Localization in Seeded PageRank

David F. Gleich, Kyle Kloster, Huda Nassar

{Computer Science, Mathematics, Computer Science} Department

Purdue University

{dgleich, kkloste, hnassar}@purdue.edu

May 23, 2017

Seeded PageRank is an important network analysis tool for identifying and studying regions nearby a given set of nodes, which are called seeds. The seeded PageRank vector is the stationary distribution of a random walk that randomly resets at the seed nodes. Intuitively, this vector is concentrated nearby the given seeds, but is mathematically non-zero for all nodes in a connected graph. We study this concentration, or localization, and show a sublinear upper bound on the number of entries required to approximate seeded PageRank on all graphs with a natural type of skewed-degree sequence—similar to those that arise in many real-world networks. Experiments with both real-world and synthetic graphs give further evidence to the idea that the degree sequence of a graph has a major influence on the localization behavior of seeded PageRank. Moreover, we establish that this localization is non-trivial by showing that complete-bipartite graphs produce seeded PageRank vectors that cannot be approximated with a sublinear number of non-zeros.

1 Introduction

The PageRank vector of a graph has a number of different derivations, but the most famous is via the random surfer process [Page et al., 1999]. As PageRank's random surfer moves between the nodes of a graph, at each node with probability α the surfer chooses to follow a random neighbor (chosen uniformly at random over all neighbors), and otherwise it resets its state with probability $1 - \alpha$. On a reset, the surfer *teleports* to a new state chosen from a *reset distribution*. The PageRank vector is the long-term fraction of time the process spends at each node, or, equivalently, the stationary distribution. (The PageRank process is a special case where these two concepts are the same.) *Seeded* PageRank refers to PageRank vectors where the reset distribution is a single node. Originally, PageRank vectors with these simple reset distributions were called personalized PageRank vectors due to PageRank's origins on the web, but we feel "seeded PageRank" is more appropriate given the vast number of uses of PageRank outside the web [Gleich, 2015]. Furthermore, large entries in seeded PageRank vectors are often used to identify *regions* nearby the seed node that support various network analysis tasks, see the survey [Gleich, 2015] for many examples.

A seeded PageRank vector is determined by three inputs:

1. the underlying graph, represented by a column-stochastic probability transition matrix $\boldsymbol{P}$ where $P_{j,i} = 1/\mathrm{degree}(i)$ for each directed edge (i, j) between nodes i and j (we label all nodes with a unique integer index and assume that all nodes have degree at least 1);

2. a parameter α that determines the reset probability $(1 - \alpha)$; and

3. a seed node, s.

In our notation, the vector $\mathbf{e}_s$ is the vector of all zeros with a single 1 in the position corresponding to node s. The seeded PageRank vector $\mathbf{x}$ is then the solution of the linear system:

$$(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{e}_s.$$

When the network is strongly connected, the solution $\mathbf{x}$ is non-zero for all nodes. This is a consequence of the fact that there is a non-zero probability of walking from the seed to any other node in a strongly connected network.

A vector or sequence is localized when we can globally approximate all of the entries using an asymptotically vanishing subset. For example, we can globally approximate a convergent infinite geometric series with only a few of the largest entries. We seek a theoretical understanding of localization in seeded PageRank vectors and are concerned with the minimal number of entries required to approximate these vectors. As a concrete example of localization, in Figure 1 we plot the seeded PageRank vector for a randomly chosen node in a 1.7 million node graph with 22 million edges. (The graph is the largest connected component of the as-skitter network [Leskovec and Sosič, 2014, Leskovec et al., 2005].) The true PageRank solution is non-zero everywhere, and the two plots illustrate different aspects of what localization looks like in a "large vector." In this case, the plots show that we can approximate the seeded PageRank vector $\mathbf{x}$ with 1.7 million entries using only roughly 10,000 of the entries and attain a global accuracy of 10^{-2}. (Formally, $\|\mathbf{x} - \mathbf{y}\|_1 \leq 10^{-2}$ where $\mathbf{y}$ is composed of just the largest 10,000 entries from $\mathbf{x}$.) Our goal is to understand when this behavior is possible.

Our results establish that all graphs with a specific, but realistic, type of degree distribution have seeded PageRank vectors that can be approximated by vectors with only a sublinear number of non-zero entries. At this point, we state a simplified version of our main theorem to give the reader an intuition for how the bound behaves. The setting, which we describe more precisely in the full theorem (Theorem 3), is an n node graph with max degree d and what we define as a rank-skewed degree sequence (Definition 1). To approximate a seeded PageRank vector with parameter α and accuracy ε, it suffices to use no more nonzeros than

$$\tilde{O}(d^{1/p}(\tfrac{\alpha^2}{\varepsilon})^{1/(1-\alpha)}), \tag{1}$$

where d is the maximum degree, p reflects a parameter of degree sequence that is empirically around 0.5 to 1, and the $\tilde{O}$ notation hides logarithmic factors. Importantly,
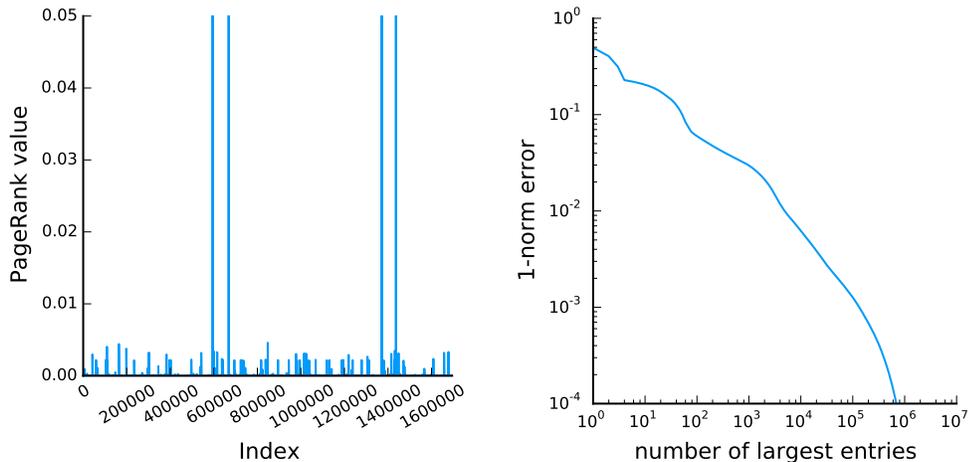
Figure 1 – At left, we display a seeded PageRank vector with $\alpha = 0.5$ from the as-skitter graph based on *index order* (the arbitrary numerical labels of each node in the graph). The peaks indicate that there are only four extremely large entries. At right, we show the same vector in terms of the *accuracy* that results from using the largest k values in an approximation. This plot shows that we can get a 10^{-2}, globally accurate, approximation of the vector with only about 10,000 entries out of 1.7 million. Our goal in this paper is to understand this type of localization in PageRank vectors.

the only quantity in this bound that may depends on n is the max degree, d. This means that the bound scales sublinearly with n as long as the $d^{1/p}$ is sublinear. Our proof is constructive and yields an algorithm to identify this sublinear-sized set. Thus, these results show that we can *locally compute seeded PageRank vectors for a realistic class of graphs*.

Our code and data are available for others to reproduce our empirical findings:

https://github.com/nassarhuda/pprlocal

1.1 Differences from the WAW2015 Conference Paper

The results in this paper are closely related to our prior WAW2015 conference paper [Nassar et al., 2015], where we showed the first localization result for PageRank vectors beyond the constant-degree case [Bar-Yossef and Mashiach, 2008]. The differences in this paper are relatively minor in that we improve the localization bounds in small, meaningful ways. Furthermore, we have stronger characterizations of the space where localization is provably impossible. Finally, we have more extensive empirical investigations that illustrate our results.

1.2 Summary of new and improved results

First, to demonstrate that the localization we study does not simply occur in all graphs, we present a characterization of localization in PageRank on complete-bipartite graphs (Section 4). Seeded PageRank vectors are combinations of uniform vectors (one for each side of the bipartition) on these graphs. This analysis has two interesting regimes: dense graphs and sparse graphs. We show that in both of these cases attaining non-trivial accuracy in the 1-norm requires dense PageRank solutions, whereas they localize in the 2-norm. We conclude that the 1-norm is the most appropriate norm to use in studying localization in PageRank, in the sense that using other norms to measure error can mistakenly report a sparse approximation of a near-uniform vector as being accurate.

Second, we sharpen our result from Nassar et al. [2015] in a way that results in a minor improvement to the bound (Theorem 3). The key to this result is a particular type of skewed degree sequence (Definition 1), and we present a new analysis of a number of real-world networks to determine where our theory makes non-trivial predictions (Section 3). In addition, in this paper we are able to further improve our previous bound under the assumption that a graph is undirected (Theorem 5).

Third, we expand our empirical evaluations of our methodologies and our theoretical predictions on graphs as follows. We increase the size of the synthetic datasets from our previous studies to graphs of size up to 10^9 nodes (Section 6.1) and use an additional graph generation mechanism to show that our theoretical bounds on localization are robust across different graph sizes and graph generators (Section 6.2). Our comparison with empirical evaluations on real-world networks shows that our theoretical bound is likely to significantly underestimate localization compared with real data. Our comparisons of localization behavior across different graphs with similar degree sequences show that different graphs with the same degree sequence will have nearly the same localization behavior, unless their clustering coefficients are markedly different (Section 6.3).

2 Related work

The most closely related work is our previous conference paper where we established a set of similar, but weaker, results [Nassar et al., 2015], as well as a previous study of the matrix exponential [Gleich and Kloster, 2015]. These all used the same type of skewed-degree sequence to show localization.

For a substantially weaker notation of approximation, seeded PageRank was also known to be localized as shown in Andersen et al. [2006]. More specifically, they prove that seeded PageRank $\mathbf{x}$ on any graph can be weakly approximated with a *constant* number of non-zero entries, and where the approximation $\hat{\mathbf{x}}$ satisfies $\varepsilon \cdot \text{degree}(j) \geq x_j - \hat{x}_j > 0$ for all entries j. A slightly relaxed version of this accuracy requirement that is more relevant to the themes of this paper can be stated as follows: $\hat{\mathbf{x}}$ satisfies a degree-normalized infinity norm, $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_\infty < \varepsilon$, where $\boldsymbol{D}$ is the diagonal matrix of node degrees. We explore

the impact of degree-weighting on our more global notions of approximations in Section 4. For the problem of computing local estimates of seeded PageRank vectors, Lofgren et al. [2014, 2015] show a number of scenarios where it is possible to determine if the i, j entry of $(1 - \alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}$ is above a value such as $1/n$.

Another related idea is to locally compute a *piece* of the global or uniform PageRank vector, which corresponds to setting the reset distribution to the uniform distribution over nodes, for the graph with bounded accuracy. Along these lines, Bar-Yossef and Mashiach [2008] proved that entries of the PageRank vector in graphs with bounded max in-degree can be computed by a local algorithm that need not explore all vertices of the graph. Without this assumption, the results usually show that $\Theta(n)$ work is required in many plausible computation scenarios for graph access, see Bressan et al. [2014] for a discussion. For instance, Borgs et al. [2014] show that the problem of computing PageRank on all nodes with value above a certain threshold requires $\Theta(n)$ work in the general case. Computing only the top-ranked nodes is no easier as shown by Bressan and Pretto [2011], where they establish that any deterministic algorithm for computing a constant number of the top *ranked* nodes for general graphs must explore $\Omega(n)$ nodes of the graph [Bressan and Pretto, 2011]. On the other hand, there are positive results on certain classes of randomized algorithms. For example, [Bressan et al., 2014] present an algorithm for computing, with probability $(1 - \varepsilon)$, a $(1 \pm \varepsilon)$-approximation of any single PageRank value in *any* graph using only $O(n^{\frac{2}{3}} \log(n)^{\frac{1}{3}})$ graph access queries.

We note that the concepts of localization and local algorithms are related to the idea of *sketching* algorithms, which are, in broad terms, small-size approximations of large-size objects such that the approximation preserves properties of interest of the larger object. Frequently, sketching algorithms take the form of using linear projections on an object with super-linear size; for example, the connectivity properties of a graph with $O(n^2)$ edges can be approximated by first computing a sparsified graph with $O(n)$ edges using a sketching procedure, and then studying the connectivity of the sketched graph [Ahn et al., 2012]. If a vector is localized, an algorithm could, in principle, compute a sparse approximation in work that scales with the localization instead of the graph size. Sketching, in this scenario, might either reflect sketching the graph data itself to make localized solutions easy to determine, or sketching a single solution vector. Under our degree-distribution assumption with $p \geq 0.5$, our results imply that the there exists a sketch of a seeded PageRank solution vector that has $O(d^2)$ entries (Theorem 3), where d is the max degree. Moreover, our results imply there is a sketch of the PageRank solution matrix $(1 - \alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}$ with nd^2 non-zero entries, compared with the trivial n^2 bound.

More generally, localization in PageRank is intimately tied to the decay of entries of functions of matrices. Here, we are concerned with the elements of the matrix $(1 - \alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}$, which corresponds to the function $f(x) = (1 - \alpha)/(1 - \alpha x)$ applied to $\boldsymbol{P}$. Study of the decay of entries of a function of a banded matrix has a long history, with early research focusing on the decay of off-diagonal entries of the inverse of

banded matrices [Demko, 1977, Demko et al., 1984]. More recent efforts generalize these decay and localization results to other functions of matrices and generalizations of the banded property to more graph theoretic scenarios involving shortest path distances and degrees [Benzi and Golub, 1999, Benzi and Razouk, 2007, Benzi et al., 2013]. Advances in the analysis of localization of functions of sparse and banded matrices hence has implications for localization behavior of PageRank.

The related work all points toward a potentially useful aspect of our present focus: namely that deterministic computation of uniform PageRank entries in a local manner is impossible in general. While we do not address the case of uniform PageRank, for the case of seeded PageRank, by making a plausible assumption on the degree distribution, we are able to beat the worst-case $\Theta(n)$ bound. It is possible that similar results would help alleviate the worst-case scenario for uniform PageRank as well.

