Technical Report

TR-2012-003

Ranking Hubs and Authorities Using Matrix Functions

by

Michele Benzi, Ernesto Estrada, Christine Klymko

Mathematics and Computer Science

EMORY UNIVERSITY

RANKING HUBS AND AUTHORITIES USING MATRIX FUNCTIONS

MICHELE BENZI*, ERNESTO ESTRADA†, AND CHRISTINE KLYMKO‡

Abstract. The notions of subgraph centrality and communicability, based on the exponential of the adjacency matrix of the underlying graph, have been effectively used in the analysis of undirected networks. In this paper we propose an extension of these measures to directed networks, and we apply them to the problem of ranking hubs and authorities. The extension is achieved by *bipartization*, i.e., the directed network is mapped onto a bipartite undirected network with twice as many nodes in order to obtain a network with a symmetric adjacency matrix. We explicitly determine the exponential of this adjacency matrix in terms of the adjacency matrix of the original, directed network, and we give an interpretation of centrality and communicability in this new context, leading to a technique for ranking hubs and authorities. The matrix exponential method for computing hubs and authorities is compared to the well known HITS algorithm, both on small artificial examples and on more realistic real-world networks. The paper also discusses the use of Gaussian quadrature rules for calculating hub and authority scores.

Key words. hubs, authorities, centrality, communicability, directed networks, digraphs, bipartite graphs, Gauss quadrature

AMS subject classifications. 05C50, 15A16, 65F60, 90B10.

1. Introduction. In recent years, the study of networks has become central to many disciplines [3, 5, 6, 9, 10, 26, 27, 28]. Networks can be used to describe and analyze many different types of interactions, from those between people (social networks), to the flow of goods across an area (transportation networks), to links between websites (the WWW graph). In general, a network is a set of objects (nodes) and the connections between them (edges). Often, research is focused on determining and describing important characteristics of a network or of the interactions among its components.

One common question in network analysis is to determine the most "important" nodes (or edges) in the network, also called *node* or *vertex* (*edge*) *centrality*. The interpretation of what is meant by "important" can change from application to application. Due to this, many different measures of centrality have been developed. For an overview, see [5].

One notion of node centrality was introduced by Estrada and Rodríguez-Velázquez in [14]; see also the review article [13]. The methods described in [13] are only directly applicable to undirected networks. However, many important real-world networks (the World Wide Web, the Internet, citation networks, food webs, certain social networks, etc.) are directed. One goal of this paper is to extend the notions of centrality and communicability described in [11, 13] to directed networks, with an eye towards developing new ranking algorithms for, e.g., document collections, web pages, and so forth.

* Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (benzi@mathcs.emory.edu). The work of this author was supported by National Science Foundation grant DMS1115692.

†Department of Mathematics and Statistics and Institute of Complex Systems, University of Strathclyde, Glasgow G1 1XQ, UK (ernesto.estrada@strath.ac.uk). The work of this author was supported in part by the New Professor's Fund, University of Strathclyde, and by the project "Mathematics of Large Technological Evolving Networks (MOLTEN)" funded by the Engineering and Physical Sciences Research Council and by the Research Councils UK Digital Economy programme, grant ref. EP/I016058/1.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (cklymko@emory.edu).

We further compare our approach with some standard algorithms, such as HITS (see [20]). Methods of quickly determining hub and authority rankings using Gauss-type quadrature rules are also discussed.

2. Digraphs.

Here we briefly review some basic graph-theoretic notions; we refer to [8] for a comprehensive treatment. A *graph* $G = (V, E)$ is formed by a set of nodes (vertices) V and edges E formed by unordered pairs of vertices. Every network is naturally associated with a graph $G = (V, E)$ where $|V|$ is the number of nodes in the network and E is the collection of edges between objects, $E = \{(i, j) \,|\, \text{there is an edge between node } i \text{ and node } j\}$. The *degree* d_i of a vertex i is the number of edges incident to i.

A *digraph* $G = (V, E)$ is formed by a set of vertices V and edges E formed by ordered pairs of vertices. That is, $(i, j) \in E \not\Rightarrow (j, i) \in E$. In the case of digraphs, which model directed networks, there are two types of degree. The *in-degree* of node i is given by the number of edges which point to i. The *out-degree* is given by the number of edges pointing out from i.

Unless otherwise specified, every (di)graph in this paper is simple (unweighted with no multiple edges or loops) and connected.

The *adjacency matrix* of a graph is a matrix $A \in \mathbb{R}^{|V|\times|V|}$ defined in the following way:

$$A = (a_{ij}); \quad a_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is an edge in } G, \\ 0, & \text{else.} \end{cases}$$

Under the conditions imposed on G, A has zeros on the diagonal. If G is an undirected graph, A will be a symmetric matrix and the eigenvalues will be real. In the case of digraphs, A is not symmetric and may have complex (non-real) eigenvalues.

3. Eigenvector-based rankings.

Here we give a brief description of eigenvector-based ranking methods, mostly HITS and its variants. We don't discuss PageRank because it is very well known and also because it does not readily provide a way to discriminate between hubs and authorities. See [22] for a good survey of HITS, PageRank, SALSA, etc.

3.1. HITS algorithm.

A highly popular method for ranking hubs and authorities was developed by Kleinberg in the late 1990s. The Hyperlink-Induced Topic Search (HITS) algorithm is based on the idea that in the World Wide Web, and in many directed networks, there are two types of important nodes: hubs and authorities [20]. Hubs are nodes which point to many nodes of the type considered important. Authorities are these important nodes. From this comes a circular definition: good hubs are those which point to many good authorities and good authorities are those pointed to by many good hubs.

Thus, the HITS ranking relies on an iterative method converging to a stationary solution. Each node i in the network is assigned two non-negative weights, an *authority weight* x_i and a *hub weight* y_i. To begin with, each x_i and y_i is given an arbitrary nonzero value. Then, the weights are updated in the following ways:

$$x_i^{(k)} = \sum_{j:(j,i)\in E} y_j^{(k-1)} \quad \text{and} \quad y_i^{(k)} = \sum_{j:(i,j)\in E} x_j^{(k)} \quad \text{for} \quad k = 1, 2, 3... \tag{3.1}$$

The weights are then normalized so that $\sum_j (x_j^{(k)})^2 = 1$ and $\sum_j (y_j^{(k)})^2 = 1$.

The above iterations occur sequentially and it can be shown that, under mild conditions, both sequences of vectors $\{x^{(k)}\}$ and $\{y^{(k)}\}$ converge as $k \to \infty$. In practice, the iterative process is continued until there is no significant change between consecutive iterates.

This iteration sequence shows the natural dependence relationship between hubs and authorities: if a node i points to many nodes with large x-values, it receives a large y-value and, if it is pointed to by many nodes with large y-values, it receives a large x-value.

In terms of matrices, the equation (3.1) becomes: $x^{(k)} = A^T y^{(k-1)}$ and $y^{(k)} = Ax^{(k)}$, followed by normalization in the 2-norm. This iterative process can be expressed as

$$x^{(k)} = c_k A^T A x^{(k-1)} \quad \text{and} \quad y^{(k)} = c'_k A A^T y^{(k-1)}, \tag{3.2}$$

where c_k and c'_k are normalization factors. A typical choice for the inizialization vectors $x^{(0)}$, $y^{(0)}$ would be the constant vector

$$x^{(0)} = y^{(0)} = [1/\sqrt{n},\, 1/\sqrt{n},\, \ldots, 1/\sqrt{n}],$$

see [15]. Hence, HITS is just an iterative power method to compute the dominant eigenvector for $A^T A$ and for AA^T. The authority scores are determined by the entries of the dominat eigenvector of the matrix $A^T A$, which is called the *authority matrix* and the hub scores are determined by the entries of the dominant eigenvector of AA^T, called the *hub matrix*. Recall that the eigenvalues of both $A^T A$ and AA^T are the squares of the singular values of A. Also, the eigenvectors of $A^T A$ are the right singular vectors of A, and the eigenvectors of AA^T are the left singular vectors of A.

3.2. HITS reformulation. In a digraph the adjacency matrix A is generally nonsymmetric, however, the two matrices used in the HITS algorithm ($A^T A$ and AA^T) are symmetric. Note that, setting

$$\mathcal{A} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix},$$

a symmetric matrix is obtained. Now,

$$\mathcal{A}^2 = \begin{pmatrix} AA^T & 0 \\ 0 & A^T A \end{pmatrix}; \quad \mathcal{A}^3 = \begin{pmatrix} 0 & AA^T A \\ A^T AA^T & 0 \end{pmatrix}.$$

In general,

$$\mathcal{A}^{2k} = \begin{pmatrix} (AA^T)^k & 0 \\ 0 & (A^T A)^k \end{pmatrix}; \quad \mathcal{A}^{2k+1} = \begin{pmatrix} 0 & A(A^T A)^k \\ (A^T A)^k A^T & 0 \end{pmatrix}.$$

Applying HITS to this matrix $\mathcal{A}$, $\mathcal{A}^T = \mathcal{A}$ so $\mathcal{A}^T\mathcal{A} = \mathcal{A}\mathcal{A}^T = \mathcal{A}^2$ and introducing the vector $u^{(k)} = \begin{pmatrix} y^{(k)} \\ x^{(k)} \end{pmatrix}$ for $k = 1, 2, 3, \ldots$, equation (3.2) becomes

$$u^{(k)} = \mathcal{A}^2 u^{(k-1)} = \begin{pmatrix} AA^T & 0 \\ 0 & A^T A \end{pmatrix} u^{(k-1)}, \tag{3.3}$$

followed by normalization of the two vector components of $u^{(k)}$ so that each has 2-norm equal to 1. Now, if A is an $n \times n$ matrix, $\mathcal{A}$ is $2n \times 2n$ and vector $u^{(k)}$ is in $\mathbb{R}^{2n}$. The first n entries of $u^{(k)}$ correspond to the hub rankings of the nodes, while the last n entries give the authority rankings. Under suitable assumptions (see the discussion in [15]), as $k \to \infty$ the sequence $\{u^{(k)}\}$ converges to the dominant nonnegative eigenvector of $\mathcal{A}$, which yields the desired hub and authority rankings.

