Journal of Computational Science 26 (2018) 205–216

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Numerically approximating centrality for graph ranking guarantees

Eisha Nathan[a,*], Geoffrey Sanders[b], Van Emden Henson[b], David A. Bader[a]

[a] School of Computational Science and Engineering, Georgia Tech, Atlanta, GA, United States
[b] Lawrence Livermore National Laboratory, Livermore, CA, United States

ARTICLE INFO

Article history:
Received 14 October 2017
Received in revised form 7 February 2018
Accepted 22 February 2018
Available online 24 February 2018

Keywords:
Graphs
Centrality measures
Data analysis
Ranking
Numerical accuracy

ABSTRACT

Many real-world datasets can be represented as graphs. Using iterative solvers to approximate graph centrality measures allows us to obtain a ranking vector on the nodes of the graph, consisting of a number for each vertex in the graph identifying its relative importance. In this work the centrality measures we use are Katz Centrality and PageRank. Given an approximate solution, we use the residual to accurately estimate how much of the ranking matches the ranking given by the exact solution. Using probabilistic matrix norms, we obtain bounds on the accuracy of the approximation compared to the exact solution with respect to the highly ranked nodes and apply numerical analysis to the computation of centrality with iterative methods. This relates the numerical accuracy of the linear solver to the data analysis accuracy of finding the correct ranking. In particular, we answer the question of which pairwise rankings are reliable given an approximate solution to the linear system. Experiments on many real-world undirected and directed networks up to several million vertices and several hundred million edges validate our theory and show that we are able to accurately estimate large portions of the approximation. We also analyze the difference between global centrality scores and personalized scores (w.r.t. specific seed vertices). By analyzing convergence error, we develop confidence in the ranking schemes of data mining. We show we are able to accurately guarantee ranking of vertices with an approximation to centrality metrics faster than current methods.

1. Introduction

Graphs are a very popular means of representing massive amounts of relational data. One of the most popular questions arising from the analysis of large graphs is to determine the most important vertices in a graph. Vertex importance is referred to as centrality, and centrality scores can be used to provide rankings on the vertices of a graph. While there exist many such centrality measures, in this work we focus on Katz Centrality and PageRank because of their analytical tractability. Efficiently solving for either of these centrality measures in a graph involves solving a linear system. Obtaining an exact solution via direct methods is prohibitively computationally expensive, since we are required to take the inverse of a matrix. Although direct methods can usually obtain high accuracy solutions, these methods tend to consume large amounts of memory or take a long time to compute. For example, when graphs are small-world and scale-free (as are many real-world networks), direct methods like Cholesky require $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ computations [1]. In many real networks the amount of data is massive and n can be as large as millions or billions of vertices, so direct methods such as these do not scale and are impractical. Moreover, there is no technique to compute an exact solution for a general graph in finite precision arithmetic, so in practice, iterative methods are often used to obtain an approximate solution. Iterative methods tend to use less memory than direct methods, where each iteration costs $\mathcal{O}(m)$, where m is the number of edges in the graph. However, in order for an iterative method to be cost effective, the number of iterations must be limited. Many real-world graphs are sparse and $m \ll n^2$ [2]. While occasionally an iterative method may require the use of a preconditioner if the system is ill-conditioned, none of the problems analyzed here are nearly ill-conditioned enough to merit the use of a preconditioner [3]. The cost required to build a preconditioner would not offset the performance benefit gained and therefore in this work we do not use any preconditioner. In this paper we provide theoretical guarantees on the accuracy of an approximate solution compared to the exact solution to certify rankings in the approximation, and explain how they can be used to limit the number of iterations in the iterative solver.

* Corresponding author.
E-mail addresses: enathan3@gatech.edu (E. Nathan), sanders29@llnl.gov (G. Sanders), henson5@llnl.gov (V.E. Henson), bader@cc.gatech.edu (D.A. Bader).

https://doi.org/10.1016/j.jocs.2018.02.010
1877-7503/

1.1. Contributions

The main contribution of this work is to relate the two research areas of numerical analysis and data mining. We show that we can approximate the centrality scores of vertices on a graph to a high enough accuracy in order to guarantee vertex ranking in graphs. Theorems 1 and 2 present a new error bound on elements of a ranking vector to provide graph ranking guarantees to the computation of centrality. We turn our numerical theory into a new stopping criterion for iterative solvers in Section 4.2 to identify top ranked vertices in a graph that reduces runtime compared to running a solver to machine precision. This paper presents the extended version of the work by Nathan et al. in [4]. Specifically, we test our method on larger datasets and extend our previous analysis of Katz Centrality to ranking using PageRank. We use Lanczos estimates to bound the $||A||_2$, the matrix 2-norm of the adjacency matrix A in Section 4.3. Our analysis is applied to the computation of both global and personalized centrality scores and we develop sound theory with empirical analysis for both undirected and directed networks. Our work allows for approximate solutions to centrality scores to be used for providing accurate guarantees of vertex ranking in graphs. By approximating the solution to a centrality metric we are able to theoretically guarantee the resultant ranking of highly ranked vertices.

2. Background

2.1. Definitions

Let $G=(V, E)$ be a graph, where V is the set of n vertices and E the set of m edges. Denote the $n \times n$ adjacency matrix A of G with entries $A(i,j)=1$ if there exists an edge from vertex i to j, 0 otherwise. For undirected graphs, $\forall i, j, A(i,j)=A(j,i)$, and in this work all edge weights are 1, although all the theory presented in this paper is easily generalized for weighted graphs.

2.2. Centrality measures

In this paper we focus on two popular linear algebra based centrality metrics, Katz Centrality and PageRank. Katz Centrality scores ($\mathbf{c}_{Katz}$) count the number of weighted walks in a network that end at each vertex in the graph. A walk of length k in a graph is a sequence of vertices $v_1, v_2, \ldots, v_k$ where vertices and edges are allowed to repeat. It is a well-known fact that powers of the adjacency matrix are used to count walks of different lengths [5], where $A^k(i,j)$ gives the number of walks of length k from vertex i to j. We can sum weighted powers of the adjacency matrix to obtain Katz scores as in Eq. (1). Successive powers of the parameter α are used to give less weight to longer walks and α must be in the range $(0, 1/||A||_2)$, where $||A||_2$ is the matrix 2-norm of A. In this work we analyze the effect of varying α in its range.

$$\sum_{k=0}^{\infty} \alpha^k A^{k+1} = A + \alpha A^2 + \alpha^2 A^3 + \cdots + \alpha^k A^{k+1} + \cdots = A(I - \alpha A)^{-1} \quad (1)$$

When Katz Centrality was first introduced, Katz used the column sums of the matrix resolvent to obtain scores as $\mathbf{c}_{Katz}=A(I-\alpha A)^{-1}\mathbf{1}$ [6]. We refer to these as *global Katz scores*. From a graph perspective, these scores count the total number of weighted walks of all lengths ending at each vertex. We can also calculate *personalized Katz scores* from a particular vertex i, or more intuitively, weighted counts of the number of walks of all lengths starting at vertex i and ending at each vertex in the graph. These scores correspond to the ith column in the matrix $A(I-\alpha A)^{-1}$ and are calculated as $\mathbf{c}_{Katz}=A(I-\alpha A)^{-1}\mathbf{e}_i$, where $\mathbf{e}_i$ is the ith canonical basis vector. Similarly, we can define personalized scores from a group of vertices $\mathcal{S} = \{v_1, v_2, \ldots, v_{|\mathcal{S}|}\}$ by defining a vector $\mathbf{e}_{\mathcal{S}} = \mathbf{e}_{v_1} + \mathbf{e}_{v_2} + \cdots + \mathbf{e}_{v_{|\mathcal{S}|}}$. The personalized scores w.r.t. $\mathcal{S}$ are then calculated as $\mathbf{c}_{Katz} = A(I-\alpha A)^{-1}\mathbf{e}_{\mathcal{S}}$. In this work when dealing with personalized scores we only use a single vertex, although the analyses presented can easily be extended to the group personalized case. The centrality scores obtained by Katz Centrality can thus be summarized as $\mathbf{c}_{Katz}=A\mathbf{x}_{Katz}$, where $\mathbf{x}_{Katz}$ is the solution to the linear system in Equation (2).

$$M_{Katz}\mathbf{x}_{Katz} = \mathbf{b}_{Katz} \quad (2)$$

We define $M_{Katz}=I-\alpha A$ and $\mathbf{b}_{Katz}$ to be either $\mathbf{1}$ or $\mathbf{e}_i$ depending on whether we are solving for the global or personalized Katz scores.

