Eur. Phys. J. B (2016) 89: 71
DOI: 10.1140/epjb/e2016-60639-0

THE EUROPEAN PHYSICAL JOURNAL B

Regular Article

Asymmetry through time dependency*

Alexander V. Mantzaris and Desmond J. Higham[a]

Department of Mathematics and Statistics, University of Strathclyde, Glasgow G1 1XH, UK

Received 5 August 2015 / Received in final form 8 November 2015
Published online 16 March 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract. Given a single network of interactions, asymmetry arises when the links are *directed*. For example, if protein A upregulates protein B and protein B upregulates protein C, then (in the absence of any further relationships between them) A may affect C but not vice versa. This type of imbalance is reflected in the associated adjacency matrix, which will lack symmetry. A different type of imbalance can arise when interactions appear and disappear over time. If A meets B today and B meets C tomorrow, then (in the absence of any further relationships between them) A may pass a message or disease to C, but not vice versa. Hence, even when each interaction is a two-way exchange, the effect of time ordering can introduce asymmetry. This observation is very closely related to the fact that matrix multiplication is not commutative. In this work, we describe a method that has been designed to reveal asymmetry in static networks and show how it may be combined with a measure that summarizes the potential information flow between nodes in the temporal case. This results in a new method that quantifies the asymmetry arising through time ordering. We show by example that the new tool can be used to visualize and quantify the amount of asymmetry caused by the arrow of time.

1 Introduction

The success of Network Science as a research discipline shows that there is great value in studying a complex system through the connectivity of its components [1]. However, even after simplifying down to the level of nodes and edges, there is typically too much information for us to digest, and we must rely on tools that further reduce the dimension of the system so that we can summarize and visualize key properties. The need to reveal hidden structure and substructure within a complex network has motivated a plethora of quantitative tools aimed at, for example, discovering significant nodes or edges, and topological features such as well-connected communities, bipartite structures, bottlenecks, motifs, hubs and authorities [2–9]. In this work we focus on the idea of quantifying and visualizing the level of asymmetry in a network, and in particular, studying asymmetry caused by the arrow of time in a dynamic network sequence.

First, we introduce some notation. To be general, we allow edges to be directed, and we consider an unweighted, directed network consisting of N nodes. We denote by $A \in \mathbb{R}^{N\times N}$ the corresponding adjacency matrix. So A has $a_{ij} = 1$ if there is an edge from node i to node j, and $a_{ij} = 0$ otherwise. We assume $a_{ii} = 0$ for $i = 1, \ldots N$, so there are no self-loops. The out and in degree of node n are defined as

$$\deg_n^{\text{out}} := \sum_j a_{nj} \quad \text{and} \quad \deg_n^{\text{in}} := \sum_i a_{in},$$

respectively. We use $\mathcal{P}$ to denote the set of all permutations of the integers $1, 2, \ldots, N$, with p_i recording the ith component of an element $p \in \mathcal{P}$. A permutation vector $p \in \mathcal{P}$ can be used to relabel the network nodes; that is, to reorder the rows and columns of the adjacency matrix, so that node n becomes node p_n. Typically, such a p is induced from a real-valued vector $v \in \mathbb{R}^N$, where the component v_n is a weight assigned to node n. From v, we generate a permutation $p \in \mathcal{P}$ by the natural procedure of ordering the nodes according to these weights. More precisely, we compute $p \in \mathcal{P}$ such that

$$v_i \le v_j \quad \Longleftrightarrow \quad p_i \le p_j, \tag{1}$$

with some rule for treating ties.

In the case where we have a time-ordered sequence of networks, based on the same set of N nodes, we introduce a superscript. So $A^{[k]}$ is the adjacency matrix at time t_k. Here $t_0 < t_1 < t_2 \cdots < t_M$ is a sequence of discrete time points. So, for example, if we are recording "who texted whom," we may choose $(A^{[k]})_{ij} = 1$ if i texted j at least once in the period $[t_{k-1}, t_k)$ and $(A^{[k]})_{ij} = 0$ otherwise.

The remainder of the article proceeds as follows. In Section 2 we explain how the widely used Fiedler vector is relevant to the problem of ordering nodes via its role in

* Contribution to the Topical Issue "Temporal Network Theory and Applications", edited by Petter Holme.

[a] e-mail: d.j.higham@strath.ac.uk

solving an optimization problem. In this case, we are reordering the network in a way that brings together nodes with similar connectivity patterns. We then introduce the "in minus out" vector that solves an alternative, one-sum, optimization problem designed to reveal network asymmetry. Section 3 then describes how a dynamic communicability matrix can be computed that summarizes, in a single matrix, the flow of information associated with a time-dependent network sequence. Based on the important observation that the one-sum optimization problem, and its solution, generalize naturally to the case of weighted edges, in Section 4 we then combine these ideas in order to develop a new tool that can reveal network asymmetry caused by time's arrow. The new technique is illustrated on a synthetic data set, with further results given for voice call data. In Section 5 we conclude with a discussion.

2 Static reordering

From the perspective of computational linear algebra, the idea of reordering rows and columns of a matrix has a long history that is typically motivated through

- avoiding the breakdown of an algorithm, for example, ensuring that a nonzero pivot sequence exists in Gaussian elimination [10];
- reducing storage and computation costs associated with "fill-in;" that is, the unnecessary creation of nonzero elements in a sparse matrix [11].

Our aim here is different. We wish to reorder as a means to reveal structure. (However, there is overlap between these aims, and hence between the resulting tools that have been developed.)

