Block Matrix Models for Dynamic Networks

In Memory of Sebastiano Seatzu.

Mohammed Al Mugahwi

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

Omar De la Cruz Cabrera

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

Caterina Fenu

Department of Mathematics and Computer Science, University of Cagliari, via Ospedale 72, 09124 Cagliari, Italy.

Lothar Reichel

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

Giuseppe Rodriguez

Department of Mathematics and Computer Science, University of Cagliari, via Ospedale 72, 09124 Cagliari, Italy.

Abstract

Networks in which connections change over time arise in many applications, e.g., when modeling phone calls and flights between airports. This paper discusses new ways to define adjacency matrices associated with this kind of networks. We propose that dynamic networks be modeled with the aid of block upper triangular adjacency matrices. Both modeling and computational aspects are discussed. Several applications to real dynamic networks are presented and illustrate the advantages of the proposed method when compared with an available approach.

Email addresses: malmugah@kent.edu (Mohammed Al Mugahwi), odelacru@kent.edu (Omar De la Cruz Cabrera), kate.fenu@unica.it (Caterina Fenu), reichel@math.kent.edu (Lothar Reichel), rodriguez@unica.it (Giuseppe Rodriguez)

Keywords: Time-dependent centrality, complex network, evolving network, graph.
2008 MSC: 05C50, 15A69.

1. Introduction

A *dynamic network*, also called a *temporal network* or *evolving network*, is a graph in which connections are formed both within and between well-determined time intervals. We can model the evolution of such a network by considering a change in the set of edges over a fixed set of nodes. Let $\{G_k\}_{k=1}^N = \big(V, \{E_k\}_{k=1}^N\big)$ be a sequence of unweighted graphs without loops and multiple edges evolving in discrete time, where V is the set of nodes of cardinality n, and E_k contains all the edges connecting the nodes in the time interval $\tau_k = [t_{k-1}, t_k)$, $k = 1, \ldots, N$. We assume the whole time period to be partitioned into N intervals, not necessarily of equal lengths. Connections in the network starting at time interval τ_i and ending at time interval τ_j are represented by the $n \times n$ adjacency matrix $A_{i,j}$, $j \geq i$. As usual for unweighted graphs without loops and multiple edges, the (r,ℓ)th entry of $A_{i,j}$ equals 1 if there is an edge from node r to node ℓ starting at τ_i and ending at τ_j, and 0 otherwise.

This kind of structure is useful when modeling both connections with a certain duration in time and those whose duration is uncertain. The first ones include phone calls, flights between airports, disease spreading, etc. The second kind of connections may be text messages, online social network interactions (Twitter, Facebook, Instagram, only to mention a few), e-mail messages, and so on; see [16] for an overview. Allowing time steps $\tau_k = [t_{k-1}, t_k)$ to have finite duration gives great generality to the model, as it allows a user to reduce the length of time intervals when an accurate localization in time is needed, and enlarge the length of the time interval for events that start and end over extended time periods.

Various recent papers discuss the definition of node centrality indices based on the notion of dynamic walks [1, 13, 14]. In [8] the authors proposed a block matrix representation in order to express centrality indices in terms of standard matrix functions. In particular, they applied the resolvent to recover some previously defined centrality indices.

The aim of this work is to present a block matrix representation for dynamic networks, whose structure is able to represent different kinds of connections, and allows one to define new centrality indices in order to identify the most important nodes in evolving networks. This representation makes it possible to recompute

centrality indices when a new time period is added at a reduced cost by a simple matrix update. Both moving time, i.e., when there is a connection between two nodes starting and ending at different time intervals, and waiting time, i.e., when a fictitious connection is added between a node and itself at different time intervals in order to simulate a stay at the same node for a certain time, can be modeled.

The plan of the paper is the following. Section 2 reports some background material that subsequently will be needed. Our model is introduced in Section 3, Section 4 discusses the issue of downweighting "old" walks, while the update of the block structure is described in Section 5. A comparison with an existing centrality measure is presented in Section 6, and Section 7 contains three real-life examples. Finally, we draw some conclusions in Section 8.

2. Background and notation

Let $|X|$ denote the cardinality of the set X. A static network, or graph, consists of two sets $G = (V, E)$, where V, with $|V| = n$, is the set of the nodes (or vertices) and E, with $|E| = m$, is the set of the edges (or links) between nodes. Each edge $e_i = (r_{k_i}, \ell_{j_i})$, $i = 1, \ldots, m$, connecting node r_{k_i} and node ℓ_{j_i}, is said to be directed if it can be traversed in only one direction; it is said to be undirected otherwise [5, 7].

A *walk* of length w from node r_{k_1} to $r_{k_{w+1}}$ is a sequence of w edges (not necessarily distinct) $(r_{k_1}, r_{k_2}), (r_{k_2}, r_{k_3}), \ldots, (r_{k_w}, r_{k_{w+1}})$. A network is said to be (strongly) connected if any two nodes are connected by a (directed) walk; see [7].

A static graph with n nodes can be represented by a square matrix $A = [a_{r,\ell}] \in \mathbb{R}^{n \times n}$, which is referred to as the adjacency matrix associated with the graph. Its entry $a_{r,\ell}$ is nonzero if there is an edge from node r to node ℓ. In this work we will, unless explicitly stated otherwise, focus on unweighted graphs without self-loops and multiple edges, i.e., we consider all edges to have the same importance. Moreover, entries $a_{r,\ell}$ are either zero or one. If the graph is undirected, then also $a_{\ell,r} = a_{r,\ell}$.

An application that has gained considerable attention in graph theory in recent years is the determination of the most important node(s) of a network according to a given ranking rule. In the case of directed networks, as well as in dynamic networks, a node can be important either as a broadcaster or as a receiver of information. The following lemma is a well known result from graph theory [6].

Lemma 2.1. *Let A^w denote the wth power of the adjacency matrix A of an unweighted graph. The quantity $(A^w)_{r,\ell}$ counts the number of walks of length w starting at node r and ending at node ℓ. If $r = \ell$, then $(A^w)_{r,r}$ counts the number*

of cycles based on node r, that is, the number of closed walks starting and ending at node r.

A dynamic network is a sequence of unweighted graphs evolving in discrete time

$$G_k = (V, E_k), \qquad k = 1, \ldots, N.$$

Here, the set of nodes V, with $|V| = n$, is fixed and the evolution in time is given by the changes in the sets of edges E_k. In the case of dynamic networks, given an ordered sequence of time points $\{t_k\}_{k=1}^{N}$, authors of [14] consider the network at time t_k to be represented by its $n \times n$ adjacency matrix $A^{[k]}$. As usual for unweighted networks, the (r,ℓ)th entry of $A^{[k]}$ equals 1 if there is an edge from node r to node ℓ at time t_k, and 0 otherwise. In the same work, the authors introduce the concept of a *dynamic walk* as follows.

Definition 2.1. *A* dynamic walk *of length w from node r_{k_1} to node $r_{k_{w+1}}$ consists of a sequence of edges (r_{k_1}, r_{k_2}), (r_{k_2}, r_{k_3}), …, $(r_{k_w}, r_{k_{w+1}})$ and a nondecreasing sequence of times $t_{k_1} \le t_{k_2} \le \cdots \le t_{k_w}$ such that $(A^{[k_s]})_{r_{k_s}, r_{k_{s+1}}} \neq 0$, $s = 1, \ldots, w$.*

This definition was used in [14] to introduce the *dynamic communicability matrix*

$$\mathcal{Q}^{[j]} = \left(I - aA^{[1]}\right)^{-1} \left(I - aA^{[2]}\right)^{-1} \cdots \left(I - aA^{[j]}\right)^{-1} = \prod_{s=1}^{j} \left(I - aA^{[s]}\right)^{-1}. \quad (2.1)$$

The factor $(I - aA^{[s]})^{-1}$, known as a resolvent of $A^{[s]}$, exists for $1 \le s \le j$, if the parameter a satisfies $0 \le a < 1/\max_s \rho(A^{[s]})$, where $\rho(A^{[s]})$ denotes the spectral radius of the matrix $A^{[s]}$. We assume this to be the case. Then the resolvent of $A^{[s]}$ may be expanded as

$$\left(I - aA^{[s]}\right)^{-1} = \sum_{w=0}^{\infty} a^w \left(A^{[s]}\right)^w.$$

In other words, $\left(\mathcal{Q}^{[j]}\right)_{r,\ell}$ may be considered a weighted sum of the number of dynamic walks from node r to node ℓ by taking into account the ordered sequence $\{A^{[k]}\}_{k=1}^{N}$. Walks of length w are scaled by a^w. Finally, the overall ability of the nodes to broadcast or receive information in this sense is given by the row and column sums

$$C^{\text{broadcast}} = \mathcal{Q}^{[j]}\mathbf{1} \qquad \text{and} \qquad C^{\text{receive}} = \mathcal{Q}^{[j]^T}\mathbf{1}, \quad (2.2)$$

respectively, where $\mathbf{1}$ is the vector of all ones; see [14] for a more detailed explanation, including a discussion on the choice of the parameter a.