The next centrality metric we analyze in this paper is PageRank ($\mathbf{c}_{PR}$). PageRank is another walk-based centrality metric that assigns high scores to vertices that are visited by a large number of random walks in the network [7]. It was first introduced to rank webpages in a web search. Given a query from the user, PageRank incorporates a measure of a webpage's importance into the results of a set of webpages that could be relevant to the user's search query. To define the PageRank problem, consider a hypothetical random web surfer navigating between pages online. After visiting a webpage, this random surfer flips a coin: if the coin comes up heads he randomly transitions to a link from the current page, otherwise if the coin comes up tails he *teleports* to a (possibly random) page independent of the current page's identity. If we let $P=A^TD^{-1}$ be the transition matrix of probabilities, then $P(i,j)$ is the probability of transitioning from page j to page i. The random surfer transitions according to the link structure of the web with probability α and teleports randomly with probability $1-\alpha$. Most applications set α to 0.85 [5], so in this work we also fix $\alpha=0.85$ when analyzing PageRank. The scores are calculated as $\mathbf{c}_{PR} = (I-\alpha A^T D^{-1})^{-1}\mathbf{b}_{PR}$, where again $\mathbf{b}_{PR}$ can be set to either $\mathbf{1}$ or $\mathbf{e}_i$ for *global* or *personalized* scores. Similar to Katz Centrality, we can define a linear system to solve for the PageRank scores ($\mathbf{c}_{PR}$) as in Equation (3), where $M_{PR}=I-\alpha A^T D^{-1}$.

$$M_{PR}\mathbf{c}_{PR} = \mathbf{b}_{PR} \quad (3)$$

2.3. Iterative methods

Since solving for many linear algebra based centrality measures directly is generally intractable, in practice we use iterative solvers to solve for them [8]. Iterative methods approximate the solution $\mathbf{x}$ in a linear system $M\mathbf{x}=\mathbf{b}$, given M and $\mathbf{b}$, by starting with an initial guess $\mathbf{x}_0$ and iteratively improving the current guess at each iteration. In this work we use $\mathbf{x}_0=\mathbf{0}$. At each iteration k of the iterative solver we obtain new approximations $\mathbf{x}^{(k)}$ and $\mathbf{c}^{(k)}$ to the unknown exact solutions $\mathbf{x}^*$ and $\mathbf{c}^*$ respectively. The error at each iteration is denoted as the difference between the exact and approximation, $||\mathbf{x}^*-\mathbf{x}^{(k)}||_2$ and the residual norm as $r_k=||\mathbf{b}-M\mathbf{x}^{(k)}||_2$, where $||\cdot||_2$ denotes the 2-norm. Since usually the exact solution is not known, typical stopping criteria for the iterative solver use the residual norm, terminating when it hits machine precision, $r_k \approx 10^{-15}$. Since we analyze both undirected and directed graphs, we use two different iterative methods, although the theory presented can be used in conjunction with other iterative techniques. For undirected graphs, we use conjugate gradient (without a preconditioner) [9] and for directed graphs we use the generalize minimum residual method (GMRES) [10]. We solve the linear systems in Eqs. (2) and (3). For Katz Centrality, the problem is more ill-conditioned and harder as $\alpha \to 1/||A||_2$ and typically requires more iterations to terminate and converge to machine precision.

3. Related work

Many data analysis problems are answered by solving an induced numerical problem [11]. In this paper, by treating the data analysis problem of identifying the highly ranked vertices as obtaining an approximate solution to a linear system, we present how error in this approximation affects the solution to the original ranking problem. Apart from Katz Centrality and PageRank, several centrality measures can be expressed as functions of the adjacency matrix of a graph [12]. We focus on linear solved based techniques in this paper. Other matrix functions can have precision based iterative stopping criterions, which can be guided by our work presented here. For example, *eigenvector centrality* ranks vertices according to the eigenvector corresponding to the largest eigenvalue of the adjacency matrix [13]. Eigenvector centrality takes into account all walks through the network by considering both direct connections to vertices (edges to neighbors) as well as indirect (paths through the network). It is defined as the solution $\mathbf{x}$ to the linear equation $A\mathbf{x} = \lambda\mathbf{x}$, where λ is the largest eigenvalue of A. The next two measures rank vertices with respect to counting walks in the network. The *subgraph centrality* of a vertex weights walks in the graph of length k by a factor of $\frac{1}{k!}$ [14]. The number of walks of length k between nodes i and j is given by $[A^k](i, j)$. This gives rise to the series $\sum_{k=0}^{\infty} A^k/k!$. The *total subgraph communicability* of a vertex is defined in terms of the row sums of matrix functions of the adjacency matrix of the network. The most common function is that of the matrix exponential: $e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots + \frac{A^k}{k!} = \sum_{k=0}^{\infty} \frac{A^k}{k!}$. The subgraph centrality of node i is given by $[e^A](i, i)$ while the subgraph communicability between nodes i and j is given by $[e^A](i, j)$ [5]. A high subgraph centrality indicates a more important vertex in the network and a high subgraph communicability between two vertices indicates that information flows more easily between those two nodes compared to other pairs of nodes with lower subgraph communicability.

Ranking vertices in graphs and finding the top ranked vertices is of very practical relevance to data analysts. Relative importance of top vertices with respect to a particular seed set and ranking in practical settings are studied in [15]. As mentioned in Section 2, solving for many linear algebra based centrality measures directly is generally intractable so iterative solvers are used to approximate them [8]. To identify highly ranked vertices using linear algebra-based centrality, previous work in the literature typically first runs an iterative solver to machine precision. The vertices returned as "highly ranked" are then the ones at the top of the sorted ranking vector (i.e., the ones with a larger centrality score) [16,17]. However, there are several problems with this approach. First, running to machine precision is slow and can require many iterations to converge. Second, there is not guarantee of correctness compared to the unknown exact solution, meaning that while we are provided with the ranking, there is no way to determine which part of the ranking we can treat with confidence. Understanding the error in the approximate solution to the numerical problem is key to understanding the error in the data mining problem of ranking.

We focus on approximating the Katz score of the vertices in the graph to a high enough accuracy to certify that the top of the ranking vector is accurate compared to the exact solution. Several other methods for approximating Katz scores across the network only examine paths up to a certain length [16] or employ low-rank approximation [17]. In [18], the authors provide theoretical guarantees for pairwise Katz scores and provide an algorithm to find the Katz scores from one vertex to the rest of the graph with reduced complexity. They use the Lanczos process to provide upper and lower bounds on the estimate of the pair-wise scores and exploit localization of the Katz matrix to provide estimates on the Katz scores. Our work differs in that we provide confidence as to which portion of the global ranking is correct and use the size of the residual to provide an accurate estimation of the ranking. Furthermore, we extend our analysis to PageRank. Several of the techniques presented in this paper can be used for other matrix estimates or functions to know when the iterative method converges.

We relate the two research areas of numerical analysis and data mining by understanding how the error in a solver affects the data analysis problem of ranking. The main contribution of this paper is bounding the error between the approximate and exact solutions to accurately certify top portions of the ranking with thorough experimentation to validate our results. We derive the bound and provide error analysis in Section 4. Numerical experiments validating the bound including analysis of both precision and performance of our method are presented in Section 5. Finally, in Section 6 we conclude and discuss further uses of this work.

4. Theory

Recall that to solve for both Katz Centrality and PageRank, we are solving a linear system. For Katz Centrality we solve for the vector $\mathbf{c}_{Katz} = A(I - \alpha A)^{-1}\mathbf{b}_{Katz}$, or equivalently solve the linear system $(I - \alpha A)\mathbf{x}_{Katz} = M_{Katz}\mathbf{x}_{Katz} = \mathbf{b}_{Katz}$ and then obtain $\mathbf{c}_{Katz}$ as $A\mathbf{x}_{Katz}$, where the right-hand side $\mathbf{b}$ is set accordingly as described in Section 2.2. For PageRank we solve for the vector $\mathbf{c}_{PR} = (I - \alpha A^T D^{-1})^{-1}\mathbf{b}_{PR}$, or equivalently we solve the linear system $(I - \alpha A^T D^{-1})\mathbf{c}_{PR} = M_{PR}\mathbf{c}_{PR} = \mathbf{b}_{PR}$. When the solution $\mathbf{c} = M^{-1}\mathbf{b}$ to either linear system is approximated, there will be differences between the approximate solution and the exact solution, where $\mathbf{c}$ is either $\mathbf{c}_{Katz}$ or $\mathbf{c}_{PR}$. We prove that these differences along with the ranking values can indicate how far down the ranking we can go before the approximation error makes it unreliable.

For iteration k of the iterative solver, define $\mathbf{d}^{(k)} = \boldsymbol{\pi}^{(k)}\mathbf{c}^{(k)}$, where $\boldsymbol{\pi}^{(k)}$ is the permutation such that $\mathbf{d}^{(k)}$ is the vector $\mathbf{c}^{(k)}$ ordered in decreasing order so that $d_i^{(k)} \geq d_{i+1}^{(k)}$. Define $\lambda_{min}(M)$ to be the smallest eigenvalue of the matrix M and $\sigma_{min}(M)$ to be the smallest singular value of the matrix M, where M is either M_{Katz} or M_{PR}. Again recall that the residual norm is given as $r_k = ||\mathbf{b} - M\mathbf{x}^{(k)}||_2$.