To begin, we consider the *two-sum* [12,13]

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2 a_{ij}. \tag{2}$$

In this expresson, any edge in the network, where $a_{ij} \neq 0$, makes a contribution to the overall sum. Edges connecting nodes that are far apart, that is, where $(i-j)^2$ is large, make correspondingly large contributions. We emphasize that this quantity is not a graph invariant – it typically depends very strongly on the given node ordering. The two-sum (2) is nonnegative, and is small when the presence of edges is biased towards nodes that have nearby indices. In other words, it is small when the nonzeros in the adjacency matrix live near the diagonal. The task of node reordering to minimize the two-sum may then be written

$$\min_{p\in\mathcal{P}} \sum_{i=1}^{N}\sum_{j=1}^{N}(p_i-p_j)^2 a_{ij}. \tag{3}$$

To make this problem feasible it is attractive to consider a relaxed version where p is replaced by a real-valued vector v. In taking this step, we must rule out the trivial solution where $v_i \equiv 0$, and also factor out the redundancy

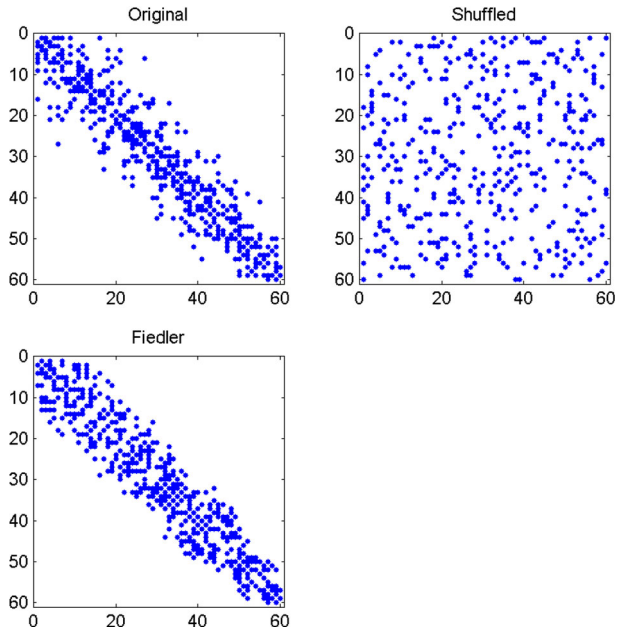

Fig. 1. Upper left: nonzero pattern in the adjacency matrix of an instance of an undirected range-dependent random graph, with two-sum of 16 518. Upper right: an arbitrarily shuffled version of this matrix, with two-sum of 269 646. Lower left: reordering of this matrix using the Fiedler vector, with two-sum of 13 304.

whereby all components of v are shifted uniformly. To do this we add the constraints $\|v\|_2 = 1$ and $v^T\mathbf{1} = 0$, where $\|\cdot\|_2$ denotes the Euclidean vector norm and $\mathbf{1} \in \mathbb{R}^N$ is the vector with all components equal to one. This gives

$$\min_{v\in\mathbb{R}^N,\, \|v\|_2=1,\, v^T\mathbf{1}=0} \sum_{i=1}^{N}\sum_{j=1}^{N}(v_i-v_j)^2 a_{ij}. \tag{4}$$

This problem can be solved analytically through the *graph Laplacian*, $L := D - B$, where $B := \frac{1}{2}(A + A^T)$ and the diagonal matrix D has $d_{ii} = \sum_k b_{ik}$. In the general case where the graph represented by $\mathrm{sign}(B)$ is connected, L is symmetric positive semi-definite with a single eigenvalue equal to zero and corresponding eigenvector proportional to $\mathbf{1}$. The problem (4) is then solved by taking v to be the *Fiedler vector* – the eigenvector corresponding to the second smallest eigenvalue of L; see [14–16] for further details. Having obtained the relaxed solution v, we may recover a permutation $p \in \mathcal{P}$ via (1).

To illustrate this idea, the upper left picture in Figure 1 shows the adjacency matrix for a network that has a strong preference for short-range edges. More precisely, we used a sample of the undirected range-dependent random graph class from [17], where the independent probability of an undirected edge between node i and j depends on $|i-j|$. In our case, we used the functional form $0.8^{|i-j|}$. Here, there are $N = 60$ nodes, so we have a symmetric adjacency matrix in $\mathbb{R}^{60\times 60}$. It is clear from the picture that, in this given ordering, nonzeros are concentrated near the diagonal – nodes tend to link to nodes that are close by in this ordering. The two-sum (2) in this

case is 16 518. The upper right picture shows an arbitrarily shuffled version of this matrix; rows and columns have undergone a common reordering with a permutation vector chosen unformly at random. In MATLAB notation, we have `A(p,p)`, where `p = randperm(60)`. This picture illustrates the type of data that we would see if we were given nodes in arbitrary order, with no information about the underlying range-dependency structure. In this case the two-sum has increased dramatically to 269 646. The lower left picture shows the effect of reordering with the Fiedler vector. Visually, we have reconstructed the concentration effect. In fact, this reordering reduces the two-sum to 13 304, below the level of the original matrix. We see that the Fiedler vector can be a powerful tool for uncovering hidden structure of this type.

In [18] a reordering approach was introduced based on the concept of the directed *one-sum*

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)a_{ij}. \tag{5}$$

Here, an edge connecting nodes i and j makes a large positive contribution to the sum if $i \gg j$; that is, if the adjacency matrix entry resides far into the lower triangle. Similarly, for $i \ll j$, where the entry is far into the upper triangle, we have a large negative contribution to the sum. So, in order to emphasize any inherent asymmetry, it is reasonable to introduce the minimization problem

$$\min_{p\in\mathcal{P}}\sum_{i=1}^{N}\sum_{j=1}^{N}(p_i-p_j)a_{ij}. \tag{6}$$

In solving this problem, we are attempting to concentrate nonzeros in the upper right-hand corner of the adjacency matrix, and in particular, to force nonzeros into the upper triangle. Whereas the two-sum minimization (3) was relaxed into (4) in order to produce a tractable problem, the one-sum version (6) is easily solved in its discrete form. A solution is given by the permutation (1) induced by the "in-degree minus out-degree" vector $v = \deg^{\text{in}} - \deg^{\text{out}}$ [18], (see also Sect. 4 for a more general case). Loosely, this reordering makes the adjacency matrix appear as upper triangular as possible, and hence reveals inherent asymmetry in the network.

