Journal of Computational and Applied Mathematics 355 (2019) 182–192

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Analysis of directed networks via the matrix exponential

Omar De la Cruz Cabrera, Mona Matar, Lothar Reichel *

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

ARTICLE INFO

Article history:
Received 6 December 2017
Received in revised form 1 August 2018

Keywords:
Network analysis
Matrix exponential
Linear algebra

ABSTRACT

The matrix exponential has been identified as a useful tool for the analysis of undirected networks, with sound theoretical justifications for its ability to model important aspects of a given network. Its use for directed networks, however, is less developed and has been less successful so far. In this article we discuss some methods to identify important nodes in a directed network using the matrix exponential, taking into account that the notion of importance changes whether we consider the influence of a given node along the edge directions (downstream influence) or how it is influenced by directed paths that point to it (upstream influence). In addition, we introduce a family of importance measures based on counting walks that are allowed to reverse their direction a limited number of times, thus capturing relationships arising from influencing the same nodes, or being influenced by the same nodes, without sacrificing information about edge direction. These measures provide information about branch points.

1. Introduction

Often, a complex system can be modeled as a *network*: a set of *nodes*, any two of which might be connected by *edges*. Mathematically, we represent a network by a graph, which may be directed or undirected [1,2]. In spite of their simplicity, which makes mathematical analysis tractable, these concepts can capture much of the complexity of the behavior of the system. Some examples are:

- Social networks: Nodes are individuals (human or animal), and edges represent social relationships (e.g., acquaintance, friendship, allegiance).
- Road networks: Each intersection or endpoint is a node, and each road section connecting one node to another one is an edge.
- In molecular biology, genes and/or proteins can be regarded as nodes, connected by relationships like regulation (directed) or interaction (undirected) [3].

In this work, we are interested in determining the relative importance of nodes, as well as communicability between pairs of nodes. In undirected networks, i.e., in networks in which each edge is a "two-way street", quantities that try to capture the intuitive notion of importance have come to be known as *notions of centrality* [1,4,5], based on the idea that important nodes should be reachable from many nodes in fairly few steps. In directed networks, a node can be important in two ways: a node can have high *downstream influence* (can reach many nodes in fairly few steps along the direction of the edges) or high *upstream influence* (is reached by many nodes in fairly few steps along the direction of the edges); a node with high centrality should have high influence both upstream and downstream.

* Corresponding author.
E-mail addresses: odelacru@kent.edu (O. De la Cruz Cabrera), mmatar2@kent.edu (M. Matar), reichel@math.kent.edu (L. Reichel).

https://doi.org/10.1016/j.cam.2019.01.015
0377-0427/

Various measures have been proposed for quantification of the centrality of a node and of the communicability between two nodes; e.g., [1,2,4,6–9]. Some approaches (see, e.g., [4,6,7,10–12]) involve the use of matrix functions, like the matrix exponential of the adjacency matrix; see Section 2.2 for further details. These methods have proved to be useful when applied to undirected networks.

For directed networks, the record is less clear. Some authors claim that the application of matrix exponential methods leads to counter-intuitive results in simple examples [13]; see Section 3.1. The *hubs and authorities* approach [9] was proposed as a way to avoid some of those perceived shortcomings. A variation of this approach has recently been described in [7]. Katz [8] proposed that the resolvent of the adjacency matrix A or its transpose A^T times the vector $\mathbf{1} = [1, 1, \ldots, 1]^T$ be used as centrality measures. Thus, Katz considered the entries of the vectors

$$(I - \mu A)^{-1}\mathbf{1}, \qquad (I - \mu A^T)^{-1}\mathbf{1}, \tag{1.1}$$

where the scalar $\mu > 0$ is chosen sufficiently small so that the power series expansions of the above expressions converge; see below for further details. More recently, Benzi et al. [4,7,11] considered analogues of the expressions (1.1) with the resolvents replaced by the exponential functions of A or A^T, as well as with the matrix A replaced by the matrix (3.1) defined below. However, for directed networks very little computational analysis has been reported in the literature that sheds light on the performance of the expressions $e^A\mathbf{1}$ and $e^{A^T}\mathbf{1}$ as measures of a node's importance and the ease of traveling to or from nodes. It is the purpose of this paper to shed some light on these issues, as well as to introduce matrix functions that allow a finite number of reversals of paths. These matrix functions are helpful in identifying branch points.

This paper is organized as follows. Section 2 introduces basic notions about graphs and matrix functions. Section 3 discusses several ways to measure the importance of a node. In particular, we discuss the application of $e^A\mathbf{1}$ and $e^{A^T}\mathbf{1}$ as measures to quantify aggregate upstream and downstream reachability, as well as relativized versions of this approach that can identify the nodes that have most influence on a predetermined set of nodes. Section 4 discusses the possibility of reverting the direction of the walk a limited number of times. We then give examples in Section 5 that show the functions $e^A\mathbf{1}$ and $e^{A^T}\mathbf{1}$ of a non-symmetric adjacency matrix A to provide meaningful ranking of the nodes in their roles as broadcasters and receivers, respectively. Numerical methods for large-scale networks with direction reversal are discussed in Section 6. We summarize our results in Section 7.

2. Basic definitions and properties

2.1. Graphs

A network can be described mathematically by a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (or vertices) and $E = \{e_1, e_2, \ldots, e_m\}$ is the set of edges; basic facts about graphs can be found, e.g., in [1,2]. We assume G to be a simple graph, i.e., G has no multiple edges and no self-loops. A directed edge e_k pointing from node v_i to node v_j can be identified with the ordered pair (v_i, v_j); for an undirected edge, this pair is not ordered. If some of the edges are directed, then we call the graph *directed*, otherwise we call it *undirected*.

The *out-degree* of a node counts the number of edges emerging from that node, and the *in-degree* counts those pointing directly at it. A (standard) *walk* of length k is a sequence $v_{i_1}, v_{i_2}, \ldots v_{i_{k+1}}$ of nodes and a sequence $e_{i_1}, e_{i_2}, \ldots, e_{i_k}$ of edges such that e_{i_j} points from v_{i_j} to $v_{i_{j+1}}$. An *alternating walk* from node v_{i_1} to node v_{i_k} is a sequence of nodes $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$, such that the direction of the edges is reversed at each step. We are interested in identifying nodes that are good *broadcasters* or good *receivers*, i.e., nodes that originate several (standard) walks (originate much information flow) or are targets of several (standard) walks. We remark that good broadcasters or receivers are not necessarily good hubs or authorities, respectively; see Section 3.1.2 for a discussion of the latter concepts and references to the literature.