4.1. Error analysis

Theorem 1 below provides guarantees as to when the rank of vertex i above j is correct from the approximate solution using Katz Centrality.

Theorem 1. *For undirected graphs, for any $i < j$, the rank of vertex i above j using Katz Centrality is correct if $|d_i^{(k)} - d_j^{(k)}| > 2\epsilon_k$ for $\epsilon_k = \frac{||A||_2}{\lambda_{\min}(M_{Katz})} r_k$. For directed graphs, for any $i < j$, the rank of vertex i above j is correct if $|d_i^{(k)} - d_j^{(k)}| > 2\epsilon_k$ for $\epsilon_k = \frac{||A||_2}{\sigma_{\min}(M_{Katz})} r_k$.*

Proof. Using foundations of error analysis in linear solvers, we can bound the point-wise error in the ranking, which will then provide a sufficient error gap in the elements of the approximation to the ranking vector.

$$
\begin{aligned}
||\mathbf{d}^*_{Katz} - \mathbf{d}^{(k)}_{Katz}||_\infty &= ||\mathbf{c}^*_{Katz} - \mathbf{c}^{(k)}_{Katz}||_\infty \\
&\leq ||\mathbf{c}^*_{Katz} - \mathbf{c}^{(k)}_{Katz}||_2 \\
&= ||A\mathbf{x}^*_{Katz} - A\mathbf{x}^{(k)}_{Katz}||_2 \\
&\leq ||A||_2 ||\mathbf{x}^*_{Katz} - \mathbf{x}^{(k)}_{Katz}||_2 \\
&= ||A||_2 ||M^{-1}_{Katz}\mathbf{b}_{Katz} - \mathbf{x}^{(k)}_{Katz}||_2 \\
&\leq ||A||_2 ||M^{-1}_{Katz}||_2 ||\mathbf{b}_{Katz} - M_{Katz}\mathbf{x}^{(k)}_{Katz}||_2 \\
&\leq ||A||_2 ||M^{-1}_{Katz}||_2 r_k
\end{aligned}
$$

For undirected graphs (with A symmetric), we have $||M_{Katz}||^{-1} \leq \frac{1}{\lambda_{\min}(M_{Katz})}$, so we can write:

$$||\mathbf{d}^*_{Katz} - \mathbf{d}^{(k)}_{Katz}||_\infty \leq \frac{||A||_2}{\lambda_{\min}(M_{Katz})} r_k \tag{4}$$

$$=: \epsilon_k$$

For directed graphs (with A nonsymmetric), $||M_{Katz}||^{-1}$ is bounded by the inverse of the minimum singular value instead of the inverse of the minimum eigenvalue:

$$\begin{aligned} ||\mathbf{d}^*_{Katz} - \mathbf{d}^{(k)}_{Katz}||_\infty &\leq \frac{||A||_2}{\sigma_{\min}(M_{Katz})} r_k \\ &=: \epsilon_k \end{aligned} \tag{5}$$

Since $d(i)^{(k)}_{Katz} - d(i)^*_{Katz} < \epsilon_k$ and $d(j)^*_{Katz} - d(j)^{(k)}_{Katz} < \epsilon_k$, this means that $d(i)^*_{Katz} - d(j)^*_{Katz} > d(i)^{(k)}_{Katz} - d(j)^{(k)}_{Katz} - 2\epsilon_k$. If $d(i)^{(k)}_{Katz} - d(j)^{(k)}_{Katz} > 2\epsilon_k$, then $d(i)^*_{Katz} - d(j)^*_{Katz} > 0$ meaning that the ranking of vertex i above j is correct.

Similarly, we can derive a corresponding bound for PageRank to guarantee the ranking of vertices in the approximate ranking vector. We again separate the bounds into the undirected and directed graph cases.

Theorem 2. *For undirected graphs, for any $i<j$, the rank of vertex i above j is correct using PageRank if $|d^{(k)}_i - d^{(k)}_j| > 2\epsilon_k$ for $\epsilon_k = \frac{1}{\lambda_{\min}(M_{PR})} r_k$. For undirected graphs, for any $i<j$, the rank of vertex i above j is correct using PageRank if $|d^{(k)}_i - d^{(k)}_j| > 2\epsilon_k$ for $\epsilon_k = \frac{1}{\sigma_{\min}(M_{PR})} r_k$.*

Proof.

$$\begin{aligned} ||\mathbf{d}^*_{PR} - \mathbf{d}^{(k)}_{PR}||_\infty &= ||\mathbf{c}^*_{PR} - \mathbf{c}^{(k)}_{PR}||_\infty \\ &\leq ||\mathbf{c}^*_{PR} - \mathbf{c}^{(k)}_{PR}||_2 \\ &= ||M^{-1}_{PR}\mathbf{b}_{PR} - \mathbf{x}^{(k)}_{PR}||_2 \\ &\leq ||M^{-1}_{PR}||_2 ||\mathbf{b}_{PR} - M_{PR}\mathbf{x}^{(k)}_{PR}||_2 \\ &\leq ||M^{-1}_{PR}||_2 r_k \end{aligned}$$

For undirected graphs (with A symmetric), we have $||M_{PR}||^{-1} \leq \frac{1}{\lambda_{\min}(M_{PR})}$, so we can write:

$$||\mathbf{d}^*_{PR} - \mathbf{d}^{(k)}_{PR}||_\infty \leq \frac{1}{\lambda_{\min}(M_{PR})} r_k \tag{6}$$

$$=: \epsilon_k$$

For directed graphs (with A nonsymmetric), $||M_{PR}||^{-1}$ is bounded by the inverse of the minimum singular value instead of the inverse of the minimum eigenvalue:

$$||\mathbf{d}^*_{PR} - \mathbf{d}^{(k)}_{PR}||_\infty \leq \frac{1}{\sigma_{\min}(M_{PR})} r_k \tag{7}$$

$$=: \epsilon_k$$

Again, since $d(i)^{(k)}_{PR} - d(i)^*_{PR} < \epsilon_k$ and $d(j)^*_{PR} - d(j)^{(k)}_{PR} < \epsilon_k$, this means that $d(i)^*_{PR} - d(j)^*_{PR} > d(i)^{(k)}_{PR} - d(j)^{(k)}_{PR} - 2\epsilon_k$. If $d(i)^{(k)}_{PR} - d(j)^{(k)}_{PR} > 2\epsilon_k$, then $d(i)^*_{PR} - d(j)^*_{PR} > 0$ meaning that the ranking of vertex i above j is correct.

We observe in practice that the bounds in Theorems 1 and 2 are tight enough to produce relevant results in many practical applications (seen in Section 5) and lend themselves to the development of a new stopping criterion for iterative solvers when identifying the highly ranked vertices in a graph.

4.2. New stopping criterion

Current methods for identifying the top vertices in a graph involve running an iterative solver to machine precision to obtain an approximation of $\mathbf{c}^*$. We introduce a new stopping criterion to find these top vertices that typically provides results much faster than existing methods, using the theory developed in Theorems 1 and 2 above. Furthermore, our method provides theoretically sound guarantees as to the correctness of the top vertices, unlike the common method of simply running a solver to machine precision and blindly hoping the resulting vector is good enough for the desired data mining task.

Suppose a user desires a set of j vertices containing the top R highly ranked vertices in a graph, with precision ϕ^*. How large does j need to be before we can accurately certify (or guarantee) that the top vertices are in the set? We are not concerned with the internal ordering of this set, but rather that the top R vertices are contained somewhere within the superset of j vertices. The desired precision ϕ^* gives a sense of how many false positives we will tolerate in our set. We answer this question using our theory.

Here, we present the implementation for the theory for Katz Centrality on undirected graphs, but the same principle can be applied to develop a stopping criterion for PageRank or directed networks. For brevity, we drop the *Katz* subscript in this section. This procedure is given in Algorithm 1, for an adjacency matrix A, right-hand side $\mathbf{b}$, number of top vertices R, desired precision ϕ^*, maximum number of iterations k_{max}, and upper bound σ_{up} on $||A||_2$. Note we discuss bounds for $||A||_2$ in the next section. For all the experiments presented in this paper, k_{max} is set to 1000 iterations but none of the trials ever reach this maximum value. At each iteration of conjugate gradient, the current solution $\mathbf{c}^{(k)}$ is ordered in decreasing order to produce the vector $\mathbf{d}^{(k)}$ as described earlier. We find the first position $j>R$ in $\mathbf{d}^{(k)}$ where we find the necessary gap of $|d^{(k)}_R - d^{(k)}_j| > 2\epsilon_k$. The precision for these values of R and j is defined as $\phi = \frac{R}{j-1}$. If for this value of j we have the desired precision ϕ^*, meaning $\phi = \frac{R}{j-1} \geq \phi^*$, then we terminate, else we iterate again using conjugate gradient to obtain a more accurate approximation.

