of vertices for which $\tilde{p}(v)/d(v) \geq q(i) - \epsilon$ has volume at least $\mathrm{Vol}(S_j)$, which proves the lemma.

The following lemma shows what happens when you repeatedly apply the Sharp Drop Lemma, but fail to find a cut with small conductance. You will find a sequence of larger and larger indices for which the rank does not drop very quickly. We give a lower bound on the rank of the final index in the sequence. We will eventually contradict this lower bound, which will show that one of the applications of the Sharp Drop Lemma finds a cut with small conductance.

Lemma 4 (Chaining Lemma). *Let $\{k_0 \ldots k_f\}$ be an increasing sequence of indices such that for each $i \in [0, f-1]$, the following holds.*

$$q(k_{i+1}) \geq q(k_i) - \alpha/\phi V(k_i) \qquad \textit{and} \qquad V(k_{i+1}) \geq (1 + \phi)V(k_i).$$

Then, the last index k_f satisfies

$$q(k_f) \geq q(k_0) - 2\alpha/\phi^2 V(k_0).$$

Proof. The bound on the change in volume implies that $V(k_i) \geq (1+\phi)^i V(k_0)$ for all $i \in [0, f-1]$. Therefore,

$$\begin{aligned} q(k_f) &\geq q(k_0) - \frac{\alpha}{\phi V(k_0)} - \frac{\alpha}{\phi V(k_1)} - \cdots - \frac{\alpha}{\phi V(k_{f-1})} \\ &\geq q(k_0) - \frac{\alpha}{\phi V(k_0)} \left(1 - \frac{1}{(1+\phi)} - \cdots - \frac{1}{(1+\phi)^{f-1}}\right) \\ &\geq q(k_0) - \frac{\alpha}{\phi V(k_0)} (\frac{1+\phi}{\phi}) \\ &\geq q(k_0) - \frac{2\alpha}{\phi^2 V(k_0)}. \end{aligned}$$

This completes the proof of the Chaining Lemma.

To contradict the lower bound from the Chaining Lemma, we will place a lower bound on $\mathrm{pr}(\alpha, v)(C)$, that depends on the conductance of C. This bound will apply to many starting vertices v within C.

Definition 3. *Given a set C, let C_α be the set of vertices v within C such that $\mathrm{pr}(\alpha, v)(\bar{C})$ is at most $2\Phi(C)/\alpha$.*

Lemma 5 (Probability Capturing Lemma). *For any set C and value of α, we have $\mathrm{Vol}(C_\alpha) \geq (1/2)\mathrm{Vol}(C)$.*

Proof. Let π_C be the probability distribution obtained by sampling a vertex v from C with probability $d(v)/\mathrm{Vol}(C)$. It is not difficult to verify the following statement, which was proved in [1].

$$\mathrm{pr}(\alpha, \pi_C) W(\bar{C}) \leq \mathrm{pr}(\alpha, \pi_C)(\bar{C}) + \Phi(C).$$

We will apply this observation to the personalized PageRank equation.

$$\begin{aligned} \mathrm{pr}(\alpha, \pi_C)(\bar{C}) &= [\alpha \pi_C + (1-\alpha)\mathrm{pr}(\alpha, \pi_C) W](\bar{C}) \\ &= (1-\alpha)[\mathrm{pr}(\alpha, \pi_C) W](\bar{C}) \\ &\leq (1-\alpha)\mathrm{pr}(\alpha, \pi_C)(\bar{C}) + \Phi(C). \end{aligned}$$

This implies

$$\mathrm{pr}(\alpha, \pi_C)(\bar{C}) \leq \Phi(C)/\alpha.$$

If we sample a vertex v from the distribution π_C, then at least half of the time $\mathrm{pr}(\alpha, v)(\bar{C})$ is at most twice its expected value of $\mathrm{pr}(\alpha, \pi_C)(\bar{C})$, and hence at least half the time v is in C_α. This implies that the volume of the set C_α is at least half the volume of C.

5 Local partitioning algorithm

The local partitioning algorithm can be described as follows:

`Local Partition`(v, ϕ, x):
The input to the algorithm is a starting vertex v, a target conductance $\phi \in (0, 1/3)$, and a target volume $x \in [0, 2m]$.