Hence, in HITS only information obtained from the dominant eigenvector of $\mathcal{A}$ is used. It is natural to expect that taking into account spectral information corresponding to the remaining eigenvalues and eigenvectors of $\mathcal{A}$ may lead to improved results.

3.3. HITS with exponentiated input. It is know that for certain networks, the HITS algorithm does not converge to unique hub and authority vectors (see [15], [22], [25] and references therein). If the dominant eigenvalue of A^TA (and, consequently of AA^T) is not simple, then the corresponding eigenspace is multidimensional. This means that the choice of the initial vector affects the convergence of the algorithm and different hub and authority vectors can be produced using different initial vectors. This can occur when A^TA is reducible, that is when the original network is not strongly connected.

In [15], Farahat *et. al.* propose a modification to the HITS algorithm in order to avoid this issue. Rather than running the HITS algorithm on the original adjacency matrix, A, the matrix $e^A - I$ is used. Both the initial hub and authority vectors are normalzied, positive vectors (e.g., constant vectors with all entries equal to $1/\sqrt{n}$)). The authors prove that, as long as the original network is weakly connected, the dominant eigenvalue of $(e^A - I)^T(e^A - I)$ is simple. Thus, HITS with this exponentiated input produces unique hub and authority rankings.

This can been seen in the following way: let G be the original network with n nodes, t of which have an in-degree of 0. The network can be relabeled such that nodes 1 through $n-t$ have a positive in-degree and nodes $n-t+1$ through n have zero in-degree. Now, the adjacency matrix can be written as:

$$A = \begin{pmatrix} \tilde{A} & 0 \\ \tilde{B} & 0 \end{pmatrix}$$

where $\tilde{A}$ is $(n-t) \times (n-t)$ and $\tilde{B}$ is $t \times (n-t)$.

When running HITS with exponentiated input, the matrix used is $e^A - I$ and the authority matrix becomes

$$(e^A - I)^T(e^A - I) = \begin{pmatrix} (e^{\tilde{A}} - I)^T(e^{\tilde{A}} - I) & 0 \\ 0 & 0 \end{pmatrix}.$$

This has both the advantage of ensuring a unique output for the HITS algorithm whenever the original network is weakly connected, and giving the t nodes with 0 in-degree an authority ranking of 0. However, the authority rankings are based on the entries of the eigenvector corresponding to the dominant eigenvector of

$$\begin{pmatrix} (e^{\tilde{A}} - I)^T(e^{\tilde{A}} - I) & 0 \\ 0 & 0 \end{pmatrix}.$$

Thus, the out-edges of the nodes which are not part of $\tilde{A}$ are less important in the calculation of authority scores than other edges. In networks where there are few nodes with an in-degree of zero or where the edges from these nodes are somewhat evenly distributed, this has little to no effect on the rankings produced using HITS.

However, when there are many nodes with 0 in-degree or their edges point to only a few other nodes, dropping these edges can greatly affect the rankings (see sections 5.4 and 5.5 for examples).

An obvious disadvantage of this algorithm is its cost, since it requires iterating with a matrix exponential and its transpose. It can be implemented using only matrix-vector products involving $\tilde{A}$ and $\tilde{A}^T$ by means of techniques, like Krylov subspace methods, for evaluating the action of a matrix function on a given vector; see, e.g., [18, Chapter 13]. This approach leads to a nested iteration scheme, with HITS as the outer iteration and the Krylov method as the inner one.

4. Subgraph centralities and communicabilities. In [13], the authors review several measures to rank the nodes in an undirected network A. The *subgraph centrality* [14] of node i is given by $[e^A]_{ii}$ and the *communicability* [11] between nodes i and j $(i \neq j)$ is given by $[e^A]_{ij}$. Nodes i corresponding to higher values of $[e^A]_{ii}$ are considered more important than nodes corresponding to lower values. Large values of $[e^A]_{ij}$ indicate that information flows more easily between nodes i and j than between pairs of nodes corresponding to lower values of the same quantity. The *Estrada index* of the graph is given by $\mathrm{Tr}\,(e^A) = \sum_{i=1}^n [e^A]_{ii}$. This index, which provides a global characterization of a network, is analogous to the *partition function* in statistical mechanics and plays an important role in the study of networks at the *macroscopic* level: quantities such as the *natural connectivity*, the *total energy*, the *Helmholtz free energy* and the *entropy* of a network can all be expressed in terms of the Estrada index [12]. Also, normalization of the diagonal entries of e^A by $\mathrm{Tr}\,(e^A)$ yields a probability distribution on the nodes of the network, which can be regarded as an analogue of the PageRank distribution.

Consider the power series expansion of e^A,

$$e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots + \frac{A^k}{k!} + \cdots \tag{4.1}$$

From graph theory, it is well known that if A is the adjacency matrix of an undirected graph, $[A^k]_{ij} = [A^k]_{ji}$ counts the number of walks of length k between nodes i and j. Thus, the subgraph centrality of node i, $[e^A]_{ii}$, counts the total number of closed walks starting at node i, penalizing longer walks by scaling walks of length k by the factor $\frac{1}{k!}$. The communicability between nodes i and j, $[e^A]_{ij}$, counts the number of walks between nodes i and j, again scaling walks of length k by a factor of $\frac{1}{k!}$.

Although the matrix exponential is certainly well-defined for any matrix, whether symmetric or not, extending the notions of subgraph centrality and communicability to directed networks is not straightforward. Moreover, computational difficulties may arise. While the computations involved do not pose a problem for small networks, many real-world networks are large enough that directly computing the exponential of the adjacency matrix is prohibitive. In [1], Benzi and Boito discuss techniques for bounding and estimating individual entries of the matrix exponential using Gaussian quadrature rules; see also [4] and section 8 below. The ability to find upper and lower bounds for the entries requires that the matrix be symmetric, thus these bounds cannot be directly computed using the adjacency matrix of a directed network. Moreover, in the directed case it is not immediately clear how to interpret the notions of subgraph centrality, communicability and Estrada index. These difficulties can be circumvented using a bipartite graph model, as discussed next.

5. Extension to digraphs. Although the techniques described in [1] cannot be directly applied to non-symmetric matrices, setting

$$\mathcal{A} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \tag{5.1}$$

produces a symmetric matrix $\mathcal{A}$ and, thus, upper and lower bounds of individual entries of $e^{\mathcal{A}}$ can be computed. In Proposition 1 below we relate $e^{\mathcal{A}}$ to the underlying hub and authority structure of the original digraph. By $B^\dagger$ we denote the Moore–Penrose generalized inverse of matrix B.

PROPOSITION 1. *Let $\mathcal{A}$ be as described in equation (5.1). Then,*

$$e^{\mathcal{A}} = \begin{pmatrix} \cosh\left(\sqrt{AA^T}\right) & A\left(\sqrt{A^TA}\right)^\dagger \sinh\left(\sqrt{A^TA}\right) \\ \sinh\left(\sqrt{A^TA}\right)\left(\sqrt{A^TA}\right)^\dagger A^T & \cosh\left(\sqrt{A^TA}\right) \end{pmatrix}.$$

Proof. Let $A = U\Sigma V^T$ be the SVD of the original (non-symmetric) adjacency matrix A. Then, $\mathcal{A}$ can be decomposed as $\mathcal{A} = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}\begin{pmatrix} 0 & \Sigma \\ \Sigma & 0 \end{pmatrix}\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix}$. Hence,

$$e^{\mathcal{A}} = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix} \exp\begin{pmatrix} 0 & \Sigma \\ \Sigma & 0 \end{pmatrix}\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix}. \tag{5.2}$$

Now,

$$\exp\begin{pmatrix} 0 & \Sigma \\ \Sigma & 0 \end{pmatrix} = \cosh\begin{pmatrix} 0 & \Sigma \\ \Sigma & 0 \end{pmatrix} + \sinh\begin{pmatrix} 0 & \Sigma \\ \Sigma & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \cosh(\Sigma) & 0 \\ 0 & \cosh(\Sigma) \end{pmatrix} + \begin{pmatrix} 0 & \sinh(\Sigma) \\ \sinh(\Sigma) & 0 \end{pmatrix}.$$

Thus,

$$\exp\begin{pmatrix} 0 & \Sigma \\ \Sigma & 0 \end{pmatrix} = \begin{pmatrix} \cosh(\Sigma) & \sinh(\Sigma) \\ \sinh(\Sigma) & \cosh(\Sigma) \end{pmatrix}. \tag{5.3}$$