Intuitively the precision shows how far past position R we must travel down the vector to find the necessary gap to ensure we are returning the top R vertices in the graph. Conjugate gradient can be organized to return $\mathbf{x}^{(k)}$, $\mathbf{c}^{(k)}$, and the residual norm r_k at each iteration (denoted CGITERATION in Algorithm 1).

Algorithm 1. Obtain top R vertices in network with precision ϕ^*.

```
1 Function Top_R
      Data: A, b, R, φ*, k_max, σ_up
      Result: Set of j vertices s.t. top R vertices are contained within this
              set
2     k = 0; j = ∞
3     M = I − αA
4     while R/(j−1) < φ* and k < k_max do
5         x^(k), c^(k), r_k = CGITERATION(M, x^(k−1), b)
6         d^(k) = π^(k) c^(k)
7         ε_k = (σ_up / λ_min(M)) r_k
8         j = argmin_{i>R} |d^(k)_R − d^(k)_i| > 2ε_k
9         k += 1
```

To solve for PageRank instead of Katz Centrality, we modify Line 3 to $M = I - \alpha A^T D^{-1}$ and change the bound accordingly in Line 7. For the directed graph case, we use GMRES (with restarts every 20 iterations) instead of conjugate gradient in Line 5 and again modify the bound in Line 7. The vector $\mathbf{b}$ is set to $\mathbf{1}$ or $\mathbf{e}_i$ accordingly if we are solving for the global or personalized scores respectively.

4.3. Bounds on ||A||$_2$

We obtain a tight bound on ϵ_k which allows us to certify that the ranking of vertex i above j is correct if the gap between two elements in the ranking vector is greater than our error bound, $|d_i^{(k)} - d_j^{(k)}| > 2\epsilon_k$. The iterative solver can be organized to readily provide the residual norm r_k at each iteration, and $\lambda_{min}(M)$ or $\sigma_{min}(M)$ can be computed provided α is chosen in the given range. To certify portions of the ranking vector, we desire ϵ_k to be as small as possible to find places in the vector where the necessary gap $|d_i^{(k)} - d_j^{(k)}|$ exists. For the bounds on Katz Centrality, obtaining a tight bound on $||A||_2$ is key to bounding ϵ_k; we present and compare two methods of bounding $||A||_2$.

The Gershgorin Circle Theorem [19] bounds the eigenvalues of the symmetric matrix A. Let $T_i = \sum_{j \neq i} |a_{ij}|$, the sum of the nondiagonal entries in row i. Then $D(a_{ii}, T_i)$ is the closed interval centered at a_{ii} with radius T_i and every eigenvalue $\lambda \in \sigma(A)$ must lie within at least one interval $D(a_{ii}, T_i)$, where $\sigma(A)$ is the spectrum of A. Since the diagonal entries a_{ii} of A are 0, the discs are all centered around the origin and $\forall i, T_i = d_i$ = the degree of vertex i. We then have $||A||_2 = \max|\lambda_i| < \max T_i = d_{max}$, where d_{max} is the largest degree in the graph. While this provides a basis for an upper bound of the matrix 2-norm of A, many real-world graphs such as social networks have a scale-free distribution and thus contain vertices with a very large degree [20]. Therefore, this is often a non-optimal bound. By using just a few matrix–vector multiplications applied to random vectors, we can compute tighter bounds with high certainty.

We next examine probabilistic matrix norm bounds [21] and consider replacing the true bound σ_{up} with an estimate of a bound with some probability. These bounds are developed using the polynomials p, q implicitly formed as a part of the Lanczos bidiagonalization process with starting vector $\mathbf{v}_1$, which is chosen randomly with unit norm. For $\beta_0 = 0$ and $\mathbf{u}^{(0)} = 0$ and $k \geq 1$, the defining relations of Lanczos bidiagonalization are stated as

$$\begin{aligned} \gamma_j \mathbf{u}^{(j)} &= A\mathbf{v}^{(j)} - \beta_{j-1}\mathbf{u}^{(j-1)} \\ \beta_j \mathbf{v}^{(j+1)} &= A^T\mathbf{u}^{(j)} - \gamma_j \mathbf{v}^{(j)}, \end{aligned}$$

where $\gamma_j = \mathbf{u}^{(j)^T} A\mathbf{v}^{(j)}$ and $\beta_j = \mathbf{u}^{(j)^T} A\mathbf{v}^{(j+1)}$ are nonnegative. Therefore the following recurrence relations hold for the recurrent polynomials derived as below:

$$\begin{aligned} \gamma_{j+1} p_j(t) &= q_j(t) - \beta_j p_{j-1}(t) \\ \beta_{j+1} q_{j+1}(t) &= t p_j(t) - \gamma_{j+1} q_j(t), \end{aligned}$$

for $p_{-1}(t) = 0$ and $q_0(t) = 1$ for $j \geq 0$. The bound is stated in Theorem 3 and the algorithm from [21] is reproduced here for clarity. Note in Algorithm 2 that the matrices U and V are the concatenated column vectors $\mathbf{u}_j$ and $\mathbf{v}_j$ respectively. The result is an upper bound $\sigma_{up}(\theta)$ for $||A||_2$ with probability $1 - \theta$, where θ is the user-chosen probability of bound failure. Define $\delta = \theta \cdot \frac{1}{2} B\left(\frac{n-1}{2}, \frac{1}{2}\right)$ where B is Euler's Beta function, $B(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt$.

Theorem 3. [21] *Suppose we have carried out k steps of the Lanczos bidiagonalization process with starting vector* $\mathbf{v}_1$*, and let* $\theta \in (0, 1)$*. Then the largest zero of the polynomials,*

$$f_1(t) = q_k(t^2) - 1/\delta, \quad f_2(t) = t p_k(t^2) - 1/\delta$$

with δ *given above, is an upper bound* $\sigma_{up}(\theta)$ *for* $||A||_2$ *with probability at least* $1 - \theta$*.*

Algorithm 2. Lanczos bidiagonalization to calculate probabilistic upper bounds.

Function *Calc_Upper_Bound*
- **Data:** A, $\mathbf{v}^{(1)}$, θ
- **Result:** Upper bound $\sigma_{up}(\theta)$ on $\|A\|_2$ with probability θ
- $\delta = \theta \cdot \frac{1}{2} B(\frac{n-1}{2}, \frac{1}{2})$
- $p_{-1}(t) = 0$, $q_0(t) = 1$
- **for** $j = 1 \cdots k$ **do**
  - $\mathbf{u} = A\mathbf{v}^{(j)}$
  - **if** $j > 1$ **then**
    - $\mathbf{u} = \mathbf{u} - \beta_{j-1}\mathbf{u}^{(j-1)}$
    - $\mathbf{u} = \mathbf{u} - U_{j-1}(\mathbf{u}^T U_{j-1})^T$
  - $\gamma_j = \|\mathbf{u}\|$
  - $\mathbf{u}_j = \mathbf{u}/\gamma_j$
  - $\mathbf{v} = A^T\mathbf{u}$
  - $\mathbf{v} = \mathbf{v} - \gamma_j \mathbf{v}^{(j)}$
  - $\mathbf{v} = \mathbf{v} - V_j(\mathbf{v}^T V_j)^T$
  - $\beta_j = \|\mathbf{v}\|$
  - $\mathbf{v}^{(j+1)} = \mathbf{v}/\beta_j$
  - $p_j(t) = \frac{q_j(t) - \beta_j p_{j-1}(t)}{\gamma_{j+1}}$
  - $q_{j+1}(t) = \frac{t p_j(t) - \gamma_{j+1} q_j(t)}{\beta_{j+1}}$

As a result of thorough experimentation, for all bounds used in this paper, we select values of $\theta = 0.01$ and $k = 10$. For $k = 10$, in order to calculate $\sigma_{up}(0.01)$ we are required to calculate the largest root of a tenth order polynomial. Since this does not change regardless of problem size n, this calculation is asymptotically a fixed cost. We use Python's SYMPY package to calculate the roots of these polynomials. The deterministic Gershgorin bounds yield large values of $||A||_2$, rendering these bounds useless. On average, these bounds return estimates of $||A||_2$ that are 30.9× greater than the true 2-norm. In contrast, the probabilistic bounds presented in Theorem 3 return estimates of $||A||_2$ that are only on average 1.07× greater than the true 2-norm, meaning that these are able to be used for practical purposes.

Remark 1. Future work will examine obtaining the bound in real-time without any additional computational cost. In the Lanczos algorithm to obtain σ_{up} we are applying A to obtain $\mathbf{u} = A\mathbf{v}$, and in conjugate gradient we are applying A to obtain $(I - \alpha A)\mathbf{x}^{(k)}$ in each iteration. These two operations can be combined and we can apply A to both vectors in the same algorithm, effectively performing both Algorithms 1 and 2 simultaneously, which is important for distributed implementations of these algorithms.