3 Our class of skewed degree sequences

We wish to make a few remarks about the class of skewed degree sequences where our results apply. Our definition of a skewed degree sequence is:

> **Definition 1 (Rank-skewed degree sequence)**
> A graph G has a (d, δ, p)–rank-skewed degree sequence if the maximum degree is d, minimum degree is δ, and the kth largest degree, $d(k)$, satisfies $d(k) \leq \max\{dk^{-p}, \delta\}$.

Perhaps the most well-known example of a skewed degree sequence in real-world networks is the power-law degree distribution where the probability that a node has degree k is proportional to $k^{-\gamma}$. These power-laws can be related to our rank-skewed sequences with $p = 1/(\gamma - 1)$ and $d = O(n^p)$ [Avrachenkov et al., 2012].

Our theoretical bounds in Theorem 3 show that there are asymptotically $\tilde{O}(d^{1/p})$ non-zeros in a seeded PageRank vector, and so this power-law setting renders our bound trivial with $\Theta(n)$ nonzeros. Nevertheless, there is evidence that some real-world networks exhibit our type of skewed degrees [Faloutsos et al., 1999] where the bound is asymptotically non-trivial. Below we explore a dozen real-world datasets to see how well our rank-skewed degree sequence setting models graphs in practice and to understand whether $d^{1/p}$ scales sublinearly in those settings. We find that there are multiple real-world networks where our theory yields non-trivial predictions and find that $0.5 \leq p \leq 1$ is a reasonable regime.

Skewed degree sequences in real-world networks. To study how practical our skewed degree sequence is, we look at the degree sequences of a variety of real-world datasets. The datasets we study are from the SNAP repository [Leskovec and Sosič, 2014]. We use collaboration networks from Arxiv Astro Physics, Arxiv Condensed Matter, Arxiv High Energy Physics [Leskovec et al., 2005], and DBLP [Yang and Leskovec, 2015];

an email communication network from Enron [Leskovec et al., 2009, Klimt and Yang, 2004]; a social network YouTube [Mislove et al., 2007]; an instance of the peer-to-peer network Gnutella [Ripeanu et al., 2002, Leskovec et al., 2005]; autonomous systems graphs Skitter and Caida [Leskovec et al., 2005]; a co-purchasing network formed from Amazon data [Leskovec et al., 2007]; and web graphs from Berkley-Stanford and Google web domains [Leskovec et al., 2009]. The results of the degree distribution and the best-fit parameters are in Figure 2 and Table 1.

We observe that the skewed degree sequence property and parameter settings we consider in our theoretical section do describe some of these real-world datasets accurately (for example, the bottom row of graphs in Figure 2). However, we also see that many of the real-world graphs are not perfectly modeled by our degree sequence assumption, and it would require a more nuanced model to accurately describe these degree sequences. In addition to showing how well our skewed degree sequence describes these real-world networks, we also want to see the extent to which our theoretical bound predicts sublinear localization for the degree sequences of these real-world graphs. With this in mind, Table 1 displays the quantity $\log_n(C_p)$ for each network, where C_p comes from Theorem 3. If this quantity is less than 1, then our theoretical bound gives an asymptotically sublinear prediction for localization. From Figure 2 and Table 1 we can see that the datasets `as-caida`, `web-BerkStan`, `web-Google`, and `youtube` all exhibit degree sequences near our skewed degree sequence, and also have a sublinear asymptotic factor C_p.

Getting the line of best fit. We use RANSAC to perform the line fitting task on the data points [Fischler and Bolles, 1981]. In order to get a better fit, we sample 500 data points from the original set that are equally-spaced on a log-log scale. Then we use RANSAC on this subset of the nodes. We do this sampling because the degree sequences we are studying have many more nodes of low degrees than nodes of high degrees, forcing any line fitting strategy to heavily bias the line toward the low degree nodes. By sampling points that are equally spaced on a log-log scale, we instead bias toward a line that fits better on the log scale. We note that similar procedures have been shown to mis-estimate the parameters of a standard power-law degree sequence compared with a maximum likelihood estimator [Clauset et al., 2009]. This aspect may merit additional investigation in the future.

4   Negative results for strong localization

The main results of this paper show that, under our rank-skewed degree distribution assumptions on a network, seeded PageRank vectors can be computed in a local manner with provable accuracy. We now show that there are classes of graphs where seeded PageRank vectors *cannot* be computed in a local manner. This shows that the localization property we study is non-trivial as it is not present in *all* networks. Furthermore, we
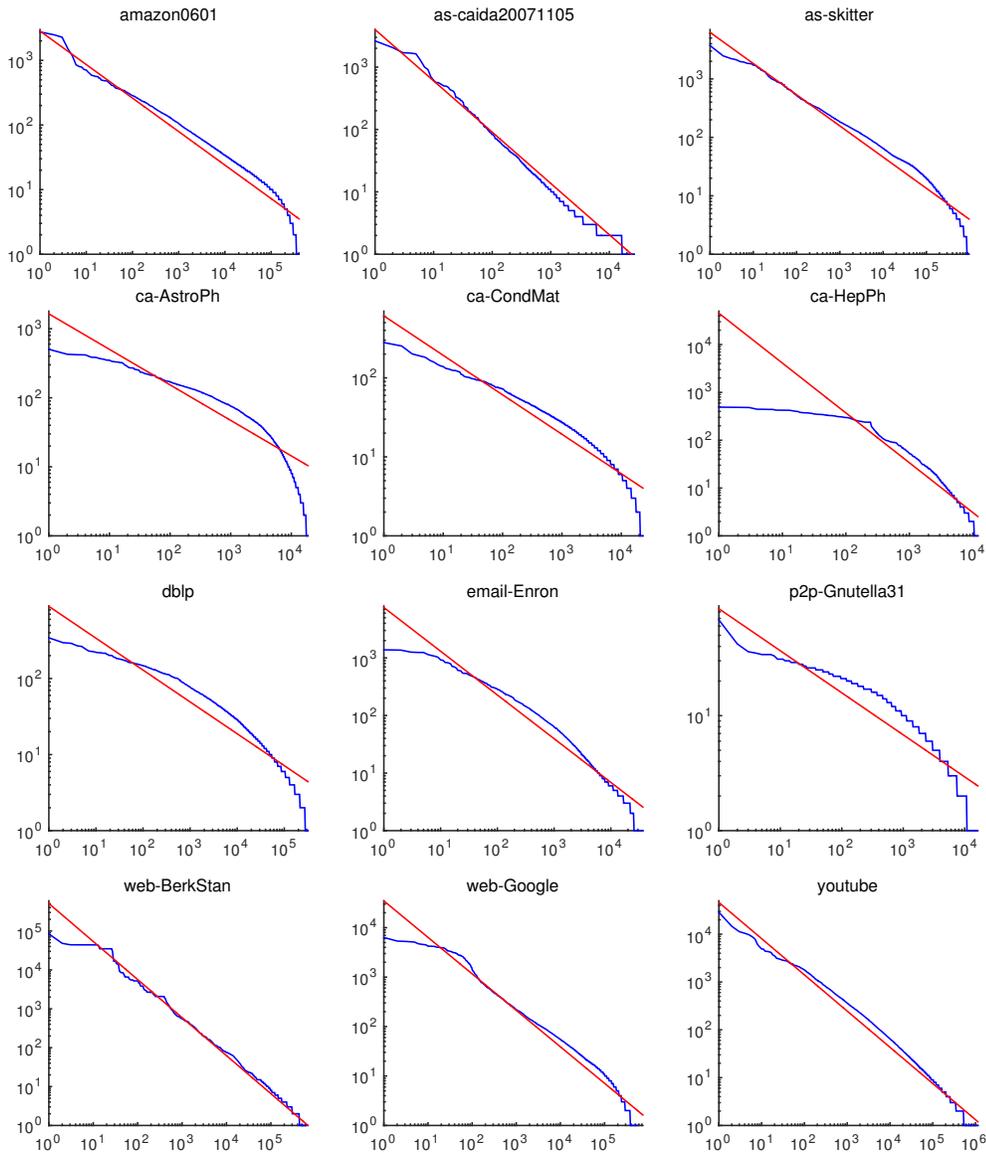
Figure 2 – Log-log plots of the degree sequences from real-world networks in terms of degree-rank (horizontal) to degree (vertical). For each dataset, we also plot a best-fit line (red) to show how the actual degree sequence (blue) compares to the rank-skewed degree sequence property that we assume in our theory. Though several datasets have degree sequences that are too lop-sided to match our skewed property (e.g. the collaboration networks in the middle two rows), a number of other graphs are well-modeled by our skewed degree sequence (the top and bottom rows). We present the max degree, the slope of each best-fit line, and the size of our theoretical bound for each best-fit line in Table 1.

Table 1 – This summary of real-world networks and their degree sequences in the context of our skewed degree sequence model and our theoretical localization bound shows that a number of real-world networks fit the regime of our model where our theory implies sublinear localization. The column p gives the (negative) slope of the best-fit lines for each dataset (pictured in Figure 2). The 3rd column displays the exponent e such that $d = n^e$. The far right column shows how the precise asymptotic factor of our localization bound from Theorem 3, C_p, compares to n. When $\log_n(C_p)$ is less than 1, this indicates our theoretical localization bound is sublinear in n and hence predicts a PageRank vector that is localized. For example, the network `web-BertStan` has $\log_n(C_p) < 1$ and is modeled well by our skewed degree sequence (see Figure 2) and so our theory predicts localization in PageRank in this case.

data	p	$\log_n(d)$	$\log_n(C_p)$	Good fit & localized
`ca-HepPh`	1.04	0.66	0.63	No
`web-BerkStan`	0.98	0.84	0.86	Yes
`as-caida20071105`	0.82	0.77	0.94	Yes
`youtube`	0.76	0.74	0.97	Yes
`email-Enron`	0.76	0.69	0.91	No
`web-Google`	0.74	0.65	0.88	Yes
`as-skitter`	0.53	0.60	1.12	No
`amazon0601`	0.52	0.61	1.18	No
`ca-AstroPh`	0.51	0.63	1.23	No
`ca-CondMat`	0.50	0.56	1.12	No
`dblp`	0.47	0.98	2.09	No
`p2p-Gnutella31`	0.37	0.43	1.19	No

show that there are natural error measures that will report *good* approximations of a near-constant vector with only a sublinear number of entries. These error measures are therefore not appropriate for detecting localization in the solution vectors.

In the rest of this section, we discuss when seeded PageRank vectors *cannot* be locally approximated. Formally, a vector $\mathbf{x}$ of length n can be locally approximated if we can find a vector $\hat{\mathbf{x}}$ with $o(n)$ non-zero entries that is an ε-accurate approximation $\mathbf{x}$. To find a class of graphs with this property, it suffices to consider complete-bipartite graphs ($K_{k,n-k}$) because their structure and symmetry enable explicit computation of seeded PageRank vectors. Once we construct these PageRank vectors, we then study the sparsity of the sparsest ε-accurate approximations when approximation error is measured using the 1-norm, 2-norm, and degree-normalized versions of the 1- and 2-norms.

Summary of negative results. Our results on localization behavior on complete bipartite graphs all rely on our explicit construction of a seeded PageRank vector. The details of that construction, and subsequent proofs of our localization bounds, are tedious,

Table 2 – Can seeded PageRank be locally approximated on complete-bipartite graphs $K_{k,n-k}$? The answer depends on how k scales with n and how we evaluate norms. When $k = O(1)$ then the graph is sparse whereas when $k = \Theta(n)$, then the graph is dense. The table summarizes our finding that when error is measured using the 2-norm, all seeded PageRank vectors are can be locally approximated – even though we prove such vectors are essentially uniform distributions with no concentration in the values (Theorem 6). This shows that only 1-norms are appropriately sensitive to our type of localization.

Graph	Error measure	Local approx.
sparse $K_{k,n-k}, k = O(1)$	$\|\mathbf{x} - \hat{\mathbf{x}}\|_1$	No
sparse $K_{k,n-k}, k = O(1)$	$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1$	No
sparse $K_{k,n-k}, k = O(1)$	$\|\mathbf{x} - \hat{\mathbf{x}}\|_2$	Yes
sparse $K_{k,n-k}, k = O(1)$	$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_2$	Yes
dense $K_{k,n-k}, k = \Theta(n)$	$\|\mathbf{x} - \hat{\mathbf{x}}\|_1$	No
dense $K_{k,n-k}, k = \Theta(n)$	$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1$	Yes
dense $K_{k,n-k}, k = \Theta(n)$	$\|\mathbf{x} - \hat{\mathbf{x}}\|_2$	Yes
dense $K_{k,n-k}, k = \Theta(n)$	$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_2$	Yes

and outside the main interests of this paper, so we include them in Appendix A, but we briefly overview the implications here and direct the interested reader to specific sections in the appendix with parantheticals.

For a complete-bipartite graph on n nodes with partitions of arbitrary sizes (denoted k and $n-k$), our analysis (Section A.1) shows that all seeded PageRank vectors consist of the seed node plus two scaled vectors of all 1s: nodes in the same partition as the seed all have the same PageRank value, and nodes outside the seed's partition all have the same PageRank value (see Theorem 6 for a precise statement). In short, complete-bipartite graphs have seeded PageRank vectors that are essentially two uniform distributions pasted together.

We use this construction to show that, for any complete-bipartite graph, all nodes have non-local seeded PageRank vectors when error is measured in the 1-norm (Section A.2.1). More concretely, we demonstrate that, in this setting, seeded PageRank vectors require $\Theta(n)$ nonzeros to attain a 1-norm accuracy of ε, where n is the number of nodes in the graph. We present our formal result here, and provide details in the appendix.

Proposition 2

Let $\boldsymbol{P}$ be the random-walk transition matrix of an n-node complete-bipartite graph, and let j be the index of any node in the graph. Fix a value $\alpha \in (0, 1)$, and a desired accuracy $\varepsilon < \alpha^2/(1+\alpha)$. Then the number of nonzeros required to approximate the seeded PageRank vector $(1 - \alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\mathbf{e}_j$ with a 1-norm accuracy of ε is bounded below by $(1 - \varepsilon(1 + \alpha)/\alpha^2)n$.

