

SCIENTIFIC REPORTS



OPEN

Using higher-order Markov models to reveal flow-based communities in networks

Received: 18 September 2015

Accepted: 02 March 2016

Published: 31 March 2016

Vsevolod Salnikov¹, Michael T. Schaub^{1,2} & Renaud Lambiotte¹

Complex systems made of interacting elements are commonly abstracted as networks, in which nodes are associated with dynamic state variables, whose evolution is driven by interactions mediated by the edges. Markov processes have been the prevailing paradigm to model such a network-based dynamics, for instance in the form of random walks or other types of diffusions. Despite the success of this modelling perspective for numerous applications, it represents an over-simplification of several real-world systems. Importantly, simple Markov models lack memory in their dynamics, an assumption often not realistic in practice. Here, we explore possibilities to enrich the system description by means of second-order Markov models, exploiting empirical pathway information. We focus on the problem of community detection and show that standard network algorithms can be generalized in order to extract novel temporal information about the system under investigation. We also apply our methodology to temporal networks, where we can uncover communities shaped by the temporal correlations in the system. Finally, we discuss relations of the framework of second order Markov processes and the recently proposed formalism of using non-backtracking matrices for community detection.

Dynamics on complex networks, such as the diffusion of information in social networks, are commonly modelled as Markov processes. An advantage of this approach is that for every (static) network with positive edge-weights we can define a corresponding Markov process by interpreting the network as the state space of a random walker, and assigning the state-transition probabilities according to the link weights. This direct correspondence between the state space of the Markov process and the network enables us to examine the interplay between structure and dynamics from two sides. On the one hand, one can assess how the topological properties of a network influence the dynamical process. On the other hand, this coupling between topology and dynamics allows us to explore the structure of a network by means of a dynamical process. Specifically, for a linear Markov process the impact of the network structure on the dynamics will be mediated by the spectral properties of the matrix governing the time-evolution of the process, e.g. the adjacency matrix or the Laplacian^{1,2}. Vice versa, spectral properties can be used to uncover salient structural properties of a network, such as modular organisation³⁻⁵.

While simple Markov models have been very successful in modelling dynamics of complex systems and found many applications, they have one obvious disadvantage. In this class of models, the future state of the system only depends on its current state and does not account for its history. In a diffusion process, for instance, the next position of a random walker only depends on the currently occupied node and its outgoing links, but not on any of the previously visited nodes. However, as it has been emphasised recently, for a broad range of networked systems, flows tend to exhibit a temporal path dependence^{6,7}. Think of human mobility: the places a person is likely to visit next, depend in most cases strongly on where the person came from. For instance, a person coming to work from home is likely to return home afterwards⁸. Other examples of processes with temporal memory include web traffic, journal citation flows and email cascades. Such processes therefore cannot be reproduced accurately by simple Markov models. However, the impact of this temporal correlations can often be already well-approximated by second-order Markov (\mathcal{M}_2) models⁶. Importantly, the transition probabilities to define these models can be obtained empirically by measuring pathways of interaction cascades, rendering such an approach suitable for applications like information spreading or human mobility.

¹naXys, University of Namur, Rempart de la Vierge 8, 5000 Namur, Belgium. ²ICTEAM, Université catholique de Louvain, Avenue George Lemaître 4, B-1348 Louvain-la-Neuve, Belgium. Correspondence and requests for materials should be addressed to V.S. (email: vsevolod.salnikov@unamur.be) or M.T.S. (email: michael.schaub@unamur.be) or R.L. (email: renaud.lambiotte@unamur.be)

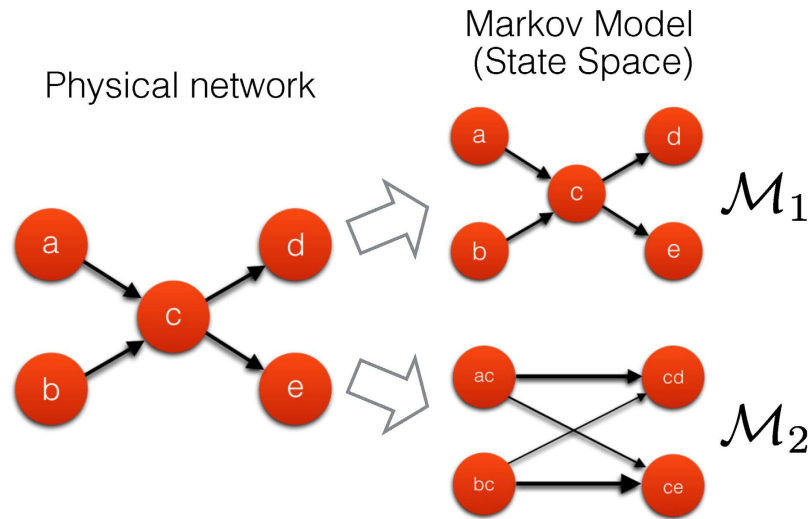


Figure 1. Schematic – dynamics on networks as Markov processes. A process on a physical network (left) may be abstracted in a number of different ways – often this is done via a Markov model. In a first order Markov model \mathcal{M}_1 (right side, top panel), the state space is isomorphic to the physical network: every node corresponds to one state, every link indicates a transition between those states. In a second order Markov model the state space is structured like a *directed line-graph* of the original network. The states in this \mathcal{M}_2 network can be identified with the directed edges in the original network and are connected if it is possible to traverse from one edge to another edge in the original directed network. Note that, when projecting back these \mathcal{M}_2 dynamics onto the physical network, the probability to move from one node to another will thus appear non-Markovian.

In the standard Markovian network model (\mathcal{M}_1), the elementary states are identified with the nodes of the original network (see Fig. 1). In the \mathcal{M}_2 model, elementary states correspond to sequences of two nodes, and thus can be identified with *directed edges* in the original network (see Fig. 1). Therefore we may think of a \mathcal{M}_2 model alternatively as a random walk between directed edges of the original network. The state space of the \mathcal{M}_2 model defines a new network describing the observed dynamics, which we call here the \mathcal{M}_2 or memory network. The structural properties of the memory network can now be studied by many tools of network science, allowing us to uncover interesting patterns of flow in an \mathcal{M}_2 model.

The purpose of this paper is to highlight how the formalism of higher order Markov models provides a simple means to extend network theoretic tools to take into account important dynamical properties as encoded in the \mathcal{M}_2 representation. Indeed many network theoretic tools may be understood as the outcome of a matrix iteration or eigenvector computation⁹, which can be naturally associated to a dynamical process.

In this paper we focus on the problem of community detection, though higher-order models can be equally applied to other problems, including the assessment of node centralities^{6,9} or the speedup (or slowdown) of spreading processes⁷. While Ref. 10,11 discussed how the map equation can account for higher-order flows in community detection, here our main example will be the Markov stability^{3,12,13} formalism, as it incorporates many commonly used community detection measures as special cases, notably, the concept of Modularity¹⁴, diverse Potts models¹⁵, and spectral clustering^{16,17}. We thereby highlight that all these classical algorithms can indeed be naturally generalised to account for memory effects by using \mathcal{M}_2 networks, showing that these representations provide a general tool for the analysis of a dynamics occurring on a latent network structure. Moreover, as spectral clustering can be shown to be a special case of the Markov Stability formalism for long times, we can draw connections to recently proposed techniques of graph partitioning based on non-backtracking random walks¹⁸, and discuss how these can be understood from the point of view of second order Markov processes. We remark that as clustering an \mathcal{M}_2 network is equivalent to finding a partition of the edges of the original system, it naturally leads to the detection of overlapping communities in the same way as the analysis of line graphs¹⁹. This can be a highly desirable feature, especially when analysing social networks, which tend to be organised in overlapping social circles^{20,21}.