Putting together equations (5.2) and (5.3),

$$e^{\mathcal{A}} = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}\begin{pmatrix} \cosh(\Sigma) & \sinh(\Sigma) \\ \sinh(\Sigma) & \cosh(\Sigma) \end{pmatrix}\begin{pmatrix} U^T & 0 \\ 0 & V^T \end{pmatrix}$$

$$= \begin{pmatrix} \cosh\left(\sqrt{AA^T}\right) & A\left(\sqrt{A^TA}\right)^\dagger \sinh\left(\sqrt{A^TA}\right) \\ \sinh\left(\sqrt{A^TA}\right)\left(\sqrt{A^TA}\right)^\dagger A^T & \cosh\left(\sqrt{A^TA}\right) \end{pmatrix}.$$

The identities involving the off-diagonal blocks can be easily checked using the SVD of A. ∎

5.1. Interpretation of diagonal entries. In the context of undirected networks, the interpretation of the entries of the matrix exponential in terms of subgraph centralities and communicabilities is well-established, see e.g. [13]. In the case of directed networks and $e^{\mathcal{A}}$, things are not as clear. The network behind $\mathcal{A}$ can be thought of as follows: take the vertices from the original network A and make two copies of them, V and V'. Then, undirected edges exist between the two sets based on the following rule: $E' = \{(i, j') |$ there is a directed edge from i to j in the original network$\}$. This creates a bipartite graph with $2n$ nodes: $1, 2, \ldots, n, n+1, n+2, \ldots, 2n$. We denote by $V(\mathcal{A})$ this set of nodes.

In the undirected case, each node had only one role to play in the network: any information that came into the node could leave by any edge. In the directed case, there are two roles for each node: that of a hub and that of an authority. It is unlikely that a high ranking hub will also be a high ranking authority, but each node can still be seen as acting in both of these roles. In the network $\mathcal{A}$, the two aspects of each node are separated. Nodes $1, 2, \ldots, n$ in $V(\mathcal{A})$ represent the original nodes in their role as hubs and nodes $n+1, n+2, \ldots, 2n$ in $V(\mathcal{A})$ represent the original nodes in their role as authorities.

Given a directed network, an *alternating walk* of length k, starting with an out-edge, from node v_1 to node v_{k+1} is a list of nodes $v_1, v_2, ..., v_{k+1}$ such that there exists edge (v_i, v_{i+1}) if i is odd and edge (v_{i+1}, v_i) if i is even:

$$v_1 \rightarrow v_2 \leftarrow v_3 \rightarrow \cdots$$

An *alternating walk* of length k, starting with an in-edge, from node v_1 to node v_{k+1} in a directed network is a list of nodes $v_1, v_2, ..., v_{k+1}$ such that there exists edge (v_{i+1}, v_i) if i is odd and edge (v_i, v_{i+1}) if i is even:

$$v_1 \leftarrow v_2 \rightarrow v_3 \leftarrow \cdots$$

From graph theory (see also [7]), it is known that $[AA^TA\ldots]_{ij}$ (where there are k matrices being multiplied) counts the number of alternating walks of length k, starting with an out-edge, from node i to node j, whereas $[A^TAA^T\ldots]_{ij}$ (where there are k matrices being multiplied) counts the number of alternating walks of length k, starting with an in-edge, from node i to node j. That is, $[(AA^T)^k]_{ij}$ and $[(A^TA)^k]_{ij}$ count the number of alternating walks of length $2k$.

In the original network A, if node i is a good hub, it will point to many good authorities, which will in turn be pointed at by many hubs. These hubs will also point to many authorities, which will again be pointed at by many other hubs. Thus, if i is a good hub, it will show up many times in the sets of hubs described above. That is, there should be many even length alternating walks, starting with an out-edge, from node i to itself. Giving a walk of length $2k$ a weight of $\frac{1}{(2k)!}$, these walks can be counted using the (i,i) entry of the matrix

$$I + \frac{AA^T}{2!} + \frac{AA^TAA^T}{4!} + \cdots + \frac{(AA^T)^k}{(2k)!} + \cdots$$

Letting $A = U\Sigma V^T$ be the SVD of A, this becomes:

$$U\left(I + \frac{\Sigma^2}{2!} + \frac{\Sigma^4}{4!} + \cdots + \frac{\Sigma^{2k}}{(2k)!} + \cdots\right)U^T$$

$$= U\cosh(\Sigma)U^T = \cosh(\sqrt{AA^T})\,.$$

The *hub centrality* of node i (in the original network) is thus given by

$$[e^{\mathcal{A}}]_{ii} = [\cosh(\sqrt{AA^T})]_{ii}.$$

This measures how well node i transmits information to the authoritative nodes in the network.

Similarly, if node i is a good authority, there will be many even length alternating walks, starting with an in-edge, from node i to itself. Giving a walk of length $2k$ a weight of $\frac{1}{(2k)!}$, these walks can be counted using the (i,i) entry of $\cosh(\sqrt{A^TA})$.

Hence, the *authority centrality* of node i is given by

$$[e^{\mathcal{A}}]_{n+i,n+i} = [\cosh(\sqrt{A^TA})]_{ii}.$$

It measures how well node i receives information from the hubs in the network.

Note that the traces of the two diagonal blocks in $e^{\mathcal{A}}$ are identical, so each accounts for half of the Estrada index of the bipartite graph. Also note that denoting by $\sigma_1, \dots, \sigma_n$ the singular values of A, we have

$$\mathrm{Tr}\,(e^{\mathcal{A}}) = 2\sum_{i=1}^{n} \cosh(\sigma_i) = \sum_{i=1}^{n} e^{\sigma_i} + \sum_{i=1}^{n} e^{-\sigma_i}.$$

5.2. Interpretation of off-diagonal entries. In discussing the off-diagonal entries of $\mathcal{A}$, there are three blocks to consider. First, there are the off-diagonal entries of the upper-left block, $\cosh(\sqrt{AA^T})$, then there are the off-diagonal entries of the lower-right block, $\cosh(\sqrt{A^TA})$. Finally, there is the off-diagonal block, $A\left(\sqrt{A^TA}\right)^{\dagger}\sinh\left(\sqrt{A^TA}\right)$ (the fourth block in $e^{\mathcal{A}}$ being its transpose).

From section 5.1, $[e^{\mathcal{A}}]_{ij} = [\cosh(\sqrt{AA^T})]_{ij}$, $1 \le i,j \le n$, counts the number of even length alternating walks, starting with an out-edge, from node i to node j, weighting walks of length $2k$ by a factor of $\frac{1}{(2k)!}$. When $i \neq j$, these entries measure how similar nodes i and j are as hubs. That is, if nodes i and j point to many of the same nodes, there will be many short even length alternating walks between them.

The *hub communicability* between nodes i and j, $1 \le i,j \le n$ is given by

$$[e^{\mathcal{A}}]_{ij} = [\cosh(\sqrt{AA^T})]_{ij}$$

This measures how similar nodes i and j are in their roles as hubs. That is, a larger value of hub communicability between nodes i and j indicates that they point to many of the same authorities. In other words, they point to nodes which are authorities on the same subjects.

Similarly, $[e^{\mathcal{A}}]_{n+i,n+j} = [\cosh(\sqrt{A^TA})]_{ij}$, $1 \le i,j \le n$ counts the number of even length alternating walks, starting with an in-edge, from node i to node j, also weighing walks of length $2k$ by a factor of $\frac{1}{(2k)!}$. When $i \neq j$, these entries measure how similar the two nodes are as authorities. If i and j are pointed at by many of the same hubs, there will be many short even length alternating walks between them.

The *authority communicability* between nodes i and j, $1 \le i, j, \le n$ is given by

$$[e^{\mathcal{A}}]_{i+n,j+n} = [\cosh(\sqrt{A^TA})]_{ij}$$

This measures how similar nodes i and j are in their roles as authorities. That is, a larger value of authority communicability between nodes i and j means that they are

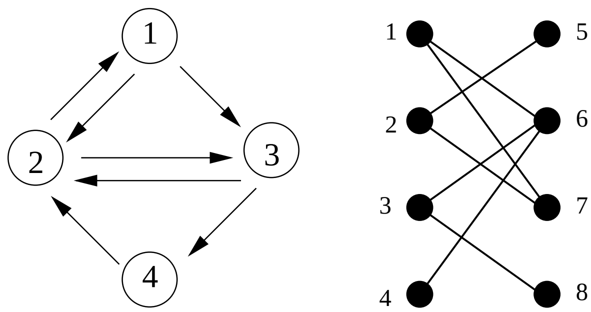

FIG. 5.1. *The original directed network from Example 1, with adjacency matrix A (left) and the bipartite network with adjacency matrix $\mathcal{A}$ (right).*

pointed to by many of the same hubs and, as such, are likely to contain information on the same subjects.

Let us now consider the off-diagonal blocks of $\mathcal{A}$. Here, $[\sinh(\sqrt{A^TA})]_{ij}$ counts the number of odd length alternating walks, starting with an out-edge, from node i to node j, weighing walks of length $2k+1$ by $\frac{1}{(2k+1)!}$. This measures the communicability between node i as a hub and node j as an authority.