This reordering approach is illustrated in Figure 2. In the upper left picture, we show the adjacency matrix for an instance of the directed range-dependent graph model introduced in [18], where a directed edge from node i to j is generated with an independent probability that depends on $i-j$. In our case, we used the functional form 0.95^{i-j+N}. We see that the resulting adjacency matrix has nonzeros concentrated in its upper right corner. In this case the one-sum is $-8\,900$, the two-sum is 365 908 and there are 30 entries in the lower triangle. The upper right picture of Figure 2 shows the effect of an arbitrary node shuffle. Here, the one-sum is $-1\,971$, the two-sum is 210 305 and 130 entries lie below the diagonal. In the lower left picture, we show the Fiedler vector reordering. It is intuitively clear that this reordering is trying to bring elements closer to the diagonal. We have a one-sum of -429,

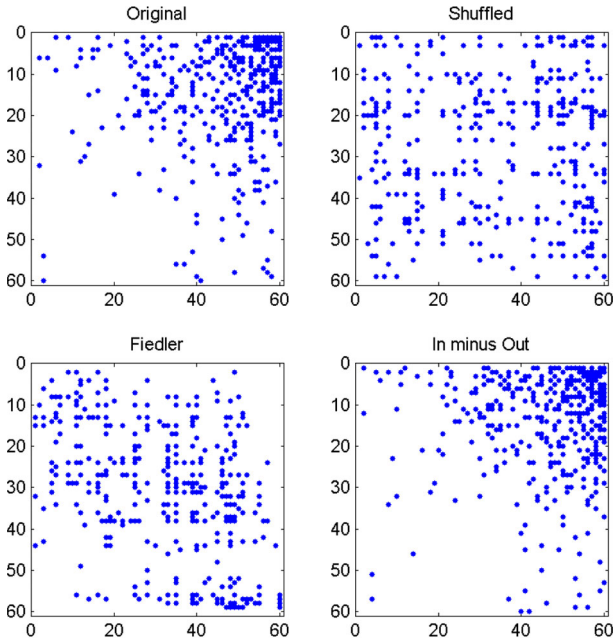

Fig. 2. Upper left: nonzero pattern in the adjacency matrix of an instance of a directed range-dependent random graph, with one-sum of $-8\,900$, two-sum of 365 908 and 30 entries below the diagonal. Upper right: an arbitrarily shuffled version of this matrix, with one-sum of -1971, two-sum of 210 305 and 130 entries below the diagonal. Lower left: reordering of this matrix using the Fiedler vector, with one-sum of -429, two-sum of 98 687 and 152 entries below the diagonal. Lower right: reordering of this matrix using in-degree minus out-degree, with one-sum of $-9\,357$, two-sum of 384 333 and 26 entries below the diagonal.

a two-sum of 98 687 and 152 nonzeros below the diagonal. Fiedler reordering has reduced the two-sum, but has not revealed the inherent asymmetry. In the lower right picture, we show the "in minus out" reordering, which solves (6). In this case, we see that nonzeros are moved up and to the right. The two-sum has increased to 384 333, showing that this reordering is not trying to put similarly connected nodes together. But the one-sum is reduced to $-9\,357$ and just 26 entries lie below the diagonal. Hence, this ordering has revealed the inherent hierarchical structure, and indeed has improved on the original ordering in terms of these two concrete measures.

3 Dynamic communicabilty

The dynamic communicability measures introduced in [19] were designed to capture the potential flow of information through the time-dependent edges over the period $[t_0, t_M]$. The idea is motivated by the concept of *dynamic walks*. A dynamic walk of length w from node i to node j is defined to be any traversal from i to j along w edges that respects the arrow of time; so, having used an edge at time t_r, the next edge that we use must exist at time t_r or later. The ability of node i to broadcast information to node j is then summarized as the total number of dynamic walks from i

to j, where a walk of length w is downweighted by the factor α^w. Here $\alpha \in (0,1)$ is a free parameter whose role is to reduce the influence of longer walks. When there is only one time point, this measure reduces to the classical Katz centrality [20], and the pairwise summaries can be computed through the matrix resolvent $(I-\alpha A)^{-1}$. This expression can be understood from the basic linear algebra fact that the matrix power entry

$$(A^w)_{ij}$$

gives a count of the number of walks from i to j using w edges. Generalizing to an ordered sequence of time points, [19] arrived at a product of resolvents

$$\mathcal{Q} = \left(I-\alpha A^{[0]}\right)^{-1}\left(I-\alpha A^{[1]}\right)^{-1}\ldots\left(I-\alpha A^{[M]}\right)^{-1}. \tag{7}$$

Here, $\mathcal{Q}_{ij}$ measures how well node i can broadcast information to node j by forming a weighted sum of all dynamic walks that start at i and finish at j, with walks using w edges downweighted by a factor α^w. To understand how $\mathcal{Q}$ arises, we note that, for example,

- $\left(A^{[0]}A^{[2]}A^{[4]}\right)_{ij}$ counts the number of walks from i to j using one edge in time window 0, one edge in time window 2 and one edge in time window 4; and
- $\left((A^{[5]})^2A^{[7]}\right)_{ij}$ counts the number of walks from i to j using two edges in time window 5 and one edge in time window 7.

The ordered product of resolvents in (7) captures all such terms – the key feature is the nondecreasing set of time indices. Just as in the original Katz version, the *dynamic communicability matrix* $\mathcal{Q}$ involves a parameter, α. In order for the matrix inverses to exist, we impose the condition that α is below

$$\alpha^\star := \min_{0\le k\le M}\left(\left(\rho\left(A^{[k]}\right)\right)^{-1}\right), \tag{8}$$

where $\rho(\cdot)$ denotes the spectral radius.