We can describe a network of n nodes by the adjacency matrix $A = [A_{ij}]_{i,j=1}^n \in \mathbb{R}^{n\times n}$ associated with the graph that defines the network. Here $A_{ij} = 1$ if an edge from node v_i points to node v_j, and $A_{ij} = 0$ otherwise. The choice of 1 for all nonzero elements of A indicates that all connections are equally important. A network is said to be directed if at least one of its edges is directed. The adjacency matrix associated with an undirected network is symmetric, while it is nonsymmetric for a directed network. The transpose, A^T, of the adjacency matrix can be thought of as the adjacency matrix of the graph obtained by reversing the direction of the edges.

2.2. The matrix exponential and other matrix functions

The (i, j)th element of A^k gives the number of walks of length k starting at node v_i and ending at node v_j. A *matrix function* is defined by a power series of the form

$$f(A) = \sum_{p=0}^{\infty} c_p A^p \tag{2.1}$$

and can be interpreted as the sum of weighted numbers of walks of various lengths between the nodes. The coefficients c_p are weights; they are nonnegative and decreasing as p increases. Hence, long walks are weighted less than short walks, i.e., long walks are considered less important than short walks. The coefficients c_p should decrease to zero quickly enough so

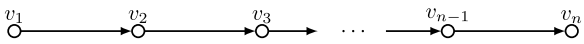

Fig. 1. Graph of a directed chain of nodes.

Table 1
Ranking the nodes in Fig. 1 by bipartizing the graph.

(a)		(b)	
Hub role		Authority role	
Node v_i	$e_{ii}^{\mathcal{A}}$	Node v_i	$e_{n+i,n+i}^{\mathcal{A}}$
v_1	1.543081	v_2	1.543081
v_2	1.543081	v_3	1.543081
v_3	1.543081	v_4	1.543081
v_4	1.543081	v_5	1.543081
v_5	1	v_1	1

that the series (2.1) converges. A nice introduction to the use of matrix functions in network analysis is provided by Estrada and Higham [12].

Commonly used matrix functions for network analysis include the *matrix exponential*, obtained by taking $c_p = 1/p!$, and the resolvent

$$f(A) = (I + \mu A)^{-1} = I + \mu A + \mu^2 A^2 + \cdots, \tag{2.2}$$

where the scalar $\mu > 0$ is chosen small enough so that the above series converges; see [12] for a discussion and illustrations. The *subgraph centrality* of node v_i, given by $[f(A)]_{ii}$, can be applied to measure the importance of the node; see [5,12]. A relatively large value indicates that the node is important.

3. Node importance and communicability using the matrix exponential

3.1. Existing methods

3.1.1. Methods for undirected networks

Measures of importance include simple notions like vertex degree (degree centrality) as well as more elaborate concepts [1,4,5,14], including subgraph centrality.

Benzi et al. [7] discussed the application of the matrix exponential to the adjacency matrix for a directed network and observed that it may not be meaningful to use the diagonal entries of the exponential as a measure of importance of the nodes of a directed network. They considered the directed network of Fig. 1.

The associated adjacency matrix is an $n \times n$ Jordan block with diagonal entries zero. All diagonal entries of the exponential of this matrix are one, giving the same importance to each of the nodes. This result was not satisfying for the authors, since the first and last nodes should not be equally important. Benzi et al. [7] therefore proposed to bipartite the network into hubs and authorities. We outline this approach in the following subsection.

3.1.2. Directed networks: Bipartization

The out-degree and in-degree are simple measures of the importance of a node in its roles as hub and authority, respectively. These measures only consider local information of each node, and may not be appropriate measures of importance.

Kleinberg [9] proposed to split a directed network into hubs and authorities and Benzi et al. [7] constructed a related symmetric matrix that makes it possible to compute the hub centrality and authority centrality of nodes using the matrix exponential. Specifically, Benzi et al. [7] considered the symmetric matrix

$$\mathcal{A} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \tag{3.1}$$

determined by the nonsymmetric adjacency matrix A associated with the given directed graph. The diagonal entry $e_{ii}^{\mathcal{A}}$, for $1 \leq i \leq n$, gives a measure of the hub centrality of node v_i. Likewise the diagonal entry $e_{n+i,n+i}^{\mathcal{A}}$, for $1 \leq i \leq n$, gives a measure for the authority centrality of node v_i.

However, while determining the importance of nodes by ranking their hub and authority roles using the diagonal entries of the exponential $e^{\mathcal{A}}$ yields valuable information in many situations, it is not satisfactory for identifying good broadcasters and receivers. For instance, consider the network of Fig. 1 and let $n = 5$. Following the process described above, we obtain the ranking displayed in Table 1.

Although this ranking shows node v_5 to be the least important node in its hub role, and node v_1 is seen to be least important in its authority role, this approach fails to give a reasonable ranking for the other nodes in the graph, which appear to be equally important as hubs and authorities. However, since node v_1 is able to send information to all the other nodes in the network, its role as broadcaster should be the largest. In the next section, we discuss an alternative way to rank the nodes of a directed network, without bipartization of the graph.

3.2. Aggregate upstream and downstream reachability

Consider the exponential of a nonsymmetric adjacency matrix A associated with a directed network. Introduce the vector $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$ for measuring how important it is to reach each node of the network; we let $u_i = 0$ if the node v_i is considered an uninteresting destination. A large value of u_i indicates that node v_i is a highly targeted node. In the following we only consider entries $u_i \in \{0, 1\}$. Now $e^A\mathbf{u}$ allows us to determine which nodes are most important for the requested flow, i.e., which nodes are the most important broadcasters. Similarly, $\left(\mathbf{u}^T e^A\right)^T = e^{A^T}\mathbf{u}$ gives a ranking of the nodes according to their role as receivers from the nodes specified by the nonzero entries of $\mathbf{u}$.