The *hub-authority communicability* between nodes i and j (that is, the communicability between node i as a hub and node j as an authority) is given by:

$$[e^{\mathcal{A}}]_{i,n+j} = [A\left(\sqrt{A^TA}\right)^{\dagger}\sinh\left(\sqrt{A^TA}\right)]_{ij}$$

$$= [\sinh\left(\sqrt{A^TA}\right)\left(\sqrt{A^TA}\right)^{\dagger}A^T]_{ji} = [e^{\mathcal{A}}]_{n+j,i}.$$

A large hub-authority communicability between nodes i and j means that they are likely in the same "part" of the directed network: node i tends to point to nodes that contain information similar to that on which node j is an authority.

5.3. Small examples. In this section we illustrate the proposed method on some simple networks of small size. We also compare our approach with HITS and, when warranted, with its exponentiated input variant.

5.3.1. Example 1. Consider the small directed network in Fig. 5.1 (left panel). The adjacency matrix is given by

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The corresponding bipartite graph is shown in Fig. 5.1 (right panel). If hubs and authorities are determined simply using in-degree and out-degree counts, the result is as follows:

node	out-degree	in-degree
1	2	1
2	2	3
3	2	2
4	1	1

Under this ranking, the hub ranking of the nodes is: $\{1, 2, 3\,(\text{tie}); 4\}$. The authority ranking of the nodes is: $\{2; 3; 1, 4\,(\text{tie})\}$. We obtain somewhat different results using the HITS algorithm. The eigenvectors of AA^T and A^TA corresponding to the largest eigenvalue $\lambda_{\max} \approx 3.9563$, which is simple, yield the following rankings for hubs and authorities:

node	hub rank	authority rank
1	.6555	.1685
2	.3351	.8058
3	.5422	.4980
4	.4051	.2726

Here, the ranking for hubs is: $\{1; 3; 4; 2\}$. The ranking for authorities is: $\{2; 3; 4; 1\}$. Note that node 2, which was given a top hub score by looking just at the out-degrees, is judged by HITS as the node with the lowest hub score.

Using $e^{\mathcal{A}}$ as described above, the rankings for hub centralities and authority centralities are:

node	hub centrality = $[e^{\mathcal{A}}]_{ii}$	authority centrality = $[e^{\mathcal{A}}]_{4+i,4+i}$
1	2.3319	1.5906
2	2.2289	3.0209
3	2.2812	2.2796
4	1.6414	1.5922

With this method, the hub ranking of the nodes is: $\{1; 3; 2; 4\}$. The authority ranking is: $\{2; 3; 4; 1\}$. On this example, our method produces the same authority ranking as HITS. The hub ranking, however, is slightly different: both methods identify node 1 as the one with the highest hub score, followed by node 3; however, our method assigns the lowest hub score to node 4 rather than node 2. This is arguably a more meaningful ranking.

5.3.2. Example 2. Consider the small directed network in Fig. 5.2 (left panel). The adjacency matrix is given by

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The corresponding bipartite graph is shown in Fig. 5.2 (right panel). If hubs and authorities are determined only using in-degrees and out-degrees, the result is as follows:

node	out-degree	in-degree
1	1	1
2	2	2
3	1	1
4	1	1

Under this criterion, the hub ranking of the nodes is: $\{2; 1, 2, 3\,(\text{tie})\}$ and the authority ranking is: $\{2; 1, 2, 3\,(\text{tie})\}$. While it is intuitive that node 2 should be given a high score (both as an authority and as a hub), just looking at the degrees does not allow one to distinguish the remaining nodes.

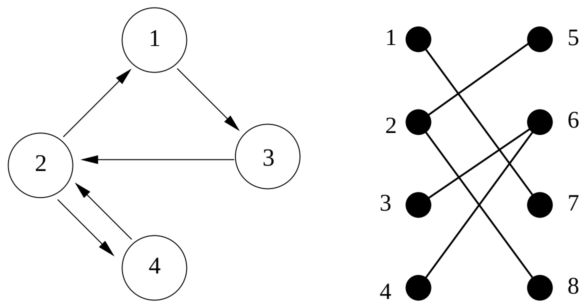

FIG. 5.2. *The original directed network from Example 2, with adjacency matrix A (left) and the bipartite network with adjacency matrix $\mathcal{A}$ (right).*

Consider now the use of HITS. The largest eigenvalue of AA^T (and A^TA) is $\lambda_{\max} = 2$ and it has multiplicity two. Thus, different starting vectors for the HITS algorithm may produce different rankings, as discussed in [15]. Starting from a constant authority vector $x^{(0)}$, as suggested in [20], produces the following scores:

node	hub rank	authority rank
1	.0002	.3332
2	.4999	.3332
3	.2499	.0003
4	.2499	.3332

The ranking for hubs is: $\{2; 3, 4\,(\text{tie}); 1\}$. The ranking for authorities is the following: $\{1, 2, 4\,(\text{tie}); 3\}$. The fact that node 2 has a higher in-degree than any of the other nodes does not affect its authority ranking under the HITS algorithm, which shows a clear drawback of HITS.

If the ranking is determined using $e^{\mathcal{A}}$ as described above, the resulting scores are:

node	hub centrality = $[e^{\mathcal{A}}]_{ii}$	authority centrality = $[e^{\mathcal{A}}]_{4+i,4+i}$
1	1.5431	1.5891
2	2.1782	2.1782
3	1.5891	1.5431
4	1.5891	1.5891

With this method, the hub ranking of the nodes is the same as in HITS: $\{2; 3, 4\,(\text{tie}); 1\}$. However, in the authority ranking, node 2 is the clear winner rather than being part of a three-way tie for first place: $\{2; 1, 4\,(\text{tie}); 3\}$. Identical rankings are obtained by the exponentiated input HITS algorithm.

5.4. Example 3. Let G be the small directed network in Fig. 5.3. The adjacency matrix is given by

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

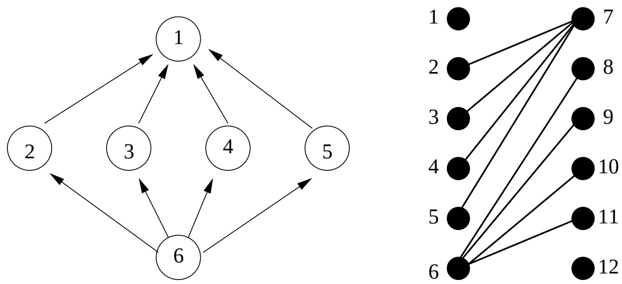

FIG. 5.3. *The original directed network from Example 3, with adjacency matrix A (left) and the bipartite network with adjacency matrix $\mathcal{A}$ (right).*

If hubs and authorities are determined using only in-degrees and out-degrees, the result is:

node	out-degree	in-degree
1	0	4
2	1	1
3	1	1
4	1	1
5	1	1
6	4	0

The hub ranking of the nodes using degrees is: {6; 2,3,4,5 (tie); 1}. The authority ranking is {1; 2,3,4,5 (tie); 6}.

If the HITS algorithm is used, the resulting rankings are similar, but not exactly the same. Starting with a constant authority vector $x^{(0)}$, the results are:

node	hub rank	authority rank
1	.000	.200
2	.125	.200
3	.125	.200
4	.125	.200
5	.125	.200
6	.500	.000

The hub ranking of the nodes is: $\{6; 2, 3, 4, 5\,(\text{tie}); 1\}$. The authority ranking is: {1,2,3,4,5 (tie); 6}. Here, HITS is unable to differentiate between node 1 and nodes 2, 3, 4, and 5 in terms of the authority score, even though node 1 has by far the highest in-degree. This appears as a failure of HITS, since it is intuitive that node 1 should be regarded as very authoritative. If HITS with exponentiated input is used, node 1 does get a higher authority score than all the other nodes:

node	Exp. input HITS hub rank	Exp. input HITS authority rank
1	.0000	.4472
2	.1382	.1382
3	.1382	.1382
4	.1382	.1382
5	.1382	.1382
6	.4472	.0000

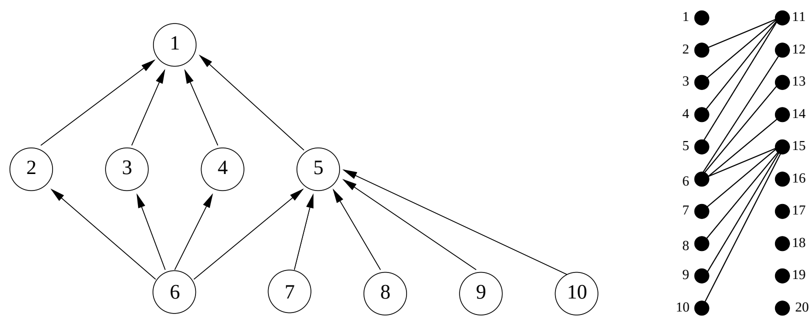

FIG. 5.4. *The original directed network from Example 4, with adjacency matrix A (left) and the bipartite network with adjacency matrix $\mathcal{A}$ (right).*

Under HITS with exponentiated input, the hub ranking is the same one obtained by HITS, namely, {6; 2,3,4,5 (tie); 1}, while the authority ranking is: {1; 2,3,4,5 (tie); 6}.