The remainder of this article is organised as follows. Initially, we review the Markov stability formalism for community detection, a general framework to detect flow-based communities in complex networks, and particularly emphasise its properties in the case of directed networks². We show how this quality function can be naturally generalised for the analysis of \mathcal{M}_2 networks, which we illustrate by studying a flight network of the United States from the perspective of second order Markov models. In this context, we also discuss the possibility to generate realistic pathway data using models of second-order Markov processes, even if only time aggregated network information is available. Subsequently, we propose a mechanism to extract pathway statistics to build second-order models from event-based, temporal network data, which do not contain pathway statistics *a priori*. Using computer-generated data and time-resolved interactions records of school-children, we demonstrate how we can uncover communities which capture important flow-constraints imposed by the temporal activation

patterns of the links. Finally, we draw connections between the \mathcal{M}_2 network representation and non-backtracking random walks to uncover communities in sparse networks.

Markov Dynamics and Community Detection

Random walks on directed networks. Let us consider a continuous-time random walk on a network with N nodes, governed by the following Kolmogorov forward equation:

$$\dot{p}_j(t) = -p_j(t) + \sum_i p_i(t) \frac{A_{ij}}{k_i^{\text{out}}}. \quad (1)$$

Here $p_j(t)$ denotes the probability of a walker to be present on node j at time t ; A_{ij} is the weight of the link from node i to node j , i.e., A is the (weighted) adjacency matrix; finally, k_j^{out} is the weighted out-degree of node j .

Clearly this dynamics is driven by the normalised Laplacian matrix L , defined as $L_{ij} = -\delta_{ij} + A_{ij}/k_i^{\text{out}}$. The Perron-Frobenius theorem guarantees that a unique stationary solution exist for the above dynamics if the network is strongly connected. For an undirected network, this stationary solution is $\pi_j = k_j/2M$, where $k_j = k_j^{\text{out}} = k_j^{\text{in}}$ is the node degree and M is the total weight of the undirected links. For a general directed network the stationary solution is given by the dominant (left) eigenvector of L , which depends on the global network properties and cannot be expressed by a simple analytical formula.

Making the dynamics ergodic (when it is not). Note that for the continuous-time random walk defined above, a strongly connected graph implies a unique stationary distribution and an ergodic dynamics. These criteria are a common requirement for dynamics-based methods for network analysis^{3,10}. However, in a majority of real-world systems the system contains not just one but several strongly connected components (SCCs), whose sizes typically have different orders of magnitude. A standard solution is to address this issue by either neglecting some nodes and focusing on the largest SCC only, or by making the system strongly connected through random teleportations²². The first solution has the advantage of not distorting the dynamics, while the second allows for the analysis of the whole system. When analyzing real-world systems in the following, we will for these reasons use the first solution if the vast majority of nodes belongs to the largest SCC, while we will opt for the second solution if the system comprises a large number of disconnected components.

Community structure from a dynamical viewpoint. The structure of a network has a strong effect on the dynamics of a diffusing random walker: in unstructured (random) networks, the diffusion process will evolve almost isotropically and the walker will quickly reach its stationary distribution. However, if the network contains structure, a walker can get trapped inside a group of nodes for a time far longer than expected. Such groups of nodes thus constitute flow-retaining, *dynamical communities* in the network. Hence a random walk dynamics can effectively be used to define a quality function for a network partition based on the persistence of the diffusion inside the groups. By optimising such a quality function, we can therefore search for a modular partition of the network. Important examples for such an approach include the map equation¹⁰ and the Markov stability framework³, which we consider in this paper. Note that, as highlighted in refs 4 and 10 this notion of community is markedly different to the commonly considered structural viewpoint of communities, in that we are interested in flow-retaining, rather than densely (homogeneously) connected substructures.

The Markov stability of a partition \mathcal{P} is defined as:

$$R(t, \mathcal{P}) = \sum_{C \in \mathcal{P}} P(C, t) - P(C, \infty) \quad (2)$$

where $P(C, t)$ is the probability for a walker to be in community C initially and at time t , for a system at stationarity. Note that as all information about the initial position is lost after infinite time, $P(C, \infty)$ also describes the probability of two independent walkers to be in C .

For the process (1), the Markov stability quality function can be written explicitly as:

$$R(t, \mathcal{P}) = \sum_{C \in \mathcal{P}} \sum_{i, j \in C} \left[\pi_i(e^{tL})_{ij} - \pi_i \pi_j \right], \quad (3)$$

where e^{tL} denotes the matrix exponential. Intuitively, with increasing time the walker will be able to explore larger and larger parts of the graph. The larger the time, the larger the modules that will be found in general. By 'zooming' through different Markov times one can thus reveal adequate scales of robust community structure, which manifest themselves as plateaux in time where the same partitions are found consistently^{4,23}. In the limit $t \rightarrow \infty$, it can be shown by eigenvalue decomposition that $R(t, \mathcal{P})$ is maximised by a bi-partition in accordance with the normalised Fiedler eigenvector, a classic method in spectral clustering¹⁷. Interestingly, the expression for Markov stability can be rewritten as the Newman-Girvan modularity¹⁴ of a *different* network, whose adjacency matrix is given by

$$Y_{ij} = \frac{1}{2} \left[\pi_i(e^{tL})_{ij} + \pi_j(e^{tL})_{ji} \right]. \quad (4)$$

Here $\pi_i(e^{tL})_{ij}$ is the flow of probability from i to j after a time interval t at stationarity. In this new network Y , the weight of a connection between two nodes is thus modulated according to its importance for the diffusion dynamics. This rewriting of Markov stability as the modularity of a different network enables us to use any

modularity based algorithm for the optimization of Markov stability, too. For the results in the present manuscript, we have made use of the Louvain algorithm²⁴, a fast greedy optimisation heuristic that has been shown to have a good performance in the literature.

In practice, it can be more efficient to consider (3) in the limit of small times $t \rightarrow 0$. Performing a Taylor expansion and keeping only linear terms in t results in the linearised Markov stability:

$$Q(t) = (1 - t) + \sum_C \sum_{i,j \in C} \left[t \frac{\pi_i A_{ij}}{k_i^{\text{out}}} - \pi_i \pi_j \right]. \tag{5}$$

Note that for an *undirected* network, we have $\pi_i = k_i/2M$ and $k_i^{\text{out}} = k_i$, and one recovers the modularity of the original network for $t = 1$ as well as the Potts-model heuristic¹⁵ for other Markov times (see Ref. 3,12 for details). However, for *directed* networks (5) is not equivalent to the directed modularity of the original network, but is given by:

$$Y_{ij} = \frac{1}{2} \left[\pi_i \frac{A_{ij}}{k_i^{\text{out}}} + \pi_j \frac{A_{ji}}{k_j^{\text{out}}} \right]. \tag{6}$$

Interestingly, similar adjusted adjacency (or Laplacian) matrices have been proposed as means for community detection in the literature based on different reasoning^{24,25}.