In many applications, recently started walks may be considered more important than those started a long time ago. For this reason, the authors in [13] introduced the *running dynamic communicability matrix* $\mathscr{S}^{[j]}$, obtained recursively, starting from $\mathscr{S}^{[0]} = 0$, as

$$\mathscr{S}^{[j]} = \left(I + e^{-b\Delta t_j}\mathscr{S}^{[j-1]}\right)\left(I - aA^{[j]}\right)^{-1} - I, \qquad j = 1,\ldots,N, \tag{2.3}$$

where $\Delta t_j = t_j - t_{j-1}$. In this recurrence, the parameter $0 < a < \min_{1\le j\le N} 1/\rho(A^{[j]})$ is used to penalize long walks and the parameter $b \ge 0$ to filter out old activities. Overall, $\mathscr{S}^{[j]}$ maintains walk counts that are scaled by a^w, in terms of length w, and by e^{-bt}, in term of chronological age t. Running versions of the broadcast and receive communicabilities are then given by the row/column sums of the matrix $\mathscr{S}^{[j]}$, that is

$$\mathscr{S}^{[j]}\mathbf{1} \qquad \text{and} \qquad \mathscr{S}^{[j]^T}\mathbf{1}. \tag{2.4}$$

In [8], given a sequence of adjacency matrices $\{A^{[k]}\}_{k=1}^{N}$, the authors proposed the application of the block bidiagonal matrix

$$B := \begin{bmatrix} \alpha A^{[1]} & \beta_2 I & & & \\ & \alpha A^{[2]} & \beta_3 I & & \\ & & \ddots & \ddots & \\ & & & \alpha A^{[N-1]} & \beta_N I \\ & & & & \alpha A^{[N]} \end{bmatrix}. \tag{2.5}$$

By using this structure, they were able to express the quantities in (2.2) and (2.4) as actions of standard matrix functions on a vector. This approach allowed them to apply efficient computational techniques for their computation; see [8] for details.

We remark that the ideas presented here are different from the models available in the literature [1, 8, 12, 13, 14]. In particular, we provide a block model for networks that the authors of [16] refer to as *interval graphs*, i.e., networks whose evolution has a certain duration in time, in order to distinguish between short and long edges. We note, however, that this terminology is not appropriate in the present paper, since interval graphs are graphs in which the nodes are intervals of the real line; see, for example, [11].

3. Block models

We consider dynamic networks discretized into a finite number of time periods. As already mentioned, we assume the time period that we are interested in to be partitioned into N time intervals, not necessarily of equal lengths, $\tau_k = [t_{k-1}, t_k)$, $k = 1, \ldots, N$. The $n \times n$ adjacency matrices $A_{i,j}$, defined in the Introduction and representing connections starting at interval τ_i and ending at interval τ_j, may be, and in many applications are, nonsymmetric. Indeed, if a connection (r, ℓ) is active in the time interval $[t_{i-1}, t_j)$, then this is not true, in general, for the reverse connection (ℓ, r).

In the following, we will present two block models that represent a dynamic network in the whole time period that we are considering. Each block matrix is specific for a given kind of connection, depending on the position of the blocks $A_{i,j}$. We will use superscripts to indicate the position of the block within the block matrix and subscripts to indicate the element within each block. In particular, if we denote the block matrix that models a dynamic network during a time period partitioned into N time intervals by B_N, then $(B_N)^{i,j}_{r,\ell}$ stands for element (r, ℓ) of the block (i, j).

By summing over all the powers w and downweighting walks of length w by $1/w!$, we obtain the exponential e^{B_N}. The (r, ℓ)th element of each block $(e^{B_N})^{i,j}$ is the weighted sum of all the walks from node r to node ℓ starting at time τ_i and ending at time τ_j, where the walks of length w are weighted by $1/w!$. Each block-row $(e^{B_N})^{i,\cdot}$ accounts for all the connections starting at time τ_i, while each block-column $(e^{B_N})^{\cdot,j}$ contains information about the connections ending at time τ_j.

Similarly to [13, 14], we use the row and column sums of the block matrix e^{B_N} to quantify the importance of the nodes as broadcasters or receivers of information, respectively. In particular, the broadcast and receive indices of node i at time k are given by

$$\text{broad}_{i,k} = (e^{B_N}\mathbf{1})_{(k-1)N+i} \quad \text{and} \quad \text{rec}_{i,k} = (e^{B_N^T}\mathbf{1})_{(k-1)N+i}, \qquad \begin{array}{l} k = 1, \ldots, N, \\ i = 1, \ldots, n, \end{array} \tag{3.1}$$

respectively. The sum

$$\mathscr{B}^i_r = \sum_{j=i}^{N} \sum_{\ell=1}^{n} (e^{B_N})^{i,j}_{r,\ell} = \left[(e^{B_N})^{i,\cdot}\,\mathbf{1}\right]_r, \qquad \begin{array}{l} r = 1, \ldots, n, \\ i = 1, \ldots, N, \end{array}$$

can be seen as a measure of the importance of the node r as a source, starting from time τ_i, in which walks of length w have been weighted by a factor $1/w!$.

Similarly, the sum

$$\mathscr{R}_\ell^j = \sum_{i=1}^{j}\sum_{r=1}^{n}(e^{B_N})_{r,\ell}^{i,j} = \left[(e^{B_N^T})^{j,\cdot}\,\mathbf{1}\right]_\ell, \qquad \begin{matrix}\ell = 1,\dots,n,\\ j = 1,\dots,N,\end{matrix}$$

may be considered a measure of the importance of node ℓ as a sink up to time τ_j, in which walks of length w have been weighted by a factor $1/w!$. It is worth noting that (3.1) is a temporal extension of the total (node) communicability introduced in [2].

Other kinds of node centralities also may be interesting. For example, one might look at the information that is sent from node r up to time τ_j, or, conversely, that is received by node ℓ starting at time τ_i. The computation of these quantities can be carried out by using the techniques from [9]. The challenging case is when the explicit computation is difficult, that is, when $Nn \gg 1$. This happens when either the number of nodes or of time intervals is large. We distinguish two significant cases, that is, when either N or n is small enough:

- If n is small and N is large, then we study a small network over many time intervals τ_i. In this case, it may be interesting to investigate the behavior of all the nodes in a given time interval. This leads to the analysis of the ith block-row or of the jth block-column of a matrix function $f(B_N)$. This situation is illustrated in Subsection 7.3, where the function f is chosen to be the matrix exponential.

- If N is small and n is large, then we observe a large network over a few time intervals. In this case, we may be interested in identifying the "most important nodes" according to their ability to broadcast or receive information during the period $[t_1, t_N]$, that is, in solving a node ranking problem. In some situations, one may want to subdivide the time interval $[t_1, t_N]$ into subintervals and solve the node ranking problem in each subinterval. This is illustrated in Subsection 7.2.

3.1. General block model

The block upper triangular matrix

$$B_N := \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & \cdots & A_{1,N} \\ 0 & A_{2,2} & & & \\ & \ddots & \ddots & & \vdots \\ & & 0 & A_{N-1,N-1} & A_{N-1,N} \\ & & & 0 & A_{N,N} \end{bmatrix} \in \mathbb{R}^{Nn\times Nn} \tag{3.2}$$

is a more general representation of the entire dynamic network than in previous work. The blocks in row i have edges starting at the interval τ_i and ending at any interval τ_j, $j = i, \ldots, N$. The main difference between the two representations (2.5) and (3.2) is that the first one does not take into account the duration in time of each edge, but only considers whether edges are present at a certain time t_i.

Both the type of connections and the discretization of the time period determine the nature of the blocks. For example, if the network represents phone calls during a whole day and each time interval concerns the phone calls that occurred in each of the 24 hours, we may have connections both within the same time interval as well as between two different time intervals. If we change the discretization, for example, by considering a month as the whole time period and each day as a time interval, then it is unlikely to find a phone call that starts during one day and ends the day after. Moreover, it is almost impossible to consider phone calls starting at time step τ_i and ending at time step τ_{i+j}, with $j > 1$.

Figure 1 and Figure 2 represent the same synthetic dynamic network modeling 15 phone calls within a group of five people during one hour; see Table 1. In Figure 1 each time step lasts for 15 minutes, while in Figure 2 it lasts for 10 minutes. In this case, the nature of the connections yields undirected networks. Indeed, if node r is involved in a phone call with node s, then node s is involved with node r. This is not true in general, for example, in networks that model flight connections. It is worth noting that a connection between nodes r and s starting and ending at the same time step τ_i, is treated, as usual, by placing two entries equal to 1 in the corresponding positions, that is, $(A_{i,i})_{r,s} = (A_{i,i})_{s,r} = 1$. If the same connection starts at τ_i and ends at a different time step τ_j, then we set $(A_{i,j})_{r,s} = (A_{i,j})_{s,r} = 1$, and this leads to the presence of two distinct edges in Figures 1 and 2.