Because matrix multiplication is not commutative, the dynamic communicability matrix $\mathcal{Q}$ depends on the time ordering of the adjacency matrices. Also, even when each $A^{[k]}$ is symmetric, the resulting summary $\mathcal{Q}$ will be unsymmetric in general, reflecting the directional nature of time's arrow.

We also note that products of temporal adjacency matrices were used later in [21] for the weaker concept of *accessibility* (the existence of at least one dynamic walk that uses at most one edge per time step).

4 Dynamic asymmetry

Our aim is now to show that the concepts in Sections 2 and 3 can be extended to provide a method for quantifying and visualizing dynamic asymmetry. To motivate this task, we note that there are a number of on-line and off-line circumstances where dynamic, pairwise human interactions have been observed to take place within a hierarchical structure. For example, in the context of online forums, Graham and Wright [22] empirically investigated three types of *superparticipants*. One type was *agenda-setters*; participants who are responsible for new thread creation, and thereby exert a disproportionate influence on *subsequent* interactions. Quoting from [22]: "the inclusion of agenda-setting reflects our view that influence is not limited to the volume of posts alone." In an empirical study of small teams tackling simulated logistics tasks, Duchon and Patterson [23] looked for *emergent thought leaders*; that is, individuals who have not been assigned to a leadership position, but emerge *over time* as dominant in power, decision-making, and communications with respect to a particular topic. Huffaker et al. [24] studied the use of chat features between players in a Massive Multiplayer Online (MMO) role-playing game, and found that in general "players send messages to higher-level experts." We point out three common features of these examples:

- there is an inherent node ordering that is either explicit (e.g., status within an on-line game) or implicit (e.g., perceived level of expertise);
- the asymmetry manifests itself not through single peer-to-peer communication but through longer threads of interactions, where knock-on effects are at play;
- to keep track of such threads we must respect the temporal nature of the interactions.

A tool for quantifying and visualizing dynamic asymmetry will therefore allow us to

- discover hidden structure and track its evolution through time: for example, who are currently the key thought leaders in a particular on-line social community?
- monitor the performance of the network players within an imposed hierarchy: for example, given the managerial structure in a company, are any employees punching above or below their weight?
- compare the inherent level of hierarchy in different social networks.

To summarize how our new algorithm works, we note that the dynamic communicability matrix $\mathcal{Q}$ in (7) defines a single weighted, unsymmetric network that represents the pairwise ability of nodes to exchange information using the time-ordered edge sequence. Having constructed $\mathcal{Q}$ for the dynamic data, we may then use a generalization of the network reordering approach described in Section 2 in order to quantify and visualize the asymmetry.

To explain this idea in more detail, we first note that in order to avoid numerical overflow, it is preferable to work with a normalized version of the dynamic communicability matrix. We will therefore compute

$$\widehat{\mathcal{Q}} := \frac{\mathcal{Q}}{\|\mathcal{Q}\|},$$

where $\|\cdot\|$ denotes the Euclidean matrix norm. To do this, we set

$$\widehat{\mathcal{Q}}^{[0]} = I \tag{9}$$

and, for each new time point t_k, form

$$\widehat{\mathcal{Q}}^{[k]} = \widehat{\mathcal{Q}}^{[k-1]}\left(I-\alpha A^{[k]}\right)^{-1}, \tag{10}$$

and then reset

$$\widehat{\mathcal{Q}}^{[k]} \mapsto \frac{\widehat{\mathcal{Q}}^{[k]}}{\|\widehat{\mathcal{Q}}^{[k]}\|}. \tag{11}$$

The final matrix

$$\widehat{\mathcal{Q}}^{[M]} =: \widehat{\mathcal{Q}} \tag{12}$$

is then our overall summary. Normalizing all entries in $\mathcal{Q}$ by the same scalar value is not an issue since we are only concerned with their relative sizes.

To reveal the asymmetry in this dynamic network summary, we may then exploit the observation that the optimization problem (6) and its exact solution continue to make sense when we have directed and weighted edges, even allowing for negative values. We make this precise as follows.

Result. Given any $B \in \mathbb{R}^{N\times N}$ the problem

$$\min_{p\in\mathcal{P}} \sum_{i=1}^{N}\sum_{j=1}^{N}(p_i - p_j)b_{ij}$$

is solved by taking p to be the ordering induced in (1) by the vector

$$v = B_{\mathrm{col}\,i} - B_{\mathrm{row}\,i},$$

where

$$B_{\mathrm{row}\,i} := \sum_{j=1}^{N} b_{ij} \quad \text{and} \quad B_{\mathrm{col}\,i} := \sum_{j=1}^{N} b_{ji}$$

are the row and column sums of B, respectively.

Proof. The sum

$$\sum_{i=1}^{N}\sum_{j=1}^{N}(p_i - p_j)b_{ij}$$

may be manipulated straightforwardly into the form

$$\sum_{i=1}^{N} p_i \left[B_{\mathrm{row}\,i} - B_{\mathrm{col}\,i}\right].$$

The result then follows immediately.

In summary, having computed the normalized dynamic communicability matrix $\widehat{\mathcal{Q}}$ from (9)–(12), we then reorder this matrix using the result above with $B = \widehat{\mathcal{Q}}$. By construction, this reordering forces large (i.e., more positive) values of $\widehat{\mathcal{Q}}$ into the top right corner. In the language of [19] we are ordering nodes according to the difference between their *broadcast* and *receive* centralities.