We define the *aggregate downstream reachability* as the vector

$$\mathrm{ADR} = e^A\mathbf{1}. \tag{3.2}$$

Thus, here $\mathbf{u} = \mathbf{1}$. This vector gives the same importance to the goal of reaching any node in the network. Similarly, the *aggregate upstream reachability* is defined as

$$\mathrm{AUR} = e^{A^T}\mathbf{1}. \tag{3.3}$$

ADR provides a reasonable ranking for the nodes in their broadcaster role, i.e., in their ability to broadcast information through the network. Likewise, AUR can be used to rank the nodes according to their receiver role. We remark that Benzi et al. [7, Section 8.1] tabulated the column sum of e^A, but did not discuss this approach in any detail.

For the graph of Fig. 1, the ADR and AUR methods determine the rankings displayed in Table 2. These rankings satisfy the requirements mentioned at the end of Section 3.1.2.

4. Walks with a bounded number of reversions

An appealing feature of the hubs-and-authorities model described by Benzi et al. [7] is that two nodes v_i, v_j can be considered related in situations when edges are connected by alternating walks; the drawback is that *only* strictly alternating walks are considered. On the other hand, measures based solely on e^A, such as ADR and AUR, fail to capture these "lateral" connections. In this section we consider an approach that allows us to recover some of these connections. Taking lateral connections into account helps us identify important branch points.

A simple way to include lateral connections is to transform the directed network into an undirected one by disregarding edge directions. This can be achieved by symmetrizing the adjacency matrix, i.e., by computing the exponential of the symmetric matrix $A_S = \frac{1}{2}(A + A^T)$ instead of the exponential of A. The matrix A_S is the closest symmetric matrix to A in the Frobenius norm. Of course, all directionality information is lost when replacing A by A_S. Nevertheless, for comparison purposes, we define the *symmetrized aggregate reachability* by

$$\mathrm{SAR} = e^{\frac{1}{2}A+\frac{1}{2}A^T}\mathbf{1}.$$

Consider now the matrix $e^{\frac{1}{2}A}e^{\frac{1}{2}A^T}$. This matrix equals the one used in SAR if and only if A and A^T commute. We have

$$e^{\frac{1}{2}A}e^{\frac{1}{2}A^T} = \left(\sum_{n=0}^{\infty}\frac{A^n}{2^n n!}\right)\left(\sum_{n=0}^{\infty}\frac{\left(A^T\right)^n}{2^n n!}\right) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{A^i\left(A^T\right)^j}{2^{i+j}i!j!}. \tag{4.1}$$

Since the entries of $A^i\left(A^T\right)^j$ count walks that move along i edges, and then in reverse along j edges, $e^{\frac{1}{2}A}e^{\frac{1}{2}A^T}$ is comprised of weighted sums of counts of walks that start in the forward direction, and have exactly one change of direction (either leg of the walk may be of length zero). Similarly, $e^{\frac{1}{2}A^T}e^{\frac{1}{2}A}$ contains weighted sums of walks that start in reverse, and then change direction exactly once. The vectors $e^{\frac{1}{2}A}e^{\frac{1}{2}A^T}\mathbf{1}$ and $e^{\frac{1}{2}A^T}e^{\frac{1}{2}A}\mathbf{1}$ are helpful in identifying branch points of directed networks. This will be illustrated in Section 5.

Analogously to (4.1), we can construct matrices containing information about walks with at most two reversions, namely

$$e^{\frac{1}{3}A}e^{\frac{1}{3}A^T}e^{\frac{1}{3}A} \quad \text{and} \quad e^{\frac{1}{3}A^T}e^{\frac{1}{3}A}e^{\frac{1}{3}A^T}.$$

In general, we can build a matrix containing weighted sums of numbers of walks with at most k reversions using alternating products of $k+1$ factors of the form $e^{\frac{1}{k+1}A}$ and $e^{\frac{1}{k+1}A^T}$.

We introduce the *bounded number of reversions* notions of reachability, denoted by $\mathrm{BNR}(x, k)$, where $x \in \{d, u\}$, by

$$\mathrm{BNR}(d, k) = \left(e^{\frac{1}{k+1}A}e^{\frac{1}{k+1}A^T}\cdots\right)\mathbf{1} \tag{4.2}$$

and

$$\mathrm{BNR}(u, k) = \left(e^{\frac{1}{k+1}A^T}e^{\frac{1}{k+1}A}\cdots\right)\mathbf{1}. \tag{4.3}$$

Here each product has $k+1$ exponential factors.

Table 2
Comparison of influence measures for the nodes in Fig. 1, including aggregate reachability using walks with a bounded number of reversions, as well as disregarding orientation altogether; see Section 4 for definitions.

Node v_i	$e_{ii}^{\mathcal{A}}$	ADR	BNR(d, 1)	BNR(d, 2)	
v_1	1.54	2.71	1.99	2.14	
v_2	1.54	2.67	2.55	2.62	
v_3	1.54	2.50	2.65	2.64	
v_4	1.54	2.00	2.47	2.39	
v_5	1.00	1.00	1.65	1.53	
Node v_i	$e_{n+i,n+i}^{\mathcal{A}}$	AUR	BNR(u, 1)	BNR(u, 2)	SAR
v_1	1.00	1.00	1.65	1.53	1.83
v_2	1.54	2.00	2.47	2.39	2.53
v_3	1.54	2.50	2.65	2.64	2.66
v_4	1.54	2.67	2.55	2.62	2.53
v_5	1.54	2.71	1.99	2.14	1.83

We are mostly interested in small values of k (say, $k = 1$ or 2), since intuition suggests that the more reversals we allow, the closer we get to losing all directional information, as in the SAR approach. This intuition is formalized by the following result:

Proposition 4.1. $\lim_{k\to\infty} \mathrm{BNR}(d, k) = \lim_{k\to\infty} \mathrm{BNR}(u, k) = \mathrm{SAR}$.