This result on complete-bipartite graphs generalizes our analysis of star graphs in our previous work [Nassar et al., 2015].

In Section A.2.1 we also consider localization of seeded PageRank vectors when error is measured in the degree-normalized 1-norm, i.e. $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1$. We demonstrate that in $K_{k,n-k}$ graphs that have partitions of the same size, seeded PageRank can be approximated with a sublinear number of nonzeros. Such instances of $K_{k,n-k}$ are dense, having $O(n^2)$ edges. In contrast, we then show that in *sparse* $K_{k,n-k}$ graphs, i.e. complete-bipartite graphs in which one partition is of constant size, seeded PageRank vectors are de-localized even when error is measured as $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1$.

In Section A.2.2 we move on to consider localization in these same seeded PageRank vectors when error is measured as $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$ and as $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_2$. In both cases we show that seeded PageRank is always localized for arbitrary complete-bipartite graphs.

We end with a toy example on a star graph to explore differences in the localization behavior of an approximate PageRank vectors and its related residual vector (Section A.2.3). A number of fast algorithms for producing sparse approximations of PageRank rely on the residual of the PageRank linear system to determine when the algorithm has converged to an accurate PageRank approximation. We show that, depending on the norm that is used, it is possible to have a sparse residual with norm less than ε even when the true solution is de-localized.

Conclusion from negative results. We conclude from this study of localization in different norms that the 1-norm is the most effective norm for analyzing localization behavior. This is because the 2-norm and degree normalized variations all identify nearly uniform distributions as being localized. We therefore recommend that future studies of localization in seeded PageRank and algorithmic termination criteria for approximating seeded PageRank use the 1-norm to measure error. Alternatively, future work could explore the utility of measuring relative error, i.e. analyzing $\|\mathbf{x} - \hat{\mathbf{x}}\|_2/\|\mathbf{x}\|_2$, for example.

5 Localization in Personalized PageRank

In this section we show for graphs with a rank-skewed degree sequence, *all* seeded PageRank vectors are localized. This result has broad utility because it implies that *every* seeded PageRank vector can be computed locally for every graph in this class.

The bounds we present in this section build on top of our recent work with seeded heat kernel vectors on graphs with a skewed degree sequence [Gleich and Kloster, 2015]. We first adapted that work to obtain localization bounds on seeded PageRank vectors in our subsequent work [Nassar et al., 2015]. Here we use similar methodologies as in our previous works, but introduce minor improvements for a sharper overall bound on localization. Compared to our PageRank localization result in [Nassar et al., 2015] we give a tighter upperbound. We then present a second result that further improves our

bound by assuming the graph is undirected. Though we use techniques from our prior work, we provide self-contained proofs in this paper for the convenience of the reader.

Theorem 3

Let $\boldsymbol{P}$ be a uniform random walk transition matrix of a graph with n nodes and with a (d, δ, p)–rank-skewed degree sequence. The Gauss-Southwell coordinate relaxation method applied to the seeded PageRank problem $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{e}_s$ from the initial solution $\mathbf{x}^{(0)} = (1 - \alpha)(\boldsymbol{I} + \alpha\boldsymbol{P})\mathbf{e}_s$ produces an approximation $\mathbf{x}_\varepsilon$ satisfying $\|\mathbf{x} - \mathbf{x}_\varepsilon\|_1 < \varepsilon$ having at most N non-zeros in the solution, where N satisfies

$$N = \min\left\{ n \ , \ 1 + d + \tfrac{1}{\delta}C_p \left(\tfrac{\alpha^2}{\varepsilon}\right)^{\frac{\delta}{1-\alpha}} \right\}, \tag{2}$$

and where we define C_p to be

$$C_p := d(1 + \log d) \qquad\qquad \text{if } p = 1$$
$$:= d\left(1 + \tfrac{1}{1-p}\left(d^{\frac{1}{p}-1} - 1\right)\right) \qquad \text{otherwise.}$$

Note that the upper bound $N = n$ is trivial because a vector cannot have more non-zeros than entries. Thus, d, δ, p, and n must satisfy certain conditions to ensure that inequality (2) is not trivial. In particular, for values of $p < 1$, it is necessary that $d = o(n^p)$ for inequality (2) to imply that $N = o(n)$. For $p > 1$, the bound guarantees sublinear growth of N as long as $d = o(n)$. Additionally, the minimum degree δ must be bounded by $O(\log\log n)$. Thus we arrive at:

Corollary 4

Let G be a class of graphs with degree sequences obeying the conditions of Theorem 3 with constant δ and $d = o(n^{\min(p,1)})$. Then seeded PageRank vectors are localized.

We also note that the theorem implies localized seeded PageRank vectors for any graph with a maximum degree $d = O(\log\log n)$.

A sharper bound for undirected graphs. If we restrict our attention to undirected graphs, we can slightly sharpen our analysis. In our first bound (2) the minimum degree δ in the exponent of (α^2/ε) can have a huge impact on the size of the overall bound. Our tighter analysis here changes this factor in the exponent to $\delta - 1$. This change affects the size of only the constant in the bound, $(\alpha^2/\varepsilon)^{(\delta-1)/(1-\alpha)}$, and so it might not seem like much of an improvement. However, for small values of δ, this can make a large difference. For example, when $\delta = 2$ the improvement makes the large constant in our new theoretical upperbound the square root of the large constant in our old upperbound:

$(\alpha^2/\varepsilon)^{2/(1-\alpha)}$ compared to $(\alpha^2/\varepsilon)^{1/(1-\alpha)}$. We state the result here, and present the proof following our proof of Theorem 3.

Theorem 5

Let the setting be the same as in Theorem 3, except we assume the graph is undirected and the minimum degree $\delta \geq 2$. The Gauss-Southwell coordinate relaxation method applied to the seeded PageRank problem $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{e}_s$ from the initial solution $\mathbf{x}^{(0)} = (1 - \alpha)(\boldsymbol{I} + \alpha\boldsymbol{P})\mathbf{e}_s$ produces an approximation $\mathbf{x}_\varepsilon$ satisfying $\|\mathbf{x} - \mathbf{x}_\varepsilon\|_1 < \varepsilon$ having at most N non-zeros in the solution, where N satisfies

$$N = \min \left\{ n , \ 1 + d + \tfrac{1}{\delta-1} C_p \left(\tfrac{\alpha^2}{\varepsilon} \right)^{\frac{\delta-1}{1-\alpha}} \right\}, \tag{3}$$

where C_p is the same as in Theorem 3.

Note that both upperbounds have a factor α^2/ε such that if $\alpha^2/\varepsilon < 1$ then the constant attached to C_p becomes very small and the bound approaches $d + 1$. This makes sense, because if $\alpha^2 < \varepsilon$, then the initial solution is already ε-accurate, making $d + 1$ a tight bound on the number of non-zeros in the solution.

Having stated our main theoretical results on the localization of PageRank, in the remainder of this section we will discuss our skewed degree sequence and present a proof of our results.

5.1 Deriving the bound

Getting back to the proof, our goal is an ε-approximation, $\mathbf{x}_\varepsilon$, to the equation $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{e}_s$ for a seed s. Given an approximation, $\hat{\mathbf{x}}$, we can express the error in terms of the residual vector $\mathbf{r} = (1 - \alpha)\mathbf{e}_s - (\boldsymbol{I} - \alpha\boldsymbol{P})\hat{\mathbf{x}}$ as follows:

$$\mathbf{x} - \hat{\mathbf{x}} \quad = \quad (\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\mathbf{r}. \tag{4}$$

Using this relationship, we can bound our approximation's 1-norm accuracy, $\|\mathbf{x} - \hat{\mathbf{x}}\|_1$, with the quantity $\frac{1}{1-\alpha}\|\mathbf{r}\|_1$. This is because the column-stochasticity of $\boldsymbol{P}$ implies that $\|(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\|_1 = \frac{1}{1-\alpha}$. Guaranteeing a 1-norm error $\|\mathbf{x} - \hat{\mathbf{x}}\|_1 < \varepsilon$ is then a matter of ensuring that $\|\mathbf{r}\|_1 < (1 - \alpha)\varepsilon$ holds. To bound the residual norm, we look more closely at a particular method for producing the approximation.

The Gauss-Southwell iteration. The Gauss-Southwell algorithm is a coordinate relaxation method for solving a linear system akin to the Gauss-Seidel linear solver. When solving a linear system, the Gauss-Southwell method proceeds by updating the entry of the approximate solution that corresponds to the largest magnitude entry of the residual, $\mathbf{r}$. Hence, one iteration consists of altering only one entry of the solution, plus

accessing one node's outlinks. This sparse update makes Gauss-Southwell convenient for seeded PageRank-style problems, and it has appeared numerous times in recent literature [Berkhin, 2007, Bonchi et al., 2012, Jeh and Widom, 2003, McSherry, 2005]. Here we describe the Gauss-Southwell update as it is used to solve the seeded PageRank linear system.

The algorithm begins by setting an initial solution and computing the associated residual. The standard choice for the initial solution is $\mathbf{x}^{(0)} = 0$, but we are able to sharpen our main theorem from [Nassar et al., 2015] by choosing the initial solution $\mathbf{x}^{(0)} = (1 - \alpha)(\boldsymbol{I} + \alpha \boldsymbol{P})\mathbf{e}_s$. Then the corresponding residual is $\mathbf{r}^{(0)} = (1 - \alpha)\mathbf{e}_s - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(0)} = (1 - \alpha)\alpha^2 \boldsymbol{P}^2 \mathbf{e}_s$.

In step k, let $j = j(k)$ be the entry of $\mathbf{r}^{(k)}$ with the largest magnitude, and let $m_k = |\mathbf{r}_j^{(k)}|$. Gauss-Southwell updates the solution $\mathbf{x}^{(k)}$ and residual as follows:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + m_k \mathbf{e}_j \tag{5}$$

$$\mathbf{r}^{(k+1)} = \mathbf{e}_s - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k+1)}, \tag{6}$$

and the residual update can be expanded to $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - m_k \mathbf{e}_j + m_k \alpha \boldsymbol{P} \mathbf{e}_j$. Since each update to the solution $\mathbf{x}^{(k)}$ alters exactly one entry of the vector, the index k is an upper-bound on the number of non-zeros that we add to the initial solution vector $\mathbf{x}^{(0)}$. (Note that the initial solution $\mathbf{x}^{(0)}$ has $d + 1$ nonzero entries.)

To continue, we show that the residual and solution vector stay nonnegative throughout this process. This fact has been proved in previous applications of Gauss-Southwell (see [Bonchi et al., 2012], Section 5.2, among others) but we include a proof for the convenience of the reader. First, note that the initial residual, $\mathbf{r}^{(0)} = (1 - \alpha)(\boldsymbol{I} + \alpha \boldsymbol{P} + \alpha^2 \boldsymbol{P}^2)\mathbf{e}_j$, is nonnegative because it is a sum of nonnegative vectors (since $\boldsymbol{P}$ is nonnegative). Now, by way of induction assume that $\mathbf{r}^{(k)}$ is nonnegative. Then the largest magnitude entry is positive, $m_k > 0$. Hence, in the update of the residual $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - m_k \mathbf{e}_j + m_k \alpha \boldsymbol{P} \mathbf{e}_j$, subtracting $m_k \mathbf{e}_j$ is equivalent to setting the j th entry of the previous residual $\mathbf{r}^{(k)}$ to zero (which is nonnegative by assumption), and then adding the scaled nonnegative vector $\boldsymbol{P} \mathbf{e}_j$ keeps the new vector $\mathbf{r}^{(k+1)}$ nonnegative. By induction, this proves that the residual is always nonnegative. Finally, the solution must also always be nonnegative because all solution updates consists of adding m_k, a positive quantity, to an entry of the solution. Since the initial solution $\mathbf{x}^{(0)} = (1 - \alpha)(\boldsymbol{I} + \alpha \boldsymbol{P})\mathbf{e}_j$ is nonnegative, this completes a proof of the claim.

Since we have shown that the residual is always nonnegative, we know the 1-norm of the residual can be expressed as $\|\mathbf{r}^{(k+1)}\|_1 = \mathbf{e}^T \mathbf{r}^{(k+1)}$, where $\mathbf{e}$ is the vector of all ones. Expanding the residual in terms of the iterative update presented above, we can write the residual norm as $\mathbf{e}^T \left(\mathbf{r}^{(k)} - m_k \mathbf{e}_j + m_k \alpha \boldsymbol{P} \mathbf{e}_j\right)$. Then, denoting $\|\mathbf{r}^{(k)}\|_1$ by r_k, we have the recurrence $r_{k+1} = r_k - m_k(1 - \alpha)$.

Next, observe that, since m_k is the largest magnitude entry in $\mathbf{r}^{(k)}$, m_k is larger than the average value of $\mathbf{r}^{(k)}$. Let $Z(k)$ denote the number of nonzero entries in $\mathbf{r}^{(k)}$; then the

average value can be expressed as $r_k/Z(k)$. Hence, we have $m_k \geq r_k/Z(k)$, and so we can bound $r_k - m_k(1-\alpha)$ above by $r_k - r_k(1-\alpha)/Z(k)$. Thus, $r_{k+1} \leq r_k\left(1 - (1-\alpha)/Z(k)\right)$, and we can recurse to find:

$$r_{k+1} \leq r_0 \prod_{t=0}^{k}\left(1 - \tfrac{1-\alpha}{Z(t)}\right), \tag{7}$$

where $r_0 = (1-\alpha)\alpha^2$ because $\mathbf{r}^{(0)} = (1-\alpha)\alpha^2 \boldsymbol{P}^2\mathbf{e}_s$. Then, using the fact that $\log(1-x) \leq -x$ for $x < 1$, we note:

$$r_k \leq (1-\alpha)\alpha^2 \prod_{t=0}^{k-1}\left(1 - \tfrac{1-\alpha}{Z(t)}\right) \leq (1-\alpha)\alpha^2 \exp\left(-(1-\alpha)\sum_{t=0}^{k-1}\tfrac{1}{Z(t)}\right). \tag{8}$$

To proceed from here we need some control over the quantity $Z(t)$, and this is where our assumptions on the degree sequence enter the proof. The proof steps below build toward a proof of both Theorem 3 and Theorem 5, so that they apply to graphs that are not necessarily undirected, as in Theorem 5. Then we explain how to modify the proof steps to obtain the sharper bound of Theorem 5 that applies specifically to undirected graphs.