Note that, as with all applications of unsupervised algorithms, care is needed when interpreting the results. For instance, the optimisation of Markov Stability for a fixed Markov time inherits the limitations of modularity maximisation, such as a resolution limit²⁷, and a high degeneracy of optimal solutions²⁸. Nevertheless, this class of methods has been applied successfully in numerous applications²⁹. Indeed, some of these problems can be circumvented by carefully sweeping through different resolution parameters^{4,29} and finding partitions that are robust over a range of parameters^{4,13,23,31}. Most importantly, however, the notion of community inherent to the methods presented here is based on flows^{3,4,10}. This is in contrast to methods like stochastic blockmodels (SBM), see e.g. refs 32–34, which employ a generative model for the whole network and aim at finding patterns of pairwise connections in the system, though very recent work aims to incorporate dynamical aspects within the SBM framework³⁵.

Dynamical Community Detection using Higher-Order Markov Models

From first to second order Markov models. Let us now show how we can incorporate second-order Markov models into the Markov stability framework. As higher-order Markov models provide more faithful representations of the dynamics observed in real world systems, this enables us to capture more of the real flow constraints in the uncovered communities. For simplicity let us initially consider undirected networks composed of N nodes and M links. The dynamics of a second order Markov process are encoded by the transition matrix $T(\vec{ij} \rightarrow \vec{jk})$, which describes the probability that a walker moves from node j to node k if it came from i in the previous step. By definition, this transition matrix is normalised such that $\sum_k T(\vec{ij} \rightarrow \vec{jk}) = 1$.

As sequences of two vertices in our original network correspond to the nodes in the \mathcal{M}_2 network, the entries in T describe precisely the transitions from \mathcal{M}_2 -node \vec{ij} to \mathcal{M}_2 -node \vec{jk} . The (second-order) \mathcal{M}_2 process on the original network is thus equivalent to a first-order Markov process, albeit on a different network: namely, the \mathcal{M}_2 network composed of $2M$ nodes. As each undirected link of the original network can be traversed in two distinct directions (from i to j , and vice versa) it accounts for 2 nodes in the \mathcal{M}_2 network, \vec{ij} and \vec{ji} . This implies that the \mathcal{M}_2 network is *directed* even if the original network is undirected. To see this, observe that if $k \neq i$, there cannot be a link between \vec{jk} and \vec{ij} , even if a transition between \vec{ij} and \vec{jk} exists. Henceforth we use Greek letters to denote \mathcal{M}_2 nodes, and Latin characters to denote the nodes in the original \mathcal{M}_1 network.

Similarly to (1), we can define a continuous-time random walk on the \mathcal{M}_2 network as:

$$\dot{p}_\beta(t) = -p_\beta(t) + \sum_\alpha p_\alpha(t) T_{\alpha\beta}, \tag{7}$$

where $p_\alpha(t)$ is the probability of finding a walker on \mathcal{M}_2 node α at time t . Likewise the stationary distribution π_α of this process is given by the left eigenvector of the corresponding Laplacian, associated with eigenvalue 0:

$$\sum_\alpha \pi_\alpha [T_{\alpha\beta} - \delta_{\alpha\beta}] = 0. \tag{8}$$

Note that we can also observe how a simple \mathcal{M}_1 Markov dynamics evolves from the point of view of an \mathcal{M}_2 network, i.e., we can view the \mathcal{M}_1 dynamics from the point of view of transitions between the (directed) edges of the graph. As this is equivalent to lifting the \mathcal{M}_1 dynamics into the larger \mathcal{M}_2 state-space, we will denote this representation as a $\mathcal{M}_1^{\text{expanded}}$ network. In this case it can be shown easily that $\pi_\alpha = w_\alpha/W$, where w_α is the weight of edge α and $W = \sum_\alpha w_\alpha$ ^{2,6}. The transition probability to any out-neighbour of α on the $\mathcal{M}_1^{\text{expanded}}$ network is simply

$$T_{\alpha\beta}^{\mathcal{M}_1} = \begin{cases} w_\beta/k_\alpha^{\text{out}} & \text{for } \beta \in \sigma_\alpha^{\text{out}}, \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

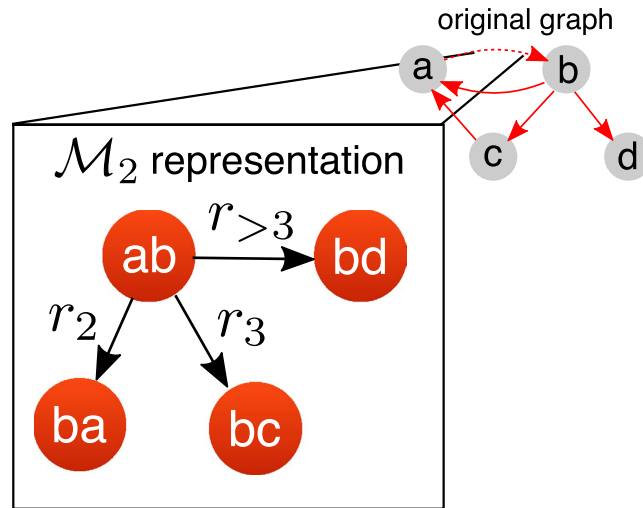


Figure 2. Construction of a \mathcal{M}_2 network using a second-order transition model. From each directed edge in the original network a walker can in principle perform three types of moves, which get weighted according to the parameters r_i : a *return step* (r_2), a *triangular step* (r_3) and an *exploratory step* ($r_{>3}$). The picture shows these different options for a generic network edge.

Here $\sigma_\alpha^{\text{out}}$ is the set of out-neighbours of α , and $k_\alpha^{\text{out}} = \sum_{\beta \in \sigma_\alpha^{\text{out}}} w_\beta$ is the out-strength of α . Note that, since the original network has been undirected we have $k_\alpha^{\text{out}} = k_j$, where j is the endpoint of memory node (the directed edge) $\alpha = \vec{ij}$.

Models of second-order Markov processes. While second order transition matrices T can be directly generated from temporal pathway data, in some cases only information about the aggregated (first-order) network might be available. However, in such cases we can still create \mathcal{M}_2 networks using a simple memory model, calibrated by the pathway statistics of similar datasets for which this temporal information is obtainable. For instance, the temporal information pathways of one Email dataset may be used to fit model-parameters to generate a second-order model for different email communication network, where only aggregated information is obtainable.

The key idea underpinning the model is to weight different types of transitions between two directed edges. As illustrated in Fig. 2, we define three different types of transitions⁶.

1. A *return step*, in which a walker coming from \vec{ij} jumps to \vec{ji} . In other words: a walker coming from node i to j returns to node i .
2. A *triangular step*, in which a walker coming from \vec{ij} moves to edge \vec{jk} , where $k \neq i$ is a neighbor of i .
3. An *exploratory step*, in which the walker moves from \vec{ij} to an edge \vec{jl} , whose endpoint l is neither i , nor any of the neighbors of i .

To account for their relative importance we assign positive weights r_2 , r_3 and $r_{>3}$ to the different types of transition as follows. Let us denote the adjacency matrix of the directed line graph associated with our network by G . Then we can decompose G into three matrices G^{ret} (return links), G^{tri} (triangular), and G^{exp} (exploratory) each containing only the links of the respective type. To obtain a weighted memory model we now define the weighted $2M \times 2M$ adjacency matrix:

$$G_{\alpha\beta}^{\text{mem}} = r_2 G_{\alpha\beta}^{\text{ret}} + r_3 G_{\alpha\beta}^{\text{tri}} + r_{>3} G_{\alpha\beta}^{\text{exp}}, \quad (10)$$

from which we can compute the associated second order transition matrix T^{model} simply by normalising each row to sum to 1.

As can be easily verified, this definition implies that when we project the resulting walk onto the node space, we obtain a (first order) Markov process when $r_2 = r_3 = r_{>3} = \text{const}$. We further remark that if the dynamics is ergodic on a graph for any set of parameters, it will remain ergodic for any value of r_2 , r_3 and $r_{>3}$, provided each parameter is strictly positive.