Proof. The statement follows from the Lie product formula (see, e.g., [15, page 35]):

$$e^{X+Y} = \lim_{m\to\infty} \left(e^{X/m} e^{Y/m}\right)^m.$$

Taking $X = \frac{1}{2}A$ and $Y = \frac{1}{2}A^T$, we obtain for $k = 2m - 1$ that

$$e^{\frac{1}{k+1}A^T} e^{\frac{1}{k+1}A} \cdots e^{\frac{1}{k+1}A} = \left(e^{\frac{1}{m}\frac{1}{2}A^T} e^{\frac{1}{m}\frac{1}{2}A}\right)^m \to e^{\frac{1}{2}A^T + \frac{1}{2}A}$$

as $m \to \infty$. The same happens for the products starting with $e^{\frac{1}{k+1}A}$. For $k = 2m$, we obtain terms of the form

$$\left(\exp\left(\tfrac{m}{m+1}\tfrac{1}{m}\tfrac{1}{2}A^T\right)\exp\left(\tfrac{m}{m+1}\tfrac{1}{m}\tfrac{1}{2}A\right)\right)^m e^{\frac{1}{2m+1}A^T}.$$

The last factor converges to I as $m \to \infty$; the remaining part can be shown to converge to $e^{\frac{1}{2}A^T + \frac{1}{2}A}$ by the same method as the proof of the Lie product formula in [15]. Finally, the result is obtained multiplying by $\mathbf{1}$ on the right. □

We remark that the matrices A and A^T commute if and only if A is normal. In this case, both BNR(d, k) and BNR(u, k) equal SAR, for all odd k. This means that the BNR measures may discard all directionality information, even for small k, but only for a restrictive class of adjacency matrices. Indeed, equality of the diagonal entries of AA^T and A^TA implies that each vertex has equal in-degree and out-degree, which is a reasonable condition if the directed network represents a volume-preserving flow; equality of the off-diagonal entries imposes an even stronger restriction. Examples of non-symmetric normal adjacency matrices include circulant matrices, which correspond to a cyclic arrangement of the nodes, and some block-circulant matrices.

5. Examples

The vector $\mathbf{u}$ in all examples of Sections 5.1 and 5.2 is chosen to be $\mathbf{1}$.

5.1. Small examples

In this subsection we give examples of small synthetic directed networks, and rank their nodes in their broadcaster and receiver roles. We compare our results with the ranking described in Section 3.1.2 for each of these networks.

5.1.1. Example 1: Simple chain

This example is illustrated in Fig. 1 for $n = 5$. The different measures discussed in this paper are summarized in Table 2.

5.1.2. Example 2: Simple chain with one branch

Regard the graph depicted in Fig. 2. Bipartization of this graph gives the nodes v_1, v_2, v_4, and v_5 the same hub rank, since they all point to only one node, that is not pointed to by any other node; see Table 3. However, the graph suggests an obvious

Table 3
Comparison of influence measures for the nodes in Fig. 2, including aggregate reachability using walks with a bounded number of reversions, as well as disregarding orientation altogether; see Section 4 for definitions.

Node v_i	$e_{ii}^{\mathcal{A}}$	ADR	BNR(d, 1)	BNR(d, 2)	
v_1	1.54	2.92	2.03	2.20	
v_2	1.54	3.33	2.79	2.95	
v_3	2.18	4.00	3.68	3.86	
v_4	1.54	2.00	2.47	2.53	
v_5	1.54	2.00	2.47	2.53	
v_6	1.00	1.00	1.65	1.55	
v_7	1.00	1.00	1.65	1.55	
Node v_i	$e_{n+i,n+i}^{\mathcal{A}}$	AUR	BNR(u, 1)	BNR(u, 2)	SAR
v_1	1.00	1.00	1.67	1.54	1.86
v_2	1.54	2.00	2.63	2.48	2.72
v_3	1.54	2.50	3.35	3.22	3.58
v_4	1.59	2.67	2.88	2.80	2.72
v_5	1.59	2.67	2.88	2.80	2.72
v_6	1.54	2.71	2.07	2.17	1.86
v_7	1.54	2.71	2.07	2.17	1.86

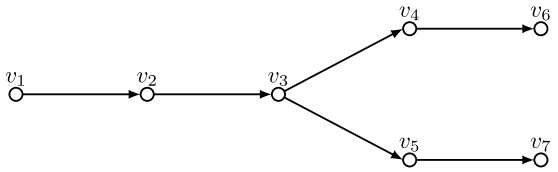

Fig. 2. Graph of Example 2.

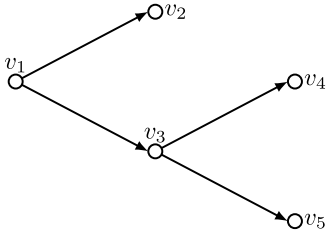

Fig. 3. Graph of Example 3.

advantage for nodes v_1 and v_2, because information from these nodes can spread to node v_3, and from there deeper into the network. The ADR ranking shows the nodes v_2 and v_1 to be the 2nd and 3rd most important broadcasters, respectively.

As far as the authority role, the nodes v_4 and v_5 get the highest ranking by the bipartization method, because of the alternating walks reaching them from v_2. However, they actually only receive information from node v_3. Using the exponential of A^T, v_6 and v_7 are ranked the highest, since they receive information from the nodes they are directly attached to, from node v_3 through a walk of length 2, from node v_2 through a walk of length 3, as well as from node v_1 through a walk of length 4. Thus, the AUR ranking determines an intuitively reasonable ordering.

The measures BNR(d, k) and BNR(u, k) identify v_3 to be the most important node for both $k = 1$ and $k = 2$, because the graph has a branch point at v_3. This example suggests that the BNR(d, k) and BNR(u, k) measures can be applied to identify important branch points. Node v_3 is the most important node also if all directed edges are replaced by undirected ones.

5.1.3. Example 3: Branching at two levels

Consider the example in Fig. 3. Using the bipartization method, Table 4 indicates that the vertices v_1 and v_3 have the same importance as hubs. However, Fig. 3 suggests that this is clearly not the case, since node v_1 broadcasts to more nodes in the network than node v_3. In fact, v_3 can reach directly two nodes v_4 and v_5, and v_1 also reaches directly two nodes v_2 and v_3, in addition to the nodes v_4 and v_5 through the hub role of node v_3. When computing the ADR ranking, the role of v_1 as a more important broadcaster is detected.

Regarding the receiver role of the nodes, we observe from Table 4 that AUR ranks the vertices v_4 and v_5 as more important receivers than v_2 and v_3, since they receive more information from the network, whereas the bipartization method does not show this difference.

Table 4
Comparison of influence measures for the nodes in Fig. 3, including aggregate reachability using walks with a bounded number of reversions, as well as disregarding orientation altogether; see Section 4 for definitions.