The (i, j)th block of the wth power of the matrix B_N contains the number of dynamic walks of length w starting at time interval τ_i and ending at time interval τ_j. For example, here is the form of some of the blocks of the 3rd power of the matrix B_3:

$$
\begin{aligned}
(B_3^3)^{i,i} &= A_{i,i}^3, \qquad i = 1, \ldots, N, \\
(B_3^3)^{1,2} &= A_{1,1}^2 A_{1,2} + A_{1,1} A_{1,2} A_{2,2} + A_{1,2} A_{2,2}^2, \\
(B_3^3)^{1,3} &= A_{1,1}^2 A_{1,3} + A_{1,1} A_{1,2} A_{2,3} + A_{1,2} A_{2,2} A_{2,3} + A_{1,1} A_{1,3} A_{3,3} \\
&\quad + A_{1,2} A_{2,3} A_{3,3} + A_{1,3} A_{3,3}^2, \\
(B_3^3)^{2,3} &= A_{2,2}^2 A_{2,3} + A_{2,2} A_{2,3} A_{3,3} + A_{2,3} A_{3,3}^2.
\end{aligned}
$$

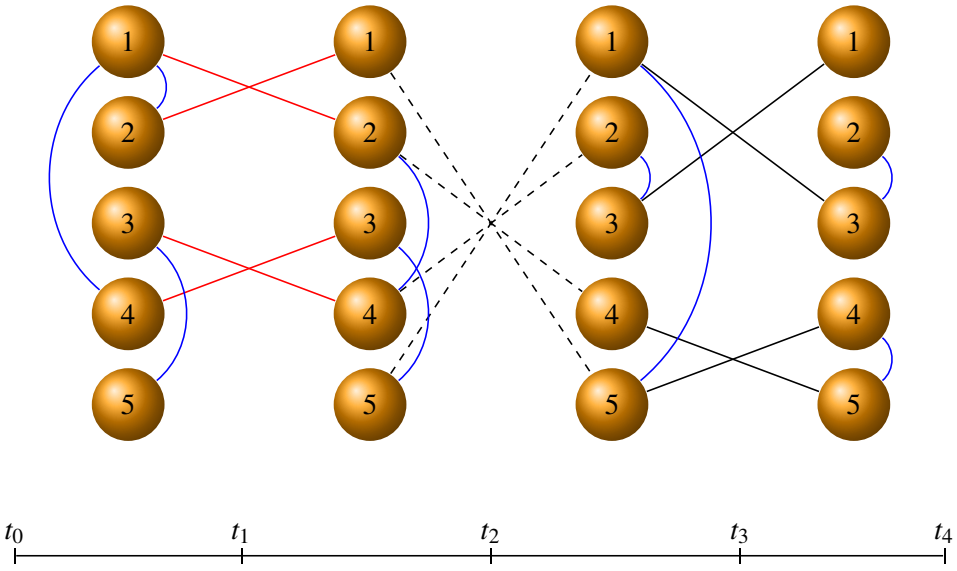

Figure 1: Phone calls between 5 people during a time period of one hour. Each time step lasts 15 minutes.

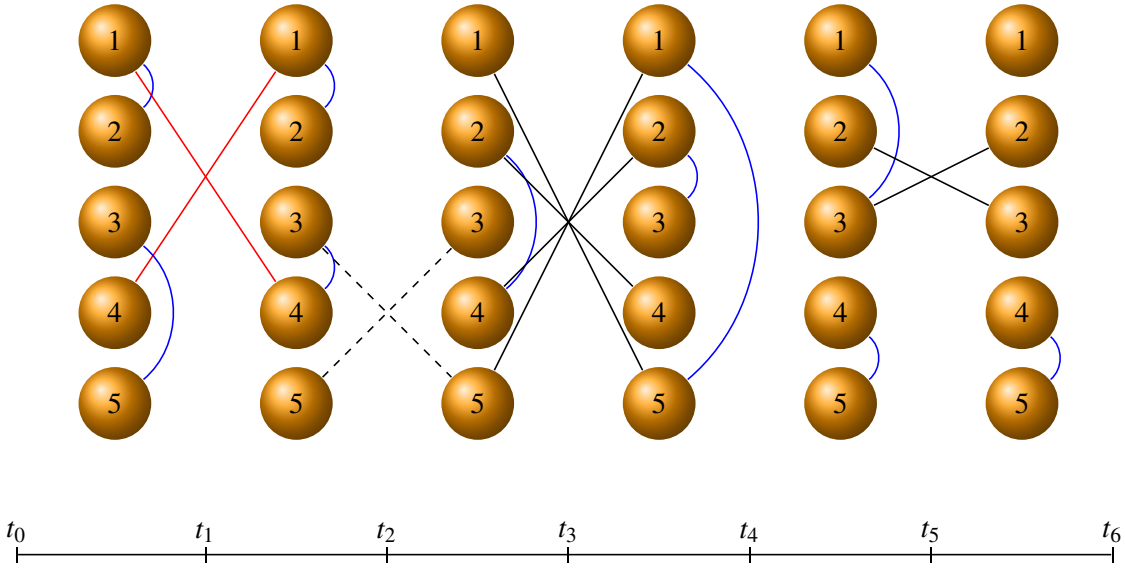

Figure 2: Phone calls between 5 people during a time period of one hour. Each time step lasts 10 minutes.

Table 1: Phone calls represented by the network in Figure 1.

9:00 - 9:14	9:15 - 9:29	9:30 - 9:44	9:45 - 10:00
3 – 5 (9:03 - 9:04)	3 – 5 (9:20 - 9:21)	2 – 3 (9:33 - 9:36)	2 – 3 (9:49 - 9:51)
1 – 2 (9:05 - 9:07)	2 – 4 (9:23 - 9:25)	1 – 5 (9:33 - 9:38)	4 – 5 (9:53 - 9:55)
1 – 4 (9:08 - 9:11)	1 – 5 (9:28 - 9:32)		
1 – 2 (9:11 - 9:20)		1 – 3 (9:44 - 9:48)	
3 – 4 (9:14 - 9:17)		4 – 5 (9:44 - 9:46)	
	2 – 4 (9:28 - 9:31)		

In the special case when no connections start and end during the same time period, the adjacency matrix for the whole time interval has the form

$$B_N := \begin{bmatrix} 0 & A_{1,2} & \dots & A_{1,N} \\ & \ddots & \ddots & \vdots \\ & & 0 & A_{N-1,N} \\ & & & 0 \end{bmatrix} \in \mathbb{R}^{Nn \times Nn}. \tag{3.3}$$

In this case, the blocks of the second and third powers of B_4 have the form

$$\begin{aligned}
&(B_4^2)^{i,j} = 0, \qquad i = \max(j-1,1), \dots, 4, \ j = 1, \dots, 4, \\
&(B_4^2)^{1,3} = A_{1,2}A_{2,3}, \\
&(B_4^2)^{1,4} = A_{1,2}A_{2,4} + A_{1,3}A_{3,4}, \\
&(B_4^2)^{2,4} = A_{2,3}A_{3,4}, \\
&(B_4^3)^{i,j} = 0, \qquad i = \max(j-2,1), \dots, 4, \ j = 1, \dots, 4, \\
&(B_4^3)^{1,4} = A_{1,2}A_{2,3}A_{3,4}.
\end{aligned}$$

An illustration is provided by Figure 3, which represents a dynamic network that models flight connections between airports.

3.2. Connections with an uncertain duration in time

Online social network interactions, such as emails and text messages, can be represented as dynamic networks characterized by connections exclusively of an uncertain duration in time. In fact, in this case, it is not meaningful to consider

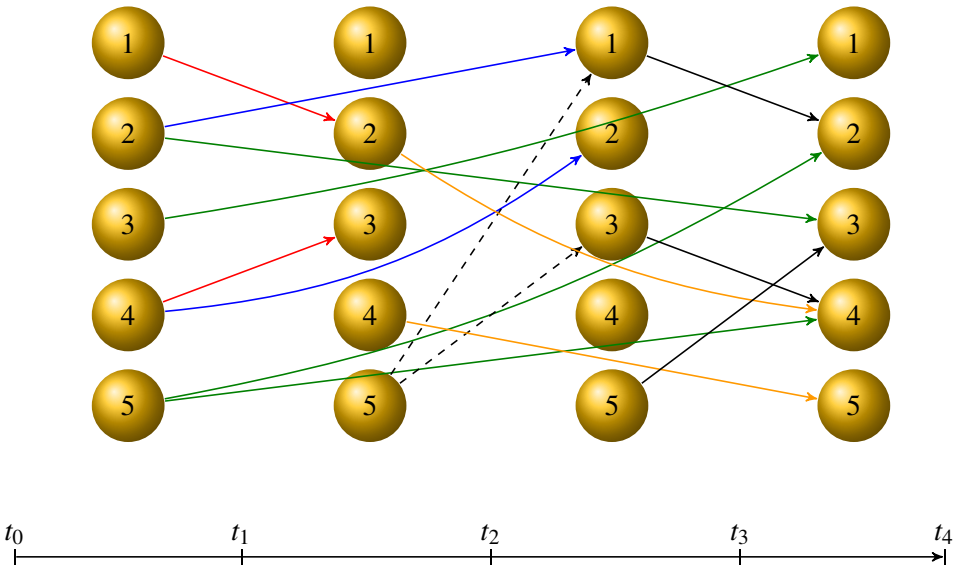

Figure 3: Flight connections between 5 airports during a time period of four hours. Each time step lasts one hour.

starting and ending times since these kind of connections do not always require a reply from the receiver, unless there is an interest in detecting when the message will actually be read, or when the receiver will take an action in response to it. In this case, any discretization of the time period is irrelevant since the information can reach any connected node in any moment and remains available indefinitely. For this reason, we consider each edge between two nodes r and ℓ at time τ_i as a link starting at time period τ_i and ending at time period τ_{i+1}.

Let $t_1 < t_2 < \cdots < t_N$ be a suitable "discretization" of time and let us denote, for notational simplicity, the adjacency matrices $A_{i,i+1}$ by A_i for $i = 1, \ldots, N-1$. Then, the adjacency matrix for this kind of dynamic network is of block super-diagonal form

$$\widehat{B}_N := \begin{bmatrix} 0 & A_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & A_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{N-2} & 0 \\ 0 & 0 & 0 & \cdots & 0 & A_{N-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{3.4}$$

In particular, $(\widehat{B}_N)^{i,i+1} = A_i \in \mathbb{R}^{n\times n}$ and $(\widehat{B}_N)^{i,j} = 0$ for every $j \neq i+1$. Then

$(\widehat{B}_N)_{r,\ell}^{i,i+1} = 1$ if and only if there is an edge between node r and node ℓ at time τ_i.