PageRank computation:

1. Let $\gamma = H(2m)$, let $\alpha = \frac{\phi^2}{8\gamma}$, and let $\epsilon = \frac{1}{2\gamma x}$.
2. Compute an ϵ-approximate PageRank vector $\tilde{p} = \mathrm{pr}(\alpha, v - r)$, using `ApproximatePR`$(v, \alpha, \epsilon)$.
3. Order the vertices so that $\tilde{q}(1) \geq \tilde{q}(2) \geq \cdots \geq \tilde{q}(n)$.

Finding a cut:
The algorithm will now examine a sequence of vertices, looking for a sharp drop. We let $\{k_i\}$ be the indices of the vertices examined by the algorithm. The first index examined by the algorithm will be k_0, and the last index examined will be k_f. We will now describe how these indices are chosen.

1. Let the starting index k_0 be the largest index such that $\tilde{q}(k_0) \geq 1/2\gamma\tilde{V}(k_0)$. If no such index exists, halt and output FAIL: NO STARTING INDEX.
2. While the algorithm is still running:
   (a) If $(1+\phi)\tilde{V}(k_i) > \mathrm{Vol}(G)$ or if $\tilde{V}(k_i) > \mathrm{Vol}(\mathrm{Support}(\tilde{p}))$, then let $k_f = k_i$, output FAIL: NO CUT FOUND and quit.
   (b) Otherwise, let k_{i+1} be the smallest index such that $\tilde{V}(k_{i+1}) \geq \tilde{V}(k_i)(1+\phi)$.
   (c) If $\tilde{q}(k_{i+1}) \leq \tilde{q}(k_i) - \alpha/\phi\tilde{V}(k_i)$, then let $k_f = k_i$, output the set S_{k_i}, and quit. Otherwise, repeat the loop.

Remarks:
When we analyze the algorithm, we will need the following observations. Regardless of whether the algorithm fails or successfully outputs a cut, the sequence of indices $\{k_0, \ldots, k_f\}$ examined by the algorithm satisfies the conditions of Lemma 4. If the algorithm fails during the loop of step 2, then k_f satisfies either $(1+\phi)\tilde{V}(k_f) > \mathrm{Vol}(G)$ or $\tilde{V}(k_f) > \mathrm{Vol}(\mathrm{Support}(\tilde{p}))$.

Theorem 2. *The running time of* `Local Partition`(v, ϕ, x) *is* $O(x\frac{\log^2 m}{\phi^2})$.

Proof. The running time of the algorithm is dominated by the time to compute and sort $\tilde{p}$. Computing $\tilde{p}$ can be done in time $O(1/\epsilon\alpha) = O(x\gamma/\alpha) = O(x\frac{\log m}{\alpha})$

using `ApproximatePR`. The support of this vector has volume $O(1/\epsilon) = O(\gamma x) = O(x \log m)$, so the time required to sort $\tilde{p}$ is

$$O(|\text{Support}(\tilde{p})| \log |\text{Support}(\tilde{p})|) = O(x \log^2 m).$$

Since we have set $\alpha = \Omega(\phi^2/\log m)$, the total running time is

$$O(x\frac{\log m}{\alpha} + x \log^2 m) = O(x\frac{\log^2 m}{\phi^2}).$$

Theorem 3. *Consider a run of the algorithm* `Local Partition` *on the input values v, ϕ, and x. Let $\tilde{p} = \text{pr}(\alpha, v - r)$ be the ϵ-approximate PageRank vector computed by the algorithm. Note that $\alpha = \phi^2/8\gamma$ and $\epsilon = 1/2\gamma x$. The following statements are true.*

1. *Let q be the rank function of the PageRank vector $p = \text{pr}(\alpha, v)$. There exists an index K such that $q(K) \geq 1/\gamma\text{Vol}(S_K)$. Furthermore, if the target volume x satisfies $x \geq \text{Vol}(S_K)$, then the algorithm finds a starting index k_0 that satisfies $\text{Vol}(\tilde{S}_{k_0}) \geq \text{Vol}(S_K)$.*
2. *Assume there exists a set C whose volume satisfies $\text{Vol}(C) \leq \frac{1}{2}\text{Vol}(G)$, whose conductance satisfies $\Phi(C) \leq \alpha/80\gamma$, and for which the starting vertex v is contained in C_α. If the target volume x satisfies $x \geq \text{Vol}(S_K)$, then the algorithm successfully outputs a set. Furthermore, the set S output by the algorithm has the following properties.*
   (a) *(Approximation guarantee) $\Phi(S) \leq 3\phi = 3\sqrt{8\gamma\alpha}$.*
   (b) *(Volume lower bound) $\text{Vol}(S) \geq \text{Vol}(S_K)$.*
   (c) *(Volume upper bound) $\text{Vol}(S) \leq (5/9)\text{Vol}(G)$.*
   (d) *(Intersection with C) $\text{Vol}(S \cap C) \geq (9/10)\text{Vol}(S)$.*