Higher order Markov dynamics reveal communities in a network of flight pathways. Let us now demonstrate how second-order Markov dynamics can help to uncover the organisation of systems where pathway data are available, and compare their outcome with the community structure obtained by first-order dynamics and by simplified models of second-order Markov dynamics, as defined in the section above. To do so, we consider a flight network in the United States. The data used to construct the network consists of individual flight trajectories of people navigating between different airports in the US. From these trajectories one can directly obtain a \mathcal{M}_1 or \mathcal{M}_2 network as discussed above and in Ref. 6. The main purpose of our analysis here is to illustrate

the differences that may arise in community detection when considering the same mobility data from different modelling perspectives.

We analysed the modular structure of this system for the following three scenarios: i) a first order Markov network ($\mathcal{M}_{1\text{expanded}}$), viewed from the perspectives of the edges; ii) a second order Markov network (\mathcal{M}_2) where the transition probabilities are directly obtained from empirical data; iii) a second order Markov network ($\mathcal{M}_{2\text{model}}$), where transitions are approximated by a simple second-order transition matrix T^{model} (see text), whose parameters have been fitted from the data. In each case, optimising the linearised stability for these different Markov models yields a clustering of the edges, which can be interpreted in terms of an overlapping community structure at the airport level. To compare the different scenarios we concentrate on the results obtained for Markov time $t = 1$ (see Fig. 3). Let us also note that, for each network, only few hardly active airports do not belong to the largest SCC. For this reason, we restrict the scope to nodes in this SCC. To be more precise, we restrict the scope to the intersection of the largest SCCs of the three processes, in order to ensure that the Markov stability is well-defined for each of them.

In order to compare the communities associated to each Markov process, we first calculate the normalised variation of information³⁶ between the three different partitions of \mathcal{M}_2 nodes obtained by optimising stability at $t = 1$. By construction, smaller values of variation of information indicate more similar community structures. We obtain a variation of information of 0.42 between \mathcal{M}_2 and $\mathcal{M}_{1\text{expanded}}$, 0.36 between \mathcal{M}_2 and $\mathcal{M}_{2\text{model}}$, and 0.38 between $\mathcal{M}_{1\text{expanded}}$ and $\mathcal{M}_{2\text{model}}$. These results confirm that the clusters obtained from $\mathcal{M}_{2\text{model}}$ provide an intermediate solution between \mathcal{M}_2 and $\mathcal{M}_{1\text{expanded}}$, with the advantage of requiring far fewer parameters than in the \mathcal{M}_2 case.

After having found the edge-communities in each scenario, each node can be characterised by the set of group labels of its incident edges. In order to quantify the apparent difference between the covers, we measure for each node the entropy of its associated group labels:

$$S_i = - \sum_c p_i(c) \ln p_i(c), \quad (11)$$

where $p_i(c)$ is its fraction of edges assigned to community c . A value $S_i = 0$ indicates that the node is associated to a single community, while higher values indicate a more diverse participation. The level of overlap of an edge partition can now be characterised by the distribution of entropy values on the set of nodes. We observe in Fig. 3 that analyzing the \mathcal{M}_2 network results in more overlapping communities than the expanded first-order model, while the $\mathcal{M}_{2\text{model}}$ network is characterized by intermediate participation values. Similarly as observed for the map equation⁶, accounting for memory via second order dynamics therefore uncovers communities with a stronger overlap, in agreement with empirical observations that higher-order dynamics tends to constrain flows within these modules.

Analyzing Time-Stamped Temporal Networks without Pathway Data.

Second-order Markov models of time-stamped temporal networks. Node or link activity in networked systems often exhibits non-trivial temporal patterns, such as heterogeneous inter-event times and correlations between activation times of neighbouring edges³⁷. Different network representations of such temporal data can be adopted, each associated to a different notion of temporal community. A first, ‘naive’ approach is to neglect the edge-timings and work with a static network representation, where the weight of each edge corresponds to an aggregation of the activity over the whole observed time interval. In this case, community detection aims at grouping nodes which have the aggregated edge-weight concentrated inside modules.

Alternatively, one can represent the system by a set of (coupled) adjacency matrices A_t , where each A_t represents the system in a short-time frame of the whole observation period. By applying community detection to this representation one tries to uncover meaningful structures in each time window, while enforcing some continuity between different time intervals³⁸. This approach accordingly aims at tracking the evolution of communities in the course of time.

As we now discuss, a third notion of community is naturally associated to the representation of dynamics on temporal networks as a second-order Markov process. In situations when the activations of neighbouring edges present temporal correlations, the dynamics of a random walker is expected to be poorly reproduced by a first-order Markov process⁷. However, by finding a second-order representation of the temporal data and using the methodology introduced above, one can capture the observed temporal constraints on flow, including causal paths and a high levels of synchrony between edge activity.

The mapping from a time series into a second-order Markov model is realized as follows. We consider a system described by an ordered set of adjacency matrices A_t defining the connections between nodes at time $t \in [1, T]$, where T is the number of observations. The static representation of the temporal network is provided by the adjacency matrix $A_{\text{static}} = \sum_t A_t$. Each directed edge within A_{static} defines a \mathcal{M}_2 node. The connection strength between \mathcal{M}_2 nodes is now obtained by simulating pathways of a random walk process on the temporal network as follows. A walker is initially randomly assigned to a node. At every time step the walker waits until at least one edge is available for transport. If an edge becomes available, the walker leaves the node with probability $(1 - p_s)$ and remains on its current node with probability p_s . If there are multiple possible transitions, the walker takes each edge with a probability proportional to its weight. This process is repeated multiple times for the observed interval $[1, T]$ in order to generate sufficiently many trajectories. From these trajectories, one can simply construct a \mathcal{M}_2 network by evaluating the transition frequencies between different edge pairs.

We remark here that several other procedures could be defined to generate \mathcal{M}_2 networks, too. For instance, we could account for the duration of the intervals between time steps, or define the walker’s leaving probability to be