5. Results

In this section we present comparisons to existing methods for identifying the top ranked vertices with respect to performance and experiments validating our bound with respect to precision. We are interested in determining if our method correctly identifies the set of top vertices and if so, how much faster we are able to certify this set. The common method of iterating to machine precision does not theoretically certify this set but our theory can be used on the machine precision solution as well. We conduct experiments on both undirected and directed networks from the KONECT [22] collection, including social networks, autonomous systems, citation, co-authorship, and web graphs. Table 1 gives the undirected networks used and Table 2 gives the directed networks used.

For the results shown here, we vary values of the desired precision as $\phi \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and the top R as $R = 10$, 100, and 1000. For Katz Centrality, we vary the α parameter as a fraction of its upper bound $1/||A||_2$. For personalized centrality results, we form the vector $\mathbf{e}_i$ by choosing a vertex i randomly from the top 10% of highest degree vertices.

Table 1
Undirected graphs used in experiments. Columns are graph name, number of vertices, number of edges, and type of graph.

Graph	$\|V\|$	$\|E\|$	Type
Douban	154,908	327,162	Social
Gowalla	196,591	950,327	Social
Dblp	317,080	1,049,866	Coauthorship
Dogster	426,820	8,546,581	Social
Catster	623,766	15,699,276	Social
Youtube	1,134,890	2,987,624	Social
Skitter	1,696,415	11,095,298	Computer
Flickr	1,715,255	15,551,250	Social
California	1,965,206	2,766,607	Infrastructure
Facebook	63,731	817,035	Social
Pgp	10,680	24,316	Online
Livejournal	5,204,175	49,174,464	Social
Orkut	3,072,441	117,184,899	Social

Table 2
Directed graphs used in experiments. Columns are graph name, number of vertices, number of edges, and type of graph.

Graph	$\|V\|$	$\|E\|$	Type
Edinburgh	23,132	312,342	Lexical
Cora	23,166	91,500	Citation
Lkml	63,399	1,096,440	Social
Epinions	75,879	508,837	Social
Enron	87,273	1,148,072	Social
Baidu	2,141,300	17,794,839	Hyperlink
Wiki-German	3,225,565	8,1626,917	Hyperlink
Wiki-English	18,268,991	172,183,984	Hyperlink

5.1. Speedup in iterations

We first analyze the effect of our stopping criterion on reducing the number of iterations taken by an iterative solver to identify the top R vertices in a graph. We denote the number of iterations taken by either conjugate gradient/GMRES to converge to machine precision as I_E and the number of iterations using our new stopping criterion as I_A and calculate speedup w.r.t. number of iterations as

$$speedup = \frac{I_E}{I_A}.$$

In this section we only show results obtained with a precision of 1.0 (so for a desired set of the top R vertices we return a set guaranteed to have no false positives) and we show results for all values of R (10, 100, and 1000). For Katz Centrality results, we sample all values of α as well. Fig. 1 plots the distribution of the speedups for undirected graphs. Figs. 1a and b plot the histograms for global and personalized Katz Centrality scores, respectively, and Figs. 1c and d show global and personalized results for PageRank, respectively. For the undirected graphs, for Katz scores we have an average of 3.99× speedup for global scores and 4.03× for personalized scores, and for PageRank an average of 6.24× speedup for global scores and 10.23× for personalized scores. Fig. 2 plots the distribution of the speedups for directed graphs, again for global and personalized Katz and PageRank scores. For the directed networks, for Katz scores we obtain an average of 4.60× speedup for global scores and 5.04× for personalized scores, and for PageRank an average of 2.52× speedup for global scores and 23.91× for personalized scores. In all cases we obtain a speedup greater than 1× (meaning our method is always faster) and up to a speedup of a maximum of over two orders of magnitude. This shows that we are able to identify the top R in a fraction of the time using our stopping criterion compared to running until machine precision, while providing a theoretical guarantee that these vertices are in the top of the ranking vector. This is especially significant because running to machine precision can sometimes take hundreds or thousands of iterations.

For all the experiments, iterating to machine precision is our baseline for comparison and the method we evaluate our algorithm against. While running to a low tolerance may suffice in many cases, without any theoretical guarantees it is impossible to know how stable and accurate the solution actually is. Furthermore, running to a low tolerance of approximately 10^{-3} is not guaranteed to return a correct set of the highly ranked vertices. For example, while iterating to a low tolerance of exactly 1e−3 certifies the highly ranked vertices in some cases, for other graphs even just an additional three or four more iterations are needed in order to accurately certify the top 10, 100, and 1000 vertices. Essentially, we are unable to know exactly the tolerance to which we should solve to until we use our stopping criterion to know when we can accurately guarantee the highly ranked vertices. In order to guarantee accurate results, our theory must be used in conjunction with an iterative method.

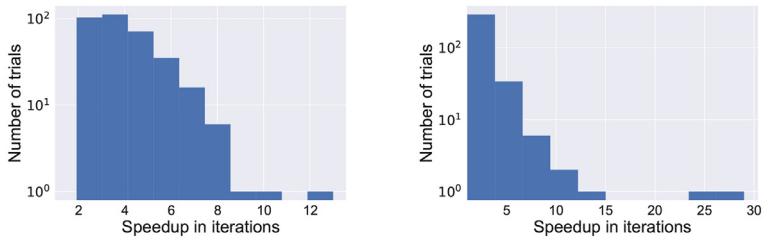
5.2. Performance vs. quality

We have shown that we are able to obtain speedups w.r.t. iteration counts using our theory versus running an iterative solver to machine precision. In this section we examine the effect varying the precision of the returned set of top vertices has on the speedup obtained.

We first explain the behavior of the sorted ranking vector $\mathbf{d}$ of a single undirected graph, *facebook*, a citation network, using Katz Centrality in Fig. 3. Fig. 3a plots the sorted values of $\mathbf{d}$ on a log-scale for all the vertices and Fig. 3b zooms in on selected regions from Fig. 3a. The top plot of Fig. 3b shows values for vertices 110–140 (vertices at the beginning of the sorted vector) and the bottom plot shows values for vertices $n-711$–$n-681$ (vertices with scores toward the end of the vector). The value of ϵ_k obtained as a part of our theory is absolute. We are able to resolve the part of the vector that the data mining task cares about, namely the top of the vector (the highly ranked vertices), with a guarantee that they are correct compared to the exact solution. However, for another use case where the user desires all the vertices in the graph to be returned correctly, since the values typically get closer to each other the further one traverses down the ranking vector, the value of ϵ_k will not be sufficient to provide the necessary gap between two elements toward the end of the vector. This is seen in Fig. 3b. For the top right plot, the two pairs of open red squares indicate pairs of vertices where the gap is sufficient to certify the ranking of one vertex above the other. Using our previous notation, this is translated into a required precision of 1.0 (where we look for gaps between successive vertices). For the first pair, the difference in the scores is $9.4 \times 10^9 \times 2\epsilon_k$ and the difference between the second pair of vertices is $9.9 \times 10^9 \times 2\epsilon_k$. However, in the bottom right plot (values for vertices further down the ranking vector) where the values are very close together, the required gap $2\epsilon_k$ is larger than the difference between successive pairs of points. The two pairs of open red squares indicate pairs of vertices with values too close together to obtain the necessary gap.

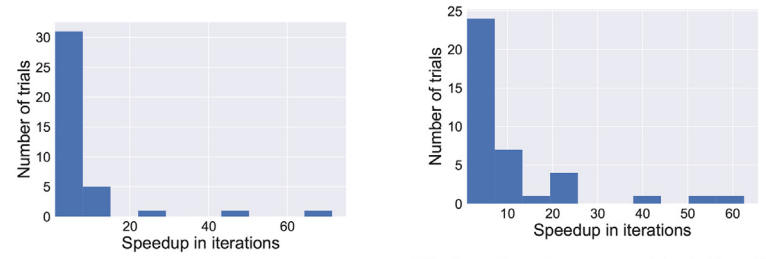
Overall the $\mathbf{d}$ vector follows an exponential decay pattern. The plateau-like behavior of the vector at certain points that is more clearly seen in Fig. 3b can be explained by the fact that the Katz vector tends to have sets of vertices grouped so tightly together around the same value that we are unable to have the necessary separation to apply the error analysis to certify individual vertices' ranking. Therefore, when we want the top R vertices, it is sometimes necessary to travel further down the ranking vector to $j = R + \Delta$ to obtain the required separation between vertices, where Δ is the number of false positives returned in the set, or equivalently, obtain highly ranked vertices with less than perfect (1.0) precision.

Next we examine the tradeoff between performance and quality of our algorithm. Recall for the top R vertices returned in a



(a) Speedup for global Katz scores.