For this network, using $e^{\mathcal{A}}$ to calculate the hub and authority scores yields the same rankings as HITS with exponentiated input:

node	hub centrality = $[e^{\mathcal{A}}]_{ii}$	authority centrality = $[e^{\mathcal{A}}]_{n+i,n+i}$
1	1.0000	3.7622
2	1.6905	1.6905
3	1.6905	1.6905
4	1.6905	1.6905
5	1.6905	1.6905
6	3.7622	1.0000

Note that, if desired, the value 1 can be subtracted from these scores since it does not affect the relative ranking of the nodes. The hub ranking is {6; 2,3,4,5 (tie); 1}, and the authority ranking is: {1; 2,3,4,5 (tie); 6}.

One may ask under which conditions, if any, the exponentiated input HITS method and the one based on $e^{\mathcal{A}}$ result in different rankings. As we will see in the next example, when there are many nodes with 0 in-degree, the rankings using $e^{\mathcal{A}}$ start to differ from those using HITS with exponentiated input.

5.5. Example 4. Let G be the 10-node directed network in Fig. 5.4. The adjacency matrix of G is given by:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Using just in-degree and out-degree, the hub and authority rankings of G are as follows:

node	out-degree	in-degree
1	0	4
2	1	1
3	1	1
4	1	1
5	1	5
6	4	0
7	1	0
8	1	0
9	1	0
10	1	0

The hub ranking is: {6; 2,3,4,5,7,8,9,10 (tie); 1}. The authority ranking is {5; 1; 2,3,4 (tie); 6,7,8,9,10 (tie)}. Using HITS, different rankings are produced. Starting with a constant authority vector $x^{(0)}$, the rankings produced are:

node	hub rank	authority rank
1	.0000	.0008
2	.0003	.1665
3	.0003	.1665
4	.0003	.1665
5	.0003	.4996
6	.3330	.0000
7	.1665	.0000
8	.1665	.0000
9	.1665	.0000
10	.1665	.0000

The hub ranking is: {6; 7,8,9,10 (tie); 2,3,4,5 (tie); 1}. The authority ranking is: {5; 2,3,4 (tie); 1; 6,7,8,9,10 (tie)}. Since nodes 2, 3, 4, and 5 are given such low hub scores, node 1 is deemed an extremely low authority, even though it has an in-degree of 4.

By comparison, when using HITS with exponentiated input, node 1 is considered very authoritative:

node	Exp. input HITS hub rank	Exp. input HITS authority rank
1	.0000	.4196
2	.0849	.1095
3	.0849	.1095
4	.0849	.1095
5	.0849	.2519
6	.2871	.0000
7	.0934	.0000
8	.0934	.0000
9	.0934	.0000
10	.0934	.0000

The hub ranking is: {6; 7,8,9,10 (tie); 2,3,4,5 (tie); 1}, identical to HITS. The authority ranking is: {1; 5; 2,3,4 (tie); 6,7,8,9,10 (tie)}. Here, even though node 5 has a higher in-degree than node 1, node 1 is given a higher authority rank.

On the other hand, using the rankings from $e^{\mathcal{A}}$, node 5 is considered a higher authority than node 1:

node	hub centrality = $[e^{\mathcal{A}}]_{ii}$	authority centrality = $[e^{\mathcal{A}}]_{n+i,n+i}$
1	1.0000	3.7622
2	1.6905	1.6974
3	1.6905	1.6974
4	1.6905	1.6974
5	1.6905	4.9203
6	4.0063	1.0000
7	1.7516	1.0000
8	1.7516	1.0000
9	1.7516	1.0000
10	1.7516	1.0000

The hub ranking is the same as with HITS and its exponentiated input variant. The authority ranking is {5; 1; 2,3,4 (tie); 6,7,8,9,10 (tie)}. Nodes 1 and 5 swap place in the authority ranking, with everything else staying the same. Whether this ranking is "better" than the one provided by HITS with exponentiated input is open to debate. What is clear is that both methods identify nodes 1 and 5 as the most authoritative ones by a considerable margin, whereas HITS with uniform starting vector completely fails in identifying node 1 as authoritative.

6. Application to web graphs. Similarly to HITS, and in analogy to subgraph centrality for undirected networks, the rankings produced by the values on the diagonal of $[e^{\mathcal{A}}]_{ii}$ can be used to rank websites as hubs and authorities in web searches (many other applications are of course also possible). The data sets tested here are small web graphs consisting of web sites on various topics and can be found at [30] along with the website associated with each node.[1] Each dataset is named after the corresponding topic. In this section, the hub and authority rankings obtained from $e^{\mathcal{A}}$ are compared with those from HITS, implemented using the Matlab code described in [23]. All experiments are performed using Matlab Version 7.9.0 (R2009b) on a MacBook Pro running OS X Version 10.6.8, a 2.4 GHZ Intel Core i5 processor and 4 GB of RAM. For the purpose of these tests we use the built-in Matlab function `expm` to compute the matrix exponential. Other approximations of $e^{\mathcal{A}}$ are discussed in section 8.

6.1. Abortion dataset. The *abortion* dataset contains $n = 2293$ nodes and $m = 9644$ directed edges. The expanded matrix $\mathcal{A} = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$ has order $N = 2n = 4586$ and contains $2m = 19288$ nonzeros. The maximum eigenvalue of $\mathcal{A}$ is $\lambda_N \approx 31.91$ and the second largest eigenvalue is $\lambda_{N-1} \approx 26.04$. In this matrix, the largest eigenvalue is fairly well-separated from the second largest so that one would expect the HITS rankings (which only use information from the dominant eigenpair of $\mathcal{A}$) to be reasonably close to the rankings from $e^{\mathcal{A}}$ (which use information from all of the eigenvalues and corresponding eigenvectors). A plot of the eigenvalues of the expanded abortion dataset matrix can be found in Fig. 6.1. Note the high multiplicity of the zero eigenvalue.

The top 10 hubs and authorities of the abortion dataset, as determined using $e^{\mathcal{A}}$,

[1] It should be noted, however, that in the node list for the adjacency matrix, the node labeling begins with 1 and in the list of websites associated with the nodes found at [30], node labeling begins at 0. Thus, node i in the adjacency matrix is associated with website $i - 1$.

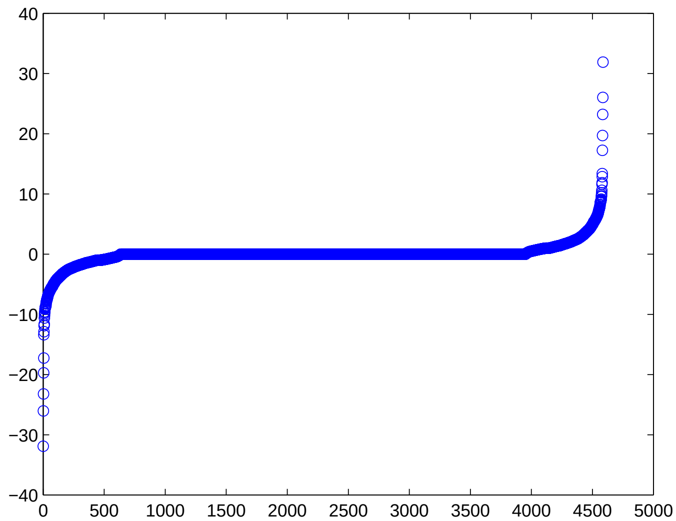

FIG. 6.1. *Plot of the eigenvalues of the expanded abortion matrix $\mathcal{A}$.*

TABLE 6.1
Top 10 hubs of the abortion web graph, ranked using $[e^{\mathcal{A}}]_{ii}$, HITS, and HITS with exponentiated input.

$[e^{\mathcal{A}}]_{ii}$ nodes	HITS nodes	Exp. input HITS nodes
48	48	80
1021	1006	1431
1007	1007	1432
1006	1021	1426
1053	1053	1425
1020	1020	1415
987	960	1388
990	968	1389
985	969	1397
989	970	1387

HITS with constant initial vector and exponentiated input HITS, can be found in tables 6.1 and 6.2. We observe that there is a good deal of agreement between the $e^{\mathcal{A}}$ rankings and the HITS ones: indeed, both methods identify the websites labeled 48, 1021, 1007, 1006, 1053, 1020 as the top 6 hubs, and both pick web site 48 as the top one. Also, there are 7 web sites identified by both methods as being among the top 10 authorities. The top authority identified by HITS is ranked third by $e^{\mathcal{A}}$, and conversely the top authority identified by $e^{\mathcal{A}}$ is third in the HITS ranking. In contrast, the exponentiated input HITS algorithm returns completely different rankings. Node 80, which is deemed the best hub by the exponentiated input HITS, is ranked 1236 by HITS and 851 using $e^{\mathcal{A}}$. Exponentiated input HITS ranks node 1430 as the top authority. This node is ranked 731 by HITS and 429 by $e^{\mathcal{A}}$. The odd behavior of exponentiated input HITS is due to the fact that the web graph abortion dataset contains many nodes with no in-links (at least, none that are included in the dataset).

TABLE 6.2
Top 10 authorities of the abortion web graph, ranked using $[e^{\mathcal{A}}]_{ii}$, HITS, and HITS with exponentiated input.