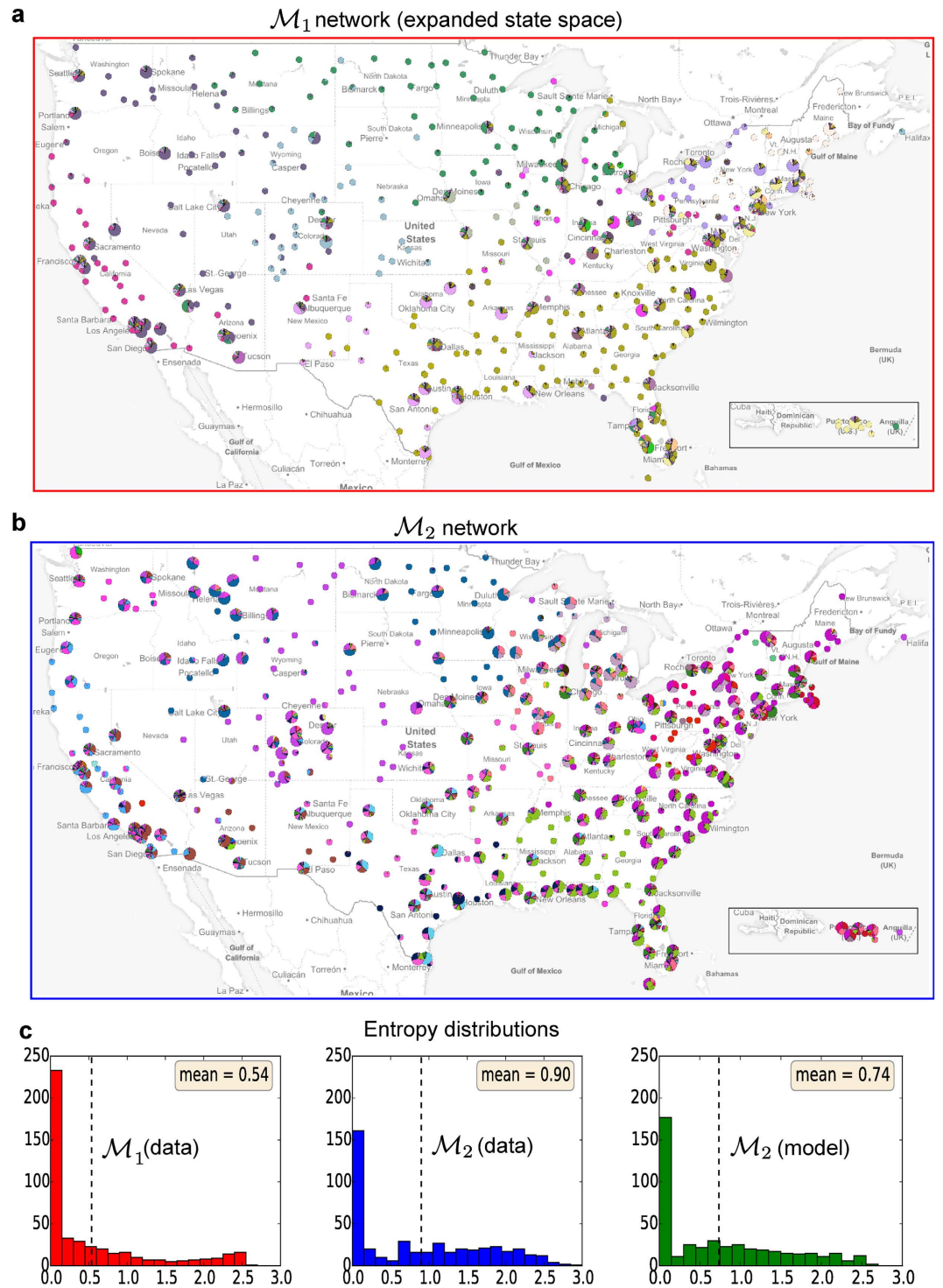


Figure 3. Clustering analysis of a passenger traffic-network between airports in the US. (a) Results based on clustering the \mathcal{M}_1 expanded system representation. (b) Clustering results obtained from the \mathcal{M}_2 system representation. In (a,b) each airport is represented by a pie chart indicating the participation of the airport in different communities. For visual clarity, nodes with a community participation entropy smaller than one are displayed as smaller nodes. (c) Distributions of the community participation entropy across the network for the \mathcal{M}_1 expanded (left), \mathcal{M}_2 model (middle) and \mathcal{M}_2 (right) network. All maps are created with <https://github.com/andrea-cuttone/geoplotlib> using map tiles and data from <https://www.mapbox.com/map-feedback/> Mapbox and <http://www.openstreetmap.org/copyright> OpenStreetMap, © OpenStreetMap contributors.

proportional to the number of available contacts. Note that in the above construction, when $p_s = 0$, the walker always takes the first available edge, and one recovers the dynamics studied in e.g. ref. 39. The use of non-zero

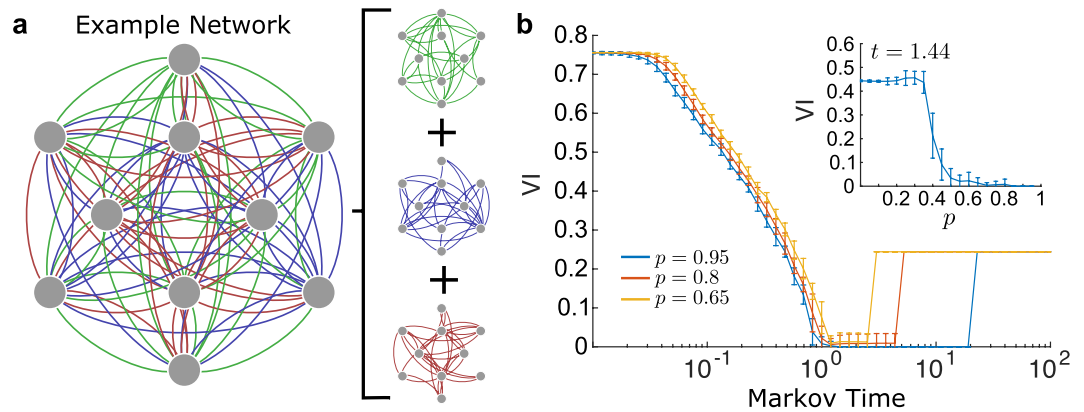


Figure 4. Analysis of an example temporal network with 10 nodes and 3 groups. (a) Shows a complete graph with 10 nodes, whose (directed) edges are divided into 3 groups, such that each node has exactly 3 outgoing edges of each type. At each time-step only edges belonging to one group are active and available for transport. The currently active group ceases to be active with probability $1 - p_k$, and is replaced by a randomly selected group. A random walker traversing the network is thus more likely to remain within the same group, the higher the probability p_k . In panel (b), the variation of information between the planted structure and the partition found by Markov stability in the \mathcal{M}_2 network is displayed as a function of Markov time. We can see that there is a plateau for values of time slightly larger than $t = 1$. As shown in the inset, the method successfully uncovers the organisation into 3 modules for sufficiently large values of p_k , i.e., when the groups remain active long enough in time such that the walker remains trapped inside a community for non-negligible periods.

values of p_s introduces a source of randomness in the dynamics, which can prevent spurious effects such as an overly strong tendency for backtracking as observed in Ref. 40. However, with increasing values of p_s , the original ordering of events becomes less important, and the impact of the exact timings is expected to be diluted⁴¹.

Community detection in \mathcal{M}_2 networks constructed from temporal data: an illustrative example.

Let us now show that a \mathcal{M}_2 representation helps at uncovering hidden structures in temporal networks. As a first illustrative example, we consider a computer-generated benchmark of a social network defined as follows. The underlying structure is a complete graph, where the (directed) social interactions are divided into g different, non-overlapping types, e.g., work relations, friends, etc. Fig. 4. A shows such a network with $N = 10$ nodes and $g = 3$ edge types. We assume that different types of relationships dominate at different times, as often observed in empirical data⁴². Therefore, at each time, all edges of one single group are simultaneously active. Initially, a randomly chosen group is active. At each step, the currently active group remains active with probability p_k ; with probability $1 - p_k$, a new active group is randomly selected among all groups (including the currently active one). For the parameters considered here, the walker thus has three outgoing edges available for transport at each step (see Fig. 4). It is important to note that for the generation of trajectories on this temporal network, a change of the probability for the walker to stay at its node p_s is effectively equivalent to a change in p_k . For the sake of simplicity, we therefore set $p_s = 0$ here.

Clearly, the trajectories of the random walker and the corresponding structure of the \mathcal{M}_2 network are affected by the value of p_k . When $p_k = 0$, a new active group is randomly selected at each step of the random walker. In this limit, the average dynamics is equivalent to a first-order random walk on a fully connected network, and thus no underlying structure can be found. Increasing p_k implies that the random walker remains for longer time periods inside one group of edges before leaving, and it is thus possible to uncover the group structure. The extreme case $p_k = 1$ is again trivial: the initial group remains active for all times. This case is therefore not considered in the following. The results of our analysis are summarised in Fig. 4, where we compare the uncovered communities with the planted solution into groups by means of normalised variation of information (VI). Our observations are twofold. First, for values of t slightly larger than $t = 1$, we find the planted partition with high accuracy. Second, one observes that the underlying communities can be successfully found when the value of p_k is sufficiently large, with a clear transition around $p_k = 0.4$ (see inset in Fig. 4).