Node v_i	$e_{ii}^{\mathcal{A}}$	ADR	BNR(d, 1)	BNR(d, 2)	
v_1	2.18	4.00	2.91	3.25	
v_2	1.00	1.00	1.50	1.59	
v_3	2.18	3.00	3.12	3.36	
v_4	1.00	1.00	1.62	1.65	
v_5	1.00	1.00	1.62	1.65	
Node v_i	$e_{n+i,n+i}^{\mathcal{A}}$	AUR	BNR(u, 1)	BNR(u, 2)	SAR
v_1	1.00	1.00	2.25	2.04	2.67
v_2	1.59	2.00	2.12	2.01	1.85
v_3	1.59	2.00	3.12	2.94	3.25
v_4	1.59	2.50	2.28	2.26	1.99
v_5	1.59	2.50	2.28	2.26	1.99

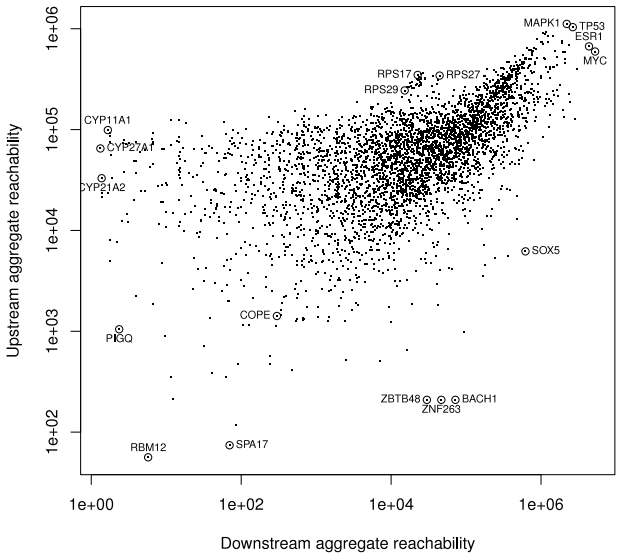

Fig. 4. Gene network: B Cell Interactome. Upstream ($\mathbf{1}^T \exp(0.25A)$) vs. downstream ($\exp(0.25A)\mathbf{1}$) aggregate reachability for genes in the network. Genes on the top right are well known, highly influential genes, like MYC and TP53. The genes on the left, like CYP27A1, perform metabolic functions but have little effect on other genes. Some ribosomal genes (RPS17, RPS27, etc.) form a small cluster. The gene COPE, which only has one incoming and one outgoing edge, appears less extreme than others.

Turning to the measures BNR(d, 1) and BNR(d, 2), we observe that both of them identify the vertices v_1 and v_3 as important; these are branch points for outflow of the graph. BNR(u, 1) and BNR(u, 2) are large for vertex v_3, because this vertex is a branch point for inflow. These measures are not large at v_1, because there is no inflow to this vertex. This example illustrates that the measures BNR(d, k) and BNR(u, k) for $k \geq 1$ reveal important information about branch points of a graph.

This subsection has compared rankings determined by bipartization, ADR, ADR, and BNR. When we are interested in broadcasters and receivers, and how information flows, the latter ranking schemes appear to be appropriate.

5.2. Real-life large examples

5.2.1. Gene regulatory network of the human B-cell interactome

We consider a network of protein–protein, protein–DNA, and modulatory interactions in human B cells [16]. There are 5737 nodes (genes/proteins) and 84,892 directed edges. Taking the matrix exponential of the adjacency matrix (multiplied by a coefficient of 0.25), we found one main strongly connected component (3891 genes) with 1833 genes downstream (grouped in singletons, pairs, triplets, or quadruplets) and 13 individual genes upstream. Analyzing the main strongly connected component, we are able to find the distribution of genes based on their aggregate downstream ($\exp(0.25A)\mathbf{1}$) and aggregate upstream ($\mathbf{1}^T \exp(0.25A)$) reachability; comparing the two, we can identify the overall role of genes as regulators, regulated, or both. Fig. 4 displays the network.

Table 5 illustrates how ADR and AUR can be used to identify important nodes in a complex directed network, and their different roles. Genes with high ADR and low AUR influence many genes, directly or indirectly, while being influenced by

Table 5
Comparison of influence measures for the nodes in the *Gene* network, including aggregate reachability using walks with a bounded number of reversions, as well as disregarding orientation altogether; see Section 4 for definitions.

Top 10 ranked nodes v_i using various measures				
e_{ii}^A	ADR	BNR(d, 1)	BNR(d, 2)	
MYC	MYC	MYC	MYC	
ESR1	ESR1	ESR1	ESR1	
CREB1	CREB1	CREB1	CREB1	
RBL2	RBL2	RBL2	RBL2	
FOXM1	TP53	TP53	TP53	
SP3	SP3	SP3	SP3	
JUND	EP300	EP300	EP300	
POU2F2	E2F4	E2F4	E2F4	
E2F4	MAPK1	MAPK1	MAPK1	
TCF1	STAT1	STAT1	STAT1	
$e_{n+i,n+i}^A$	AUR	BNR(u, 1)	BNR(u, 2)	SAR
EP300	MAPK1	MAPK1	MAPK1	MYC
CREBBP	TP53	TP53	TP53	ESR1
CDC2	GRB2	GRB2	GRB2	CREB1
PCNA	FYN	FYN	FYN	RBL2
BRCA1	CDC2	CDC2	CDC2	JUND
AURKA	SRC	SRC	SRC	SP3
CCNA2	MAPK8	MAPK8	MAPK8	FOXM1
LYN	JUNB	JUNB	JUNB	POU2F2
LTK	STAT3	STAT3	STAT3	E2F4
JUN	TRAP1	TRAP1	TRAP1	STAT1

relatively few other genes; typically, these are transcription factors, that control the expression level of many other genes through regulatory pathways (ZBTB48, ZNF263, BACH1, and SOX5 are transcription factors). Genes with low ADR and high AUR can be expected to be "workhorse" genes, which perform important duties, and are therefore controlled by many upstream genes, but do not have a regulatory function (CYP11A1, CYP27A1, and CYP21A2, for example, encode enzymes with metabolic functions). Finally, genes with high ADR and high AUR can be regarded as very central in the network, brokers of influence that collect information from many genes upstream and control the expression of many genes downstream; unsurprisingly, crucial master genes like TP53 and MAPK1 have extreme values in both measures. (The information about specific genes mentioned above was obtained from Gene Cards [17].)