5.2 Using the degree sequence

We show that, for a graph with a particular kind of skewed degree sequence, the number of entries in the residual obeys the following:

$$Z(t) \leq C_p + \delta t, \tag{9}$$

where the term C_p is defined in the statement of Theorem 3. We follow our analysis in [Nassar et al., 2015] that improved on our previous approach in [Gleich and Kloster, 2015]. The bound in (9) is proved below, but first we use this upperbound on $Z(t)$ to control the bound on r_k.

For the next step of our proof, we prove the following inequality

$$-\sum_{t=0}^{k-1}\tfrac{1}{Z(t)} \leq -\tfrac{1}{\delta}\log\left((\delta k + C_p)/C_p\right)$$

via using a left-hand rule integral approximation. To do this, first we use (9) to bound $1/Z(t) \geq 1/(C_p + \delta t)$. Then we sum and use the integral approximation:

$$\sum_{t=0}^{k-1}\tfrac{1}{Z(t)} \geq \sum_{t=0}^{k-1}\tfrac{1}{C_p+\delta t} \tag{10}$$

$$\geq \int_0^k \tfrac{1}{C_p+\delta x}\mathrm{d}x, \tag{11}$$

15

which equals $\log\left((C_p + \delta k)/C_p\right)/\delta$, as claimed. Continuing with the residual bound, we plug the bound on $\sum_{t=0}^{k-1} 1/Z(t)$ into (8) as follows. We obtain

$$r_k \leq (1-\alpha)\alpha^2 \exp\left(-\frac{(1-\alpha)}{\delta}\log\left(\frac{\delta k + C_p}{C_p}\right)\right),$$

which simplifies to $r_k \leq (1-\alpha)\alpha^2\left((\delta k + C_p)/C_p\right)^{(\alpha-1)/(\delta-1)}$. Finally, to guarantee $r_k < \varepsilon(1-\alpha)$, it suffices to choose k so that $((\delta k + C_p)/C_p)^{(\alpha-1)/\delta} \leq \varepsilon/\alpha^2$. This holds if and only if $(\delta k + C_p) \geq C_p\left(\alpha^2/\varepsilon\right)^{\delta/(\alpha-1)}$ holds. This inequality is guaranteed by $k \geq \frac{1}{\delta}C_p\left(\alpha^2/\varepsilon\right)^{\delta/(1-\alpha)} - \frac{1}{\delta}C_p$. Thus, Gauss-Southwell will produce an ε-approximation as long as the number of steps used is at least $k = \frac{1}{\delta}C_p\left(\alpha^2/\varepsilon\right)^{\delta/(1-\alpha)} - \frac{1}{\delta}C_p$. Note that if $\alpha^2/\varepsilon > 1$ then this number is negative – this simply means that $k = 0$ steps of Gauss-Southwell suffice to reach the target accuracy. (This occurs when the initial solution $\mathbf{x}^{(0)}$ already satisfies the desired accuracy.)

Each step of Gauss-Southwell introduces at most one non-zero in the solution vector, and we know that the initial solution vector $\mathbf{x}^{(0)} = (1-\alpha)(\boldsymbol{I} + \alpha\boldsymbol{P})\mathbf{e}_s$ has no more than $1 + d$ non-zeros. This implies that if $1 + d + k < n$, then there is an approximation $\mathbf{x}_\varepsilon$ with $N = 1 + d + k < n$ non-zeros. If $1 + d + k \geq n$, then this analysis produces the trivial bound $N = n$.

Proving the degree sequence bound. Finally, we justify the inequality in (9) used in the proof of Theorem 3. Recall that $Z(t)$ is the number of nonzeros in the residual after t steps. This means that $Z(0)$ is the sparsity of the initial residual, $(1-\alpha)\alpha^2\boldsymbol{P}^2\mathbf{e}_j$. To upperbound $Z(t)$, we will consider the number of nonzeros in this initial residual, as well as the maximum number of new nonzeros that could be introduced in the residual during the t steps of Gauss-Southwell. Essentially, we upperbound the number of nonzeros in the residual using a sum of the largest degrees in the graph.

To upperbound the number of nonzeros in the vector $\boldsymbol{P}^2\mathbf{e}_j$, first note that the vector $\boldsymbol{P}\mathbf{e}_j$ can have no more than d nonzero entries (recall that d is the maximum degree in the graph). Thus, $\boldsymbol{P}^2\mathbf{e}_j = \boldsymbol{P}(\boldsymbol{P}\mathbf{e}_j)$, which is a weighted sum of no more than d columns of $\boldsymbol{P}$, can have no more nonzeros than $\sum_{m=1}^{d} d(m)$, i.e. the sum of the d largest degrees in the graph. In every iteration of Gauss-Southwell, no more than d_{new} new nonzeros can be introduced in the residual, where d_{new} is the largest degree of any node that has not yet been operated on. Thus, we can upperbound the sparsity of the residual in step t as follows: $Z(t) \leq \sum_{m=1}^{d+t} d(m)$. When we substitute the decay bound $d(m) \leq dm^{-p}$ into this expression, $d(m)$ is only a positive integer when $m \leq (d/\delta)^{1/p}$. Hence, we split the summation $Z(t) \leq \sum_{m=1}^{t} d(m)$ into two pieces,

$$Z(t) \quad \leq \quad \sum_{m=1}^{t} d(m) \quad \leq \quad \left(\sum_{m=1}^{\lfloor(d/\delta)^{1/p}\rfloor} dm^{-p}\right) + \left(\sum_{m=\lfloor(d/\delta)^{1/p}\rfloor+1}^{t} \delta\right).$$

16

We want to prove that this implies $Z(t) \leq C_p + \delta t$. The second summand is straight-forward to majorize by δt. To bound the first summand, we use a right-hand integral approximation:

$$\sum_{m=1}^{\lfloor (d/\delta)^{1/p} \rfloor} dm^{-p} \leq d \left(1 + \int_1^{(d/\delta)^{1/p}} x^{-p} \mathrm{d}x \right).$$

This integral is straightforward to bound above with the quantity C_p defined in Theorem 3. This completes the proof.

5.3 The undirected case

Here we describe how to modify the above proof of Theorem 3 so that it implies a slightly stronger result, restricted to undirected graphs. (The stronger result is Theorem 5).

Recall that the inequality (9), $Z(t) \leq C_p + \delta t$, gives an upperbound on the number of nonzeros in the residual at step t. We obtain the upperbound by summing together the t largest degrees in the graph. The term C_p contains the sum of the degrees that are larger than δ – this consists of the $\lceil (d/\delta)^{1/p} \rceil$ largest-degree nodes. All nodes that are introduced in the residual after these must then have degree δ.

We alter this proof as follows. Although it is true that δ is the *degree* of any node introduced in the residual after the $\lceil (d/\delta)^{1/p} \rceil$ nodes of largest degree, these nodes of degree δ do not actually introduce δ new *nonzero entries* in the residual. This is because of the additional assumption that the graph is undirected: any node that is nonzero in the residual "received" its residual mass because one of that node's neighbors was operated on in a previous iteration of Gauss-Southwell. Thus, when we operate on a particular nonzero node v in the residual and make the neighbors of v become nonzero in the residual, at least one of the neighbors of node v was already nonzero. Hence, the number of nonzeros in the residual does not increase by $d(v)$, but by at most $d(v) - 1$. Hence, in (9) we can replace δ with $(\delta - 1)$. It is straight-forward to follow the rest of the steps of the proof of Theorem 3 and replace δ with $(\delta - 1)$ to obtain the sharper inequality in Theorem 5.

Minimum degree of 1. We want to remark on the problem that this modification appears to cause in the case that $\delta = 1$. The issue arises when we try to divide by $\delta - 1$, which would equal 0 if we have $\delta = 1$. Because we introduce the notation δ as the "minimum degree" in the graph, this issue would seem to be problematic, since many graphs have 1 as their minimum degree. However, in reality the above proofs work for any constant δ that upperbounds the smallest degree in the graph. Thus, the simplest way around this apparent difficulty is just to set $\delta = 2$ even when the minimum degree is 1. We mention this simple workaround for the case that $d_{\mathrm{minimum}} = 1$ to demonstrate that this common situation does not break our theory. However, we remark that this workaround makes the resulting theoretical bound more loose.

6 Experiments

We address three goals in our experimental evaluations and summarize our findings.

1. *Do graphs have localized solutions when our theory predicts they should?* We find that the results from Theorem 3 are conservative, but do match features of empirical localization on random graphs with our rank-skewed degree sequences. The experiments are discussed in Section 6.1.

2. *Are these findings consistent across different choices of graph generators and the random samples?* In short, yes, the findings are robust to looking at multiple samples from the same graph generator and robust to the choice of random graph generator. Further discussion is in Section 6.2.

3. *How does localization in synthetic graphs compare to real-world graphs?* Our synthetic networks exhibit worse localization than real-world samples, and this can be attributed to the lack of local clustering in the types of synthetic graphs we study. See Section 6.3 for the details.

Overall, the experiments we present in this section show that, although our theoretical localization bound seems to describe PageRank localization behavior more accurately on asymptotically large graphs, there remains a gap between the existing theoretical explanations for PageRank localization and the behavior we see in practice. Nevertheless, the bound we present in this paper brings the literature closer to an explanation of the graph properties responsible for the localization so often observed in seeded PageRank in practice.

Procedure used for generating graphs. Here we describe the routines used to generate our synthetic graphs with specified degree sequences in this paper. As a general pipeline, we construct a rank-skewed degree sequence with minimum degree $\delta = 2$, maximum degree $d = \lceil \sqrt{n} \rceil$. Given a parameter p for a decay exponent, we produce the degree sequence $d(k) = \max\{\lfloor dk^{-p} \rfloor, \ \delta\}$ for $k = 1, \cdots, n$. We then use the Erdős-Gallai conditions and the Havel-Hakimi algorithm to check if the generated degree sequence is graphical. Once we obtain a degree sequence that is graphical, we proceed to the graph generation step. The failure of the graphical sequence check is often due to the degree sequence having degree values that sum to an odd number. In all the experiments we have performed, perturbing one of the nodes with degree δ and increasing it by 1 was sufficient. We use the Bayati-Kim-Saberi procedure [Bayati et al., 2010] as implemented in the bisquik software package [Gleich, 2010] to generate graphs that have our skewed degree sequence. We also use the Chung-Lu model [Chung and Lu, 2002] to produce graphs with the correct degree sequence in expectation.

Computing localization. We use the following procedure to compute an ε-accurate seeded PageRank vector to get the minimal number of non-zeros. First, we compute a highly accurate estimate of the true PageRank vector $\mathbf{x}$. Then we sort the values from largest to smallest and continue adding entries in sorted order to our approximation until the difference between the approximation and the true vector is smaller than ε. (To compute the "true" vector we perform enough iterations of a standard power method for computing PageRank to guarantee 10^{-12} accuracy.) This always produces the smallest possible of non-zero entries that gives an ε-accurate approximation.

Changes from conference version. We note that, although this analysis follows a similar experimental setup as in our previous study [Nassar et al., 2015], the results presented here are from entirely new sets of experiments, expand on the parameter regimes of our previous work, and include an additional graph generation mechanism to study robustness of the results across different graph generators.

6.1 Our localization theory in synthetic graphs

In this section we empirically evaluate our theoretical bound on PageRank localization by plotting our bound alongside actual PageRank vectors computed on large synthetic graphs. For each parameter setting $p = \{0.5, 0.75, 0.95\}$ we generate a single graph with $n = 10^9$ nodes and the other parameters as described above. In all experiments we make sure that the largest connected component of the graph is of size at least $0.95 \cdot n$. Then we compute seeded PageRank on each such graph, seeded on the node of maximum degree, for each parameter setting $\alpha = \{0.25, 0.5, 0.85\}$. Figure 3 compares our theoretical bound to the actual PageRank localization curve on each different graph. The experiments show that, although the theoretical bound is far from tight, it gives a meaningful prediction of localization for smaller values of α on the graphs other than the densest graph, i.e. the graph generated with $p = 0.5$.

We encounter empirical localization in Figure 3 under parameter setting $\alpha = 0.85$: PageRank stays localized for $p = 0.75, 0.95$ where we would not expect localization given our theory. We believe this occurs for two reasons. First, our theory is obviously too coarse to precisely describe localization. Second, this behavior is a consequence of the way our graphs are constructed. For the parameters in question, the graphs have many long chains of nodes of degree 2 linked to higher degree nodes such that the PageRank diffusion takes many steps to reach a vast portion of the graph. Thus, the diffusion stays localized in these graphs beyond what we would expect.