A real-world case study: analyzing temporal interaction pattern of school children. To demonstrate the utility of our approach with a real-world example, we have considered an empirical dataset from the SocioPatterns project⁴³. The data, described in detail in⁴⁴, consists of time-dependent face-to-face contacts, captured by wireless wearable sensors, between children and teachers in a primary school. In total, the dataset is made of 77,602 contact events between 242 individuals during two consecutive days. As the children belong to 10 different classes, we expect to find strong structural communities associated to these classes in the dynamic contact network. In the following analysis, the pathways were generated with a probability for the walker to stay at his node at each step set to $p_s = 0.05$, but similar results were obtained when varying this parameter.

By scanning through Markov times we can find several persisting partitions. As validation we first identified the Markov time $t_{C=10} \approx 0.631$ around which the algorithm robustly detects 10 communities, and verified that the

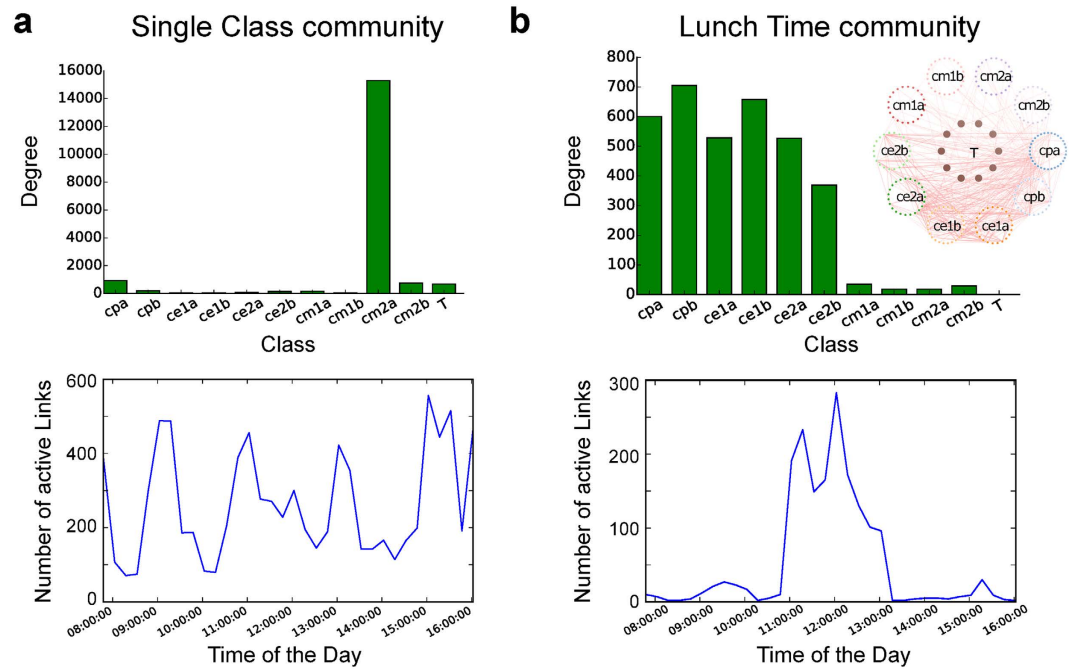


Figure 5. Analysis of the temporal community structure of a school network. Two types of communities can be found by analyzing the \mathcal{M}_2 network generated with $p_{stay} = 0.05$. Around Markov time equal to $t = 0.56$ we find a robust set of the directed edges into 15 communities. 10 of these communities are ‘single class communities’, in the sense that most connections are concentrated in one single class (cm2a in the example shown in (a)). These communities do not display a particular temporal structure in their activation patterns. This is clear from the temporal profile of activity (bottom panel in (a)), which displays the number of active links as a function of time, averaged over 2 days. (b) The remaining 5 communities are temporally localized ‘lunchtime communities’. While they extend over a range of class labels, connecting pupils from different groups, they are highly coherent and synchronised in time. These communities represent the lunch-time interactions between students of different classes.

communities overlap with the partition into classes of the students. One should note here that the partition into classes can in fact be uncovered even from the aggregated, weighted network associated to the data⁴⁴.

The advantage of our approach becomes apparent when looking at further stable partitions found for smaller Markov times. For instance, we find another robust split of the \mathcal{M}_2 network at Markov time $t = 0.56$ into 15 communities. For this partition most communities are still identified with classes, as can be seen in Fig. 5. However, there are also additional modules in which students from different classes are mixed. These modules display activity patterns which are highly localized in time and therefore are bound to get lost when averaging over the temporal dimension in the data. Interestingly, these latter communities correspond in fact to interactions taking place between students during lunch breaks. Notably, these interactions are not detectable from the aggregate network or when using a memoryless representation of the dynamics (e.g., the $\mathcal{M}_{1\text{expanded}}$ network). By using the approach outlined here, we can thus not only detect ‘structural’ communities, in which actors maintain a higher amount of interactions between each other throughout time, but also ‘temporally localized’ communities, which are associated to groups that interact with each other strongly over certain time-periods.

Second-Order Models, Non-Backtracking Walks and Spectral Clustering

Recently, there has been an increased interest in spectral methods based on the so-called non-backtracking matrix of a network. While standard spectral algorithms for graph partitioning tend to fail when applied to very sparse networks, Krzakala *et al.*¹⁸ demonstrated how the non-backtracking matrix can be used to design spectral algorithms which behave optimally right until the theoretical limit of detectability of a stochastic block model. As the name suggests, the non-backtracking matrix is intimately related to the non-backtracking random walk on a network. The non-backtracking random walk is a diffusion process in which a particle behaves just like a simple random walker on the network, albeit with one additional constraint: after arriving at a node the walker is not allowed to immediately return to the node from which she originated (she cannot ‘backtrack’).

Interestingly, this non-backtracking random walk can be simply phrased in terms of a memory dynamics. Following the notation of section the non-backtracking matrix is defined on the set of \mathcal{M}_2 -nodes and may be simply written as

$$B = G^{\text{tri}} + G^{\text{exp}}. \quad (12)$$

Clearly this matrix describes the line graph for a memory walker with parameters set to $r_2 = 0, r_3 = 1, r_{>3} = 1$ (see section). The associated transition matrix T^B of the non-backtracking random walk is simply obtained through normalisation:

$$T_{\alpha,\beta}^B = \frac{B_{\alpha\beta}}{k_{\alpha}^{\text{out}}}, \quad (13)$$

where k_{α}^{out} is the weighted out-degree of \mathcal{M}_2 -node α . As this is just a diagonally scaled version (by the degree) of the non-backtracking operator, we may employ the flow based transition matrix T^B for spectral partitioning of the nodes⁴⁵. In fact, T^B is just one particular instance of a whole continuum of possible transition matrices, each with varying amounts of memory. We can thus assess the effect the introduced memory has on the clustering by varying the parameters r_i and performing a spectral clustering analysis.