An approach described by Croft and Higham [18] is closely related, but designed with the goal of extracting a hierarchical structure. In our notation, their measure becomes ADR − AUR, and it can be used to identify putatively influential transcription factors; however, it would fail to distinguish between TP53 and a relatively unimportant gene like COPE. One could define a measure for centrality given by ADR + AUR which would distinguish between TP53 and COPE; of course, ADR − AUR and ADR + AUR are essentially an axis rotation of the measures ADR and AUR. It seems clear that a full picture requires at least two separate measures. In this example the BNR measures do not seem to add much beyond what is provided by ADR and AUR, at least for the top ranked genes.

We remark that scaling of an adjacency matrix may enhance the usefulness of the ordering determined. An interpretation of graphs as oscillator networks in which the scaling coefficient corresponds to inverse temperature is provided in [19]. Although the proper choice of scale is an important issue, it falls outside the scope of this work; our choice of 0.25 as scaling factor is approximately the largest value that prevents the entries of e^A from growing to the point of numerical overflow.

5.3. Bus route network targeting specific nodes

In this example we rank the nodes according to their downstream or upstream influence on particular nodes. Consider the Kent State University main campus bus system, illustrated in Fig. 5, left panel. The route consists of four working loops: Front Campus/Summit East (in blue), Reverse Loop (in green), Gateway Loop (in orange), and Alberton (in purple). Due to road construction, Campus Loop (in red) was not running at the time of writing, and is not included in our example.

The bus stops are the nodes $v_1, v_2, \ldots, v_{10}$, which are connected by directed edges according to the map and schedule information [20]. We assign only one node to each named bus stop, even when the bus stops on both sides of the street. The graph of the directed network is shown in Fig. 5, right panel. We assign the same weight to all edges regardless of travel length.

We compute $e^A\mathbf{u}$ for two vectors $\mathbf{u}$. As mentioned in Section 3.2, the entries of the vector $\mathbf{u}$ indicate how important it is to reach each node in the network. In this example we let these entries be either 0 or 1, depending on whether we are interested in reaching a node or not.

For Table 6, we let all entries of $\mathbf{u}_1$ be zero except for the fourth entry. Thus, we are interested in ranking the nodes according to how much they contribute to reaching node v_4. Table 7 shows how much each node contributes to receiving from node v_4. This is a new way of calculating the communicability among the nodes.

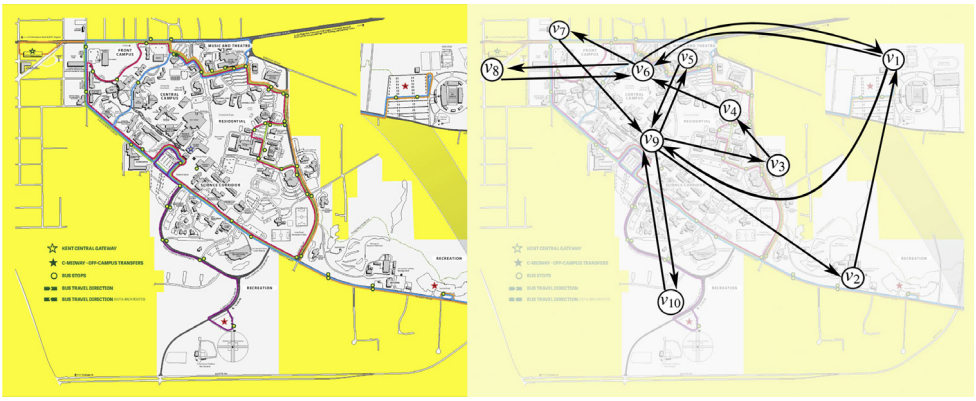

Fig. 5. Left: Kent State University main campus bus route [20]. Right: Network graph of Kent State University main campus bus route . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Comparing the top 5 ranked nodes in Fig. 5 as broadcasters where $\mathbf{u}_1 = [0,0,0,1,0,0,0,0,0,0]^T$ and $\mathbf{u}_2 = [0,0,0,1,0,0,1,0,0,0]^T$.

Broadcasting					
Node v_i	$(e^A\mathbf{1})_i$	Node v_i	$(e^A\mathbf{u}_1)_i$	Node v_i	$(e^A\mathbf{u}_2)_i$
v_9	10.00	v_4	1.02	v_4	1.61
v_6	9.56	v_3	1.00	v_6	1.47
v_1	9.31	v_9	0.60	v_7	1.20
v_7	5.34	v_1	0.20	v_3	1.19
v_{10}	5.34	v_7	0.19	v_1	0.81

Table 7
Comparing the top 5 ranked nodes in Fig. 5 as receivers, where $\mathbf{u}_1 = [0,0,0,1,0,0,0,0,0,0]^T$ and $\mathbf{u}_2 = [0,0,0,1,0,0,1,0,0,0]^T$.

Receiving					
Node v_i	$(e^{A^T}\mathbf{1})_i$	Node v_i	$(e^{A^T}\mathbf{u}_1)_i$	Node v_i	$(e^{A^T}\mathbf{u}_2)_i$
v_9	11.51	v_6	1.37	v_9	1.83
v_6	8.33	v_4	1.02	v_7	1.61
v_1	7.23	v_1	0.61	v_6	1.47
v_3	5.74	v_7	0.59	v_4	1.20
v_2	5.74	v_8	0.59	v_1	0.81

Let the vector $\mathbf{u}_2$ have the fourth and the seventh entries equal to 1 and the other entries zero. Thus, we are interested in determining the best node to place our information in order to reach nodes v_4 or v_7. This is displayed by Table 6. The best node to gather information coming from nodes v_4 or v_7 is shown by Table 7.

Take the scenario of a driver who would like to drop off four students at the university. One of the students is going to node v_4, the second one to v_7, the third to v_8, and the last one to node v_{10}. The driver can only take them to one bus stop. Where should he/she stop the car? Table 6 indicates that it is best to take the students to node v_6, from where each one of them rides the bus to his/her destination. Table 7 shows that it is best that they all ride the bus to node v_9, where the driver picks all of them up at the same time.

6. Numerical considerations

When a network has fairly few nodes and, therefore, the associated adjacency matrix A is small, evaluation of the matrix exponential is quite inexpensive. We then can calculate expressions of the forms (3.2), (3.3), (4.2) and (4.3) by first evaluating e^A and then computing the desired expression(s), where we may use that $e^{A^T} = (e^A)^T$. However, when the network has many nodes and the adjacency matrix A is large, the explicit calculation of e^A is too expensive to be attractive. This section discusses how approximations of the expressions (3.2), (3.3), (4.2) and (4.3) can be evaluated fairly inexpensively for large adjacency matrices with the aid of the Arnoldi process.

