Network Science **6** (4): 469–493, 2018.
doi:10.1017/nws.2018.19

Choosing the number of groups in a latent stochastic blockmodel for dynamic networks

RICCARDO RASTELLI
School of Mathematics and Statistics, University College Dublin, Dublin, Ireland
(*e-mail:* riccardo.rastelli@ucd.ie)

PIERRE LATOUCHE
Laboratoire MAP5, UMR CNRS 8145, Université Paris Descartes & Sorbonne Paris Cité, Paris, France
(*e-mail:* pierre.latouche@univ-paris1.fr)

NIAL FRIEL
School of Mathematics and Statistics and Insight: Centre for Data Analytics, University College Dublin, Dublin, Ireland
(*e-mail:* nial.friel@ucd.ie)

Abstract

Latent stochastic blockmodels are flexible statistical models that are widely used in social network analysis. In recent years, efforts have been made to extend these models to temporal dynamic networks, whereby the connections between nodes are observed at a number of different times. In this paper, we propose a new Bayesian framework to characterize the construction of connections. We rely on a Markovian property to describe the evolution of nodes' cluster memberships over time. We recast the problem of clustering the nodes of the network into a model-based context, showing that the integrated completed likelihood can be evaluated analytically for a number of likelihood models. Then, we propose a scalable greedy algorithm to maximize this quantity, thereby estimating both the optimal partition and the ideal number of groups in a single inferential framework. Finally, we propose applications of our methodology to both real and artificial datasets.

Keywords: *stochastic blockmodels, dynamic networks, greedy optimization, Bayesian inference, integrated completed likelihood*

1 Introduction

In the last few years, there has been an increasing amount of data stored characterizing interactions between individuals or, more generally, units of interest. An interaction, represented by a triple (i, j, η), indicates that units i and j have a connection at a specific time point η. Interactions can for instance describe email exchanges between individuals or posts on social media. In Biology, units can correspond to genes and interactions to regulation events between the genes. A natural approach to model the set of all observed interactions is to rely on a dynamic graph where each unit is associated with a node and where an edge (i, j) is present between two units i and j, at time η, if the interaction (i, j, η) is recorded.

A long series of methods have been recently proposed to cluster the nodes of dynamic networks in order to summarize the information hidden in such data. A

large number of these consider the stochastic blockmodel (SBM) (Wang & Wong, 1987; Nowicki & Snijders, 2001) as a starting point and propose extensions to the dynamic framework.

We recall that the SBM was originally introduced to cluster the nodes of static networks where nodes and edges do not evolve through time. The model assumes that nodes are spread in latent clusters, which, in practice, have to be inferred from the data. The probability for two nodes to connect is then only dependent on their respective clusters. Since no constraints are imposed on the connection probabilities, clusters may correspond to various types of profiles, making the methodologies based on SBM applicable to networks with disassortative structures (Daudin *et al.*, 2008).

An important strand of literature on the dynamic extensions of SBMs focuses on the modeling of networks evolving continuously in the time dimension. For example, Matias *et al.* (2018) and Corneli *et al.* (2017) consider the interactions between any two nodes as events of a non-homogeneous Poisson point process. Zhou *et al.* (2013), Tran *et al.* (2015), and Farajtabar *et al.* (2015) adopt instead Hawkes processes to capture the self-exciting nature of the interactions.

An alternative modeling approach considers instead networks which are discrete in time: this is achieved by partitioning the time dimension in intervals and aggregating interactions according to such grid. This leads to a number of network snapshots indexed by a discrete time variable t. In the case of a binary dynamic network, two nodes are connected in a snapshot if they have at least one interaction in the corresponding time interval.

The statistical models for discrete temporal networks may not have the flexibility of continuous-time models, however, their simpler structure often permits scalable inferential procedures capable of handling large datasets. A discrete dynamic SBM-like model is that of Yang *et al.* (2011), which allows, for example, each node to switch its cluster at time t depending on its state at time $t-1$. A transition matrix is employed to characterize the switching probabilities. While clusters can change over time, fixed connection probabilities are used. Conversely, the model of Xu & Hero (2014) relies on evolving connection probabilities, and temporal changes are described through a state space model. Therefore, the inference requires the use of optimization tools such as the Kalman filter and the Rauch–Tung–Striebel smoother. Xu (2015) extends this approach to include a direct dependence between any two contiguous network snapshots, and hence capture the persistence of edges over time. The work of Yang *et al.* (2011) has been extended by Matias & Miele (2017) for the clustering of nodes in dynamic networks where edges are not necessarily binary. We emphasize that theoretical results are also provided in this paper to show that dynamic SBM-like models should not let both the clusters and connection parameters evolve through time without incurring into identifiability issues. A similar type of temporal model is also proposed by Ishiguro *et al.* (2010), where the latent partitioning of the nodes is determined by a dynamic version of the Dirichlet Process, and model inference is performed through a slice sampler.