$[e^{\mathcal{A}}]_{ii}$ nodes	HITS nodes	Exp. input HITS nodes
967	939	1430
958	958	1387
939	967	1425
962	961	1426
963	962	1409
964	963	1417
961	964	1429
965	965	1406
966	966	1396
587	1582	1405

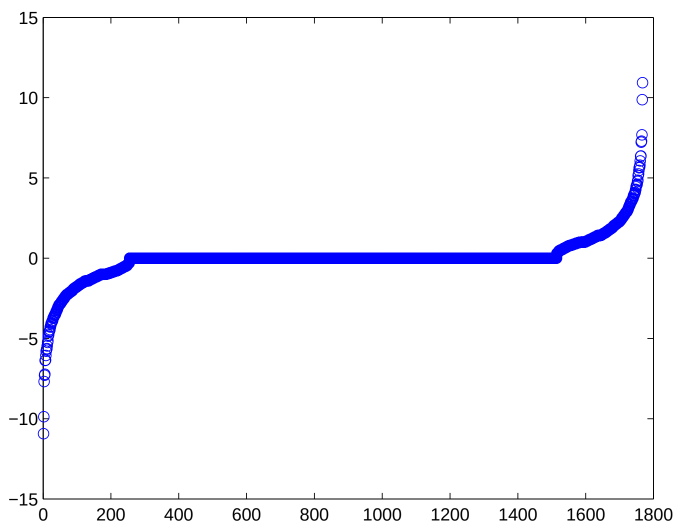

FIG. 6.2. *Plot of the eigenvalues of the expanded computational complexity matrix $\mathcal{A}$.*

As seen in section 5.5, nodes with no in-links are given less value by this algorithm and thus will not be reported as top hubs (nor will nodes pointed to by many nodes with no in-links be reported as top authorities). Since exponentiated input HITS behaved unreliably on the remaining two data sets as well, we do not show the corresponding results.

6.2. Computational complexity dataset. The *computational complexity* data set contains $n = 884$ nodes and $m = 1616$ directed edges. The expanded matrix $\mathcal{A}$ has order $N = 2n = 1768$ and contains $2m = 2232$ nonzeros. The maximum eigenvalue of $\mathcal{A}$ is $\lambda_N \approx 10.93$ and the second largest eigenvalue is $\lambda_{N-1} \approx 9.86$. Here, the (relative) spectral gap between the first and the second eigenvalue is smaller than in the previous example; consequently, we expect the rankings produced using $e^{\mathcal{A}}$ and HITS to be less similar than for the abortion dataset. A plot of the eigenvalues of the expanded computational complexity dataset matrix can be found in Fig. 6.2.

Table 6.3
Top 10 hubs of the computational complexity web graph, ranked using $[e^{\mathcal{A}}]_{ii}$ and HITS.

$[e^{\mathcal{A}}]_{ii}$ nodes	$[e^{\mathcal{A}}]_{ii}$ score	HITS nodes	HITS score
57	2.6518e04	57	0.276589
17	7.2059e02	634	0.035592
644	6.6561e02	644	0.020557
643	6.1256e02	721	0.018340
634	5.5558e02	643	0.017880
106	4.7486e02	554	0.014191
119	4.2791e02	632	0.013106
529	3.8451e02	801	0.012383
86	3.6528e02	640	0.011566
162	3.5502e02	639	0.010893

Table 6.4
Top 10 authorities of the computational complexity web graph, ranked using $[e^{\mathcal{A}}]_{ii}$ and HITS.

$[e^{\mathcal{A}}]_{ii}$ nodes	$[e^{\mathcal{A}}]_{ii}$ score	HITS nodes	HITS score
1	4.8958e03	719	0.012155
315	1.4747e03	717	0.011501
673	8.0015e02	727	0.009972
148	7.3093e02	723	0.009131
719	6.6746e02	808	0.008828
717	5.8437e02	735	0.008785
2	5.5637e02	737	0.008721
45	4.0969e02	1	0.008550
727	4.0315e02	722	0.008491
534	3.4473e02	770	0.008491

The top 10 hubs and authorities of the computational complexity dataset, determined by both rankings, can be found in Tables 6.3 and 6.4. We also report the actual scores obtained for these nodes. As expected, we see less agreement between the two ranking methods. Concerning the hubs, both methods agree that the web site labelled 57 is by far the most important hub on the topic of computational complexity. However, the method based on $e^{\mathcal{A}}$ identifies as the second most important hub the web site corresponding to node 17, which is not even in the top 10 according to HITS. The two methods agree on the next three hubs, but after that they return completely different results. The difference is even more pronounced for the authority rankings. The method based on $e^{\mathcal{A}}$ clearly identifies web site 1 as the most authoritative one, whereas HITS relegates this node to 8th place. The top authority according to HITS, web site 719, places 5th in the ranking obtained by $e^{\mathcal{A}}$. The two methods agree on only two other web sites as being in the top 10 authorities (717 and 727).

6.3. Death penalty dataset. The *death penalty* dataset contains $n = 1850$ and $m = 7363$ directed edges. The expanded matrix $\mathcal{A}$ has order $N = 2n = 3700$ and contains $m = 14726$ nonzeros. The maximum eigenvalue of $\mathcal{A}$ is $\lambda_N \approx 28.02$ and the second largest eigenvalue $\lambda_{N-1} \approx 17.68$. In this case, the largest and second largest eigenvalues are quite far apart, and the relative gap is larger than in the previous examples. A plot of the eigenvalues of the expanded death penalty matrix can be

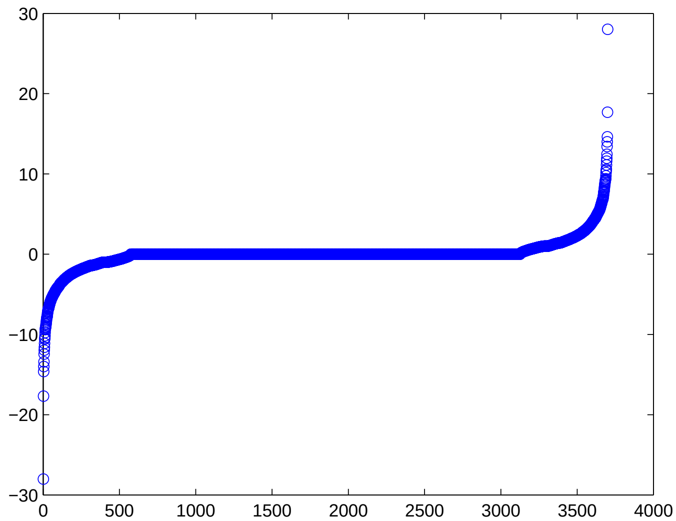

FIG. 6.3. *Plot of the eigenvalues of the expanded death penalty matrix $\mathcal{A}$.*

TABLE 6.5
Top 10 hubs of the death penalty web graph, ranked using $[e^{\mathcal{A}}]_{ii}$ and HITS.

$[e^{\mathcal{A}}]_{ii}$ nodes	$[e^{\mathcal{A}}]_{ii}$ score	HITS nodes	HITS score
210	6.7731e10	210	0.017562
637	3.5805e10	637	0.012769
413	3.2347e10	413	0.012137
1586	1.9676e10	1586	0.009466
552	1.7611e10	552	0.008955
462	1.1998e10	462	0.007392
930	1.1772e10	930	0.007322
542	1.1713e10	542	0.007303
618	1.1163e10	618	0.007130
1275	1.0556e10	1275	0.006933

found in Fig. 6.3.

Due to the presence of a large spectral gap, much of the information used in forming the rankings of $e^{\mathcal{A}}$ is also used in the HITS ranking, and we expect the two methods to produce similar results. Indeed, as shown in Table 6.5 (hubs) and Table 6.6 (authorities), in this case the top 10 rankings produced by the two methods are actually identical.

7. Other functions.

Besides the matrix exponential, another function that has been successfully used to define centrality and communicability measures for an undirected network is the matrix resolvent, which can be defined as

$$R(A; c) = (I - cA)^{-1} = I + cA + c^2A^2 + \cdots + c^kA^k + \cdots ,$$

with $0 < c < 1/\lambda_{\max}(A)$; see, e.g., [19, 13, 12]. Here A is the symmetric adjacency matrix of the undirected network. The condition on the parameter c ensures that $R(A; c)$ is well defined (i.e., that $I - cA$ is invertible and the geometric series converges

TABLE 6.6
Top 10 authorities of the death penalty web graph, ranked using $[e^{\mathcal{A}}]_{ii}$ and HITS.

$[e^{\mathcal{A}}]_{ii}$ nodes	$[e^{\mathcal{A}}]_{ii}$ score	HITS nodes	HITS score
4	6.4460e10	4	0.023556
1	5.4816e10	1	0.021723
6	3.8451e10	6	0.018193
7	3.0091e10	7	0.016094
10	2.8139e10	10	0.015564
16	2.6059e10	16	0.014977
2	2.5742e10	2	0.014886
3	2.4149e10	3	0.014418
44	1.9763e10	44	0.013043
27	1.8591e10	27	0.012651

to its inverse) and nonnegative; indeed, $I - cA$ will be a nonsingular M-matrix. It is hardly necessary to mention the close relationship existing between the resolvent and the exponential function, which can be expressed via the Laplace transform. For the adjacency matrix $\mathcal{A}$ of a bipartite graph given by (5.1), the resolvent is easily determined to be

$$R(\mathcal{A};c) = \begin{pmatrix} I + c^2A(I - c^2A^TA)^{-1}A^T & cA(I - c^2A^TA)^{-1} \\ c(I - c^2A^TA)^{-1}A^T & (I - c^2A^TA)^{-1} \end{pmatrix}. \quad (7.1)$$

The condition on c can be expressed as $0 < c < 1/\sigma_1$, where $\sigma_1 = \|A\|_2$ denotes the largest singular value of A, the adjacency matrix of the undirected network. This ensures that the matrix in (7.1) is well-defined and nonnegative, with positive diagonal entries. The diagonal entries of $I + c^2A(I - c^2A^TA)^{-1}A^T$ provide the hub scores, those of $(I - c^2A^TA)^{-1}$ the authority scores. A drawback of this approach is the need to select the parameter c, and the fact that different values of c may lead to different rankings.