Consider the powers of the matrix $\widehat{B}_N$. Each block in the matrix

$$\widehat{B}_N^2 = \begin{bmatrix} 0 & 0 & A_1A_2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & A_2A_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & A_{N-2}A_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

is such that

$$(\widehat{B}_N^2)^{i,j} = \delta_{i,j-2}A_iA_{i+1}, \qquad i = 1,\ldots,N-1, \quad j = 1,\ldots,N,$$

and 0 otherwise, where $\delta_{k,\ell}$ denotes the Kronecker delta, and $(\widehat{B}_N^2)_{r,\ell}^{i,i+2}$ is the number of dynamic walks of length 2 from node r to node ℓ starting at time τ_i. For a generic positive integer power w of $\widehat{B}_N$, we have

$$(\widehat{B}_N^w)^{i,j} = \delta_{i,j-w} \prod_{p=i}^{i+w-1} A_p, \qquad w = 1,\ldots,N,$$

and $(\widehat{B}_N^w)_{r,\ell}^{i,i+w}$ is the number of dynamic walks of length w from node r to node ℓ starting at time τ_i.

Since $\widehat{B}_N^w = 0$ for every $w \geq N$, the exponential function becomes the finite sum

$$e^{\widehat{B}_N} = \sum_{w=0}^{N-1} \widehat{B}_N^w / w!, \tag{3.5}$$

where $1/w!$ is used to downweigh longer (that is, older) walks. In particular,

$$e^{\widehat{B}_N} = \begin{bmatrix} I & A_1 & A_1A_2/2! & \cdots & \prod_{p=1}^{N-1} A_p/(N-1)! \\ 0 & I & A_2 & \cdots & \prod_{p=2}^{N-1} A_p/(N-2)! \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{N-1} \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}.$$

We note that the data structures presented in this paper do not require much computer memory. When each network is large and only has fairly few edges

(which is a common situation), the matrix B_N can be stored as a sparse matrix. If the networks are small and dense, all the matrices $A_{i,j}$ can be stacked in a column, and the product of B_N times a vector can be evaluated by a suitably designed function. The latter situation happens, for example, in electroencephalography, where measurements are taken on a small number of channels, but are repeated over a quite large time interval.

3.3. Waiting edges

By using the block models (3.2) and (3.4), we take into account only dynamic walks for which each edge is present in a given interval, that is we only consider "moving" edges. If we want to allow edges to stay at the same node "waiting" for several time periods before moving to another node, then we can add a link between a node and itself across successive time intervals. In other words, we replace $A_{i,i+1}$ by $A_{i,i+1}+I$ in (3.2) and A_i by A_i+I in (3.4), that is, we add self-loops to each superdiagonal block. We will talk about "waiting" edges by referring to this kind of self-loops. Adding an identity to each superdiagonal block implies that all nodes that determine this block are equipped with self-loops.

4. Downweighting old walks and short time intervals

By applying the matrix exponential function to the block matrix (3.2), each walk of length w is downweighed by a factor $1/w!$, regardless of the beginning time period τ_i or the ending time period τ_j. Walks that started at an earlier time period have the same weight as walks that start later. The same happens with the ending time periods. Let us consider the matrix B_3 discussed above. All walks in B_3^3 are of length 3 and are downweighed by 1/3!, but they have three different beginning time periods: τ_1, τ_2, and τ_3.

One way of downweighting older walks (walks that start during an earlier time period) is by scaling the block rows of e^{B_N}. Define the block diagonal matrix

$$\mathscr{D}_L = \mathrm{diag}(a_1 I_n, a_2 I_n, \ldots, a_N I_n), \tag{4.1}$$

where the a_i can be chosen at will, and consider $\mathscr{D}_L e^{B_N}$. Then, the ith block-row of e^{B_N} is multiplied by a_i and walks of the same length starting at different times (say, τ_1 and τ_3) get different weights (a_1 and a_3).

If we are interested in the ending time period, rather than in the beginning time period, then we define

$$\mathscr{D}_R = \mathrm{diag}(b_1 I_n, b_2 I_n, \ldots, b_N I_n), \tag{4.2}$$

and work with the matrix $e^{B_N}\mathscr{D}_R$.

A similar approach may be used to weigh time intervals of distinct length differently. To act on the starting intervals, the scaling factors in (4.1) may be chosen as

$$a_i = \alpha(t_i - t_{i-1})^\beta, \qquad i = 1,\dots,N,$$

where α is a chosen scaling factor and $\beta = 1$ or -1, depending on the desire to give lesser or larger weights to smaller time intervals. The same definition can be applied to (4.2) to weigh the end intervals.

The model introduced allows one also to penalize walks with respect to the duration of the single edge. For example, if we are interested in giving more importance to connections which last longer, we define the matrix

$$D_k = \mathrm{diag}(c_k I_n, k), \qquad k = 0,\dots,N-1,$$

where $\mathrm{diag}(M_n,k)$ denotes the $Nn \times Nn$ matrix whose kth upper block-diagonal contains the block M_n repeated $N-k$ times. Then, we work with the matrix $\mathscr{T}\,\square e^{B_N}$, where

$$\mathscr{T} = D_0 + D_1 + \cdots + D_{N-1}, \tag{4.3}$$

and $\square$ denotes the block Hadamard product between matrices defined in [17]. Thus, let the block matrices $M = [M_{i,j}]_{i,j=1}^k \in \mathbb{R}^{kn\times kn}$ and $\widehat{M} = [\widehat{M}_{i,j}]_{i,j=1}^k \in \mathbb{R}^{kn\times kn}$ have the blocks $M_{i,j}, \widehat{M}_{i,j} \in \mathbb{R}^{n\times n}$. Then $M\square\widehat{M} = [M_{i,j}\widehat{M}_{i,j}]_{i,j=1}^k \in \mathbb{R}^{kn\times kn}$.

The following example illustrates the application of the proposed approaches. Let $N = 4$, $\mathscr{D}_L = \mathrm{diag}(\rho^3 I_n, \rho^2 I_n, \rho I_n, I_n)$, with $0 < \rho < 1$, and

$$e^{B_4} = \begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} & C_{1,4} \\ 0 & C_{2,2} & C_{2,3} & C_{2,4} \\ 0 & 0 & C_{3,3} & C_{3,4} \\ 0 & 0 & 0 & C_{4,4} \end{bmatrix}.$$

Then

$$\mathscr{D}_L e^{B_4} = \begin{bmatrix} \rho^3 C_{1,1} & \rho^3 C_{1,2} & \rho^3 C_{1,3} & \rho^3 C_{1,4} \\ 0 & \rho^2 C_{2,2} & \rho^2 C_{2,3} & \rho^2 C_{2,4} \\ 0 & 0 & \rho C_{3,3} & \rho C_{3,4} \\ 0 & 0 & 0 & C_{4,4} \end{bmatrix}.$$

Now let instead $\mathscr{D}_R = \mathrm{diag}(\rho^3 I_n, \rho^2 I_n, \rho I_n, I_n)$. Then

$$e^{B_4}\mathscr{D}_R = \begin{bmatrix} \rho^3 C_{1,1} & \rho^2 C_{1,2} & \rho C_{1,3} & C_{1,4} \\ 0 & \rho^2 C_{2,2} & \rho C_{2,3} & C_{2,4} \\ 0 & 0 & \rho C_{3,3} & C_{3,4} \\ 0 & 0 & 0 & C_{4,4} \end{bmatrix}.$$

Eventually, let $\mathscr{T} = \mathrm{diag}(\omega^3 I_n, 0) + \mathrm{diag}(\omega^2 I_n, 1) + \mathrm{diag}(\omega I_n, 2) + \mathrm{diag}(I_n, 3)$. Then,

$$\mathscr{T} \square e^{B_4} = \begin{bmatrix} \omega^3 C_{1,1} & \omega^2 C_{1,2} & \omega C_{1,3} & C_{1,4} \\ 0 & \omega^3 C_{2,2} & \omega^2 C_{2,3} & \omega C_{2,4} \\ 0 & 0 & \omega^3 C_{3,3} & \omega^2 C_{3,4} \\ 0 & 0 & 0 & \omega^3 C_{4,4} \end{bmatrix}.$$

We see that considering $\mathscr{D}_L e^{B_N}$ and $e^{B_N}\mathscr{D}_R$ allows one to separate downweighting with respect to walk length $(1/w!)$ from downweighting with respect to age (ρ^j). Instead, by considering $\mathscr{T} \square e^{B_N}$ we can distinguish between penalization with respect to walk length $(1/w!)$ and edge length (ω^j).

5. Computation and updating

One of the main computational issues when dealing with dynamic networks is updating of the quantities (3.1), when an extra time period is available, without recomputing them.

The new adjacency matrix B_{N+1} can be obtained by appending a new block-column to B_N and adjusting the size. Let

$$B_{N+1} := \begin{bmatrix} B_N & V_{N+1} \\ 0 & A_{N+1,N+1} \end{bmatrix} \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{5.1}$$

where $V_{N+1} = [A_{1,N+1}^T, A_{2,N+1}^T, \ldots, A_{N,N+1}^T]^T \in \mathbb{R}^{Nn \times n}$. The powers of B_{N+1} can be written as

$$B_{N+1}^w := \begin{bmatrix} B_N^w & \sum_{j=0}^{w-1} B_N^j V_{N+1} A_{N+1,N+1}^{w-j-1} \\ 0 & A_{N+1,N+1}^w \end{bmatrix} = \begin{bmatrix} B_N^w & S_{N,w} \\ 0 & A_{N+1,N+1}^w \end{bmatrix}. \tag{5.2}$$

We use the relation $S_{N,w} = B_N S_{N,w-1} + V_{N+1} A_{N+1,N+1}^{w-1}$ to determine the last block column of B_{N+1}^w (for $w = 2, 3, \ldots$) with $S_{N,1} = V_{N+1}$ [18]. Then, the matrix exponential of B_{N+1} can be updated as follows

$$e^{B_{N+1}} := \begin{bmatrix} e^{B_N} & X_N \\ 0 & e^{A_{N+1,N+1}} \end{bmatrix} \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{5.3}$$

with $X_N = \sum_{w=1}^{\infty} S_{N,w}/w!$.