We first illustrate this idea on a synthetic data set where the results can be interpreted clearly. We take $N = 31$ nodes and allow edges to evolve according to a temporal, directed binary tree, with some added noise; in a similar way to the test in [25]. We cycle with period four across 16 time windows:

- at time windows $1, 5, 9, 13$, a directed edge is inserted from node $1 \to 2$ and $1 \to 3$;
- at time windows $2, 6, 10, 14$, a directed edge is inserted from node $2 \to 4$, $2 \to 5$, $3 \to 6$ and $3 \to 7$;
- at time windows $3, 7, 11, 15$, a directed edge is inserted from node $4 \to 8$, $4 \to 9$ up to $7 \to 14$, $7 \to 15$;
- at time windows $4, 8, 12, 16$, a directed edge is inserted from node $8 \to 17$, $8 \to 18$ up to $15 \to 30$, $15 \to 31$.

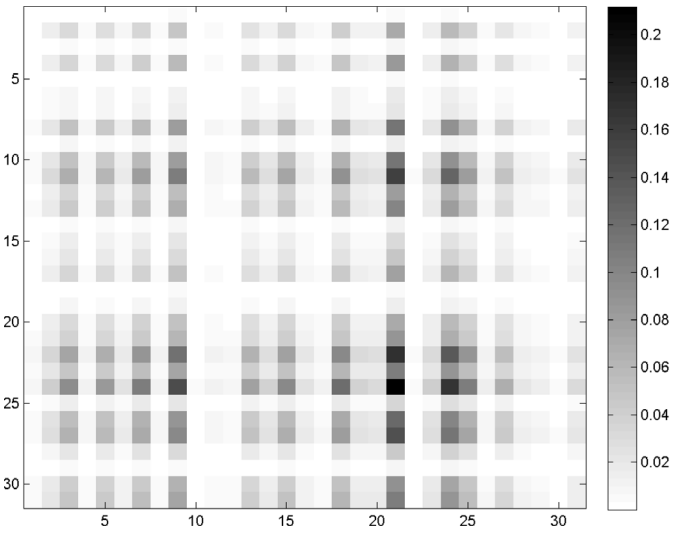

Fig. 3. Heat map showing entries in the dynamic communicability matrix for a binary tree cascade. Here the nodes are ordered according to the aggregate number of directed edges that they have produced. Because this ordering does not take account of the temporal hierarchy, there is no evidence of asymmetry. The one-sum for this ordering is 39.1.

In this structure nodes 1 to 15 have the same level of activity, in terms of aggregate out degree. However, there is a built-in cascade of walks down the directed binary tree. From a message passing perspective, we may view this structure as arising from a hierarchy between the nodes: there is a chain of command such that nodes pass information to lower-level individuals. To disrupt this structure slightly, we also add some noise: at each time window any other edge has an independent probability $1/N$ of arising. So we expect one unstructured directed edge per node per time window.

Taking the Katz parameter α to be 90% of the upper limit $\alpha^\star$ in (8), Figure 3 shows the entries of $\widehat{\mathcal{Q}}$ using a gray-scale heatmap. Here, we have ordered the nodes according to their overall activity, that is, the aggregate out degree. We see that ordering with respect to this measure does not uncover the temporal pattern of information flow. The one-sum of $\widehat{\mathcal{Q}}$ in this ordering is 39.1.

In Figure 4 we show the result from in-minus-out reordering. Comparing with Figure 3, we see that the hierarchical structure has been highlighted, as quantified by the one-sum of $\widehat{\mathcal{Q}}$ being reduced to -199.4.

To investigate further, in Figure 5 we show how Figure 3 changes when we arbitrarily shuffle the time points. From an application of MATLAB's `randperm` function, we took timepoints in the order 5, 16, 1, 11, 9, 14, 13, 3, 7, 10, 4, 2, 6, 12, 8, 15. In this case, the cascade effect is

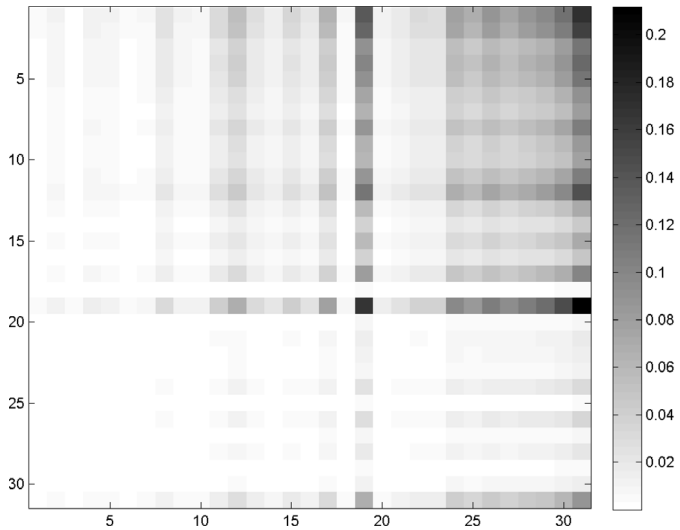

Fig. 4. As for Figure 3, but with the nodes ordered according to the difference between the row and column sums of the dynamic communicability matrix. The hierarchy caused by the temporal flow of edges is now apparent. The one-sum for this ordering is −199.4.