Other functions that have been used for the analysis of complex networks include variants of the exponential, such as $f(A) = e^{\beta A}$, where β can be interpreted as a (negative) inverse "temperature", as well as similar functions involving the graph Laplacian $L = D - A$, where D is the degree matrix, $D = \mathrm{diag}\,(d_1, \ldots, d_n)$. We refer to [12] for a detailed study and justification of these matrix functions in the study of undirected networks. A comparison of the various hub and authority rankings obtained using these functions is beyond the scope of this paper, and will be the subject of a separate study.

8. Approximating the matrix exponential. Several approaches are available for computing the matrix exponential [18]. A commonly used scheme is the one based on Padé approximation combined with the scaling and squaring method [17, 18], implemented in Matlab by the `expm` function. For an $n \times n$ matrix, this method requires $O(n^2)$ storage and $O(n^3)$ arithmetic operations; any sparsity in A, if present, is not exploited in currently available implementations. Evaluation of the matrix exponential based on diagonalization also requires $O(n^2)$ storage and $O(n^3)$ operations. Furthermore, these methods cannot be easily adapted to the case where only selected entries (e.g., the diagonal ones) of the matrix exponential are of interest.

For the purpose of ranking hubs and authorities in a directed network, only the main diagonal of $e^{\mathcal{A}}$ is required. This can be done without having to compute *all* the entries in $e^{\mathcal{A}}$. If some of the off-diagonal entries (communicabilities) are desired, for example those between the highest ranked hubs and/or authorities, it is also possible to compute them without having to compute the whole matrix $e^{\mathcal{A}}$, which would be prohibitive even for a moderately large network. We further emphasize that in most applications one is not so much interested in computing an exact ranking of *all* the nodes in a digraph, but only in identifying the top k ranked nodes, where the integer k is small compared to n (for example, $k = 10$ or $k = 20$). It is highly desirable to develop methods that are capable of quickly identifying the top k hubs/authorities without having to compute accurate hub/authority scores for each node.

Efficient, accurate methods for estimating (or, in some cases, bounding) arbitrary entries in a matrix function $f(A)$ have been developed by Golub, Meurant and collaborators (see [16] and references therein) and first applied to problems of network analysis by Benzi and Boito in [1]; see also [4]. Here we limit ourselves to a brief description of these methods, referring the reader to [1] and [16] for further details. Let A be a real, symmetric, $n \times n$ matrix and let f be a function defined on the spectrum of A. Consider the eigendecompositions $A = Q\Lambda Q^T$ and $f(A) = Qf(\Lambda)Q^T$, where $Q = [\phi_1, \ldots, \phi_n]$. For given vectors u and v we have

$$u^T f(A)v = u^T Q f(\Lambda) Q^T v = w^T f(\Lambda) z = \sum_{k=1}^{n} f(\lambda_k) w_k z_k, \tag{8.1}$$

where $w = Q^T u = (w_k)$ and $z = Q^T v = (z_k)$. In particular, for $f(A) = e^A$ we obtain

$$u^T e^A v = \sum_{k=1}^{n} e^{\lambda_k} w_k z_k. \tag{8.2}$$

Choosing $u = v = e_i$ (the vector with the ith entry equal to 1 and all the remaining ones equal to 0) we obtain an expression for the subgraph centrality of node i:

$$SC(i) := \sum_{k=1}^{n} e^{\lambda_k} \phi_{k,i}^2 \,,$$

where $\phi_{k,i}$ denotes the ith component of vector ϕ_k. Likewise, choosing $u = e_i$ and $v = e_j$ we obtain the following expression for the communicability between node i and node j:

$$C(i,j) := \sum_{k=1}^{n} e^{\lambda_k} \phi_{k,i} \phi_{k,j} \,.$$

Analogous expressions hold for other matrix functions, such as the resolvent.

Hence, the problem is reduced to evaluating bilinear expressions of the form $u^T f(A)v$. Such bilinear forms can be thought of as Riemann- Stieltjes integrals with respect to a (signed) spectral measure:

$$u^T f(A)v = \int_a^b f(\lambda) d\mu(\lambda), \quad \mu(\lambda) = \begin{cases} 0, & \text{if } \lambda < a = \lambda_1, \\ \sum_{k=1}^{i} w_k z_k, & \text{if } \lambda_i \le \lambda < \lambda_{i+1}, \\ \sum_{k=1}^{n} w_k z_k, & \text{if } b = \lambda_n \le \lambda. \end{cases}$$

This integral can be approximated by means of a Gauss-type quadrature rule:

$$\int_a^b f(\lambda) d\mu(\lambda) = \sum_{j=1}^{p} c_j f(t_j) + \sum_{k=1}^{q} v_k f(\tau_k) + R[f], \tag{8.3}$$

where $R[f]$ denotes the error. Here the nodes $\{t_j\}_{j=1}^p$ and the weights $\{c_j\}_{j=1}^p$ are unknown, whereas the nodes $\{\tau_k\}_{k=1}^q$ are prescribed. We have

- $q = 0$ for the Gauss rule,
- $q = 1$, $\tau_1 = a$ or $\tau_1 = b$ for the Gauss–Radau rule,
- $q = 2$, $\tau_1 = a$ and $\tau_2 = b$ for the Gauss–Lobatto rule.

For certain matrix functions, including the exponential and the resolvent, these quadrature rules can be used to obtain lower and upper bounds on the quantities of interest; prescribing additional quadrature nodes leads to tighter and tighter bounds, which (in exact arithmetic) converge monotonically to the true values [16]. The evaluation of these quadrature rules is mathematically equivalent to the computation of orthogonal polynomials via a three-term recurrence, or, equivalently, to the computation of entries and spectral information of a certain tridiagonal matrix via the Lanczos algorithm. Here we briefly recall how this can be done for the case of the Gauss quadrature rule, when we wish to estimate the ith diagonal entry of $f(A)$. It follows from (8.3) that the quantity of interest has the form $\sum_{j=1}^p c_j f(t_j)$. This can be computed from the relation (Theorem 3.4 in [16]):

$$\sum_{j=1}^{p} c_j f(t_j) = e_1^T f(J_p) e_1,$$

where

$$J_p = \begin{pmatrix} \omega_1 & \gamma_1 & & & \\ \gamma_1 & \omega_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{p-2} & \omega_{p-1} & \gamma_{p-1} \\ & & & \gamma_{p-1} & \omega_p \end{pmatrix}$$

is a tridiagonal matrix whose eigenvalues are the Gauss nodes, whereas the Gauss weights are given by the squares of the first entries of the normalized eigenvectors of J_p. The entries of J_p are computed using the Lanczos algorithm with starting vectors $x_{-1} = 0$ and $x_0 = e_i$. Note that it is not required to compute all the components of the eigenvectors of J_p if one uses the Golub–Welsch QR algorithm; see [16].

For small p (i.e., for a small number of Lanczos steps), computing the $(1, 1)$ entry of $f(J_p)$ is inexpensive. The main cost in estimating one entry of $f(A)$ with this approach is associated with the sparse matrix-vector multiplies in the Lanczos algorithm applied to the adjacency matrix A. If only a small, fixed number of iterations are performed for each diagonal element of $f(A)$, as is usually the case, the computational cost (per node) is at most $O(n)$ for a sparse graph, resulting in a total cost of $O(n^2)$ for computing the subgraph centrality of every node in the network. If only $k < n$ subgraph centralities are wanted, with k independent of n, then the overall cost of the computation will be $O(n)$ provided that sparsity is carefully exploited in the Lanczos algorithm and that only a small number p of iterations (independent of n) is carried out. Note, however, that depending on the connectivity characteristics of the network under consideration, the prefactor in the $O(n)$ estimate could be large. The algorithm can be implemented so that the storage requirements are $O(n)$ for a sparse network—that is, a network in which the total number of links grows linearly in the number n of nodes.

As already mentioned, a nice feature of the approach based on Gauss quadrature is that it yields monotonically converging lower and upper bounds. As shown in

Table 8.1
The number of iterations necessary for the top 10 hubs or authorities to be determined (not necessarily in the correct order).

Dataset	hub (lower bound)	hub (upper bound)
Abortion	> 40	> 40
Comp. Complex.	3	3
Death Penalty	5	3

Dataset	authority (lower bound)	authority (upper bound)
Abortion	2	2
Comp. Complex.	4	5
Death Penalty	4	2

section 8.1 below, in a typical network this allows for the rapid identification of nodes with high hub and authority scores, which often is all one needs. Indeed, as soon as the lower bound for a node becomes larger than the upper bound for another node, it is known that the former is ranked higher than the latter, and additional iterations cannot alter that fact. This is especially useful when we want to compare pairs of nodes in terms of their hub or authority rankings.