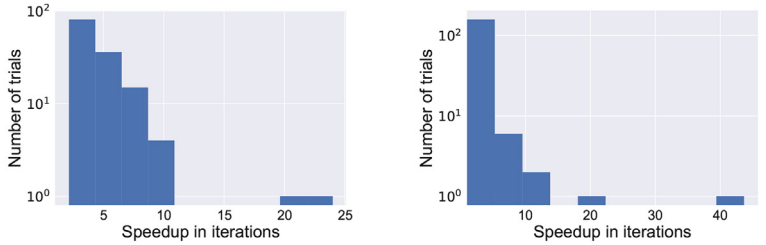
(b) Speedup for personalized Katz scores.



(c) Speedup for global PageRank scores.

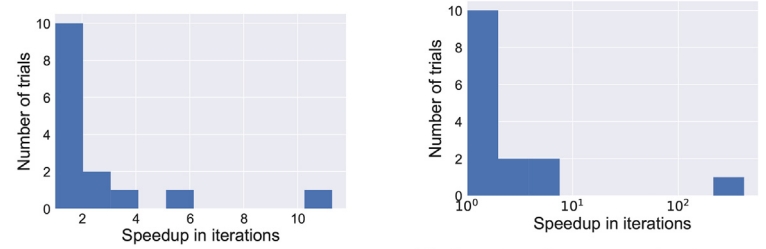
(d) Speedup for personalized PageRank scores.

Fig. 1. Histograms of speedups in iterations for undirected graphs with precision 1.0. All speedup values are above 1.



(a) Speedup for global Katz scores.

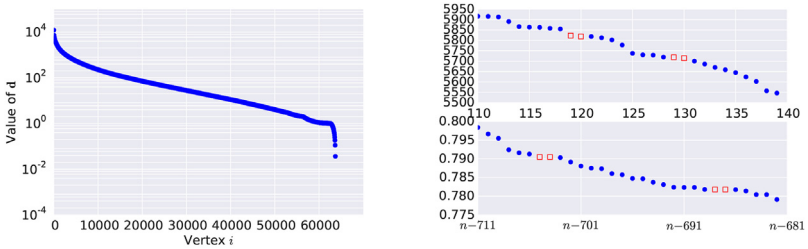
(b) Speedup for personalized Katz scores.



(c) Speedup for global PageRank scores.

(d) Speedup for personalized PageRank scores.

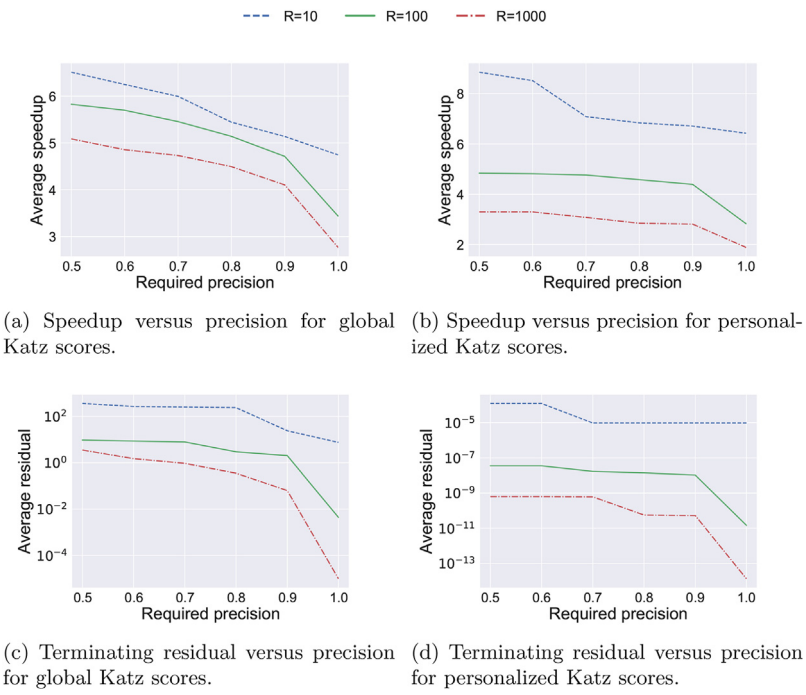
Fig. 2. Histograms of speedups in iterations for directed graphs with precision 1.0. All speedup values are above 1.



(a) All vertices.

(b) Specific subsets of vertices.

Fig. 3. Sorted ranking vector d_{Katz} for *facebook* graph. Values are plotted in blue circles while selected points with an extremely close error gap are shown in red squares. Left plot is on a log-scale; right plots are on a linear scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Speedup versus precision for global Katz scores.

(b) Speedup versus precision for personalized Katz scores.

(c) Terminating residual versus precision for global Katz scores.

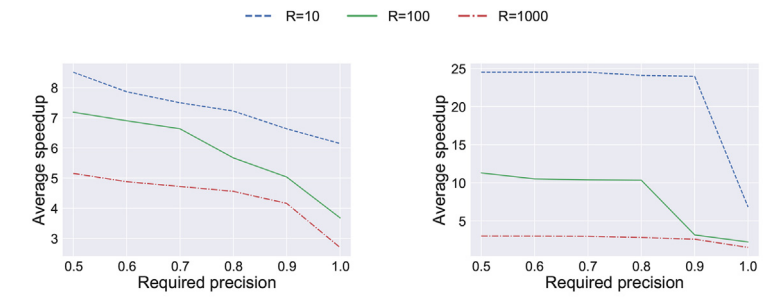
(d) Terminating residual versus precision for personalized Katz scores.

Fig. 4. Performance versus required precision for Katz Centrality on undirected graphs (with $\alpha = 0.9/||A||_2$).

superset of j vertices, we define precision as $\frac{R}{j-1}$. Requiring a predetermined precision of ϕ^* means we want $\frac{R}{j-1} > \phi^*$. Performance is again measured by speedup (in iterations) comparing our method to iterating to machine precision. Results shown are averaged over all the datasets. Fig. 4 plots the average speedup and terminating residual for global and personalized scores for Katz Centrality on undirected graphs, where the terminating residual is the residual upon terminating at our new stopping criterion (iteration $k = I_A$). We plot results for $\alpha = \frac{0.9}{||A||_2}$, although trends seen for other values of α are similar. Fig. 4a and b plot the average speedup versus required precision in iterations for global and personalized scores respectively, and Fig. 4c and d plot the terminating residual versus required precision for global and personalized scores respectively. All plots show results for the top $R = 10$, 100, and 1000 vertices.

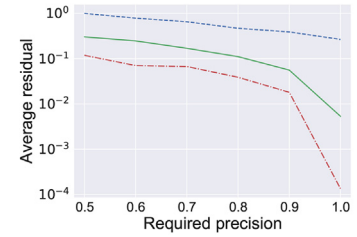
In all cases (for both speedup and terminating residual), we have more of an improvement using our stopping criterion for smaller values of R. More specifically, we obtain greater speedups and are able to terminate at a higher residual (obtaining a less accurate numerical solution) for smaller values of R. This behavior can be attributed to the shape of the centrality vector as explained by Fig. 3 previously. While the gap $2\epsilon_k$ that we are looking for in between elements of the centrality vector is fixed, elements in the vector themselves decrease exponentially. Therefore, for larger values of R we need to traverse further down the ranking vector to obtain the necessary gap. Nevertheless, we still see significant speedups for larger values of R such as 1000. In all cases, even for large R and high precision rates, we are able to terminate at a residual significantly above machine precision. For the personalized results (Fig. 4b and d), we see a greater speedup but lower terminating residual than their global counterparts (Fig. 4a and c). Intuitively, we obtain smaller terminating residuals for the personalized results because the values in the ranking vector themselves are smaller. For a possible reason behind the greater speedup in the personalized case, we turn our attention back to the theory presented in Theorem 1. Our stopping criterion terminates the iterative solver when we have a necessary gap between elements in the ranking vector of $2\epsilon_k = 2\frac{||A||_2}{\lambda_{\min}} r_k$, where r_k is the residual norm. The gap ϵ_k differ in the global and personalized case only in the residual norm. Therefore, the residual dictates how far we need to traverse down the ranking vector until we can guarantee the top vertices in the returned set. Since the residual in the personalized case is several orders of magnitude smaller than the residual in the global case, we seek a smaller gap between elements in the ranking vector. We are therefore able to stop after fewer iterations, relative to machine precision, in the personalized case. Finally as expected, as we increase the required precision we see reduced speedups and smaller terminating residuals. Increasing the required precision means we desire a tighter set of the top R vertices to be returned. For example, for a precision of 1.0 we are looking for a gap of $2\epsilon_k$ between elements R and $R+1$, whereas for a precision of 0.5 we are only looking for a gap between elements R and $2R+1$. Clearly we will be able to find a gap between elements that are farther apart such as R and $2R+1$ much faster than successive elements R and $R+1$, so larger speedups for smaller precisions is not surprising. However, we note that the difference in speedups for required precisions from about 0.5 to 0.9 is about the same as the difference in speedups for required precisions from about 0.9 to 1.0. This means that we are able to quickly obtain highly ranked vertices without sacrificing too much quality.