It is also possible to update the broadcast centrality $e^{B_N}\mathbf{1}$ when a new time step is added. In this case, we have

$$e^{B_{N+1}}\mathbf{1} = \begin{bmatrix} e^{B_N}\mathbf{1} + X_N\mathbf{1} \\ e^{A_{N+1,N+1}}\mathbf{1} \end{bmatrix},$$

where $\mathbf{1}$ is the vector of all ones of appropriate size.

The algorithm for the computation of the new block-column X_N in matrix $e^{B_{N+1}}$ is described in [18], where the authors present a scaling and squaring method for the update of the exponential of a block triangular matrix. An alternative algorithm based on an iterative method is reported in Algorithm 1, which can be applied recursively starting from $B_1 = A_{1,1}$.

Algorithm 1 Recursive computation of the broadcast centrality.

Require: Starting block $A_{1,1}$ of size n, successive updates $(V_k, A_{k,k})$, $k = 2,3,\ldots$
Require: tolerance τ, maximum number of iterations $i_{\max}$
Ensure: block matrix B_N (3.2), broadcast centrality vector $\mathbf{y} = e^{B_N}\mathbf{1}$

$B = A_{1,1}$, $\mathbf{y} = e^{B}\mathbf{1}$
$k = 1$
while a new update $(V_{k+1}, A_{k+1,k+1})$ is available **do**
  $k = k+1$
  $S = V_k$, $X = V_k$, $P = I_n$
  flag $= 1$, $i = 1$, fact $= 1$
  while flag **and** $i < i_{\max}$ **do**
    $i = i+1$, fact $= i \cdot$ fact
    $P = A_{k,k} \cdot P$
    $S = B \cdot S + V_k \cdot P$
    $X = X + S/\text{fact}$
    flag $= \|S\|_\infty > \tau \cdot \text{fact} \cdot \|X\|_\infty$
  end while
  $\mathbf{y} = \begin{bmatrix} \mathbf{y} + X\mathbf{1} \\ e^{A_{k,k}}\mathbf{1} \end{bmatrix}$, $B = \begin{bmatrix} B & V_k \\ O & A_{k,k} \end{bmatrix}$
end while

Recomputing centrality indices is particularly useful in a real-time monitoring setting. In this situation, when information about the time interval τ_{N+1} becomes available, the matrix B_N is augmented by a block-row and a block-column as in (5.1). To follow the network evolution, we proceed as follows. Let us assume a connection exists between node r and node ℓ, started at time τ_i, $i \le N$, which does not end at time step τ_N. To keep track of this, we introduce a temporary block-column $\widetilde{V}_N = [\widetilde{A}_{1,N}^T, \widetilde{A}_{2,N}^T, \ldots, \widetilde{A}_{N,N}^T]^T \in \mathbb{R}^{Nn \times n}$ and set $(\widetilde{A}_{i,N})_{r,\ell} = 1$. Then, if the connection is still active at the end of time interval τ_{N+1}, we set $(\widetilde{A}_{i,N+1})_{r,\ell} = 1$ in $\widetilde{V}_{N+1}$. If, on the contrary, the connections ends at step τ_{N+1}, we set $(\widetilde{A}_{i,N+1})_{r,\ell} = 0$ and $(A_{i,N+1})_{r,\ell} = 1$ to update the network dynamically. The temporary array $\widetilde{V}_N$

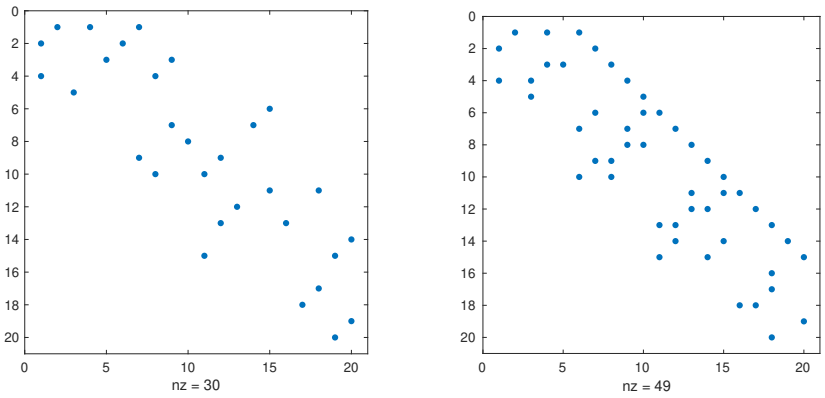

Figure 4: Spy plots of the adjacency matrix associated to the network represented in Figure 1 by using the block model (3.2) (left) and by using the block model defined in [8] (right).

can be discarded when time step τ_{N+1} ends.

6. Comparison with existing centrality measures

To validate the introduction of the new block model, we compare it with existing centrality measures. In particular, we compute the broadcast index for all the nodes in the phone call network in Figure 1 by using both the exponential defined in this paper and the resolvent defined in [13, 14]. We recall that centrality indices defined in [13, 14] do not take into account the duration of the connections, but only consider whether a connection is present or not at a certain time step.

Figure 4 reports the spy plot associated to the network represented in Figure 1. In this kind of graph each nonzero entry of the adjacency matrix is represented by a blue dot, and the quantity "nz" on the x-axis denotes the number of nonzero entries. The plot on the left-hand side is drawn by using the block model (3.2) and the plot on the right-hand side by using the block model defined in [8]. We remark that, while the new block model contains a number of nonzero elements equal to the number of connections in the network (in this case doubled since the network is undirected), the block model in [8] does not have this feature, that is, the number of nonzero elements in the matrix is independent of the number of connections. This happens because each connection in the network has its own position in the matrix depending on the starting and ending time.

Figure 5 shows the spy plot of the exponential of the network defined by Figure 1. The exponential of a matrix of the form (3.2) generally is a fairly dense upper block triangular matrix.

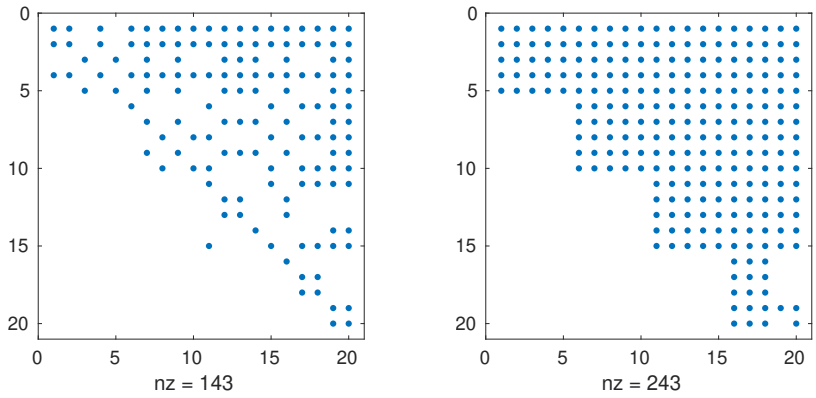

Figure 5: Spy plots of the exponential of the adjacency matrix associated to the network represented in Figure 1 by using the block model (3.2) (left) and by using the block model defined in [8] (right).

Figure 6 reports the spy plot associated to the network represented in Figure 2. The plot on the left-hand side is drawn by using the block model (3.2) and the plot on the right-hand side by using the block model defined in [8]. Again, the number of nonzero entries of the new block model equals the number of connections in the network (in fact, the number of connections is independent of the discretization), while in the block model [8] the number of nonzero entries is unrelated to the number of connections in the network and this number changes with the discretization. This depends on that in the latter model, a connection that traverses a time step is present in more than one block, while in our new block model, as already said, each edge corresponds to only one (or two if the network is undirected) nonzero element(s) in the whole matrix.

Table 2 reports the ranking of the nodes of the network represented in Figure 1 obtained by using the two approaches: the resolvent-based centrality defined in [14] and the exponential-based centrality introduced in this paper. It can be seen that the centrality index based on the exponential of the adjacency matrix identifies node v_1 as the most important node, while the resolvent-based centrality does not. The importance of node v_1 is in agreement with intuition, since v_1 is involved in more minutes of conversation than the other nodes, but this measure of importance does not necessarily coincide with the simple notion of degree in larger examples, as it takes into consideration the downweighting of long walks in the nodes ranking. We remark that, in this example, the resolvent-based centrality is computed by setting $a = 0.9/\max\rho(A^{[s]})$ in (2.1), and that the ranking of the nodes does not change using a different value of the parameter a in the interval

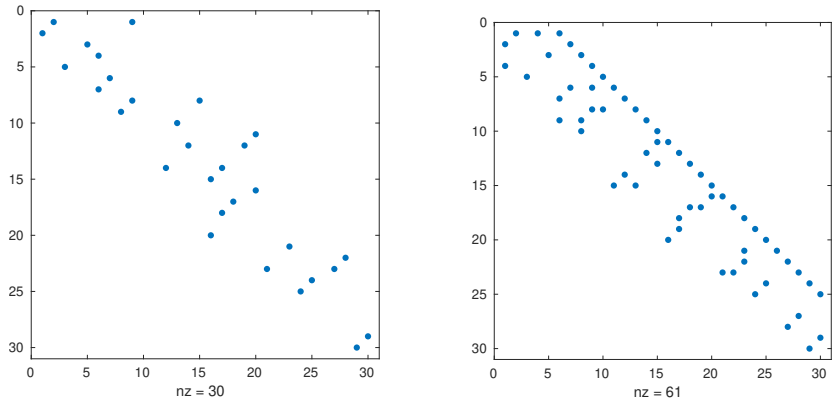

Figure 6: Spy plots of the adjacency matrix associated to the network represented in Figure 2 by using the block model (3.2) (left) and by using the block model defined in [8] (right).