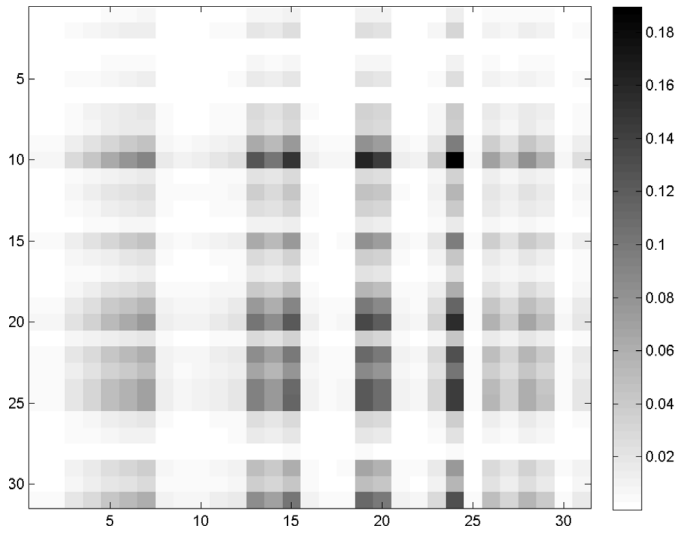

Fig. 5. As for Figure 3, but with the time points arbitraily shuffled to dilute the cascade effect. The one-sum for this ordering is 27.2.

likely to be reduced and it is of interest to quantify how much asymmetry remains. Ordering on aggregate activity is uniformative again, in this case giving a one-sum of 27.2. Figure 6 shows the result of "broadcast minus receive" ordering on $\widehat{\mathcal{Q}}$. Because the directed flow of information has been diluted, the algorithm reveals a reduction in the visual evidence of asymmetry, as reflected in the one-sum of −171.9.

We next give results on voice call interactions from the IEEE VAST 2008 Challenge [26]. This synthetically generated set was designed to reflect interactions between members of a socio-political movement. We are given time-

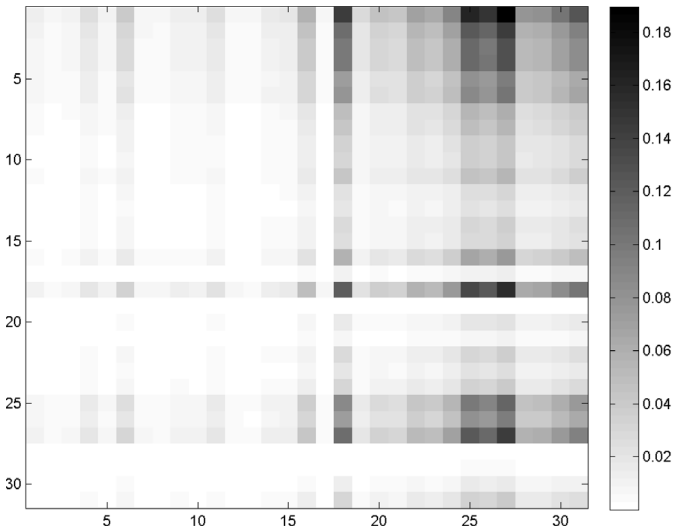

Fig. 6. As for Figure 4, but with the time points arbitraily shuffled to dilute the cascade effect. The one-sum for this ordering is −171.9.

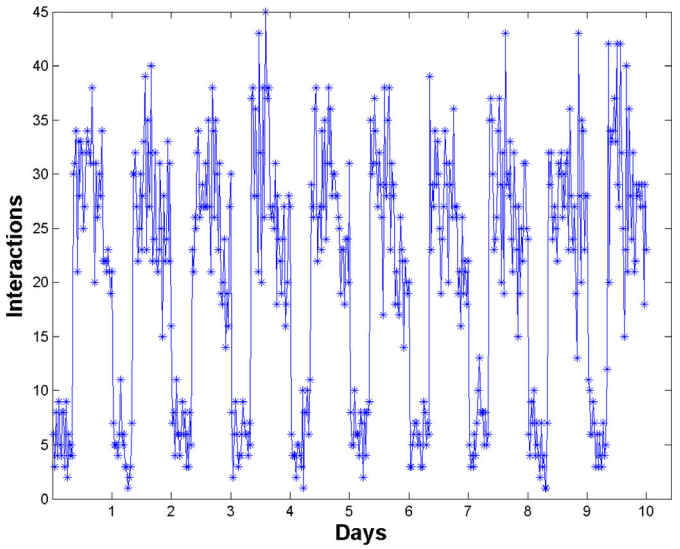

Fig. 7. Number of directed network edges (voice call interactions) in each 30 min time window.

stamped information for 400 cell phone users over a ten day period. There are 9 834 calls, with IDs for the send and receive nodes, a start time in hours/minutes and a duration in seconds. For our experiment, we discretized into 30 min time windows, so the directed adjacency matrix $A^{[k]}$ is such that $(A^{[k]})_{ij} = 1$ when ID number i initiated a phone conversation with ID number j at least once in that 30 min period.

In Figure 7, we show the number of calls in each time window. There is a clear 24 h period in the activity.

In Figure 8 we highlight the large entries in the matrix $\widehat{\mathcal{Q}}$ by showing those entries that are greater that the mean (that is, the arithmetic average over all entries). We see that in the ordering supplied with the data, there is no visual sign of asymmetry. However, the matrix is highly

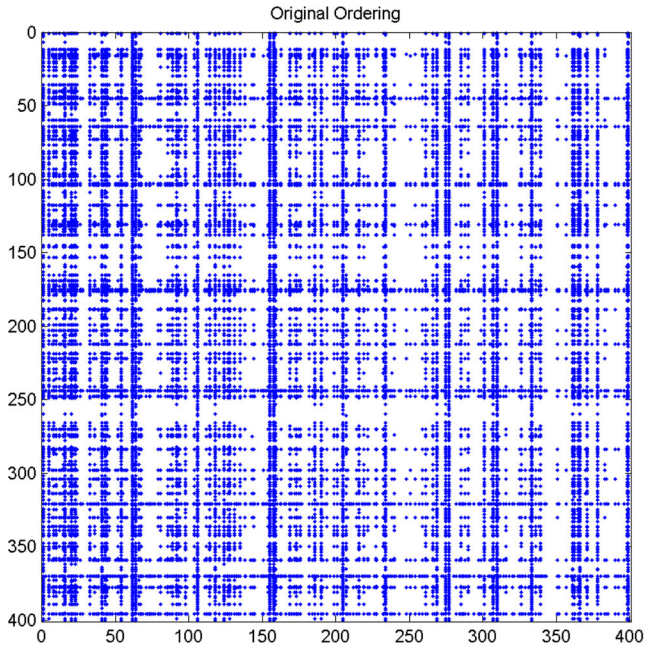

Fig. 8. Dominant entries in the normalized dynamic communicability matrix $\widehat{\mathcal{Q}}$ in its original ordering. Dots denote entries that are greater than the overall mean. The one-sum for this ordering is 1750.9.