As regards the static framework, many extensions have been considered for the SBM model. Many of them have then been adapted to deal with dynamic networks. For example, the well-known mixed-membership SBM of Airoldi *et al.* (2008) has been extended to a dynamic framework in several works including Xing *et al.* (2010),

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

Ho *et al.* (2011), and Kim & Leskovec (2013). Note that the latent position model of Hoff *et al.* (2002), which is also popular in the network community, was also adapted by Sarkar & Moore (2005) and Friel *et al.* (2016) to deal with dynamic interactions.

In this paper, we consider a statistical model for discrete-time temporal networks: following the approaches of Yang *et al.* (2011) as well as Matias & Miele (2017), we model the evolution of the cluster memberships over time by relying on a Markovian property. We develop a Bayesian hierarchical structure which allows one to analytically integrate out most of the model parameters by exploiting the conjugacy of the prior distributions. This so-called collapsing (previously employed in a number of works including Nobile & Fearnside, 2007, McDaid *et al.*, 2013, and Côme & Latouche, 2015) allows one to obtain an analytical expression for the integrated completed data likelihood for a number of likelihood models. A scalable greedy optimization algorithm is then employed to perform inference. A crucial advantage of the approach proposed is that it allows the estimation of both the number of clusters and the cluster memberships of the nodes within the same algorithmic framework.

Section 2 introduces the model and the notation. The Bayesian hierarchical structure is then described in Section 3. Finally, the optimization procedure is given in Section 5 and experiments are carried out in Sections 6–8 to assess the proposed methodology.

2 Markovian stochastic blockmodel

The observed data consist of a sequence of network objects $\mathscr{X} = \left\{\mathbf{X}^{(t)}\right\}_{t\in\mathscr{T}}$ defined on the same set of nodes $\mathscr{V} = \{1,\ldots,N\}$. We focus our attention on temporal networks, where $\mathscr{T} = \{1,\ldots,T\}$ identifies a set of contiguous time intervals.

The random variable $X_{ij}^{(t)}$ models the value exhibited by the edge from i to j at time t, $\forall t \in \mathscr{T}$, and $\forall i,\ j \in \mathscr{V}$. We outline our methodology on directed networks, however it may also be applied to undirected ones, as we illustrate in the applications we consider. A typical scenario is that of a binary dynamic network, where

$$x_{ij}^{(t)} = \begin{cases} 1, & \text{if an edge from } i \text{ to } j \text{ appears at time } t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

In addition, self-edges are not allowed, i.e. $x_{ii}^{(t)} = 0,\ \forall t \in \mathscr{T}$ and $\forall i \in \mathscr{V}$.

The random variable $Z_i^{(t)}$ characterizes the allocation of a node, whereby $Z_i^{(t)} = g$ indicates that node i is allocated to group g at time t, for a certain $g \in \mathscr{K} = \{1,\ldots,K\}$. The set $\mathbf{Z}^{(t)} = \left\{Z_i^{(t)}\right\}_{i\in\mathscr{V}}$ therefore corresponds to a random hard clustering (i.e. a partitioning) of $\mathscr{V}$, $\forall t \in \mathscr{T}$. We also denote the full set of allocations with $\mathscr{Z} = \left\{\mathbf{Z}^{(t)}\right\}_{t\in\mathscr{T}}$. The total number of groups in this underlying structure is denoted by K, hence $Z_i^{(t)} \in \mathscr{K},\ \forall t \in \mathscr{T}$ and $\forall i \in \mathscr{V}$. Note that K represents the total number of groups in $\mathscr{Z}$, implying that, at each time frame t, the current number of non-empty groups $K^{(t)}$ may be any number in $\mathscr{K}$.