$[0.5/\max\rho(A^{[s]}), 1/\max\rho(A^{[s]}))$.

Table 2: Comparison between the dynamic centrality indices of the network in Figure 1. The resolvent-based centrality is computed by setting $a = 0.9/\max\rho(A^{[s]})$ in (2.1).

Resolvent-based centrality		Exponential-based centrality	
Ranking	Value of centr.	Ranking	Value of centr.
4	1349.75	1	10.05
3	1220.49	4	7.20
1	1219.78	2	7.19
5	801.22	3	5.78
2	800.75	5	3.79

Table 3 reports the ranking of the nodes of the network represented in Figure 2 obtained by using the two approaches. It can be seen that, by decreasing the length of each time step, both the centrality indices correctly identify node v_1 as the most important one in terms of number of minutes of conversation. The resolvent-based centrality is computed by setting $a = 0.9/\max\rho(A^{[s]})$ in (2.1). Again, in this example, the ranking of the nodes by using the resolvent-based centrality does not change using different values of the parameter a in the interval $[0.5/\max\rho(A^{[s]}), 1/\max\rho(A^{[s]}))$. It is worth noting that, except for the first and last nodes, the rankings obtained by the two methods do not coincide.

Table 3: Comparison between the dynamic centrality indices of the network in Figure 2. The resolvent-based centrality is computed by setting $a = 0.9/\max\rho(A^{[s]})$ in (2.1).

Resolvent-based centrality		Exponential-based centrality	
Ranking	Value of centr.	Ranking	Value of centr.
1	2103.36	1	4.90
4	1672.69	2	3.54
2	1405.66	3	2.72
3	954.57	4	2.72
5	826.43	5	2.72

7. Computed examples

To illustrate the performance of the models, Algorithm 1 has been implemented in the Matlab programming language and applied to three real-world data sets obtained from airport logistics, telecommunication, and bioinformatics. The numerical experiments were performed on an Intel Xeon Gold 6136 computer (16 cores, 32 threads) equipped with 128 Gbyte RAM, running Matlab R2019a. The Matlab code and some of the data sets can be downloaded from the web page [4].

7.1. Airports

The first example concerns a data set that describes the flights through 305 US airports during one day (June 1st, 2008) [3]. Only flights that take off and arrive at the same day are considered. Airports are represented by nodes and flights by edges. In some cases, the arrival time is earlier than the departure time because of different time zones. For instance, there is a flight from South Bend International Airport (SBN) to Chicago O'Hare International Airport (ORD) that takes off at 11:51 am Eastern Standard Time (EST) and arrives at 11:36 am Central Standard Time (CST). To avoid this difficulty, all departure and arrival times were transferred into EST. We discretized the day into 24 one-hour intervals.

Then, we represent these flights by edges in the adjacency matrices $A_{i,j} \in \mathbb{R}^{305\times 305}$, where the (r,ℓ)th entry of $A_{i,j}$ is 1 if there is a flight that takes off during the time period τ_i and arrives during the time period τ_j from airport r to airport ℓ. The block matrix $B_N \in \mathbb{R}^{7320\times 7320}$ in (3.2), which is composed by these adjacency matrices, is formed and Algorithm 1 is applied to obtain the vector $\mathbf{y} = e^{B_N}\mathbf{1}$, whose entries measure the capacity of each airport as a source. To be more specific, the first 305 entries of the vector $\mathbf{y}$ show the broadcasting ability of

the 305 airports starting from the first hour of the day, the second 305 entries show the broadcasting ability of the airports starting form the second hour of the day, and so on. Figure 7 displays the broadcasting centrality of these airports during 24 hours.

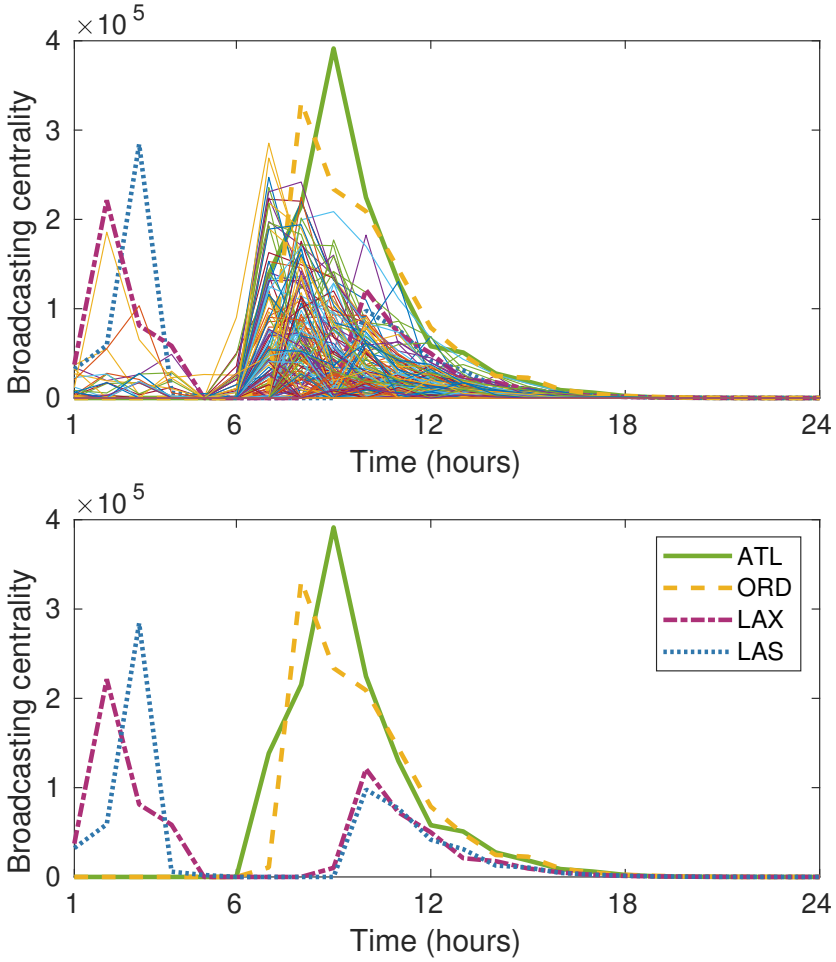

Figure 7: Broadcasting centrality for all airports (top) and for the four main broadcasters (bottom).

At the beginning of the day, between 1:00 am and 3:00 am, western airports, such as the Los Angeles Airport (LAX) and the Las Vegas Airport (LAS), are particularly active, because these times correspond to 11:00 pm and 1:00 am in the Pacific Standard Time (PST) zone. We can see that from 6:00 am most airports start to be active and hit a peak at 7:00 am. These are the airports in the EST zone. Chicago (ORD) Airport hits its peak at 8:00 am because it is in the CST zone.

Although Atlanta (ATL) is in the EST zone, it hits its peak at 9:00 am. Hence, when starting to fly from Atlanta at 9:00 am, one has many options to reach other destinations during the same day.

In this example, Algorithm 1 is faster than the built-in Matlab function `expm`. It takes 2.3 seconds, compared to 8.8 seconds for `expm`, but the advantage is not only a matter of speed. Indeed, the sparse data matrix $B_N \in \mathbb{R}^{7320\times 7320}$ requires 32 Kbytes of the computer memory, while its exponential is a full matrix which demands 429 Mbytes. Furthermore, the model allows us to track the sending ability for each airport over time during the computations.

We remark that, while the exponential-based centrality identifies the Los Angeles Airport as the most important broadcaster in the whole time period, the node with the highest value of the resolvent-based centrality is the McCarran International Airport in Las Vegas. This shows that the two centralities produce different rankings of the nodes.

7.2. Cell phone calls

In this subsection, we apply our models to a data set that describes a dynamic network of cell phone calls. It was used in the IEEE VAST Challenge in 2008 [15]. The data set consist of 400 cell phone IDs that are connected by 9834 pairwise calls over 10 days. For each call, the sender, the receiver, the starting time, and the call duration are given.

Call durations range between 7 seconds and 36 minutes 11 seconds. In our experiment, we discretize the ten days into 1 hour intervals, to obtain 240 intervals. The calls are represented by symmetric adjacency matrices $A_{i,j} \in \mathbb{R}^{400\times 400}$, where the (r,ℓ) and (ℓ,r) entries of $A_{i,j}$ are 1 if there is a call from node r to ℓ, or from node ℓ to r, starting at the time interval τ_i and ending at the time interval τ_j.

We applied Algorithm 1 to the block matrix B_N to compute $\mathbf{y} = e^{B_N}\mathbf{1}$. The size of the block matrix B_N is in this case 96000 and the execution time was 332 seconds. The size of the problem is too large to compute the vector $\mathbf{y}$ by the `expm` function of Matlab. We run a test processing half of the data set, that is, for a time period of 5 days. In this case, the size of the matrix B_N was 48000, and `expm` completed the computation in 389 seconds, using almost all the available 128 GB of memory, while Algorithm 1 took 86 seconds. To have an idea of the memory storage required, when $n = 48000$ the sparse matrix B_N occupies 500 Kbytes, while e^{B_N} requires 18 Gbyte. When the full data set is processed ($n = 96000$), the two sizes are 1 Mbyte and 74 Gbyte, respectively.