Like in Ref. 18, we consider the problem of spectral clustering here in the simplest possible setup, which is the detection of a bipartition of equally sized groups in a large, sparse network. The networks we consider here consist of $N = 10^4$ nodes, divided into two groups according to a simple stochastic block model¹⁸. We denote the average degree of each node by c , the average degree to nodes inside its group by c_{in} , and the average degree to nodes outside its own group by c_{out} . Note that these quantities are coupled by the simple relation $c = (c_{in} + c_{out})/2$.

Similar to Ref. 18, while keeping the ratio $c_{out}/c_{in} = 0.3$ fixed, we vary the average degree c and observe the results we obtain from different operators via spectral clustering. Now, each node is grouped using the following protocol^{18,45}. We first compute the second dominant (left) eigenvector of the respective matrix operator. Second, for each node we sum over the eigenvector components corresponding to all of its incoming links. The nodes are grouped according to the sign of this sum: if it is positive the node is assigned to the first group, if not to the second group. In case the respective matrix operator is directly defined on the node space, the nodes are simply partitioned according to the signs of the eigenvector. Note that in contrast to the problems considered above we here look at the problem of community detection from the perspective of a hard clustering of the nodes rather than the edges and moreover assume that there is a fixed, known number of non-overlapping node-communities. We thus do not expect that using second-order transition matrices will necessarily improve the performance (for the non-overlapping node-clustering), indeed accounting for memory can also be detrimental for this purpose as we will see in the following. However, our main purpose here is not to design an improved node clustering algorithm but to better understand the connection between spectral clustering and memory matrices.

In Fig. 6a, we initially perform a spectral clustering analysis the non-backtracking matrix B ¹⁸, the adjacency matrix A , the normalised Laplacian matrix $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$. As has been observed by Krzakala *et al.*, the clustering based on the backtracking matrix B performs clearly better than the one based on the adjacency or Laplacian matrix, in particular when the graph becomes very sparse (see Fig. 6a). Figure 6b shows the results for the spectral clustering based on the memory walk matrices $T_{\{r_i\}}$ with various parameter settings r_i . There appears to be a clear trend: the more weight there is assigned to exploratory steps, the easier it appears to decide on the node groupings. Spectral clustering based on the non-backtracking second-order transition matrix $T(r_2 = 0, r_3 = 1, r_{>3} = 1)$ performs almost as well as the clustering based on B for very sparse networks. For networks with a higher average degree ($c \approx 8$), the results are even slightly better. As the second eigenvector of T , which is used for the clustering, describes the approach of a memory dynamics towards stationarity it should not be too surprising that the results deteriorate if more memory is introduced: as very sparse graphs have a tendril like structure, additional memory can lead to a localisation of the dynamics in parts of the graph which correspond to strong 'local' bottlenecks. Stated differently, the slowest time-scale of the diffusion may not be correlated with moving from one planted (node) community to the other, but local obstacles become more important making the second eigenvector a bad predictor of the bi-partition.

This is in particular interesting, as for most memory dynamics reported⁶ the return flow appears to be significantly large, which would imply that the dynamical constraints on the flow are much more pronounced on a local level than one may expect from the perspective of an aggregated network. This observation appears to be aligned with the fact that many real-world systems tend to be composed of overlapping communities²¹.

Conclusions

Network-based models have been extremely successful for the analysis of complex systems, and have led to the discovery of structural features with profound impacts on its dynamics⁴⁶. However, to accurately capture the complexity of real-world interacting systems, simple network models are often not sufficient. For this reason, different approaches have been proposed recently, which increase the model dimension and enrich the network paradigm. One approach that has gained prominence in the literature, is to account for the different types of interactions present in the system by means of multiplex or multilayer networks⁴⁷. Our approach developed in the work presented here proceeds in a similar spirit: by providing a larger state space representation in the temporal dimension, we aim to obtain new insights about the dynamical processes taking place in the system. Despite their similarities (see Ref. 2), multiplex networks and higher-order Markov models differ, as the former model tends to emphasise differences in the connections of the system, while the latter emphasises pathways.

In this paper, we have focused on the applicability of second-order Markov dynamics in the context of flow-based community detection. Second-order Markov dynamics model real-world temporal dynamics more accurately, so the found communities are expected to better capture the actual flow constraints in the system. In particular, we have shown how the Markov stability framework, and thus techniques like Potts models and spectral clustering, can be based on a second-order Markov dynamics, which is equivalent to considering transitions between directed edges in the original network. As clustering of second order models yields a partition of the edges, one can readily interpret the results as an overlapping clustering of the nodes in the original system, which is a beneficial feature of the approach.

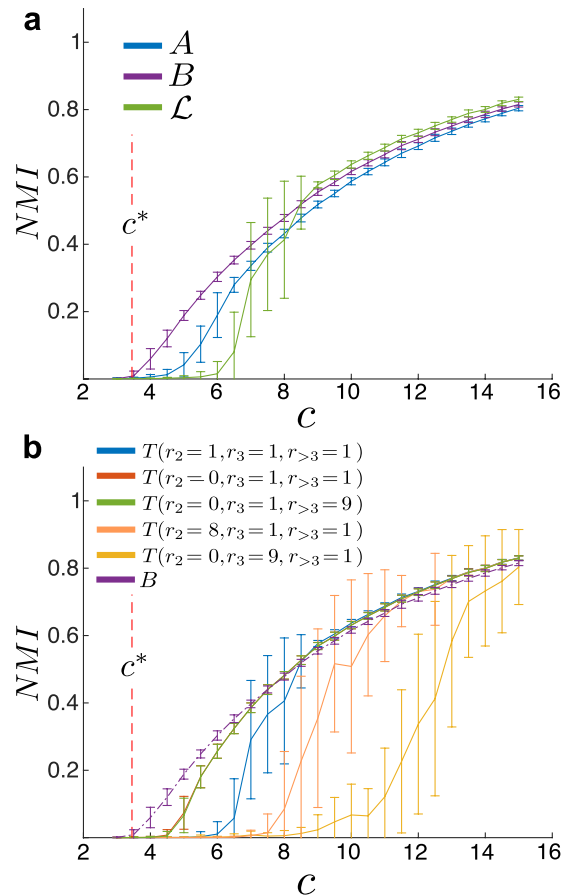


Figure 6. Spectral Clustering of sparse networks using memory matrices. The networks consist of 10^4 nodes with a planted bipartition. The average degree of the network vs. the spectral clustering performance for classifying the *nodes* as measured by the normalised mutual information (NMI). Note that the theoretical detectability limit is given by $c^* \approx 3.45$ (red dashed line). (a) Clustering performance based on the Adjacency matrix A , the non-backtracking matrix B , and the normalised Laplacian matrix \mathcal{L} . (b) Clustering performance based on different $\mathcal{M}_{2\text{model}}$ transition matrices with different parameter settings.

Some practical concerns are associated with moving to a second-order model. Naturally, there is an increased computational cost, as the number of nodes in the second-order model is equivalent to the number of directed edges in the original network. However, as most networks are sparse, this cost tends to be outweighed by the benefits gained from a higher-order dynamical representation. Another practical issue is that pathway statistics, needed to construct a second-order transition matrix directly from data, may not always be immediately accessible. To make the toolkit of second order models available in such scenarios, we have presented two different strategies. First, we have analysed a simple model able to generate realistic second-order dynamics, which can be calibrated from similar datasets. Second, we have demonstrated how to represent temporal network data as a second order network. We have tested these two strategies by focusing on a flight network in the United States, and a (temporal) social interaction network in a school environment, respectively. In both cases, the second order dynamics, even approximated, allowed us to extract temporal patterns in the data that would have been missed by an aggregated first-order model.