Fig. 5 broadly plots the same results as above except for directed graphs. We again plot results for $\alpha = \frac{0.9}{||A||_2}$. Fig. 5a and b plot the average speedup versus required precision in iterations for global and personalized scores respectively, and Fig. 5c and d plot the terminating residual versus required precision for global and personalized scores respectively. Most of the same trends discussed from the undirected results are applicable for the directed graphs. In fact, for the personalized speedups (Fig. 4b), there is a much stronger trend of obtaining a relatively constant speedup for precisions of 0.5–0.9 and then a sharp drop in speedup for a precision of 1.0. This suggests that while there are vertices in the ranking vector with these necessary gaps to guarantee ranking, in order to find the gap between successive vertices the solver needs to reach a high level of accuracy. From this we can conclude that if the use case can tolerate a few false positives in the set of the top R highly ranked vertices, then we can obtain the top vertices in a graph quickly with relatively high precision.

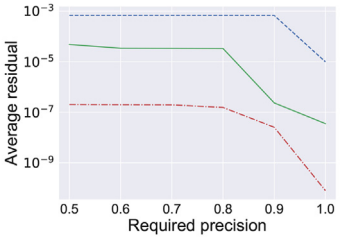


(a) Speedup versus precision for global Katz scores.

(b) Speedup versus precision for personalized Katz scores.

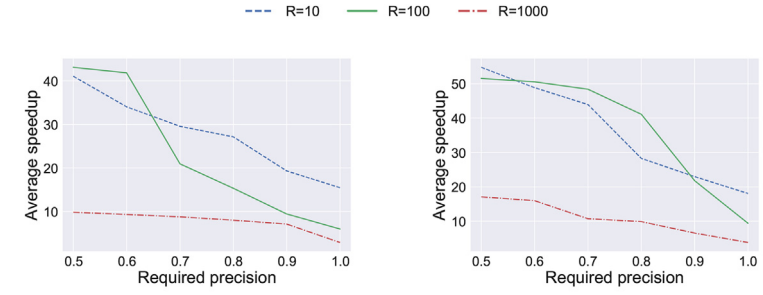


(c) Terminating residual versus precision for global Katz scores.



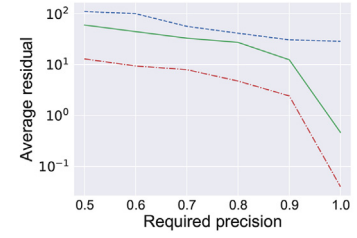
(d) Terminating residual versus precision for personalized Katz scores.

Fig. 5. Performance versus required precision for Katz Centrality on directed graphs (with $\alpha = 0.9/||A||_2$).

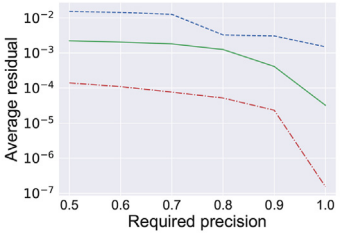


(a) Speedup versus precision for global PageRank scores.

(b) Speedup versus precision for personalized PageRank scores.



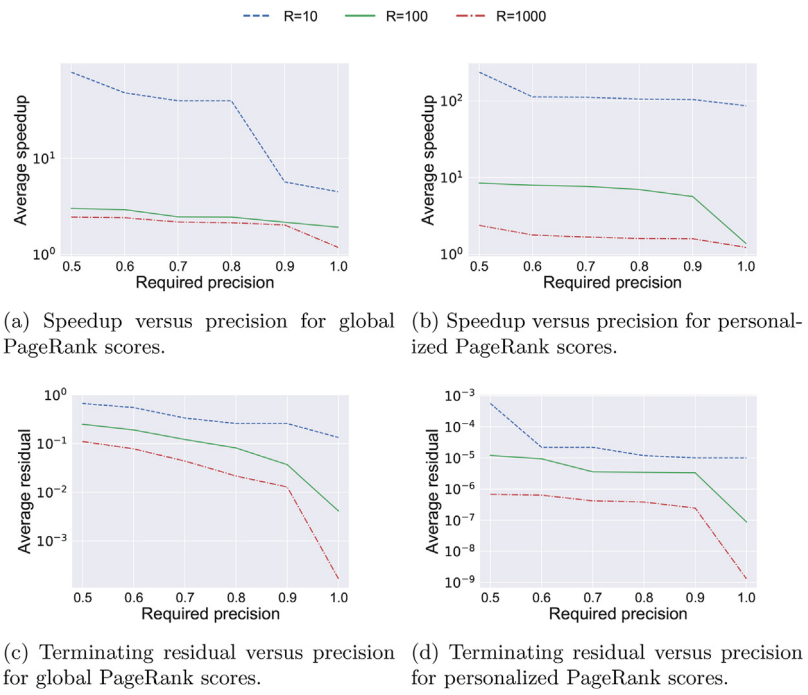
(c) Terminating residual versus precision for global PageRank scores.



(d) Terminating residual versus precision for personalized PageRank scores.

Fig. 6. Performance versus required precision for PageRank on undirected graphs.

Next we analyze the effect of our stopping criterion on PageRank. Here we use the theory from Theorem 2 for both undirected (Fig. 6) and directed (Fig. 7) graphs. Similar to the results for Katz Centrality earlier, we see higher speedups and lower terminating residuals for the personalized results (Fig. 6b and d) compared to their global counterparts (Fig. 6a and c). For PageRank, however, the speedups in the personalized case are considerably higher than the respective global ones. We also see similar trends of larger speedups and higher terminating residuals for smaller values of R. Note that in Fig. 6a and b there are regions in the plot where the speedup for $R = 10$ is less than the speedup for $R = 100$ (for the same precision). This is likely due to the behavior of the ranking vector for these values. For example, if the centrality values of vertices in the top 10–20 vertices are very similar, our stopping criterion would have to iterate further in order to obtain that required gap of $2\epsilon_k$. Likewise, if the values for vertices 100 and 101 are very far apart and the gap is found almost immediately, then the stopping criterion will be able to terminate sooner. This behavior of the centrality vector would lead to cases where speedup for higher values of R is greater than that of lower values of R. Finally, we examine our



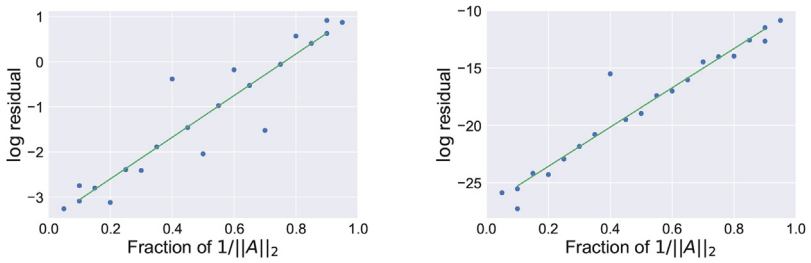
(a) Speedup versus precision for global PageRank scores.

(b) Speedup versus precision for personalized PageRank scores.

(c) Terminating residual versus precision for global PageRank scores.

(d) Terminating residual versus precision for personalized PageRank scores.

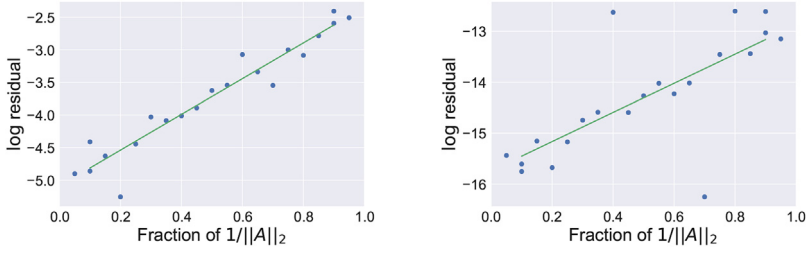
Fig. 7. Performance versus required precision for PageRank on directed graphs.



(a) Global Katz scores.

(b) Personalized Katz scores.

Fig. 8. Terminating residual obtained as we increase α for Katz scores in undirected graphs.



(a) Global Katz scores.

(b) Personalized Katz scores.

Fig. 9. Terminating residual obtained as we increase α for Katz scores in directed graphs.

stopping criterion on PageRank for directed graphs. Like Katz Centrality on directed graphs, the terminating residual (both global and personalized rankings) stays relatively constant for a required precision between 0.5–0.8 or 0.9 and then sharply drops for a required precision of 1.0.