The first 400 entries of $\mathbf{y}$ show the broadcasting ability for the 400 nodes starting from the first hour, the second 400 entries show the broadcasting ability for

the 400 nodes starting from the second hour, and so on. Figure 8 displays the broadcasting ability for all the callers during ten days.

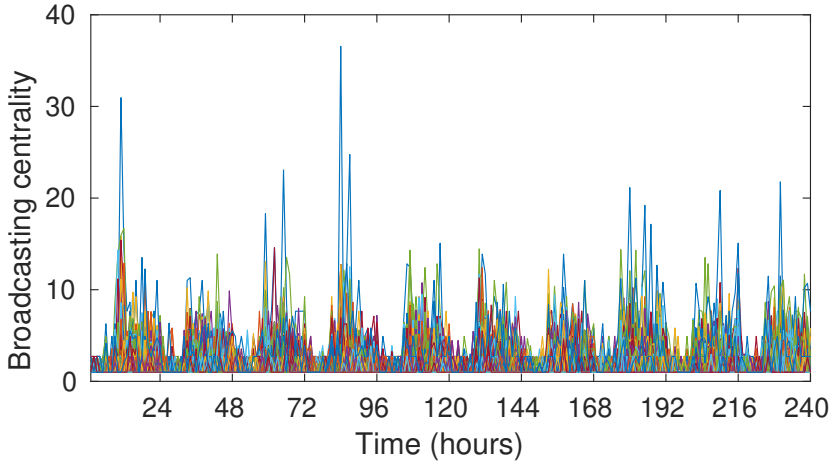

Figure 8: Broadcasting centrality for all the 400 telephone calls.

The broadcasting activity of node 200 was intense and concentrated in the first 7 days. It was connected directly to nodes 1, 2, 3, and 5, that had the highest broadcasting connectivity. In the last 3 days, the above five nodes were substituted by the nodes 300, 306, 309, 360, and 397, for what concerns broadcast capacity [8, 12, 19, 28]. Our model shows that the same nodes were quite inactive in the first week. Figure 9 displays the broadcasting ability of the two groups of nodes.

7.3. Brain networks

Associating a graph to brain activity can be done by associating nodes to brain regions and links to anatomical or functional connections [23]. Modern diagnostic devices, such as electroencephalography (EEG), magnetic resonance imaging (MRI), and magnetoencephalography (MEG), are able to produce large data sets across variable time intervals. Taking into account the variability in time is essential in analyzing the brain response to external stimulation, and in detecting particular diseases, such as Alzheimer's disease and epilepsy [10, 21, 22, 23, 24].

To construct a test brain graph, we considered the data set "Psychophysics (4Mb)", available at [26], which lists EEG data collections available for public download. This particular data set is reported to belong to the EEGLAB library [25]. It contains data from a 32-channel EEG, measured at 128 Hz over 238 seconds, on a subject committed to a visual attention task.

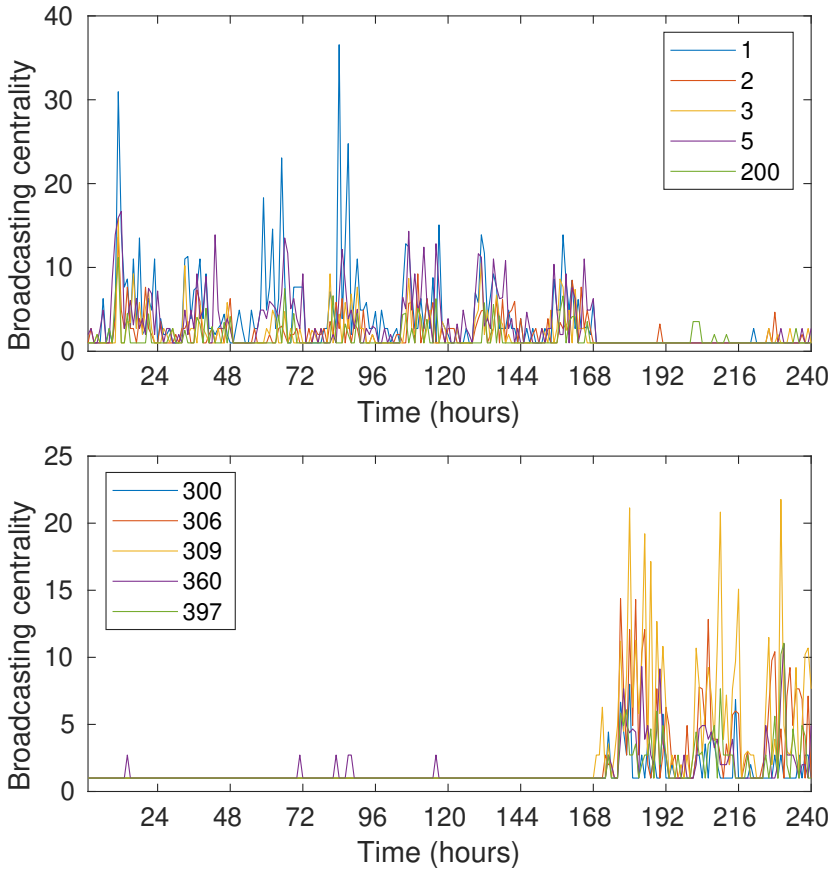

Figure 9: Broadcasting centrality for calls 1, 2, 3, 5, and 200 (top), and calls 300, 306, 309, 360, and 397 (bottom).

We divide the 32 signals corresponding to the nodes (EEG channels) into $N = 238$ data chunks, each reporting 1 second of activity, i.e., 128 measurements. To each nodes pair (r, ℓ), for each time duration (τ_i, τ_j), we associate the correlation $(B_N)^{i,j}_{r,\ell}$ between the corresponding time series over the chosen time duration. The resulting adjacency matrix is weighted, but we binarize it by applying a threshold of 0.9 to the values of the correlation. It is quite sparse, as its density, that is, the ratio between the number of nonzero elements of the matrix and the square of its size, is 0.2. We display a spy plot of two sections of it in Figure 10, including 4 and 16 diagonals blocks, respectively.

We remark that correlation is only one of the possible measures that can be used to construct a brain network. Another possibility is to resort to coherence,

which allows one to investigate specific physiologically relevant frequency bands in the signal [27, 20].

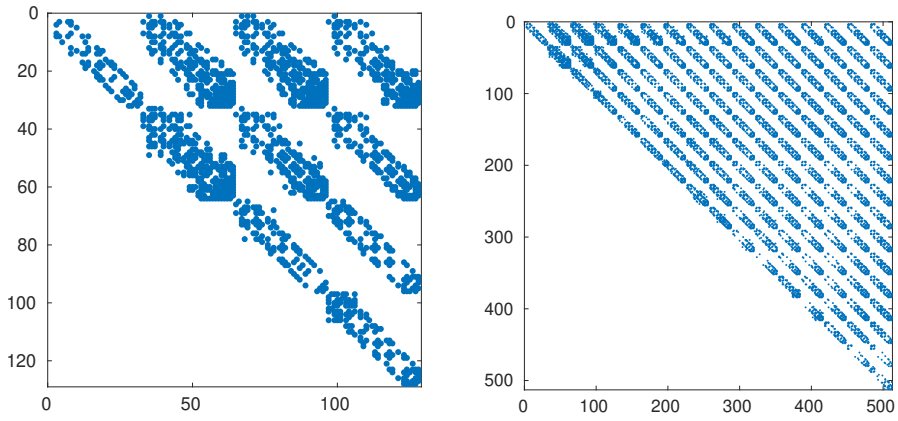

Figure 10: Model (3.2) for the brain time-network: Sections including 4 (left) and 16 (right) diagonal blocks.

We used Algorithm 1 to compute the broadcast centrality $\mathbf{y} = e^{B_N}\mathbf{1}$ for all the nodes in the network. This experiment does not pretend to have any clinical significance; it is only a mean to illustrate the capability of model (3.2). The computation, for the sparse matrix B_N of size 7616, whose storage space is 22 Mbytes, took 87 seconds. Performing the same computation with the `expm` function of Matlab required 274 seconds of computing time and 464 Mbytes to store the matrix exponential.

The centralities are displayed in Figure 11, in semi-logarithmic scale. The graph shows clearly that the broadcasting activity decays faster than exponentially over time. It appears that different nodes have different behavior; we illustrate this in detail in Figure 12, where we selected some nodes exhibiting such features. As it can be seen, the centrality of some nodes behaves smoothly, while other nodes exhibit an intermittent activity; see the graph on the left-hand side. The graph on the right-hand side of the same figure shows that some nodes are mostly inactive, and activate just for very short time periods, while some other nodes are active almost continually, and interrupt their broadcasting only during very short time intervals.

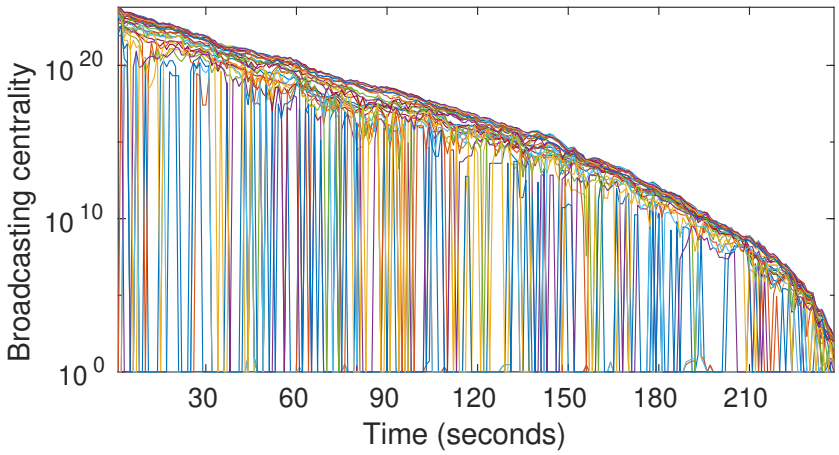

Figure 11: Broadcasting centrality for all the nodes in the brain time-network.