Finally, we have highlighted the relationship between second-order Markov models and the recently introduced spectral clustering formalism based on the non-backtracking matrix. Interestingly, the non-backtracking matrix corresponds to a scaled version of the transition matrix of a non-backtracking random walk, which is a special case of the second order dynamics discussed in this manuscript. As we have demonstrated, this connection opens up the possibility to investigate further second order Markov processes and their relationships to spectral clustering. Investigating these issues in more detail will be the subject of future work.

References

1. Newman, M. E. J. *Networks: An Introduction* (Oxford University Press, USA, 2010).
2. Lambiotte, R., Salnikov, V. & Rosvall, M. Effect of memory on the dynamics of random walks on networks. *J. Complex Netw.* **3**, 177–188 (2015).
3. Delvenne, J.-C., Yaliraki, S. N. & Barahona, M. Stability of graph communities across time scales. *Proc. Natl. Acad. Sci.* **107**, 12755–12760 (2010).
4. Schaub, M. T., Delvenne, J.-C., Yaliraki, S. N. & Barahona, M. Markov Dynamics as a Zooming Lens for Multiscale Community Detection: Non Clique-Like Communities and the Field-of-View Limit. *PLoS ONE* **7**, e32210 (2012).

5. Newman, M. E. Spectral methods for community detection and graph partitioning. *Phys. Rev. E* **88**, 042822 (2013).
6. Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D. & Lambiotte, R. Memory in network flows and its effects on spreading dynamics and community detection. *Nat. Commun.* **5** (2014).
7. Scholtes, I. *et al.* Causality-driven slow-down and speed-up of diffusion in non-Markovian temporal networks. *Nat. Commun.* **5** (2014).
8. Song, C., Qu, Z., Blumm, N. & Barabási, A.-L. Limits of Predictability in Human Mobility. *Science* **327**, 1018–1021 (2010).
9. Estrada, E. & Higham, D. Network Properties Revealed through Matrix Functions. *SIAM Rev.* **52**, 696–714 (2010).
10. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**, 1118–1123 (2008).
11. Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D., & Lambiotte, R. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, **5**, (2014).
12. Delvenne, J.-C., Schaub, M. T., Yaliraki, S. N. & Barahona, M. The Stability of a Graph Partition: A Dynamics-Based Framework for Community Detection. In Mukherjee, A., Choudhury, M., Peruani, F., Ganguly, N. & Mitra, B. (eds.) *Dynamics On and Of Complex Networks, Volume 2, Modeling and Simulation in Science, Engineering and Technology*, 221–242 (Springer, New York, 2013).
13. Lambiotte, R., Delvenne, J.-C. & Barahona, M. Random Walks, Markov Processes and the Multiscale Modular Organization of Complex Networks. *IEEE Trans. Netw. Sc. Eng.* **1**, 76–90 (2014).
14. Newman, M. E. J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**, 8577–8582 (2006).
15. Reichardt, J. & Bornholdt, S. Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Phys. Rev. Lett.* **93**, 218701 (2004).
16. Fiedler, M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Math. J.* **25**, 619–633 (1975).
17. Shi, J. & Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 888–905 (2000).
18. Krzakala, F. *et al.* Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci.* **110**, 20935–20940 (2013).
19. Evans, T. S. & Lambiotte, R. Line graphs, link partitions, and overlapping communities. *Phys. Rev. E* **80**, 016105 (2009).
20. Friggeri, A., Chelius, G. & Fleury, E. Triangles to Capture Social Cohesion. In Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), MIT, Boston, USA (2011).
21. Ahn, Y.-Y., Bagrow, J. P. & Lehmann, S. Link communities reveal multiscale complexity in networks. *Nature* **466**, 761–764 (2010).
22. Lambiotte, R. & Rosvall, M. Ranking and clustering of nodes in networks with smart teleportation. *Phys. Rev. E* **85**, 056107 (2012).
23. Delmotte, A., Tate, E. W., Yaliraki, S. N. & Barahona, M. Protein multi-scale organization through graph partitioning and robustness analysis: application to the myosin? myosin light chain interaction. *Phys. Biol.* **8**, 055010 (2011).
24. Blondel, V., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **2008**, P10008 (2008).
25. Chung, F. Laplacians and the Cheeger inequality for directed graphs. *Ann. Comb.* **9**, 1–19 (2005).
26. Satuluri, V. & Parthasarathy, S. Symmetrizations for clustering directed graphs. In Proceedings of the 14th International Conference on Extending Database Technology, Uppsala, Sweden. Publisher: ACM Digital Library (2011).
27. Fortunato, F. & Barthélémy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA* **104**, 36–41 (2006).
28. Good, B. H., de Montjoye, Y.-A. & Clauset, A. The performance of modularity maximization in practical contexts. *Phys. Rev. E* **81**, 046106 (2010).
29. Fortunato, F. Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
30. Schaub, M. T., Lambiotte, R. & Barahona, M. Encoding dynamics for multiscale community detection: Markov time sweeping for the map equation. *Phys. Rev. E* **86**, 026112 (2012).
31. Lambiotte, R. Multi-scale modularity in complex networks. In Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on, 546–553 (IEEE, 2010).
32. Holland, P., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Netw.* **5**, 109–137 (1983).
33. Karrer, B. & Newman, M. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011).
34. Peixoto, T. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **4**, 011047 (2014).
35. Peixoto, T. & Rosvall, M. Modeling sequences and temporal networks with dynamic community structures. *arXiv:1509.04740* (2015).
36. Meila, M. Comparing clusterings—an information based distance. *J. Multivariate Anal.* **98**, 873–895 (2007).
37. Holme, P. & Saramki, J. Temporal networks. *Phys. Rep.* **519**, 97–125 (2012).
38. Mucha, P., Richardson, T., Macon, K., Porter, M. & Onnela, J.-P. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. *Science* **328**, 876–878 (2010).
39. Delvenne, J.-C., Lambiotte, R. & Rocha, L. Diffusion on networked systems is a question of time or structure. *Nat. Commun.* **6** (2015).
40. Saramaki, J. & Holme, P. Exploring Temporal Networks with Greedy Walks. *arXiv:1508.00693* (2015).
41. Gueuning, M., Delvenne, J.-C. & Lambiotte, R. Imperfect diffusion on temporal networks. *arXiv:1508.04006* (2015).
42. Eagle, N., Pentland, A. S. & Lazer, D. Inferring friendship network structure by using mobile phone data. *Proc. Natl. Acad. Sci.* **106**, 15274–15278 (2009).
43. Gemmetto, V., Barrat, A. & Cattuto, C. Mitigation of infectious disease at school: targeted class closure vs school closure. *BMC Infect. Dis.* **14**, 695 (2014).
44. Stehlé, J. *et al.* High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS ONE* (2011).
45. Newman, M. Spectral community detection in sparse networks. *arXiv:1308.6494* (2013).
46. Newman, M., Barabasi, A.-L. & Watts, D. *The structure and dynamics of networks* (Princeton University Press, 2006).
47. Kivela, M. *et al.* Multilayer networks. *J. Complex Netw.* **2**, 203–271 (2014).

Acknowledgements

We acknowledge financial support from F.R.S-FNRS, ARC and from the COST Action TD1210 KnowEscape. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimisation), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

Author Contributions

All authors conceived the study; V.S. and M.T.S. performed the numerical simulations and created the figures; All authors wrote and reviewed the manuscript.

Additional Information

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Salnikov, V. *et al.* Using higher-order Markov models to reveal flow-based communities in networks. *Sci. Rep.* **6**, 23194; doi: 10.1038/srep23194 (2016).



This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>