Let $\|\cdot\|$ denote the Euclidean vector norm. Application of ℓ steps of the Arnoldi process to the matrix A with initial vector $\mathbf{w} \neq \mathbf{0}$ gives the decomposition

$$AW_\ell = W_\ell H_\ell + \mathbf{g}_\ell \mathbf{e}_\ell^T, \tag{6.1}$$

where the matrix $W_\ell = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_\ell] \in \mathbb{R}^{n\times\ell}$ has orthonormal columns that span the Krylov subspace $\mathbb{K}_\ell(A, \mathbf{w}) = \text{span}\{\mathbf{w}, A\mathbf{w}, \ldots, A^{\ell-1}\mathbf{w}\}$ with $\mathbf{w}_1 = \mathbf{w}/\|\mathbf{w}\|$. The matrix $H_\ell \in \mathbb{R}^{\ell\times\ell}$ is of upper Hessenberg form, $\mathbf{g}_\ell \in \mathbb{R}^n$ satisfies $W_\ell^T\mathbf{g}_\ell = \mathbf{0}$, and $\mathbf{e}_\ell = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ denotes the ℓth column of an identity matrix of appropriate order; see, e.g., Saad [21, Chapter 6] for details on the Arnoldi process. We assume that ℓ is small enough so that the decomposition (6.1) with the stated properties exists. This is the generic situation. The computation of this decomposition requires the evaluation of ℓ matrix–vector products with the matrix A.

Expressions of the form $e^A\mathbf{w}$ are commonly approximated by the right-hand side of

$$e^A\mathbf{w} \approx W_\ell e^{H_\ell}\mathbf{e}_1\|\mathbf{w}\|,$$

see, e.g., [22,23] for discussions. In particular, we obtain an approximation of (3.2) by letting $\mathbf{w} = \mathbf{1}$. When A is large, the dominating computational work for calculating this approximation is the evaluation of the ℓ matrix–vector products required to determine the decomposition (6.1).

An approximation of the expression (3.3) can be determined similarly: we apply the Arnoldi process to the matrix A^T with initial vector $\widehat{\mathbf{w}} = \mathbf{1}$. This gives the decomposition

$$A^T\widehat{W}_\ell = \widehat{W}_\ell\widehat{H}_\ell + \widehat{\mathbf{g}}_\ell\mathbf{e}_\ell^T, \tag{6.2}$$

which is analogous to (6.1). We then evaluate the right-hand side of

$$e^{A^T}\widehat{\mathbf{w}} \approx \widehat{W}_\ell e^{\widehat{H}_\ell}\mathbf{e}_1\|\widehat{\mathbf{w}}\|. \tag{6.3}$$

When A is large, the dominating computational effort to calculate this approximation is the evaluation of the ℓ matrix–vector products with A^T needed to determine the decomposition (6.2).

We turn to the approximation of the expression (4.2) for $k = 1$. Extension to the situation when $k > 1$ is straightforward. The expression (4.3) can be computed in a similar fashion. We first compute the Arnoldi decomposition (6.2) with initial vector $\widehat{\mathbf{w}} = \mathbf{1}$ and then evaluate the Arnoldi decomposition (6.1) with initial vector $\mathbf{w} = \widehat{W}_\ell e^{\widehat{H}_\ell}\mathbf{e}_1\|\mathbf{1}\|$. This gives the approximation

$$W_\ell e^{H_\ell}\mathbf{e}_1\|\mathbf{w}\| \tag{6.4}$$

of (4.2). The following results shed some light on this approximant.

Proposition 6.1. *Let f be a polynomial of degree at most $\ell - 1$ and let $\widehat{\mathbf{w}}$ be an initial vector for the Arnoldi decomposition (6.2). Consider the approximation*

$$\widehat{W}_\ell f(\widehat{H}_\ell)\mathbf{e}_1\|\widehat{\mathbf{w}}\|$$

of $f(A^T)\widehat{\mathbf{w}}$. Use the above vector as initial vector $\mathbf{w}$ for the decomposition (6.1) and compute the approximation

$$W_\ell f(H_\ell)\mathbf{e}_1\|\mathbf{w}\| \tag{6.5}$$

of $f(A)f(A^T)\widehat{\mathbf{w}}$. Then this approximation is exact. We assume that the required Arnoldi decompositions can be computed without breakdown of the Arnoldi process.

Proof. Consider the decomposition (6.1). It is well known that for any polynomial f of degree at most $\ell - 1$, we have

$$f(A)\mathbf{w} = W_\ell f(H_\ell)\mathbf{e}_1\|\mathbf{w}\|;$$

see, e.g, [22]. Clearly, an analogous result holds if the decomposition (6.1) is replaced by the decomposition (6.2). The desired result follows. □

The approximation (6.5) of $f(A)f(A^T)\widehat{\mathbf{w}}$ requires the evaluation of 2ℓ matrix–vector products, ℓ with each one of the matrices A and A^T. When the adjacency matrix A is stored in a format that makes the evaluation of matrix–vector products with A^T more expensive than with A, it may be tempting to apply 2ℓ steps of the Arnoldi process to A with initial vector $\widehat{\mathbf{w}}$ and then use the low-rank matrices

$$e^A \approx W_{2\ell}e^{H_{2\ell}}W_{2\ell}^T, \qquad e^{A^T} \approx W_{2\ell}e^{H_{2\ell}^T}W_{2\ell}^T, \tag{6.6}$$

to approximate $e^Ae^{A^T}\widehat{\mathbf{w}}$. The evaluation of this approximation requires the same number of matrix–vector product evaluations as the approach described in Proposition 6.1. However, no analogue of this proposition is available for the approximations (6.6) and, indeed, the approximation of $e^Ae^{A^T}\widehat{\mathbf{w}}$ typically is of significantly worse quality than an approximation computed by the approach of Proposition 6.1.

We conclude this section with a comparison of the evaluation of the expression (4.2) for $k = 1$ for the matrix A of Section 5.2.1 by explicitly computing the matrix exponential e^A and by evaluating Arnoldi decompositions as described by Proposition 6.1. The matrix A is of order 3891. The computation of the matrix exponential $M = e^{A/2}$ using the MATLAB

function expm required 1222.15 s ($\approx$ 20.37 min).[1] This is the dominating work. Let $\mathbf{w} = [1, 1, \ldots, 1]^T \in \mathbb{R}^{3891}$. Having the matrix M, we can calculate BNR(d, 1) by evaluating two matrix–vector products BNR(d, 1) $= M(M^T\mathbf{w})$. The top 10 ranked nodes obtained in this manner are shown in column 3 in the top part of Table 5.