8. Conclusion

This paper introduces a new block structure for representing dynamic networks. The structure allows the modeling of edges with a limited duration in time. Often long walks are less important than short walks, and old walks are less important than recent walks. Our model model allows downweighting walks that are long and old to account for their reduced importance. The model allows entities to move through the network as quickly as possible, as well as to include waiting times.

The importance of the nodes is measured with the aid of the exponential function of the adjacency matrix for the dynamic network. This matrix may be quite large in applications. We therefore discuss its efficient computation. In particular, we exploit that the adjacency matrix is an upper triangular block matrix. This makes it possible to evaluate the matrix exponential by an iterative method based on the approach described in [18]. This method is faster than using the `expm` function of Matlab, and allows updating the broadcast centrality when an additional time slice is added.

Some aspects of the model proposed would appear to be applicable to the analysis of higher-order dynamic networks, that display higher-order features and higher organization. Higher-order networks often are modeled by nonnegative tensors. We plan to extend the technique developed in the resent paper to higher-order dynamic networks, but hasten to add that this extension is not straightforward.

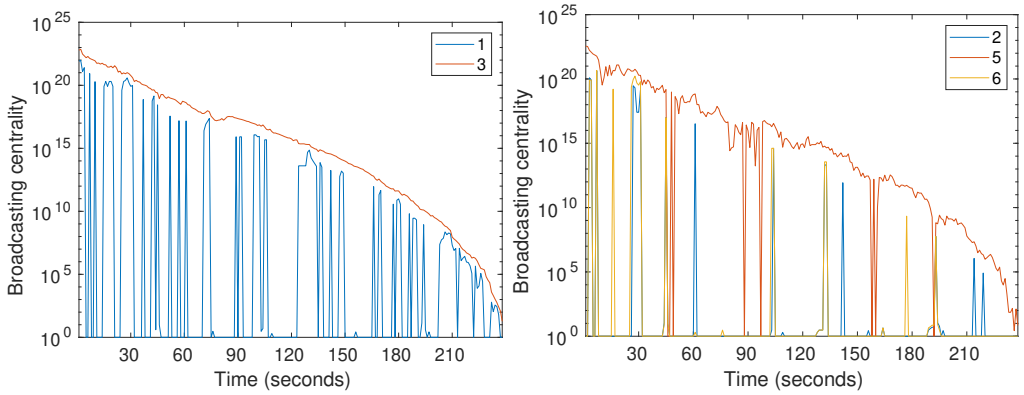

Figure 12: Broadcasting centrality for nodes 1, 3 (left) and nodes 2, 5, 6 (right) in the brain time-network.

Acknowledgment

The authors would like to thank the referees for comments that resulted in an improved presentation. This research is partially supported by the Fondazione di Sardegna 2017 research project "Algorithms for Approximation with Applications [Acube]", the INdAM-GNCS research project "Tecniche numeriche per l'analisi delle reti complesse e lo studio dei problemi inversi", and the Regione Autonoma della Sardegna research project "Algorithms and Models for Imaging Science [AMIS]" (RASSR57257, intervento finanziato con risorse FSC 2014-2020 - Patto per lo Sviluppo della Regione Sardegna). CF gratefully acknowledges Regione Autonoma della Sardegna for the financial support provided under the Operational Programme P.O.R. Sardegna F.S.E. (European Social Fund 2014-2020 - Axis III Education and Formation, Objective 10.5, Line of Activity 10.5.12). Research by ODC and LR was supported in part by NSF grant DMS-1720259.

[1] A. Alsayed, D. J. Higham, *Betweenness in time dependent networks*, Chaos Solitons Fractals, 72 (2015), pp. 35–48.

[2] M. Benzi, C. Klymko, *Total communicability as a centrality measure*, J. Complex Networks, 1(2) (2013), pp. 124–149.

[3] Bureau of Transportation Statistics, U.S. Department of Transportation (2009) Airline On-Time Statistics and Delay Causes, year 2008 [raw dataset]. Retrieved on February 20, 2018, from `https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp`

[4] CaNA Software Repository, University of Cagliari, `http://bugs.unica.it/cana/software/`

[5] E. Estrada, *The Structure of Complex Networks*, Oxford University Press, Oxford, 2012.

[6] E. Estrada, D. J. Higham, *Network properties revealed through matrix functions*, SIAM Rev., 52 (2010), pp. 696–714.

[7] E. Estrada, P. Knight, *A First Course in Network Theory*, Oxford University Press, Oxford, 2015.

[8] C. Fenu, D. J. Higham, *Block matrix formulation for evolving networks*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 343–360.

[9] C. Fenu, D. Martin, L. Reichel, G. Rodriguez, *Block Gauss and anti-Gauss quadrature with application to networks*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1655–1684.

[10] M. Fraschini, M. Demuru, M. Puligheddu, S. Floridia, L. Polizzi, A. Maleci, M. Bortolato, A. Hillebrand, F. Marrosu, *The re-organization of functional brain networks in pharmaco-resistant epileptic patients who respond to VNS*, Neurosci. Lett., 580 (2014), pp. 153–157.

[11] M. C. Golumbic, *Interval graphs and related topics*, Discrete Math., 55 (1985), pp. 113–121.

[12] P. Grindrod, D. J. Higham, *A dynamic systems view of network centrality*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 470 (2014), 20130835.

[13] P. Grindrod, D. J. Higham, *A matrix iteration for dynamic network summaries*, SIAM Rev., 55 (2013), pp. 118–128.

[14] P. Grindrod, D. J. Higham, M. C. Parsons, E. Estrada, *Communicability across evolving networks*, Phys. Rev. E, 83 (2011), pp. 046120.

[15] G. G. Grinstein, C. Plaisant, S. J. Laskowski, T. O'Connell, J. Scholtz, M. A. Whiting, *Vast 2008 challenge: Introducing mini-challenges*, in Proceedings of the 2008 IEEE Symposium on Visual Analytics Science and Technology, IEEE, (2008), pp. 195–196.

[16] P. Holme, J. Saramäki, *Temporal networks*, Phys. Rep., 519 (2012), pp. 97–125.

[17] R. A. Horn, R. Mathias, Y. Nakamura, *Inequalities for unitarily invariant norms and bilinear matrix products*, Linear and Multilinear Algebra, 30 (1991), pp. 303–314.

[18] D. Kressner, R. Luce, F. Statti, *Incremental computation of block triangular matrix exponentials with application to option pricing*, Electron. Trans. Numer. Anal., 47 (2017), pp. 57–72.

[19] J. Ma, J. Liu, W. Ma, M. Gong, L. Jiao, *Decomposition-based multiobjective evolutionary algorithm for community detection in dynamic social networks*, Sci. World J., vol. (2014), Article ID 402345, 22 pages, Jan. 2014.

[20] J. McBride, X. Zhao, N. Munro, C. Smith, G. Jicha, Y. Jiang, *Resting EEG discrimination of early stage Alzheimer???s disease from normal aging using inter-channel coherence network graphs*, Ann. Biomed. Eng., 41 (2013), pp. 1233–1242.

[21] S. M. Pani, M. Ciuffi, M. Demuru, S. M. La Cava, G. Bazzano, E. D'Aloja, M. Fraschini, *Subject, session and task effects on power, connectivity and network centrality: a source-based EEG study*, Biomed. Signal Process. Control, 59 (2020), Article ID 101891.

[22] S. C. Ponten, F. Bartolomei, C. J. Stam, *Small-world networks and epilepsy: graph theoretical analysis of intracerebrally recorded mesial temporal lobe seizures*, Clin. Neurophysiol., 118 (2007), pp. 918–927.

[23] M. Rubinov, O. Sporns, *Complex network measures of brain connectivity: Uses and interpretations*, Neuroimage, 52 (2010), pp. 1059–1069.

[24] C. J. Stam, W. de Haan, A. Daffertshofer, B. F. Jones, I. Manshanden, A. M. van Cappellen van Walsum, T. Montez, J. P. A. Verbunt, J. C. de Munck, B. W. van Dijk, H. W. Berendse, P. Scheltens, *Graph theoretical analysis of magnetoencephalographic functional connectivity in Alzheimer's disease*, Brain, 132 (2009), pp. 213–224.

[25] Swartz Center for Computational Neuroscience, University of California San Diego, EEGLAB, `https://sccn.ucsd.edu/eeglab/index.php`

[26] Swartz Center for Computational Neuroscience, University of California San Diego, EEG/ERP data available for free public download, https://sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html

[27] B. C. M. Van Wijk, C. J. Stam, A. Daffertshofer, *Comparing brain networks of different size and connectivity density using graph theory*, PloS One, 52 (2010), pp. 1059–1069.

[28] Q. Ye, T. Zhu, D. Hu, B. Wu, N. Du, B. Wang, *Cell phone mini challenge award: Social network accuracy??? exploring temporal communication in mobile call graphs*, 2008 IEEE Symposium on Visual Analytics Science and Technology, Columbus, OH, (2008), pp. 207–208.