When applying the approach based on Gauss quadrature rules to the $2n \times 2n$ matrix $\mathcal{A}$, only matrix-vector products with A and its transpose are required, just like in the HITS algorithm. If only the hub scores are wanted, it is also possible to apply the described techniques to the symmetric matrix AA^T using the function $f(\lambda) = \cosh(\sqrt{\lambda})$; the same applies if only the authority scores are wanted, working this time with the matrix A^TA. The problem with this approach is that only estimates (rather than increasingly accurate lower and upper bound) can be obtained, due to the fact that the function $f(\lambda) = \cosh(\sqrt{\lambda})$ is not strictly completely monotonic on the positive real axis. We refer to [2] for details. In our experiments we always work with the matrix $\mathcal{A}$, since we are interested in computing both hub and authority scores.

8.1. Test results. Accurate evaluation of *all* the diagonal entries of $e^{\mathcal{A}}$ using quadrature rules is too expensive for truly large-scale graphs. In most applications, fortunately, it is not necessary to rank all the nodes in the network: only the top few hubs and authorities are likely to be of interest. When using quadrature rules, the number of quadrature nodes (Lanczos iterations) required to correctly rank the nodes as hubs or authorities varies and depends on both the eigenvalues of $e^{\mathcal{A}}$ and how close the diagonal entries are in value. If the rankings of the nodes are very close, it can take many iterations for the ordering to be exactly determined. However, since estimates for diagonal entries are calculated individually, once the top 10 (say) nodes have been identified, additional iterations can be performed only on these nodes in order to determine their exact ranking.

The number of iterations necessary to identify the top 10 hubs and authorities, using Gauss-Radau lower and upper bounds, for the three datasets from section 6 is given in Table 8.1. From this table it can be seen that, in most cases, only 2-5 iterations are needed. An exception is the determination of the top 10 hubs of the abortion dataset, for which the number of iterations is large (> 40). This is due to a cluster of nodes (nodes 960 and 968-990) that have nearly identical hub rankings. These nodes' scores agree to 15 significant digits. However, for most applications, if a

Table 8.2
The number of iterations necessary for the top 10 hubs or authorities to be ranked in the top 30.

Dataset	hub (lower bound)	hub (upper bound)
Abortion	5	4
Comp. Complex.	2	2
Death Penalty	2	2

Dataset	authority (lower bound)	authority (upper bound)
Abortion	2	2
Comp. Complex.	4	2
Death Penalty	2	2

subset of nodes are so closely ranked, their exact ordering may not be so important. Table 8.2 reports the number of Lanczos iterations neeeded for the top 10 hubs and authorities to be ranked at least in the top 30. Here, the number of iterations needed is always no more than 5. Using a simple implementation based on G. Meurant's Matlab code [24], which has not been optimized, the running times range from 0.58s for the computational complexity data set to 6.41s for the abortion data set.

9. Conclusions and outlook. In this paper we have presented a new approach to ranking hubs and authorities in directed networks using functions of matrices. Bipartization is used to transform the original directed network into an undirected one with twice the number of nodes. The adjacency matrix of the bipartite graph is symmetric, and this allows the use of subgraph centrality (and communicability) measures for undirected networks. We showed that the diagonal entries of the matrix exponential provide hub and authority rankings, and we gave an interpretation for the off-diagonal entries (communicabilities). Unlike HITS, the results are independent of any starting vectors, and the proposed method appears to be superior to the variant of HITS known as "exponentiated input HITS", which is unreliable in the presence of a large number of nodes with 0 in-degree (a common occurrence in many real-life networks).

Numerous examples, both synthetic and corresponding to real datasets, have been used to demonstrate the effectiveness of the proposed ranking algorithms relative to HITS and its exponentiated input variant. In particular, our experiments indicate that our method results in rankings that are different, and arguably better, than those computed by HITS in the absence of large gaps between the dominant singular value of the adjacency matrix of the digraph and the remaining ones. This is to be expected, since our method uses information from all the singular spectrum of the network, not just the dominant left and right singular pairs.

The price to pay from the more refined rankings obtained is a higher computational cost than HITS. We showed how Gaussian quadrature rules can be used to quickly identify the top ranked hubs and authorities for networks involving thousands of nodes.

Future work should include a comparison of the matrix exponential with other possible matrix functions, and tests on large networks. It is likely that the proposed approach based on Gaussian quadrature will prove to be too expensive for truly large-scale networks with millions of nodes. We hope to explore techniques similar to those presented in [4] and [29] in order to extend our methodology to truly large-scale networks. Finally, we are currently investigating the rate of convergence of the

Lanczos algorithm for estimating bilinear forms associated with adjacency matrices of graphs of different types.

Acknowledgments. We are indebted to Tammy Kolda (Sandia National Laboratories) and Stratis Gallopoulos (University of Patras, Greece) for insightful conversations on the topic of this paper.

REFERENCES

[1] M. Benzi and P. Boito, *Quadrature rule-based bounds for functions of adjacency matrices*, Lin. Algebra Appl., 433 (2010), pp. 637–652.

[2] M. Benzi and G. H. Golub, *Bounds on the entries of matrix functions with applications to preconditioning*, BIT, 39 (1999), pp. 417–438.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, *Complex networks: structure and dynamics*, Phys. Rep., 424 (2006), pp. 175–308.

[4] F. Bonchi, P. Esfandiar, D. F. Gleich, C. Greif, and L. V. S. Lakshmanan, *Fast matrix computations for pair-wise and column-wise commute times and Katz scores*, Internet Math., to appear, 2012.

[5] U. Brandes and T. Erlebach, eds., *Network Analysis: Methodological Foundations*, Lecture Notes in Computer Science Vol. 3418, Springer, New York, 2005.

[6] G. Caldarelli, *Scale-free Networks*, Oxford University Press, Oxford, UK, 2007.

[7] J. J. Crofts, E. Estrada, D. J. Higham, and A. Taylor, *Mapping directed networks*, Electr. Trans. Numer. Anal., 37 (2010), pp. 337–350.

[8] R. Diestel, *Graph Theory*, Springer-Verlag, Berlin, 2000.

[9] E. Estrada, *The Structure of Complex Networks*, Oxford University Press, Oxford, UK, 2011.

[10] E. Estrada, M. Fox, D. Higham, and G.-L. Oppo, eds., *Network Science. Complexity in Nature and Technology*, Springer, New York, 2010.

[11] E. Estrada and N. Hatano, *Communicability in complex networks*, Phys. Rev. E, 77 (2008), article 036111.

[12] E. Estrada, N. Hatano, and M. Benzi, *The Physics of Communicability in Complex Networks*, Preprint arXiv:1109.2950, September 2011. To appear in *Physics Reports*.

[13] E. Estrada and D. J. Higham, *Network properties revealed through matrix functions*, SIAM Rev., 52 (2010), pp. 671–696.

[14] E. Estrada and J. A. Rodríguez-Velázquez, *Subgraph centrality in complex networks*, Phys. Rev. E, 71 (2005), 056103.

[15] A. Farahat, T. Lofaro, J. C. Miller, G. Rae, and L. A. Ward, *Authority rankings from HITS, PageRank, and SALSA: existence, uniqueness, and effect of initialization*, SIAM J. Sci. Comput., 27 (2006), pp. 1181–1201.

[16] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, NJ, 2010.

[17] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Applic., 26 (2006), pp. 1179–1193.

[18] N. J. Higham, *Functions of Matrices. Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008.

[19] L. Katz, *A new status index derived from sociometric data analysis*, Psychometrika, 18 (1953), pp. 39–43.

[20] J. Kleinberg, *Authoritative sources in a hyperlinked environment*, J. ACM, 46 (1999), pp. 604–632.

[21] R. Lempel and S. Moran, *The stochastic approach for link-structure analysis (SALSA) and the TKC effect*, in Proceedings of the Ninth International Conference on the World Wide Web, 2000, pp. 387–401.

[22] A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for Web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.

[23] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.

[24] G. Meurant, *MMQ toolbox for MATLAB*, http://pagesperso-orange.fr/gerard.meurant/.

[25] J. C. Miller, G. Rae, F. Schaefer. L. A. Ward, A. Farahat, and T. LoFaro *Modifications of Kleinberg's HITS algorithm using matrix exponentiation and web log records*, Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 2001, pp. 444–445

[26] M. E. J. Newman, *The structure and function of complex networks*, SIAM Rev., 45 (2003), pp. 167–256.

[27] M. E. J. Newman, *Networks: An Introduction*, Cambridge University Press, Cambridge, UK, 2010.

[28] M. E. J. Newman, A.-L. Barabasi, and D. J. Watts, *The Structure and Dynamics of Networks*, Princeton University Press, Princeton, NJ, 2003.

[29] B. Savas and I. Dhillon, *Clustered low rank approximation of graphs in information science applications*, Proceedings of the 2011 SIAM Conference on Data Mining, April 2011.

[30] P. Tsaparas, *Datasets for Experiments on Link Analysis Ranking Algorithms*, available at http://www.cs.toronto.edu/∼tsap/experiments/datasets/index.html (last accessed January 2012).