5.3. *Effect of stopping criterion on harder problems*

Finally we investigate on what problems our method proves to be the most useful. For these results, we focus our analysis exclusively on Katz Centrality. We know as $\alpha \to \frac{1}{||A||_2}$, the problem becomes more ill-conditioned and typically requires more iterations to converge to machine precision. Since $\alpha \in (0, 1/||A||_2)$, we apply our stopping criterion to the different graphs for various α in this range. Fig. 8 plots the relationship between α and the residual norm obtained when the solver terminates using our criterion for undirected graphs for global (Fig. 8a) and personalized (Fig. 8b) rankings. The blue scatterplot points show the averaged values and the green line in the plots is a line fitted using regression analysis. We use values of $\alpha \in \{\frac{.05}{||A||_2}, \frac{.1}{||A||_2}, \ldots, \frac{.95}{||A||_2}\}$. For each value of α, the log of the averaged residual norm obtained upon termination using our stopping criterion is plotted across graphs. All results are aver-

aged over values of $R = 10$, 100, and 1000 and over all the graphs. When running to machine precision, the residual norm upon termination is typically $r_k \approx 10^{-15}$, but we see that we never have to iterate until machine precision using our new stopping criterion if we are interested in only the top vertices in a graph. Regression analysis of these results shows a strong linear correlation with a slope of 4.617 and mean sum of squares of 0.724 for the global values and a slope of 17.110 and mean sum of squares of 0.862 for the personalized values. We repeat the same analysis for the directed networks in Fig. 9, with the global results plotted in Fig. 9a and the personalized results plotted in Fig. 9b. The slope of the line plotted for the global results is 2.74 with a mean sum of squares of 0.804 and the slope for the personalized results is 2.86 with a mean sum of squares of 0.544. The linear relationship suggests that we need less accurate approximate solutions for harder problems as $\alpha \to \frac{1}{||A||_2}$ to obtain the top vertices in the graph. Typically the harder problems tend to take thousands of iterations to converge with the standard stopping criterion of iterating until a residual norm of 10^{-15}, but with our stopping criterion we can converge faster at a lower tolerance to solve the desired data mining task for the global scores. The low residual norm suggests we are able to certify the top R correctly with low fidelity solutions and we are able to use this technique to turn harder linear algebra problems into easier data mining problems.

6. Conclusions

This work bridges the two research areas of numerical accuracy of solvers and network analysis by understanding how the error in a solver affects the data analysis problem of ranking. We extended our previous work of certifying ranking in undirected graphs using global Katz scores by developing additional theory to guarantee ranking using PageRank. Additionally, we show our theory holds for directed graphs and furthermore compare results for global versus personalized centrality scores. We turned the data analysis problem of ranking vertices in graph into the numerical problem of understanding accuracy in a linear solver. This allows us to provide guarantees as to how accurate of a solution to the numerical problem we need to certify highly ranked vertices in graphs. We provided theoretical guarantees to bound the error in an approximate solution from an iterative method to the exact centrality scores (either using Katz or PageRank) of vertices and are able to identify the most central vertices with high confidence. Using the theory and error analysis, we developed a new stopping criterion that can be used in conjunction with any iterative solver to determine when to terminate given a desired number of highly ranked vertices with some preset precision, where the precision provides a bound on how many false positives we will tolerate being returned. With our new stopping criterion, we see a reduction in the number of iterations taken to solve the data analysis problem of ranking while maintaining a high precision rate in identifying top vertices. In fact, for personalized PageRank scores we obtain speedups of several orders of magnitude. As evidenced by the close relationship between the theory for Katz Centrality and PageRank, the results from this paper can be applied to any linear solver based ranking. Identifying top ranked vertices by Katz Centrality or PageRank are just two examples in practice presented in this work, but the theory is generalizable to other linear algebra based ranking metrics.

Acknowledgements

Eisha Nathan is in part supported by the National Physical Science Consortium Graduate Fellowship. The work depicted in this paper was sponsored in part by the National Science Foundation under award #1339745. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. This work was in part performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 with release number LLNL-JRNL-739840.

References

[1] D.A. Spielman, Algorithms, graph theory, and linear equations in Laplacian matrices, in: Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (in 4 volumes) vol. I: Plenary Lectures and Ceremonies vols. II–IV: Invited Lectures, World Scientific, 2010, pp. 2698–2722.
[2] R. Albert, H. Jeong, A.-L. Barabási, The Diameter of the World Wide Web, 1999 arXiv preprint cond-mat/9907038.
[3] O. Livne, A. Brandt, Lean algebraic multigrid (LAMG): fast graph Laplacian linear solver, SIAM J. Sci. Comput. 34 (4) (2012) B499–B522.
[4] E. Nathan, G. Sanders, J. Fairbanks, V.E. Henson, D. Bader, Graph ranking guarantees for numerical approximations to Katz centrality, Proc. Comput. Sci. 108 (2017) 68–78.
[5] M. Benzi, C. Klymko, Total communicability as a centrality measure, J. Complex Netw. 1 (2) (2013) 124–149.
[6] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.
[7] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, 1999.
[8] U. Brandes, C. Pich, Centrality estimation in large networks, Int. J. Bifurc. Chaos 17 (07) (2007) 2303–2318.
[9] Y. Saad, Iterative Methods for Sparse Linear Systems, Siam, 2003.
[10] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[11] E. Kokiopoulou, J. Chen, Y. Saad, Trace optimization and eigenproblems in dimension reduction methods, Numer. Linear Algebr. Appl. 18 (3) (2011) 565–602.
[12] M. Benzi, E. Estrada, C. Klymko, Ranking hubs and authorities using matrix functions, Linear Algebr. Appl. 438 (5) (2013) 2447–2474.
[13] P. Bonacich, Some unique properties of eigenvector centrality, Soc. Netw. 29 (4) (2007) 555–564.
[14] E. Estrada, J. Rodriguez-Velazquez, Subgraph centrality in complex networks, Phys. Rev. E 71 (5) (2005) 056103.
[15] S. White, P. Smyth, Algorithms for estimating relative importance in networks, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 266–275.
[16] K. Foster, S. Muth, J. Potterat, R. Rothenberg, A faster Katz status score algorithm, Comput. Math. Organ. Theory 7 (4) (2001) 275–285.
[17] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Am. Soc. Inf. Sci. Technol. 58 (7) (2007) 1019–1031.
[18] F. Bonchi, P. Esfandiar, D. Gleich, C. Greif, L. Lakshmanan, Fast matrix computations for pairwise and columnwise commute times and Katz scores, Internet Math. 8 (1-2) (2012) 73–112.
[19] R. Varga, Gershgorin and his Circles in Springer Series in Computational Mathematics, vol. 36, 2004.
[20] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[21] M. Hochstenbach, Probabilistic upper bounds for the matrix two-norm, J. Sci. Comput. 57 (3) (2013) 464–476.
[22] J. Kunegis, Konect: the Koblenz Network Collection, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 1343–1350.

Eisha Nathan is a PhD candidate in Computational Science and Engineering (CSE) at Georgia Institute of Technology in Atlanta, GA. She received her Master's degree (2017) in CSE from Georgia Tech and dual Bachelor's degrees in Computer Engineering and Mathematics (2014) from the University of Maryland in College Park (UMD). Her current research is focused on graph analysis and data mining. Her publication record includes several conference and journal papers in national and international proceedings.

Geoffrey Sanders is a Staff Scientist in the Computational Mathematics group at the Center for Applied Scientific Computing. Geoff's current research focus is on developing distributed graph analysis techniques for the efficient computation of challenging data mining tasks that involve topology and metadata jointly. A native of Reno, Nevada, Geoffrey earned his Bachelor's degree in Mathematics at the University of California, San Diego in 2002, his Master's in Applied Mathematics at CU Boulder in 2005, and his PhD at the same institution in 2008.

Van Emden Henson is a staff researcher in the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory (LLNL). He received his Ph.D. (1990) and M.S. (1988) in Applied Mathematics from the University of Colorado-Denver. His current research is focused on linear algebra, computational linear algebra, graph analysis, and data mining, while his earlier research focus was on multigrid and multilevel methods. In addition to three dozen or so journal and conference publications, he is a co-author of two books, namely, The DFT: an owner's manual for the discrete Fourier transform (1995) and A Multigrid Tutorial, 2nd edition (2000), both published by SIAM. Prior to joining LLNL Van was on the Mathematics faculty at the United States Naval Postgraduate School from 1990 to 1997. Van received B.S. degrees in both Geophysics and in Geology from the University of Utah in 1979 and spent most of the 1980s working as a seismologist for Cities Service Oil and Gas and for Occidental Petroleum.

David A. Bader is Professor and Chair of the School of Computational Science and Engineering, College of Computing, at Georgia Institute of Technology. He is a Fellow of the IEEE and AAAS and served on the White House's National Strategic Computing Initiative (NSCI) panel. Dr. Bader serves as a board member of the Computing Research Association, on the NSF Advisory Committee on Cyberinfrastructure, on the Council on Competitiveness High Performance Computing Advisory Committee, on the IEEE Computer Society Board of Governors, and on the Steering Committees of the IPDPS and HiPC conferences. He is the editor-in-chief of IEEE Transactions on Parallel and Distributed Systems, and is a National Science Foundation CAREER Award recipient. Dr. Bader is a leading expert in data sciences. His interests are at the intersection of high-performance computing and real-world applications, including cybersecurity, massive-scale analytics, and computational genomics, and he has co-authored over 210 articles in peer-reviewed journals and conferences.