Proof. We begin by considering the PageRank vector $p = \text{pr}(\alpha, v)$ in order to prove claim 1 of the theorem. Lemma 2 shows that there is some index K for which $q(K) \geq 1/\gamma\text{Vol}(S_K)$, which proves part of the claim. To prove the other part, we assume that $x \geq \text{Vol}(S_K)$, and show that the algorithm finds a starting index k_0 that satisfies $\text{Vol}(\tilde{S}_{k_0}) \geq \text{Vol}(S_K)$. Lemma 3 shows that there exists an index j such that $\text{Vol}(\tilde{S}_j) \geq \text{Vol}(S_K)$ and

$$\tilde{q}(j) \geq q(K) - \epsilon \geq \frac{1}{\gamma\text{Vol}(S_K)} - \epsilon.$$

Since $x \geq \text{Vol}(S_K)$, we have $\epsilon = 1/2\gamma x \leq 1/2\gamma\text{Vol}(S_K)$, which implies the following.

$$\tilde{q}(j) \geq \frac{1}{\gamma\text{Vol}(S_K)} - \frac{1}{2\gamma x} \geq \frac{1}{2\gamma\text{Vol}(S_K)} \geq \frac{1}{2\gamma\text{Vol}(\tilde{S}_j)}.$$

This shows that j may be chosen as a starting index, so the algorithm is assured of choosing some starting index $k_0 \geq j$, which we know will satisfy

$$\mathrm{Vol}(\tilde{S}_{k_0}) \geq \mathrm{Vol}(S_K) \qquad \text{and} \qquad \tilde{q}(k_0) \geq \frac{1}{2\gamma \mathrm{Vol}(\tilde{S}_{k_0})}.$$

This proves Claim 1 of the theorem.

We now move on to the proof of Claim 2. Let k_f be the index of the last vertex considered by the algorithm. We will give a lower bound on $\tilde{q}(k_f)$. Because the rank $\tilde{q}(k_{i+1})$ is not much smaller than $\tilde{q}(k_i)$ at each step, Lemma 4 shows that

$$\tilde{q}(k_f) \geq \tilde{q}(k_0) - \frac{(2\alpha/\phi^2)}{\mathrm{Vol}(\tilde{S}_{k_0})}.$$

We have set $\alpha = \phi^2/8\gamma$ to ensure that $2\alpha/\phi^2 \leq 1/4\gamma$, and so

$$\begin{aligned}
\tilde{q}(k_f) &\geq \tilde{q}(k_0) - \frac{(2\alpha/\phi^2)}{\mathrm{Vol}(\tilde{S}_{k_0})} \\
&\geq \frac{1}{2\gamma \mathrm{Vol}(\tilde{S}_{k_0})} - \frac{1}{4\gamma \mathrm{Vol}(\tilde{S}_{k_0})} \\
&\geq \frac{1}{4\gamma \mathrm{Vol}(\tilde{S}_{k_0})}.
\end{aligned}$$

We now use the assumptions that $v \in C_\alpha$ and $\Phi(C) \leq \alpha/80\gamma$, and apply Lemma 5 to give the following bound on $\tilde{p}(\bar{C})$.

$$\tilde{p}(\bar{C}) \leq p(\bar{C}) \leq 2\Phi(C)/\alpha \leq 1/40\gamma.$$

Combining our lower bound on $\tilde{q}(k_f)$ with our upper bound on $\tilde{p}(\bar{C})$ gives the following bound on the intersection of $\tilde{S}_{k_f}$ with $\bar{C}$.

$$\begin{aligned}
\mathrm{Vol}(\tilde{S}_{k_f} \cap \bar{C}) &\leq \frac{\tilde{p}(\bar{C})}{\tilde{q}(k_f)} \\
&\leq \frac{4\gamma \mathrm{Vol}(\tilde{S}_{k_0})}{40\gamma} \\
&\leq \frac{1}{10}\mathrm{Vol}(\tilde{S}_{k_f}).
\end{aligned}$$

This implies

$$\mathrm{Vol}(\tilde{S}_{k_f}) \leq \mathrm{Vol}(C) + \mathrm{Vol}(\tilde{S}_{k_f} \cap \bar{C}) \leq \mathrm{Vol}(C) + \frac{1}{10}\mathrm{Vol}(\tilde{S}_{k_f}).$$

We now use the fact that $\mathrm{Vol}(C) \leq (1/2)\mathrm{Vol}(G)$,

$$\mathrm{Vol}(\tilde{S}_{k_f}) \leq (10/9)\mathrm{Vol}(C) \leq (5/9)\mathrm{Vol}(G) \leq \frac{\mathrm{Vol}(G)}{1+\phi}.$$

The last step follows by assuming that $\phi \leq 1/3$. We can do so without loss of generality because the approximation guarantee of the theorem is vacuous if $\phi \geq 1/3$.