6.2 Consistency of localization across samples and graph generators

We would like to know whether the seeded PageRank localization behavior changes significantly on different graph outputs, even though the graphs have the same, or roughly
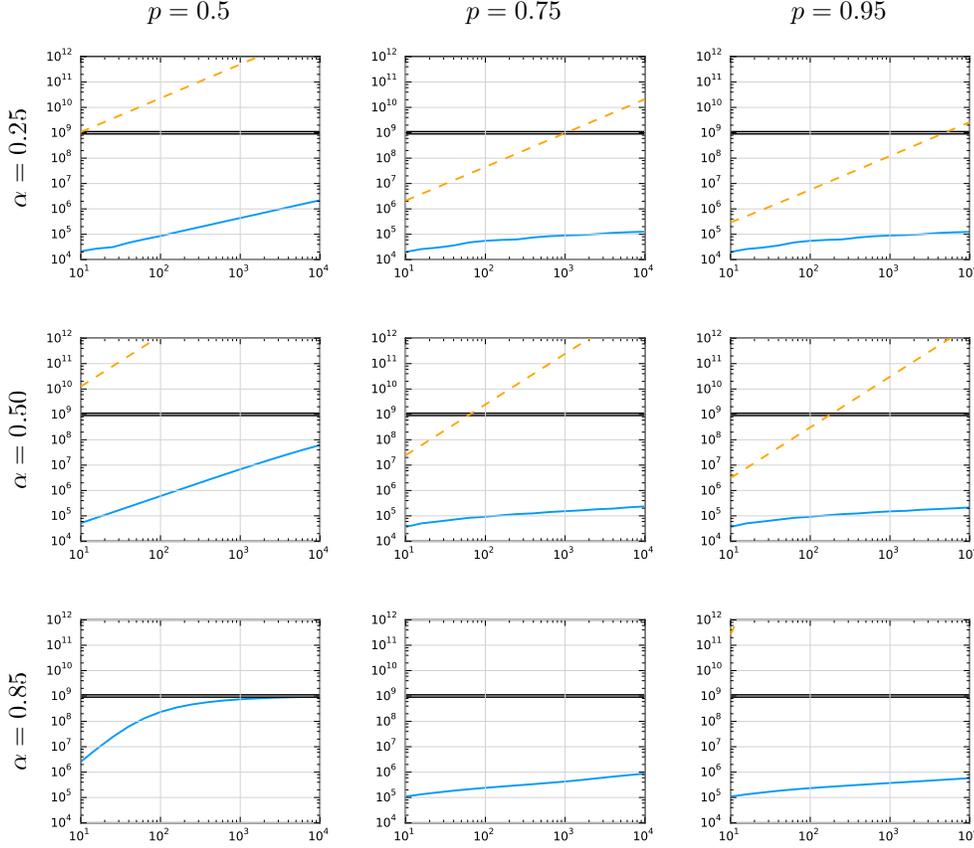
Figure 3 – In each sub-plot the x-axis represents $1/\epsilon$ and the y-axis represents the number of nonzeros present in the PageRank approximation. The blue curve represents the number of nonzeros in the sparsest PageRank approximation with 1-norm accuracy ϵ. All plots in this figure display results computed on graphs generated with bisquik [Gleich, 2010] with $n = 10^9$, max degree $d = \lceil n^{1/2} \rceil$, and min degree $\delta = 2$. The dashed orange line shows the theoretical prediction from Theorem 5. The thick black line indicates when a solution vector is de-localized, i.e. has n nonzeros. We observe far more empirical localization in these graphs than we would predict based on our theory, illustrating that our theory is too coarse. But the we see an increase in localization as p increases, as expected via our theory. In this figure, the size of the graph in all these plots is 10^9, but the only experiment that delocalizes is $p = 0.5$ with $\alpha = 0.85$, which uses parameters where our theory does not predict a sublinear bound on the number of nonzeros for any n.
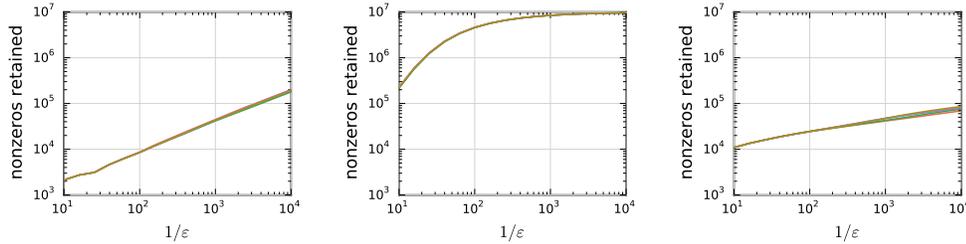
Figure 4 – These plots display the localization behavior of seeded PageRank vectors computed on multiple different graphs generated by bisquik using the same parameter settings (in particular, the same rank-skewed degree sequence). All graphs used for these plots were generated with $n = 10^7$ nodes, maximum degree $d = \lceil n^{1/2} \rceil$, and minimum degree $\delta = 2$. For each plot, five graphs were generated, and for each graph we computed PageRank seeded on the node of maximum degree. (*At left.*) PageRank localization curves computed with $\alpha = 0.25$ on five different graphs generated with $p = 0.5$. (*Middle.*) PageRank localization curves computed with $\alpha = 0.85$ on the five different graphs with $p = 0.5$ used in the left plot. (*At right.*) PageRank localization curves computed with $\alpha = 0.85$ on five different graphs generated with $p = 0.95$. All three plots show that PageRank behaves nearly identically on different graphs with the same degree sequence.

the same, degree sequence. To explore this question, we use bisquik [Gleich, 2010] to generate multiple graphs with identical parameter settings and observe PageRank localization on each such instance. We also compare bisquik graphs with an exact degree sequence to Chung-Lu random graphs where the degree sequence occurs in expectation. Figure 4 shows plots of localization curves for five random samples of a bisquik graph with the same degree sequence. Figure 5 compares bisquik to Chung-Lu. There is only minimal variation in the behavior of the PageRank localization across the different graphs and between Chung-Lu and bisquik. These results show that the empirical behavior can be inferred from just a single sample and there do not appear to be extreme sensitivities or variations between samples or graph generation schemes.

6.3 Localization in real-world and synthetic graphs

In the next set of experiments, we continue to explore how much PageRank localization behavior can change across different graphs that have similar degree sequence. For these experiments we use three real-world datasets (YouTube, Email-Enron, and Amazon, from Section 3) and, for each of the datasets, we use the Chung-Lu model to generate a synthetic graph with a degree sequence that is nearly identical to that of the real-world network. The output of the Chung-Lu procedure is not guaranteed to have exactly the degree sequence specified. With this in mind, in Figures 6a, 6b, 6c (top row) we display the degree sequence of each of the real-world networks alongside the degree sequence of the Chung-Lu graph we produced to mimic the original degree sequences. We then
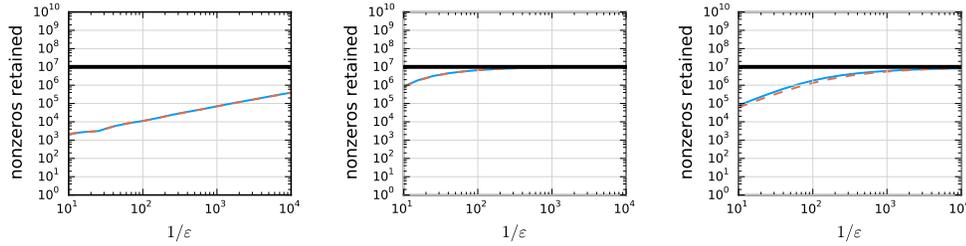
Figure 5 – Comparison of PageRank localization on graphs produced by the bisquik procedure (blue) and the Chung-Lu procedure (red-dashed). For this experiment we generated a Chung-Lu graph on $n = 10^7$ nodes, set to have our rank-skewed degree sequence in expectation. We then used bisquik to produce a graph with degree sequence identical to that of the Chung-Lu graph. We did this for rank-skewed degree sequences with $p = 0.5$ (the left and center plots) and $p = 0.95$ (the right plot). These plots show localization curves for PageRank seeded on the maximum degree node of the synthetic graphs. The left plot shows PageRank with $\alpha = 0.25$ while the center and right plots show PageRank with $\alpha = 0.85$. In all cases the localization behavior of PageRank is nearly identical on graphs generated under the Chung-Lu model and the bisquik procedure.

compute PageRank seeded on the maximum degree node for each graph, with parameter values $\alpha = \{0.25, 0.5, 0.85\}$. The results are shown in the remaining three rows of those figures.

These experiments show two graphs, YouTube and Email-Enron, where the localization in the real-world and synthetic graph behave in closely related ways. However, for the Amazon graph, the real-world network exhibits starkly more localization. While a precise explanation of the difference is beyond the scope of this paper, we believe this is due to the lack of clustering structure in synthetic networks such as Chung-Lu [Kolda et al., 2014] (where each edge is generated independently), and the presence of local clustering structure in real-world data (where there are far more locally dense structures such as triangles than would be expected). Local clustering structure can be quantified by looking at the clustering coefficient of the network, which is the number of triangles (3-cliques) divided by the number of wedges (paths of length 2). Local clustering structure can only *improve* localization because it concentrates the PageRank mass in a smaller region of the graph. In fact, large clustering coefficients and highly-skewed degree sequences can be formally shown to produce vertex neighborhoods with small conductance in a graph [Gleich and Seshadhri, 2012, Yin et al., 2017].

Table 3 shows that the graphs with localization behavior closely related to that of their synthetic counterparts (i.e. YouTube and Email-Enron) have clustering coefficients that are also not too different from that of their counterparts. YouTube's global clustering coefficient is nearly the same as its Chung-Lu partner's, Enron's coefficient is a factor of 3 larger than its counterpart's, but Amazon's clustering coefficient is almost 10,000 times

larger than its Chung-Lu clone's coefficient. The high level of clustering in the Amazon graph is responsible for the high level of localization we observe in its seeded PageRank vectors (apparent in Figure 6c); in contrast, the Amazon Chung-Lu clone has almost no clustering, and thus we witness its seeded PageRank vector delocalize rapidly.

Table 3 – Each real-world dataset has a synthetic counterpart created via Chung-Lu. Despite having nearly identical degree sequences, the synthetic counterparts can have clustering behavior that differs, sometimes markedly, from the real-world graph. This table shows that the synthetic graphs have lower clustering coefficients than their real-world counterparts, with a particularly sharp difference for the Amazon graph. This difference in clustering behavior likely explains the difference in localization behavior observed in Amazon (Figure 6c), compared to that of YouTube (Figure 6a) and Email-Enron (Figure 6b).

data	clustering coefficient	clustering coefficient of Chung-Lu sample
youtube	0.019	0.015
email-Enron	0.26	0.085
amazon0601	0.50	0.00054

7 Discussion and Future Work

We have studied localization behavior in approximate PageRank vectors and its relationship both to the degree sequence of the underlying graph and the norm used to measure the error of the approximation. In particular we proved that, when error is measured in the 1-norm, seeded PageRank is always de-localized for complete-bipartite graphs, whereas for graphs with a rank-skewed degree sequence it is always localized. Our experiments show that, although this skewed degree sequence does accurately model some real-world networks, there are also many real-world datasets with degree sequences that do not decay sufficiently rapidly for our theory to apply. Thus, there remains an intermediate regime of degree sequences which neither our de-localization theory nor our localization theory describes. To extend our theory into this regime, we plan to extend our negative results on complete-bipartite graphs to slight edge-perturbations of the highly structured complete-bipartite graphs. This would make progress toward closing the gap between complete-bipartite graphs and graphs with our skewed degree sequence.

Additionally, we are investigating extending our localization theory to apply to degree sequences that follow a more relaxed property. Rather than require a strictly log-linear skew, we believe that a piece-wise log-linear degree sequence, or possibly even more general, will suffice to prove localization results nearly identical to our current theory. Such a result could advance the literature to explain localization in a majority of real-world graphs. Our experiments also make it clear that the local clustering structure plays an important role in the behavior on real-world graphs.

(a) YouTube   (b) Email-Enron   (c) Amazon
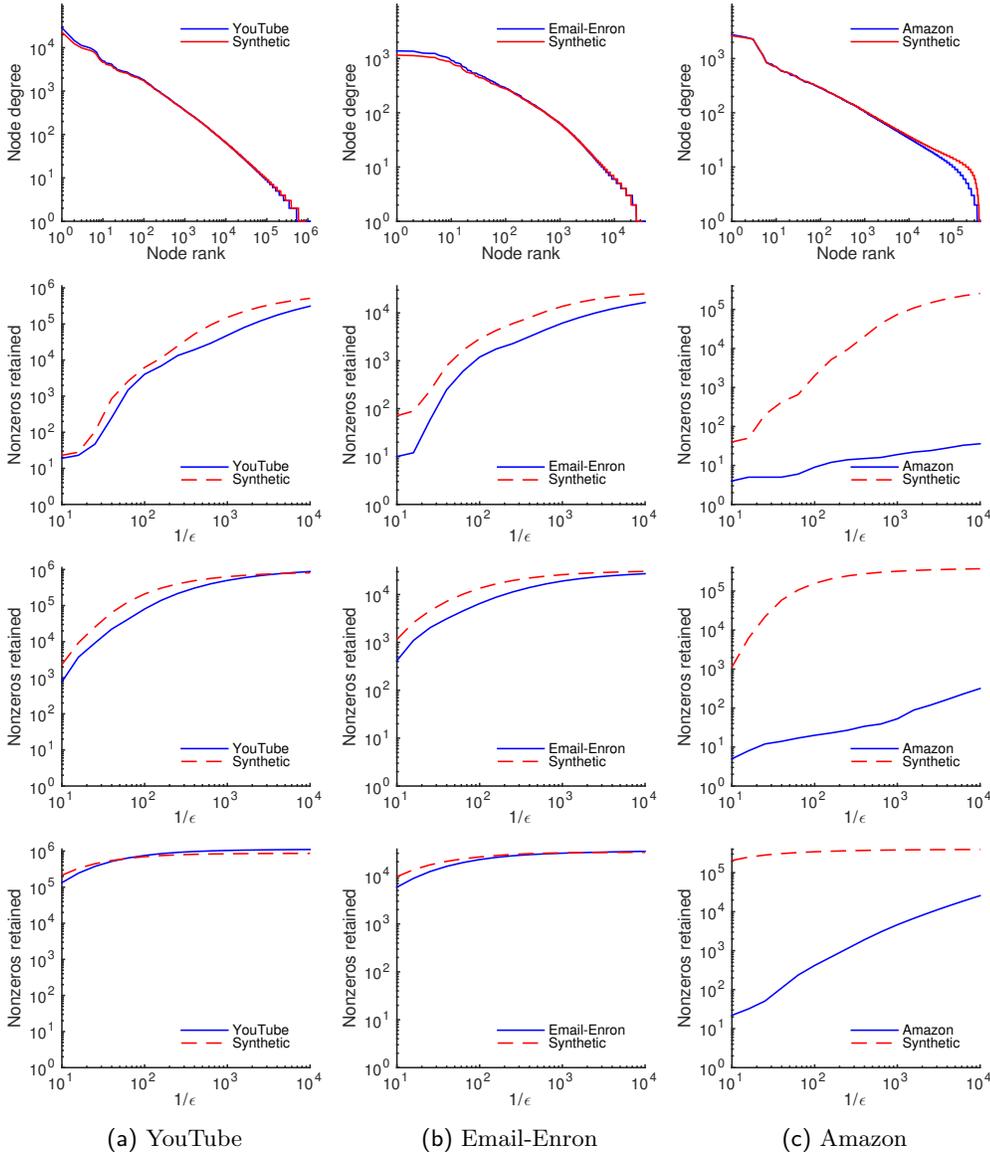
Figure 6 – Comparison of PageRank localization in YouTube (6a), Enron-Email (6b), and Amazon0601 (6c) datasets and synthetic graphs generated to have similar degree sequences via Chung-Lu. (*Top Row*) The degree sequences of the three networks and their synthetic counterparts are nearly identical. The other plots show localization in seeded PageRank vectors, seeded on the nodes of maximum degree in the three networks and their synthetic counterparts, for different values of α. We use the values $\alpha = 0.25$ (*Second Row*), $\alpha = 0.5$ (*Third Row*), and $\alpha = 0.85$ (*Last Row*). Columns 1 and 2 (YouTube and Enron) show similar localization between the synthetic and real-world graphs. The third column (Amazon) provides an example of two graphs that have nearly identical degree sequences yet starkly different localization. Namely, the real-world graph has substantially more localization than the synthetic graph. We study clustering coefficients in the text to explain this difference.