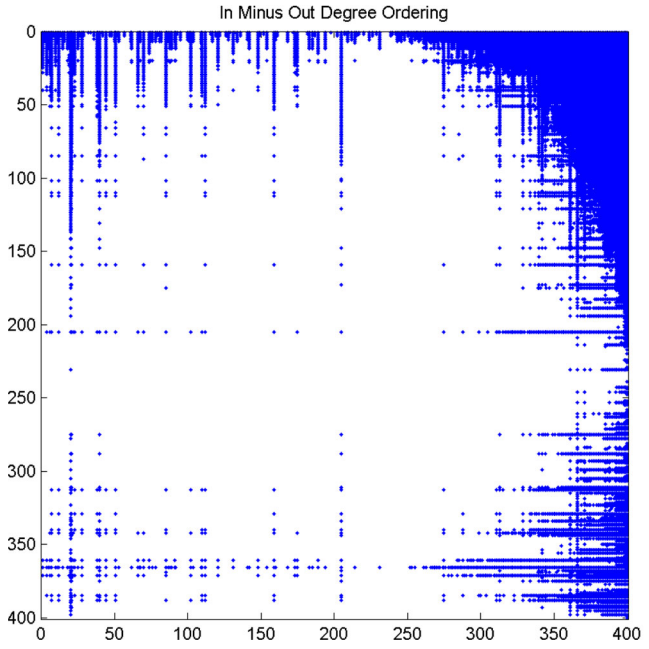

Fig. 9. Dominant entries in the normalized dynamic communicability matrix $\widehat{\mathcal{Q}}$ with node ordering determined by in minus out degree. Dots denote entries that are greater than the overall mean. The one-sum for this ordering is −9 547.7.

asymmetric, in the sense that

$$\|\widehat{\mathcal{Q}}\| = 1 \quad \text{and} \quad \|\widehat{\mathcal{Q}} - \widehat{\mathcal{Q}}^T\| = 0.9968.$$

In Figure 9 we show the result of the "broadcast minus receive" reordering exercise, again marking entries that

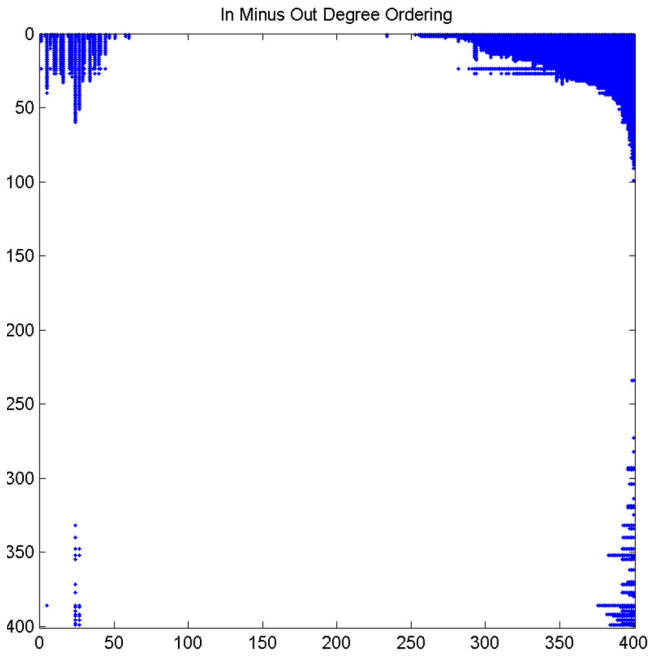

Fig. 10. As in Figure 9, but with all voice call interactions regarded as undirected. Dominant entries in the normalized dynamic communicability matrix $\widehat{\mathcal{Q}}$ with node ordering determined by in minus out degree. Dots denote entries that are greater than the overall mean. The one-sum for this ordering is −4 543.2.

exceed the overall mean. The one-sum of this reordered version of $\widehat{\mathcal{Q}}$ is −9 547.7, in contrast with the value 1 750.9 arising from the original ordering in Figure 8.

In Figure 10 we repeat the computations leading to Figure 9, except that voice call interactions are regarded as undirected. So each $A^{[k]}$ is such that $(A^{[k]})_{ij} = 1$ and $(A^{[k]})_{ji} = 1$ when i engaged in a phone conversation with j at least once in that 30 min period, independently of who initiated the call. While the directed edge construction used above retains information about the "chain of command," it can be argued that this undirected alternative is more relevant when we are concerned about the possible flow of information. We see in Figure 10 that there remains a visually striking asymmetry – since each $A^{[k]}$ is now symmetric, this feature of the dynamic communicability matrix $\widehat{\mathcal{Q}}$ arises solely from the time-ordering of the interactions. The one-sum for the broadcast minus receive ordering is −4 543.2 and we now have

$$\|\widehat{\mathcal{Q}}\| = 1 \quad \text{and} \quad \|\widehat{\mathcal{Q}} - \widehat{\mathcal{Q}}^T\| = 0.9995.$$