The equation above shows that the algorithm will not experience a failure caused by $(1+\phi)\text{Vol}(\tilde{S}_{k_f}) > \text{Vol}(G)$, and our lower bound on $\tilde{q}(k_f)$ ensures that the algorithm will not experience a failure caused by $\text{Vol}(\tilde{S}_{k_f}) > \text{Vol}(\text{Support}(\tilde{p}))$. This ensures that the algorithm does not fail and output FAIL: NO CUT FOUND. We have already ensured that the algorithm does not fail and output FAIL: NO STARTING INDEX. Since we have ruled out all of the possible failure conditions, the algorithm must successfully output a set.

We must still prove that the set output by the algorithm satisfies all the properties in claim 2. We have already proved that $\text{Vol}(\tilde{S}_{k_f}) \leq (5/9)\text{Vol}(G)$, which proves claim 2(b). We have proved $\text{Vol}(\tilde{S}_{k_f} \cap \bar{C}) \leq \frac{1}{10}\text{Vol}(\tilde{S}_{k_f})$, which proves claim (2d). We have proved $\text{Vol}(\tilde{S}_{k_f}) \geq \text{Vol}(\tilde{S}_{k_0}) \geq \text{Vol}(S_K)$, which proves claim (2c).

For the coup de grâce, we will apply the Sharp Drop Lemma. Since the set $\tilde{S}_{k_f}$ was output by the algorithm, that set must have the following property: if k_{f+1} is the smallest index such that $\text{Vol}(\tilde{S}_{k_{f+1}}) \geq \text{Vol}(\tilde{S}_{k_f})(1+\phi)$, then $\tilde{q}(k_{f+1}) < \tilde{q}(k_f) - \alpha/\phi \text{Vol}(\tilde{S}_{k_f})$. We can then apply the Sharp Drop Lemma to show that the number of edges leaving S_{k_f} satisfies $e(\tilde{S}_{k_f}, \bar{\tilde{S}}_{k_f}) < 2\phi\text{Vol}(\tilde{S}_{k_f})$. Since $\text{Vol}(\tilde{S}_{k_f}) \leq \frac{5}{9}\text{Vol}(G)$, we have $\text{Vol}(G) - \text{Vol}(\tilde{S}_{k_f}) \geq \frac{4}{9}\text{Vol}(G) \geq \frac{4}{5}\text{Vol}(\tilde{S}_{k_f})$, and so

$$\begin{aligned}
\Phi(\tilde{S}_{k_f}) &= \frac{e(\tilde{S}_{k_f}, \bar{\tilde{S}}_{k_f})}{\min(\text{Vol}(\tilde{S}_{k_f}), \text{Vol}(G) - \text{Vol}(\tilde{S}_{k_f}))} \\
&\leq \frac{2\phi\text{Vol}(\tilde{S}_{k_f})}{(4/5)\text{Vol}(\tilde{S}_{k_f})} \\
&\leq 3\phi.
\end{aligned}$$

This proves the approximation guarantee of claim (2a).

References

1. R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *Proc. 47th Annual Symposium on Foundations of Computer Science*, (2006).
2. P. Berkhin, Bookmark-Coloring Approach to Personalized PageRank Computing, *Internet Mathematics*, to appear.
3. S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, (1998), 107–117.
4. D. Fogaras and B. Racz, Towards scaling fully personalized PageRank, *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW)*, (2004), 105–117.

5. T. H. Haveliwala, Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search, *IEEE Trans. Knowl. Data Eng.*, (2003), 784-796.
6. G. Jeh and J. Widom, Scaling personalized web search, *Proceedings of the 12th World Wide Web Conference (WWW)*, (2003), 271-279.
7. L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, *Stanford Digital Library Technologies Project*, 1998.
8. D. A. Spielman and S.-H. Teng, Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems, *ACM STOC* (2004), 81–90.