Acknowledgments

References

Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: Sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '12, pages 5–14, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1248-6. doi: 10.1145/2213556.2213560. URL http://doi.acm.org/10.1145/2213556.2213560. Cited on page 5.

Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using PageRank vectors. In *FOCS2006*, 2006. Cited on pages 4 and 38.

Konstantin Avrachenkov, Nelly Litvak, Marina Sokol, and Don Towsley. Quick detection of nodes with large degrees. In Anthony Bonato and Jeannette Janssen, editors, *Algorithms and Models for the Web Graph*, volume 7323 of *Lecture Notes in Computer Science*, pages 54–65. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-30541-2_5. Cited on page 6.

Ziv Bar-Yossef and Li-Tal Mashiach. Local approximation of pagerank and reverse pagerank. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 279–288. ACM, 2008. Cited on pages 3 and 5.

Mohsen Bayati, JeongHan Kim, and Amin Saberi. A sequential algorithm for generating random graphs. *Algorithmica*, 58(4):860–910, 2010. ISSN 0178-4617. Cited on page 18.

M Benzi and N Razouk. Decay bounds and O(n) algorithms for approximating functions of sparse matrices. *ETNA*, 28:16–39, 2007. Cited on page 6.

Michele Benzi and Gene H Golub. Bounds for the entries of matrix functions with applications to preconditioning. *BIT Numerical Mathematics*, 39(3):417–438, 1999. Cited on page 6.

Michele Benzi, Paola Boito, and Nader Razouk. Decay properties of spectral projectors with applications to electronic structure. *SIAM Review*, 55(1):3–64, 2013. Cited on page 6.

Pavel Berkhin. Bookmark-coloring algorithm for personalized PageRank computing. *Internet Mathematics*, 3(1):41–62, 2007. Cited on page 14.

Francesco Bonchi, Pooya Esfandiar, David F. Gleich, Chen Greif, and Laks V.S. Lakshmanan. Fast matrix computations for pairwise and columnwise commute times and Katz scores. *Internet Mathematics*, 8(1-2):73–112, 2012. Cited on page 14.

Christian Borgs, Michael Brautbar, Jennifer Chayes, and Shang-Hua Teng. Multiscale matrix sampling and sublinear-time pagerank computation. *Internet Mathematics*, 10 (1-2):20–48, 2014. Cited on page 5.

Marco Bressan and Luca Pretto. Local computation of pagerank: the ranking side. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 631–640. ACM, 2011. Cited on page 5.

Marco Bressan, Enoch Peserico, and Luca Pretto. Approximating pagerank locally with sublinear query complexity. *arXiv preprint arXiv:1404.1864*, 2014. Cited on page 5.

Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002. Cited on page 18.

Fan R. L. Chung. *Spectral Graph Theory*. American Mathematical Society, 1992. Cited on page 29.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009. doi: 10.1137/070710111. URL http://link.aip.org/link/?SIR/51/661/1. Cited on page 7.

Stephen Demko. Inverses of band matrices and local convergence of spline projections. *SIAM Journal on Numerical Analysis*, 14(4):616–619, 1977. Cited on page 6.

Stephen Demko, William F Moss, and Philip W Smith. Decay rates for inverses of band matrices. *Mathematics of computation*, 43(168):491–499, 1984. Cited on page 6.

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, 1999. Cited on page 6.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL http://doi.acm.org/10.1145/358669.358692. Cited on page 7.

David F. Gleich. bisquik: bisquik randomly samples a graph with a prescribed degree distribution. https://github.com/dgleich/bisquik, November 2010. Cited on pages 18, 20, and 21.

David F. Gleich. PageRank beyond the web. *SIAM Review*, 57(3):321–363, August 2015. doi: 10.1137/140976649. Cited on page 1.

David F. Gleich and Kyle Kloster. Sublinear column-wise actions of the matrix exponential on social networks. *Internet Mathematics*, 11(4-5):352–384, 2015. Cited on pages 4, 11, and 15.

David F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *KDD2012*, pages 597–605, August 2012. doi: 10.1145/2339530.2339628. Cited on page 22.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008. Cited on page 29.

G. Jeh and J. Widom. Scaling personalized web search. In *WWW*, pages 271–279, 2003. Cited on page 14.

Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *CEAS*, 2004. Cited on page 7.

Tamara G. Kolda, Ali Pinar, Todd Plantenga, and C. Seshadhri. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing*, 36(5): C424–C452, 2014. doi: 10.1137/130914218. Cited on page 22.

Jure Leskovec and Rok Sosič. SNAP: A general purpose network analysis and graph mining library in C++. http://snap.stanford.edu/snap, June 2014. Cited on pages 2 and 6.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 177–187, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081893. URL http://doi.acm.org/10.1145/1081870.1081893. Cited on pages 2, 6, and 7.

Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007. Cited on page 7.

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. Cited on page 7.

Peter Lofgren, Siddhartha Banerjee, and Ashish Goel. Bidirectional pagerank estimation: From average-case to worst-case. In *Algorithms and Models for the Web Graph*, pages 164–176. Springer, 2015. Cited on page 5.

Peter A Lofgren, Siddhartha Banerjee, Ashish Goel, and C Seshadhri. Fast-ppr: Scaling personalized pagerank estimation for large graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1445. ACM, 2014. Cited on page 5.

Frank McSherry. A uniform approach to accelerated PageRank computation. In *WWW*, pages 575–582, 2005. ISBN 1-59593-046-9. Cited on page 14.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007. Cited on page 7.

Huda Nassar, Kyle Kloster, and David F Gleich. Strong localization in personalized pagerank vectors. In *Algorithms and Models for the Web Graph*, pages 190–202. Springer International Publishing, 2015. Cited on pages 3, 4, 11, 14, 15, and 19.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford University, 1999. Cited on page 1.

Matei Ripeanu, Ian Foster, and Adriana Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *arXiv preprint cs/0209028*, 2002. Cited on page 7.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015. Cited on page 6.

Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. Local higher-order graph clustering. Accepted at KDD2017, 2017. Cited on page 22.

A   Appendix

A.1   Constructing seeded PageRank on complete-bipartite graphs

To prove our de-localization results for seeded PageRank vectors on complete-bipartite graphs (Section 4) we use an explicit construction of these vectors. We present that construction in this section. Before going through the details of the construction, we first briefly outline the steps.

First we use the known structure of the spectrum of a complete-bipartite graph to construct a quadratic polynomial $q(x)$ that interpolates the scaled resolvent function $R(x) = (1-\alpha) \cdot (1-\alpha x)^{-1}$ on the spectrum of any complete-bipartite graph. Then we study

the block structure of the random-walk transition matrix $\boldsymbol{P}$ for an arbitrary complete-bipartite graph. Finally, we use this work to explicitly compute $q(\boldsymbol{P})\mathbf{e}_j = R(\boldsymbol{P})\mathbf{e}_j$ for any seed node j.

The construction We begin by noting that the random-walk transition matrix $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{D}^{-1}$ of an arbitrary complete-bipartite graph is necessarily diagonalizable. This is because it is a similarity transformation of a symmetric matrix: $\boldsymbol{D}^{-1/2}\boldsymbol{P}\boldsymbol{D}^{1/2} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{A}$ is symmetric because complete-bipartite graphs are undirected. Consequently we can use the following simple result from the theory of functions of matrices ([Higham, 2008], Chapter 1.2): for any function $f(x)$ defined on the spectrum of $\boldsymbol{P}$, we can express $f(\boldsymbol{P}) = q(\boldsymbol{P})$ for any polynomial $q(x)$ that interpolates the function $f(x)$ on the spectrum of $\boldsymbol{P}$. If $\boldsymbol{P}$ has r distinct eigenvalues (i.e. $\boldsymbol{P}$ has a minimal polynomial of degree r), then this implies that $f(\boldsymbol{P})$ can be expressed using an interpolating polynomial on r values, and so $q(x)$ can be a polynomial of degree $r-1$.

Using the above facts, any matrix function $f(\boldsymbol{P})$ (such that $f(x)$ is defined on the spectrum of $\boldsymbol{P}$) can be expressed in the form

$$f(\boldsymbol{P}) = c_0\boldsymbol{I} + c_1\boldsymbol{P} + \cdots + c_{r-1}\boldsymbol{P}^{r-1}$$

for appropriate coefficients c_j. Next we show how to combine this fact with structural properties of complete-bipartite graphs to produce our de-localization results. Note that we this addresses a broader set of functions than just the resolvent function, but later on we will study specifically the case of the resolvent function to obtain a more explicit bound on the de-localization of PageRank.

Specializing to complete-bipartite graphs The random-walk transition matrix $\boldsymbol{P}$ of any complete-bipartite graph has eigenvalues $-1, 0$, and 1 ([Chung, 1992], Chapter 1.2). As noted above, $\boldsymbol{P}$ is diagonalizable, so we can determine $f(\boldsymbol{P})$ by the action of $f(x)$ on the spectrum of $\boldsymbol{P}$. Thus, for any function $f(x)$ that is defined on the values $\{-1, 0, 1\}$, and any polynomial $q(x)$ that satisfies

$$q(-1) = f(-1), \quad q(0) = f(0), \quad p(1) = f(1) \tag{12}$$

the polynomial will also satisfy $q(\boldsymbol{P}) = f(\boldsymbol{P})$. Below we give the structure of the interpolating polynomial for the resolvent function before showing how such polynomials can be used to prove that these functions are not local on complete-bipartite graphs.

The intuition of our results is that, because complete-bipartite graphs have 3 eigenvalues, any function of the graph can be expressed using an interpolating polynomial of degree 2; the matrices $\boldsymbol{P}$ and $\boldsymbol{P}^2$ then determine the structure of any function $f(\boldsymbol{P})$, and because of the uniform nature of complete-bipartite graphs, the expressions $\boldsymbol{P}$ and $\boldsymbol{P}^2$ (and, hence, $f(\boldsymbol{P})$) are roughly the sum of uniform distributions and so cannot be approximated using only a small number of entries that are non-zero.

A.1.1 Interpolating polynomials

Given any function $f(x)$ defined on the values $\{-1, 0, 1\}$, the degree 2 interpolating polynomial $q(x)$ for this function, defined by Equation (12), has coefficients $q(x) = c_0 + c_1 x + c_2 x^2$ given by

$$c_0 = f(0) \tag{13}$$

$$c_1 = \tfrac{1}{2}(f(1) - f(-1)) \tag{14}$$

$$c_2 = \tfrac{1}{2}(f(1) + f(-1) - 2f(0)). \tag{15}$$

The value $c_0 = f(0)$ follows from plugging 0 into $q(x)$ and noting $f(0) = q(0) = c_0 + 0 + 0$. The other values follow from similar straight-forward algebra. Thus, for a general function $f(x)$ defined on the spectrum of a complete-bipartite graph, we can study the localization of

$$f(\boldsymbol{P}) = f(0) \cdot \boldsymbol{I} + \left(\tfrac{1}{2}(f(1) - f(-1))\right) \cdot \boldsymbol{P} + \left(\tfrac{1}{2}(f(1) + f(-1) - 2f(0))\right) \cdot \boldsymbol{P}^2, \tag{16}$$

by looking at the structure of the matrices $\boldsymbol{P}$ and $\boldsymbol{P}^2$. Because we intend to focus on the particular choice of the resolvent function, $f(x) = (1 - \alpha) \cdot (1 - \alpha x)^{-1}$, we compute

$$c_0 = f(0) = (1 - \alpha) \tag{17}$$

$$c_1 = \tfrac{1}{2}(f(1) - f(-1)) = \tfrac{\alpha}{1+\alpha} \tag{18}$$

$$c_2 = \tfrac{1}{2}(f(1) + f(-1) - 2f(0)) = \tfrac{\alpha^2}{1+\alpha} \tag{19}$$

so that we can write the more specific expression

$$f(\boldsymbol{P})\mathbf{e}_j = (1 - \alpha)\mathbf{e}_j + \tfrac{\alpha}{1+\alpha}\boldsymbol{P}\mathbf{e}_j + \tfrac{\alpha^2}{1+\alpha}\boldsymbol{P}^2\mathbf{e}_j$$

for a seeded PageRank vector on a complete-bipartite graph. All that remains is to analyze the block structure of $\boldsymbol{P}$ itself to determine the exact de-localization behavior of PageRank in this setting.