We turn to the approximation of the expression (4.2) with the aid of Arnoldi decompositions. First we compute the decomposition (6.2) for $\ell = 6$ and initial vector $\mathbf{w}$. This required only 0.037 s and gives the approximation $\widehat{\mathbf{z}} := \widehat{W}_6 e^{\widehat{H}_6}\mathbf{e}_1\|\mathbf{w}\|$ of $M^T\mathbf{w}$. The total time needed to compute $\widehat{\mathbf{z}}$ was 0.077 s. Next we evaluate the Arnoldi decomposition (6.1) with $\ell = 6$ and initial vector $\widehat{\mathbf{z}}$. The calculation of this decomposition required 0.023 s. The difference in time required to compute the decompositions (6.2) and (6.1) depends on the storage format for the matrix A used by MATLAB. The total time needed to compute an approximation of (4.2) for $k = 1$ in this manner is only 0.129 s. The top 10 ranked nodes are those of column 3 in the top part of Table 5. Thus, the application of the Arnoldi process twice with $\ell = 6$ as described by Proposition 6.1 gives the same ranking of the nodes as the evaluation of $e^{\frac{1}{2}A}e^{\frac{1}{2}A^T}\mathbf{w}$ and requires much less time.

7. Discussion and conclusion

Until now the use of matrix functions based on the exponential has not received much attention for ranking the nodes of a directed network. Differently from the situation for undirected networks, it is generally not so useful to tabulate the diagonal entries of the exponential of the adjacency matrix. This already has been observed in the literature; see, e.g., Benzi et al. [7].

An important difference between directed and undirected networks is that in the former one measure of importance is rarely sufficient. Instead, a combination of measures will often provide a more complete picture. The gene network example (Section 5.2.1) illustrates this by combining upstream and downstream aggregate reachabilities to identify genes that play different roles and are important in different ways.

We also introduced a family of reachability measures that consider walks that are allowed to change direction a finite number of times. This allows us to take into account "lateral" relationships between nodes.

Acknowledgments

This research was supported in part by the National Science Foundation, USA grants DMS-1729509 and DMS-1720259.

References

[1] E. Estrada, The Structure of Complex Networks: Theory and Applications, Oxford University Press, Oxford, 2012.
[2] M.E.J. Newman, Networks: An Introduction, Oxford University Press, Oxford, 2010.
[3] G.A. Pavlopoulos, M. Secrier, C.N. Moschopoulos, T.G. Soldatos, S. Kossida, J. Aerts, R. Schneider, P.G. Bagos, Using graph theory to analyze biological networks, BioData Min. 4 (2011) 10.
[4] M. Benzi, C. Klymko, On the limiting behavior of parameter-dependent network centrality measures, SIAM J. Matrix Anal. Appl. 36 (2015) 686–706.
[5] E. Estrada, J.A. Rodriguez-Velazquez, Subgraph centrality in complex networks, Phys. Rev. E 71 (2005) 056103.
[6] J. Baglama, C. Fenu, L. Reichel, G. Rodriguez, Analysis of directed networks via partial singular value decomposition and gauss quadrature, Linear Algebra Appl. 456 (2014) 93–121.
[7] M. Benzi, E. Estrada, C. Klymko, Ranking hubs and authorities using matrix functions, Linear Algebra Appl. 438 (2013) 2447–2474.
[8] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1953) 39–43.
[9] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM 46 (1999) 604–632.
[10] M. Benzi, P. Boito, Quadrature rule-based bounds for functions of adjacency matrices, Linear Algebra Appl. 433 (2010) 637–652.
[11] M. Benzi, C. Klymko, Total communicability as a centrality measure, J. Complex Netw. 1 (2013) 124–149.
[12] E. Estrada, D.J. Higham, Network properties revealed through matrix functions, SIAM Rev. 52 (2010) 696–714.
[13] A. Farahat, T. LoFaro, J.C. Miller, G. Rae, L.A. Ward, Authority rankings from HITS, pagerank, and SALSA: Existence, uniqueness, and effect of initialization, SIAM J. Sci. Comput. 27 (2006) 1181–1201.
[14] D.F. Gleich, Pagerank beyond the web, SIAM Rev. 57 (2015) 321–363.
[15] B. Hall, Lie Groups, Lie Algebras, and Representations: An Elementary Introduction, Springer, New York, 2003.
[16] C. Stark, B.J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, M. Tyers, Biogrid: a general repository for interaction datasets, Nucleic Acids Res. 34 (suppl. 1) (2006) D535–D539.
[17] G. Stelzer, N. Rosen, I. Plaschkes, S. Zimmerman, M. Twik, S. Fishilevich, T.I. Stein, R. Nudel, I. Lieder, Y. Mazor, S. Kaplan, D. Dahary, D. Warshawsky, Y. Guan-Golan, A. Kohn, N. Rappaport, M. Safran, D. Lancet, The geneCards suite: From gene data mining to disease genome sequence analyses, in: Current Protocols in Bioinformatics, Wiley, 2016.
[18] J.J. Crofts, D.J. Higham, Googling the brain: Discovering hierarchical and asymmetric network structures, with applications in neuroscience, Internet Math. 7 (2011) 233–254.
[19] E. Estrada, N. Hatano, M. Benzi, The physics of communicability in complex networks, Phys. Rep. 514 (2012) 89–119.
[20] PARTA, Kent State Campus Bus Service, URL http://www.partaonline.org/ride-parta/campus-bus-service/.
[21] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.
[22] B. Beckermann, L. Reichel, Error estimation and evaluation of matrix functions via the Faber transform, SIAM J. Numer. Anal. 47 (2009) 3849–3883.
[23] L.A. Knizhnerman, CaLculation of functions of unsymmetric matrices using Arnoldi's method, USSR Comput. Math. Math. Phys. 31 (1991) 1–8.

[1] All computations of this paper were carried out in MATLAB with about 15 significant decimal digits on a Lenovo ideapad 510 laptop computer with a 2.5 GHz Intel Core i7 processor and 6 GB 2133 MHz DDR4 memory.