5 Discussion

Techniques that summarize complex time-dependent networks must necessarily discard information in order to focus on certain key aspects. Our emphasis here is on the pairwise temporal asymmetry between nodes, defined

via their propensity to exchange information through dynamic links. Alternative methods for extracting information may of course be derived, depending on what features are deemed important. Such choices of emphasis are essentially a modelling issue. We note that one particular choice is whether to respect the arrow of time. For example, the recent work [27] is aimed at quantifying network centrality based on a *supra-centrality matrix* of the form

$$\mathbb{M} = \begin{bmatrix} \epsilon M^{[1]} & I & 0 & \dots \\ I & \epsilon M^{[2]} & I & \ddots \\ 0 & I & \epsilon M^{[3]} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{13}$$

Here, $M^{[k]}$ is an N by N centrality matrix that is relevant to time k and I is the N by N identity. The authors use the simple choice $M^{[k]} \equiv A^{[k]}$ to illustrate the idea. The parameter ϵ is included to account for the fact that the identity matrices represent "between layer" connections (generally linking node i at time k to node i at times $k-1$ and $k+1$) which are inherently different to the "within layer" weights arising from the network data. The key idea in [27] is to apply a standard static network centrality algorithm to the supra-centrality matrix (13). However, we note that if the individual network slices are undirected, then with $M^{[k]} \equiv A^{[k]}$ (or with $M^{[k]} \equiv (I - \alpha A^{[k]})^{-1}$ in the Katz-based setting), the supra-centrality matrix $\mathbb{M}$ is symmetric. It then becomes apparent that any static network algorithm applied to $\mathbb{M}$ must be oblivious to the type of temporal asymmetry that we have discussed.

To see this another way, from a message-passing perspective the use of the identity matrices in (13) serves to introduce links both backwards and forwards through time, so that combinatoric or random walk-based algorithms violate time's arrow when interpreted on the larger matrix.

Overall, we believe that the work described here adds value to the current range of tools for summarisizing temporal networks, and we note that there are several promising directions in which these ideas could be extended, including

(a) the development of an algorithm that applies over arbitrarily long time intervals by discarding chronologically old information, based on the technique in [25];
(b) following on from (a), the study and categorization of individual nodes as they move dynamically through the nodal hierarchy;
(c) using ideas from [28], a related algorithm that uses a continuous-time network representation, thereby avoiding the need to specify time windows.

DJH acknowledges support from a Royal Society Wolfson Award and an EPSRC/Digital Economy Fellowship ref. EP/M00158X/1. AVM was supported by the EPSRC and Stipso (Edinburgh) via an Impact Acceleration Account secondment ref. EP/K503861/1.

References

1. M.E.J. Newman, *Networks: An Introduction* (Oxford University Press, 2010)
2. L.d.F. Costa, F.A. Rodrigues, G. Travieso, P.R.V. Boas, Adv. Phys. **56**, 167 (2007)
3. S. Fortunato, Phys. Rep. **486**, 75 (2010)
4. M. Girvan, M.E. Newman, Proc. Natl. Acad. Sci. **99**, 7821 (2002)
5. P. Holme, F. Liljeros, C.R. Edling, B.J. Kim, Phys. Rev. E **68**, 056107 (2003)
6. A.V. Mantzaris, EPJ Data Sci. **3**, 26 (2014)
7. M.P. Rombach, M.A. Porter, J.H. Fowler, P.J. Mucha, SIAM J. Appl. Math. **74**, 167 (2014)
8. M.A. Riolo, M.E.J. Newman, J. Complex Netw. **2**, 121 (2014)
9. O. Sporns, R. Kotter, Plos Biol. **2**, 1910 (2004)
10. G.H. Golub, C.V. Loan, *Matrix Computations*, 4th edn. (Johns Hopkins University Press, 2012)
11. T.A. Davis, *Direct Methods for Sparse Linear Systems* (*Fundamentals of Algorithms 2*) (Society for Industrial and Applied Mathematics, Philadelphia, 2006)
12. S.T. Barnard, A. Pothen, H.D. Simon, Numerical Linear Algebra **2**, 317 (1995)
13. D.J. Higham, J. Comput. Appl. Math. **158**, 61 (2003)
14. F. Chung, *Spectral Graph Theory* (American Mathematical Society, Providence, 1997)
15. G. Strang, *Computational Science and Engineering* (Wellesley-Cambridge Press, 2008)
16. R. Van Driessche, D. Roose, Parallel Comput. **21**, 29 (1995)
17. P. Grindrod, Phys. Rev. E **66**, 066702 (2002)
18. J.J. Crofts, D.J. Higham, Internet Math. **7**, 233 (2011)
19. P. Grindrod, D.J. Higham, M.C. Parsons, E. Estrada, Phys. Rev. E **83**, 046120 (2011)
20. L. Katz, Psychometrika **18**, 39 (1953)
21. H.H.K. Lentz, T. Selhorst, I.M. Sokolov, Phys. Rev. Lett. **110**, 118701 (2013)
22. T. Graham, S. Wright, J. Comput. Mediated Commun. **19**, 625 (2014)
23. A. Duchon, E.S. Patterson, in *Social Computing, Behavioral-Cultural Modeling and Prediction* (Springer, 2014), pp. 51–58
24. D. Huffaker, J. Wang, J. Treem, M. Ahmad, L. Fullerton, D. Williams, M. Poole, N. Contractor, IEEE Int. Conf. Comput. Sci. Eng. **4**, 326 (2009)
25. P. Grindrod, D.J. Higham, SIAM Rev. **55**, 118 (20013)
26. G.G. Grinstein, C. Plaisant, S.J. Laskowski, T. O'Connell, J. Scholtz, M.A. Whiting, VAST 2008 Challenge: Introducing mini-challenges, in *IEEE VAST* (2008), pp. 195–196
27. D. Taylor, S.A. Myers, A. Clauset, M.A. Porter, P.J. Mucha, arXiv:1507.01266 (2015)
28. P. Grindrod, D.J. Higham, A dynamical systems view of network centrality, Proc. R. Soc. A **470**, 20130835 (2014)

Open Access This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.