A.1.2 Using graph structure

Here we will consider a complete-bipartite graph with n nodes divided into a partition of k nodes and $n - k$ nodes. In particular we want to understand the structure of the random-walk transition matrix, $\boldsymbol{P}$, as well as $\boldsymbol{P}^2$, for such graphs. The nodes can be ordered so that the adjacency matrix has the block form $\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{E}^T \\ \boldsymbol{E} & 0 \end{bmatrix}$, where $\boldsymbol{E}$ is the $(n - k) \times k$ matrix of all 1s. The degree matrix is then $\boldsymbol{D} = \begin{bmatrix} (n - k) \cdot \boldsymbol{I}_{n-k} & 0 \\ 0 & k \cdot \boldsymbol{I}_k \end{bmatrix}$. Letting $\begin{bmatrix} \boldsymbol{I}_k \\ 0 \end{bmatrix}$ be a $n \times k$ matrix with a $k \times k$ identity block in the upper-most block, we

can express the relationships

$$P = AD^{-1} = \tfrac{1}{k} \begin{bmatrix} I_k \\ 0 \end{bmatrix} \begin{bmatrix} 0 & E^T \end{bmatrix} + \tfrac{1}{n-k} \begin{bmatrix} 0 \\ E \end{bmatrix} \begin{bmatrix} I_k & 0 \end{bmatrix} \tag{20}$$

$$= \tfrac{1}{k} \begin{bmatrix} 0 & E^T \\ 0 & 0 \end{bmatrix} + \tfrac{1}{n-k} \begin{bmatrix} 0 & 0 \\ E & 0 \end{bmatrix} \tag{21}$$

and

$$P^2 = \tfrac{1}{n-k}\tfrac{1}{k} \begin{bmatrix} 0 & 0 \\ 0 & EE^T \end{bmatrix} + \tfrac{1}{n-k}\tfrac{1}{k} \begin{bmatrix} I_k \\ 0 \end{bmatrix} E^T E \begin{bmatrix} I_k & 0 \end{bmatrix} \tag{22}$$

$$= \tfrac{1}{n-k} \begin{bmatrix} 0 & 0 \\ 0 & J_{n-k} \end{bmatrix} + \tfrac{1}{k} \begin{bmatrix} J_k & 0 \\ 0 & 0 \end{bmatrix} \tag{23}$$

where J_i is a square block of 1s of dimension $i \times i$. We use the notation J_i specifically to denote square matrices of all 1s, whereas we use E to denote matrices of all 1s that are possibly not square. Equation (23) uses the relationships $EE^T = k \cdot J_{n-k}$ and $E^T E = (n-k) \cdot J_k$, which follow from the fact that the matrix E is all 1s and dimension $(n-k) \times k$.

Next we use the expressions for P and P^2 in Equations (21) and (23) to show that columns of matrix functions $f(P)e_j$ are de-localized, except for extreme cases which we discuss below. To complete our proof, we will simply substitute the above expressions for P^t in Equations (21) and (23) into Equation (16) and perform a little algebra. First consider any node j in the partition containing k nodes. From the structure of P described above, we get that $Pe_j = \tfrac{1}{n-k} \begin{bmatrix} 0 \\ e \end{bmatrix}_{n-k}$, where $\begin{bmatrix} 0 \\ e \end{bmatrix}_{n-k}$ is a vector with k entries that are 0s, and $(n-k)$ entries that are 1s in the lower block. Similarly we obtain that $P^2 e_j = \tfrac{1}{k} \begin{bmatrix} e \\ 0 \end{bmatrix}_k$, where $\begin{bmatrix} e \\ 0 \end{bmatrix}_k$ is a vector with k entries that are 1s in the upper block, followed by $(n-k)$ zero entries.

Now we use this information to compute the magnitude of the individual entries of the vectors $f(P)e_j$. Substituting into Equation (16) the structure of Pe_j and $P^2 e_j$ that we just computed, we can see that $f(P)e_j$ has k entries of magnitude exactly $\tfrac{1}{k}c_2$, and $n-k-1$ entries of magnitude exactly $\tfrac{1}{n-k}c_1$, and finally entry j itself has magnitude $f(0) + \tfrac{1}{n-k}c_1$.

The purpose of this arithmetic is that it allows us to see that a vector $f(P)e_j$ has too many non-trivial entries, and so cannot be approximated to an accuracy of ε with fewer than $O(n)$ of its non-zero entries, unless the function $f(x)$ is such that at least one of the coefficients $\tfrac{1}{n-k}c_1$ or $\tfrac{1}{k}c_2$ is very small relative to ε (where the coefficients c_t are as defined above in Equation (13)). We make this notion rigorous in the following proposition, which the above discussion has proved:

Theorem 6

Let $\boldsymbol{P}$ be the random-walk transition matrix of a complete-bipartite graph on n nodes, and let the partition sizes be k and $n-k$. Fix a value $\alpha \in (0,1)$ and let $f(x)$ be the resolvent function $f(x) = (1-\alpha) \cdot (1-\alpha x)^{-1}$. Then columns of $f(\boldsymbol{P})$ can be expressed as follows. For any node j in the partition of size k, we have, in the notation of Section A.1.2,

$$f(\boldsymbol{P})\mathbf{e}_j = (1-\alpha) \cdot \mathbf{e}_j + \tfrac{\alpha}{(1+\alpha)}\tfrac{1}{(n-k)} \cdot \begin{bmatrix} 0 \\ \mathbf{e} \end{bmatrix}_{n-k} + \tfrac{\alpha^2}{(1+\alpha)}\tfrac{1}{k} \cdot \begin{bmatrix} \mathbf{e} \\ 0 \end{bmatrix}_k. \qquad (24)$$

This result implies that every column of $R(\boldsymbol{P})$ consists of one large entry at the seed node, $R(\boldsymbol{P})_{jj}$, and one sub-vector for each of the two graph partitions, with each sub-vector having entries of uniform value. Note that the sum of the entries in the partition containing the seed is $\frac{1}{1+\alpha}$, and the sum of the entries in the other partition is then $\frac{\alpha}{1+\alpha}$.

When studying the degree-normalized error it will be useful for us also to have explicit computation of the vector $\boldsymbol{D}^{-1}\mathbf{x} = \boldsymbol{D}^{-1}R(\boldsymbol{P})\mathbf{e}_j$. In complete-bipartite graphs, nodes in the partition of size k each have degree $n-k$, and nodes in the partition of size $n-k$ have degree k. Thus, scaling the PageRank vector in (24) by node degrees yields the following.

Corollary 7

In the notation of Theorem 6, the degree-normalized seeded PageRank vector seeded on node j is

$$\boldsymbol{D}^{-1}R(\boldsymbol{P})\mathbf{e}_j = (1-\alpha)\tfrac{1}{n-k} \cdot \mathbf{e}_j + \tfrac{\alpha}{(1+\alpha)}\tfrac{1}{k(n-k)} \cdot \begin{bmatrix} 0 \\ \mathbf{e} \end{bmatrix}_{n-k} + \tfrac{\alpha^2}{(1+\alpha)}\tfrac{1}{k(n-k)} \cdot \begin{bmatrix} \mathbf{e} \\ 0 \end{bmatrix}_k.$$
$$(25)$$

Having stated these preliminary constructions, we next proceed with proofs of our localization and de-localization results.

A.2   Proofs of de-localization results on complete-bipartite graphs

A.2.1   Localization behavior in the 1-norm

Recall that the overarching goal of the paper is to study the sparsest possible approximation to a seeded PageRank vector, such that the approximation meets a desired accuracy criterion. In this section we use the explicit PageRank construction in Theorem 6 to study the sparsity of approximations that are ε-accurate in the 1-norm and degree-normalized 1-norm, where ε is the desired accuracy.

We construct our sparse approximation as follows. Let $\hat{\mathbf{x}}$ be the vector $(1-\alpha)\mathbf{e}_j$ plus a certain number of the nonzero entries in the true solution $\mathbf{x}$. More specifically, $\hat{\mathbf{x}}$ equals $(1-\alpha)\mathbf{e}_j$ plus z_1 entries of $\mathbf{x}$ corresponding to nodes in the partition of size $n-k$, and

z_2 entries of $\mathbf{x}$ corresponding to nodes in the partition of size k. Note that we design $\hat{\mathbf{x}}$ including the component $(1-\alpha)\mathbf{e}_j$ simply to make the error analysis cleaner; this only affects the sparsity of the approximation by 1 and makes no difference in the asymptotic results. Thus, the approximation we are constructing is within 1 nonzero entry of being the sparsest possible.

Next we want to measure the 1-norm and 2-norm error of the approximation $\hat{\mathbf{x}}$ that results from retaining $z_1 + z_2$ nonzeros as described above. We begin with the 1-norm analysis. We claim that the error satisfies

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_1 = \tfrac{\alpha}{1+\alpha}\left(1 - \tfrac{z_1}{n-k}\right) + \tfrac{\alpha^2}{1+\alpha}\left(1 - \tfrac{z_2}{k}\right). \tag{26}$$

To see this, note that the entries in $\mathbf{x}$ corresponding to nodes in the seed's partition sum to $\alpha^2/(1+\alpha)$, whereas nodes outside the seed's partition sum to $\alpha/(1+\alpha)$. If z_1 entries outside the seed's partition are retained, then $(n-k) - z_1$ entries are omitted from that sub-vector, so the error accrued from that sub-vector alone is $((n-k) - z_1)/(n-k) = (1 - z_1/(n-k))$ times the total error contained in those nodes, which is $\alpha/(1+\alpha)$. Similar arithmetic yields the second summand of Equation (26).

To construct a *sparse* approximation of PageRank we want to retain the *largest* nonzeros in $\mathbf{x}$. This requires that we know which entries have the largest and smallest value. There are two cases: the case in which the nodes in the seed's partition are larger in magnitude, i.e. when $\frac{\alpha^2}{k(1+\alpha)} > \frac{\alpha}{(n-k)(1+\alpha)}$, and the case in which they are not. Simple algebraic manipulation shows that these cases are determined by the following inequality. Nodes in the partition of the seed (the partition of size k) have larger PageRank value if and only if $k < \alpha(n-k)$.

First we consider that case that the seed's partition has larger PageRank values, i.e. $k < \alpha(n-k)$. If this is the case, we want to retain nodes in the seed's partition (because they are larger in magnitude). In the context of Equation (26) this means we are increasing z_2. However, even if we retain all nodes in the seed's partition (by setting $z_2 = k$), the total 1-norm error can still be as large as $\alpha/(1+\alpha)$ if we do not retain any nonzeros in the other partition (this corresponds to setting $z_1 = 0$). If $\varepsilon < \alpha/(1+\alpha)$, then we must increase z_1 in order to obtain an ε-accurate approximation.

We remark that if $\varepsilon < \alpha/(1+\alpha)$, then we *must* retain $z_2 = k$ nonzeros just to attain an accuracy of $\alpha/(1+\alpha)$, and if $k = \Theta(n)$, then our approximation is already de-localized. However, to cover the case that k is small, we continue our analysis.

If we retain all k nodes in the seed's partition, and only z_1 nodes in the other partition, then by Equation (26) the 1-norm error is $\alpha/(1+\alpha)(1 - z_1/(n-k))$. This is bounded above by ε if and only if

$$\left(1 - \varepsilon\tfrac{1+\alpha}{\alpha}\right)(n-k) \le z_1.$$

The total number of nonzeros in the approximation $\hat{\mathbf{x}}$ would then be $z_1 + z_2$, which is

$$z_1 + k \geq \left(1 - \varepsilon \tfrac{1+\alpha}{\alpha}\right)(n - k) + k \tag{27}$$

$$= \left(1 - \varepsilon \tfrac{1+\alpha}{\alpha}\right) n + \left(\varepsilon \tfrac{1+\alpha}{\alpha}\right) k, \tag{28}$$

which completes a proof in the first case (the case $k < \alpha(n - k)$) that $\Theta(n)$ nonzeros are required for an ε-accurate approximation.

In the second case, $k \geq \alpha(n - k)$, nodes outside the seed's partition are greater in magnitude, and so we retain those first. Similar to the previous case, setting $z_1 = (n - k)$ and $z_2 = 0$ in Equation (26) yields an approximation with a 1-norm accuracy of $\frac{\alpha^2}{1+\alpha}$. Again, $(n - k)$ might already be enough nonzeros to be de-localized, but we continue our analysis for the case that $(n - k)$ is small.

If $\frac{\alpha^2}{(1+\alpha)} > \varepsilon$ then additional nonzeros must be retained in our approximation. Hence, we must increase z_2. Note that the 1-norm error of our approximation is $\frac{\alpha^2}{(1+\alpha)}(1 - z_2/k)$. This is bounded above by ε if and only if

$$\left(1 - \varepsilon \tfrac{1+\alpha}{\alpha^2}\right) k \leq z_2.$$

The total number of nonzeros in the approximation $\hat{\mathbf{x}}$ is, again, $z_1 + z_2$, which is

$$n - k + z_2 \geq (n - k) + \left(1 - \varepsilon \tfrac{1+\alpha}{\alpha^2}\right) k \tag{29}$$

$$= n - \left(\varepsilon \tfrac{1+\alpha}{\alpha^2}\right) k. \tag{30}$$

Since $k \leq n$ we have $-k \geq -n$, and so we can write $n - k + z_2 \geq (1 - \varepsilon(1+\alpha)/\alpha^2)n$ which completes the second case, $k \geq \alpha(n-k)$. Finally, both cases guarantee that approximating seeded PageRank with a 1-norm accuracy of ε requires at least $(1 - \varepsilon(1+\alpha)/\alpha^2)n$ nonzero entries, proving the following result.

> **Proposition 2**
>
> Let $\boldsymbol{P}$ be the random-walk transition matrix of an n-node complete-bipartite graph, and let j be the index of any node in the graph. Fix a value $\alpha \in (0, 1)$, and a desired accuracy $\varepsilon < \alpha^2/(1+\alpha)$. Then the number of nonzeros required to approximate the seeded PageRank vector $(1 - \alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\mathbf{e}_j$ with a 1-norm accuracy of ε is bounded below by $(1 - \varepsilon(1 + \alpha)/\alpha^2)n$.

This statement implies that any seeded PageRank vector in any complete-bipartite graph is de-localized when error is measured in the 1-norm. The bound given in the proposition is not tight – rather, we relax the bounds that we gave in our proofs above for the sake of a cleaner, more intuitive statement.

Degree-normalized 1-norm Next we consider localization behavior when we normalize error by node degrees. More specifically, here we study the localization when we measure error using the expression $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1$.

This analysis builds on our previous analysis simply by scaling the nodes' Page-Rank values by the node degrees. Recalling Corollary 7, we will analyze the error in approximating the following scaled PageRank vector:

$$\boldsymbol{D}^{-1}R(\boldsymbol{P})\mathbf{e}_j = (1-\alpha)\tfrac{1}{n-k}\cdot\mathbf{e}_j + \tfrac{\alpha}{(1+\alpha)}\tfrac{1}{k(n-k)}\cdot\begin{bmatrix}0\\\mathbf{e}\end{bmatrix}_{n-k} + \tfrac{\alpha^2}{(1+\alpha)}\tfrac{1}{k(n-k)}\cdot\begin{bmatrix}\mathbf{e}\\0\end{bmatrix}_k. \quad (31)$$

First we show that the approximation $\hat{\mathbf{x}} = (1-\alpha)\mathbf{e}_j$ can be ε-accurate, depending on the sizes of n and the partitions of the complete-bipartite graph. This will give an example of a sparse approximation of PageRank that is ε-accurate, showing that, when error is measured as $\|\boldsymbol{D}^{-1}(\cdot)\|_1$, seeded PageRank can be localized in the context of complete-bipartite graphs.

Observe from Equation (31) that the error of this sparse approximation is

$$\left\|\boldsymbol{D}^{-1}\left(\mathbf{x} - (1-\alpha)\mathbf{e}_j\right)\right\|_1 = \tfrac{\alpha}{(1+\alpha)}\tfrac{1}{k} + \tfrac{\alpha^2}{(1+\alpha)}\tfrac{1}{(n-k)}$$
$$= \tfrac{\alpha}{(1+\alpha)}\left(\tfrac{1}{k} + \tfrac{\alpha}{(n-k)}\right).$$

For a complete-bipartite graph with balanced partitions (partitions with sizes $k = n/2 = (n-k)$) the error of this approximation is $(\alpha/(1+\alpha))(1+\alpha)2/n = 2\alpha/n$. As n increases this quantity goes to 0, proving that, for large enough graphs, the sparse approximation $\hat{\mathbf{x}} = (1-\alpha)\mathbf{e}_j$ has error less than ε when measured in the degree-normalized 1-norm. This result demonstrates that seeded PageRank can be localized on dense graphs (since a complete bipartite graph with partitions of size $n/2$ has $O(n^2)$ edges) when error is measured in the degree-normalized 1-norm.

Sparse $K_{k,n-k}$. In contrast with the above example of a dense graph, this paper is primarily concerned with *sparse* graphs – graphs with $O(n)$ edges. This is because real-world graphs that exhibit localization behavior tend to be sparse. Hence, next we consider specifically complete-bipartite graphs in which at least one partition is of a constant size, as this class of graph has only $O(n)$ edges.

Forcing one partition to have constant size creates two cases: the case in which the seed is in the partition of constant size ($k = O(1)$) and the case in which it is not ($(n-k) = O(1)$). Next we show in each of these two cases that seeded PageRank vectors are de-localized (in the context of sparse complete-bipartite graphs) when error is measured in the degree-normalized 1-norm.

First we address the case that the seed is in the small partition, so $k = O(1)$. In this case, our approximation $\hat{\mathbf{x}}$ can retain all k nonzeros in $\mathbf{x}$ that correspond to nodes in the small partition, and the approximation will still be localized. This leaves $n - k$ entries in $\mathbf{x}$ that we have not yet retained, all of magnitude $\alpha/(k(n-k)(1+\alpha))$. If our approximation $\hat{\mathbf{x}}$ retains z of these nonzeros (in addition to the other k nonzeros already retained), then the error is

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1 = \tfrac{\alpha}{(1+\alpha)}\tfrac{1}{k}\left(1 - \tfrac{z}{n-k}\right).$$

Thus, in order to guarantee $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1 \leq \varepsilon$, the number of nonzeros retained must satisfy

$$\left(1 - \varepsilon \frac{(1+\alpha)k}{\alpha}\right)(n-k) \leq z.$$

Because we assumed that $k = O(1)$, we know $(n-k) = O(n)$, and so this proves that the PageRank vector $\mathbf{x}$ is de-localized, as long as $\varepsilon < \frac{\alpha}{k(1+\alpha)}$. (We note that this is a constant because k is assumed to be a constant.)

Next we address the case that the seed is in the large partition, which is the case $k = \Theta(n)$. In this case our approximation can retain all $n - k = O(1)$ nonzeros in the small partition and the approximation will still be sparse. This leaves $k - 1$ entries of magnitude $\alpha^2/(k(n-k)(1+\alpha))$, and the seed node. To make the analysis a little cleaner, let our approximation be $\hat{\mathbf{x}} = (1 - \alpha)\mathbf{e}_j$ plus the $n - k$ nonzeros in the small partition of the graph. If, in addition, our approximation retains z of the k remaining nonzeros, then the error is

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1 = \frac{\alpha^2}{(1+\alpha)} \frac{1}{n-k} \left(1 - \frac{z}{k}\right).$$

Thus, in order to guarantee $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_1 \leq \varepsilon$, the number of nonzeros retained must satisfy

$$\left(1 - \varepsilon \frac{(1+\alpha)(n-k)}{\alpha^2}\right)k \leq z.$$

Because we assumed that $k = \Theta(n)$, we know $(n-k) = O(1)$, and so this proves that the PageRank vector $\mathbf{x}$ is de-localized, as long as $\varepsilon < \frac{\alpha}{(1+\alpha)(n-k)}$. (We note again that this is a constant because here we assume $n - k$ is a constant.) This completes our proof that seeded PageRank vectors on sparse complete-bipartite graphs are de-localized in the degree-normalized 1-norm.

A.2.2 Localization behavior in the 2-norm

To study how localization behaves when error is measured in the 2-norm, we return to Equation (24). The analysis we performed above (to relate sparsity of the approximation to α and ε) carries over to the 2-norm, with only mild changes.

The big difference (that seeded PageRank on complete-bipartite graphs is *always* localized when error is measured in the 2-norm) is a consequence of the following relationship of the 1-norm and 2-norm. The best example of a de-localized vector is the uniform distribution, the length n vector $(1/n)\mathbf{e}$. Note that the 1-norm of this vector is 1, but the 2-norm is $1/\sqrt{n}$. Thus, as n increases, this uniform distribution stays de-localized in the 1-norm, but can be "accurately approximated" with the 0 vector if error is measured in the 2-norm: $\|(1/n)\mathbf{e} - 0\|_2 = 1/\sqrt{n}$, which is bounded by ε as n increases. This same effect makes seeded PageRank vectors "localized" when error is measured in the 2-norm, because, in our setting of complete-bipartite graphs, the PageRank vectors are essentially two uniform distributions glued together (see the equation in Theorem 6).

Here we demonstrate this localization behavior more rigorously. Following the analysis in Section A.2.1, consider the squared 2-norm error resulting from the approximation $\hat{\mathbf{x}} = (1 - \alpha)\mathbf{e}_j$ plus z_1 non-zero entries retained outside the seed's partition, and z_2 nonzeros retained in the seed's partition. To compute the squared 2-norm error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$, we must square all entries that we do not retain and then sum the result. This produces the following modification of Equation (26) which we derived when measuring error in the 1-norm:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \left(\frac{\alpha}{1+\alpha}\right)^2 \frac{1}{(n-k)} \left(1 - \frac{z_1}{n-k}\right) + \left(\frac{\alpha^2}{1+\alpha}\right)^2 \frac{1}{k} \left(1 - \frac{z_2}{k}\right). \tag{32}$$

The factor of $1/(n-k)$ comes from the fact that in $\mathbf{x}$ there are $(n-k)$ entries of magnitude $(\alpha/(1+\alpha))(1/(n-k))$ that must be squared and summed to compute the 2-norm. (The factor of $1/k$ has a similar origin.)

Equation (32) allows us to show that seeded PageRank can always be sparsely approximated with a 2-norm accuracy of ε. If both $(n-k)$ and k are large, then we can set $z_1 = 0$ and $z_2 = 0$ and obtain an ε accurate approximation; in other words, $\hat{\mathbf{x}} = (1 - \alpha)\mathbf{e}_j$ would be an ε-accurate approximation in the 2-norm. This is because setting $z_1, z_2 = 0$ results in the error

$$\|\mathbf{x} - (1 - \alpha)\mathbf{e}_j\|_2^2 = \left(\frac{\alpha}{1+\alpha}\right)^2 \frac{1}{(n-k)} + \left(\frac{\alpha^2}{1+\alpha}\right)^2 \frac{1}{k}$$

which goes to 0 if both k and $(n - k)$ increase.

In the case that one partition stays constant size, then we set $\hat{\mathbf{x}}$ to retain all nonzeros in the partition of constant size. If $k = O(1)$, then this results in an approximation $\hat{\mathbf{x}}$ with no more than $k + 1$ nonzeros, and an error of $(\alpha/(1 + \alpha))^2/(n - k)$, which goes to zero as n increases (since we have assumed k is bounded by a constant). On the other hand, if $(n - k) = O(1)$, then retaining all $(n - k)$ nonzeros still results in an approximation with a constant number of nonzeros, and the error will be $(\alpha^2/(1 + \alpha))^2/k$. Again, this goes to 0 as n increases, because $(n - k) = O(1)$ implies $k = (n)$, and so k would go to infinity as well.

This completes a proof that seeded PageRank on complete-bipartite graphs can be approximated with a 2-norm accuracy of ε using only a constant number of nonzeros.

Degree-normalized 2-norm. Here we see the same localization behavior for seeded PageRank vectors on complete-bipartite graphs when measuring error in a degree-normalized 2-norm. The proof follows the proofs above with only minor modifications to account for the degree scaling.

From Equation (25) we have

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 = \left(\frac{\alpha}{1+\alpha}\right)^2 \frac{1}{k^2(n-k)} \left(1 - \frac{z_1}{n-k}\right) + \left(\frac{\alpha^2}{1+\alpha}\right)^2 \frac{1}{(n-k)^2 k} \left(1 - \frac{z_2}{k}\right), \tag{33}$$

where again z_1 and z_2 are the number of nonzeros from the two graph partitions that we retain in the approximate solution. Setting $z_1, z_2 = 0$, (once again using the sparse

approximation $\hat{\mathbf{x}} = (1 - \alpha)\mathbf{e}_j$) we get the squared, degree-normalized 2-norm error

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2 = \frac{1}{k(n-k)} \left(\left(\frac{\alpha}{1+\alpha}\right)^2 \frac{1}{k} + \left(\frac{\alpha^2}{1+\alpha}\right)^2 \frac{1}{(n-k)} \right)$$
$$\leq \frac{1}{k(n-k)},$$

which is bounded by $1/n$. Hence, we have that $\|\boldsymbol{D}^{-1}(\mathbf{x} - \hat{\mathbf{x}})\|_2 \leq 1/\sqrt{n}$, which goes to 0 as n increases. This proves that seeded PageRank can be accurately approximated with a constant number of non-zeros when error is measured in the degree-normalized 2-norm.

A.2.3   Localization behavior in residual vectors

As a final observation on the relationship of norms to de-localization behavior, we look at the localization of *residual* vectors related to approximate PageRank vectors. Many algorithms for rapidly approximating seeded PageRank vectors in a sparse manner (such as the Gauss-Southwell linear solver we use in this paper) operate by maintaining sparse iterative solution $\hat{\mathbf{x}}$ and residual vectors $\mathbf{r} = (1 - \alpha)\mathbf{e}_j - (\boldsymbol{I} - \alpha\boldsymbol{P})\hat{\mathbf{x}}$. If convergence is desired in the 1-norm then often the exact error can be computed simply by studying $\|\hat{\mathbf{x}}\|_1$. This is possible if $\mathbf{x} \geq \hat{\mathbf{x}} \geq 0$ holds entry-wise. However, if convergence is desired in other norms (for example, $\|\boldsymbol{D}^{-1}(\cdot)\|_\infty$ is used in [Andersen et al., 2006]) then the residual is used to determine convergence via the relationship $\mathbf{x} - \hat{\mathbf{x}} = (1 - \alpha)(\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\mathbf{r}$.

Here we construct a toy example to show that, depending on which norm is used, it is possible to have a sparse residual with norm less than ε, even if the exact solution is de-localized and the approximate solution has poor accuracy. The purpose of such an example is to show that we must carefully choose the norms we used to determine convergence of our algorithms for seeded PageRank, especially when the residual is used to determine convergence.

The toy example is simply the seeded PageRank vector for the star graph on n nodes, where the seed is the center node, which we will index as node 1. In this setting, the trivial approximation $\hat{\mathbf{x}} = 0$ has residual $\mathbf{r} = (1 - \alpha)\mathbf{e}_1$. Using the degree-normalized 1- or 2-norm suggests that the approximation is good:

$$\|\boldsymbol{D}^{-1}\mathbf{r}\|_1 = \|\boldsymbol{D}^{-1}(1 - \alpha)\mathbf{e}_1\|_1 = \frac{(1-\alpha)}{(n-1)}$$
$$\|\boldsymbol{D}^{-1}\mathbf{r}\|_2 = \|\boldsymbol{D}^{-1}(1 - \alpha)\mathbf{e}_1\|_2 = \frac{(1-\alpha)}{(n-1)}.$$

In both norms, the residual has norm $(1 - \alpha)/(n - 1)$, which goes to 0 as n increases. This is problematic because the actual solution vector, computed via Theorem 6, is $\mathbf{x} = (1/(1 + \alpha))\mathbf{e}_1 + (\alpha/(1 + \alpha))/(n - 1)(\mathbf{e} - \mathbf{e}_1)$ – as a uniform distribution with one altered entry, this vector is totally de-localized, yet a sparse, inaccurate approximation ($\hat{\mathbf{x}} = 0$) yields a sparse residual $((1 - \alpha)\mathbf{e}_1)$ with seemingly high accuracy, $(1 - \alpha)/(n - 1)$, when viewed through the degree-normalized 1- and 2-norms. In contrast, both the 1- and 2-norms report a large norm for this residual: $\|(1 - \alpha)\mathbf{e}_1\|_1 = \|(1 - \alpha)\mathbf{e}_1\|_2 = (1 - \alpha)$.

Again, this suggests that non-degree-normalized norms are more trustworthy than degree-normalized norms for detecting localization behavior in seeded PageRank vectors.