The allocations characterize the connection profiles of the nodes of the network, in that nodes allocated to the same group have their edges drawn from the same probability distribution. The edges of the dynamic network are marginally dependent,

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

however they are conditionally independent given the allocations:

$$p\left(\mathscr{X}|\mathscr{Z}\right) = \prod_{t\in\mathscr{T}} p\left(\mathbf{X}^{(t)}|\mathbf{Z}^{(t)}\right) \tag{2}$$

and $\forall t \in \mathscr{T}$:

$$p\left(\mathbf{X}^{(t)}|\mathbf{Z}^{(t)}\right) = \prod_{i\in\mathscr{V}} \prod_{j\in\mathscr{V}:j\neq i} p\left(X_{ij}^{(t)}|Z_i^{(t)}, Z_j^{(t)}\right) \tag{3}$$

The distribution for the value of a single edge is determined by the allocations of the nodes as follows:

$$p\left(X_{ij}^{(t)} = x|z_i^{(t)} = g, z_j^{(t)} = h, \boldsymbol{\Theta}\right) = f\left(x;\ \boldsymbol{\theta}_{gh}\right) \tag{4}$$

where $\boldsymbol{\Theta}$ and $\boldsymbol{\theta}_{gh}$ are collections of model parameters. In the binary network case, f corresponds to the mass probability function of a Bernoulli variable with success probability $\theta_{gh} \in [0, 1]$. In such case, the likelihood of the model may be written as

$$\mathscr{L}_{\mathscr{X}} = \prod_{g\in\mathscr{K}} \prod_{h\in\mathscr{K}} \prod_{t\in\mathscr{T}} \prod_{\left\{i\in\mathscr{V}:\ z_i^{(t)}=g\right\}} \prod_{\left\{j\in\mathscr{V}:\ j\neq i;\ z_j^{(t)}=h\right\}} \theta_{gh}^{x_{ij}^{(t)}} \left(1-\theta_{gh}\right)^{1-x_{ij}^{(t)}} \tag{5}$$

Concerning the modeling of the temporal evolution of the network, a Markovian property on the nodes' allocations is adopted. The sequences $\mathbf{Z}_i = \left\{Z_i^{(t)}\right\}_{t\in\mathscr{T}}$ are assumed to be independent Markov chains which are realized using the same $K \times K$ transition probability matrix Π. Therefore, the prior structure on the allocations factorizes as follows:

$$p\left(\mathscr{Z}|\Pi\right) = p\left(\mathbf{Z}^{(1)}\right) \prod_{t\in\mathscr{T}\setminus\{1\}} p\left(\mathbf{Z}^{(t)}|\mathbf{Z}^{(t-1)}, \Pi\right) \tag{6}$$

where $\forall t \in \mathscr{T} \setminus \{1\}$ and $\forall i \in \mathscr{V}$:

$$p\left(Z_i^{(t)} = h|Z_i^{(t-1)} = g, \Pi\right) = \pi_{gh} \in [0, 1] \tag{7}$$

Note that the rows of Π must sum to 1, whereas no constraint is imposed on the columns of the same matrix. Combining Equations (6) and (7), the prior on the allocations may be alternatively written as

$$p\left(\mathscr{Z}|\Pi\right) = p\left(\mathbf{Z}^{(1)}\right) \prod_{g\in\mathscr{K}} \prod_{h\in\mathscr{K}} \pi_{gh}^{R_{gh}} \tag{8}$$

where R_{gh} denotes the total number of switches from group g to h.

We note that a similar modeling framework has been recently proposed by Matias & Miele (2017). In fact, our formulation corresponds to a special case of their model, whereby identifiability of the parameters is guaranteed by fixing the connection probabilities over time. In the next section, we propose an extension of their model by adding a Bayesian hierarchical structure on the parameters.

3 Bayesian hierarchical structure

We propose a hierarchical structure to further model the parameters $\boldsymbol{\theta}$ and Π. In particular, we focus on the use of conjugate priors, since these allow one to integrate out (*collapse*) most of the model parameters.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

Table 1. A list of edge types that can be accounted for using the methodology proposed. For each case, the corresponding Bayesian hierarchical structure is shown.

Edge type	Likelihood	Conjugate prior
Binary	Bernoulli	Beta
Categorical	Multinomial	Dirichlet
Counts (positive integers)	Poisson	Gamma
Positive weights	Gamma	Gamma
Truncated counts or proportions	Binomial	Beta
Heavy tailed positive weights	Pareto	Gamma
Real numbers	Normal	Normal-Gamma
Real numbers with covariates	Multivariate normal	Normal-Gamma

Concerning the transition probabilities, we specify a Dirichlet distribution over the rows of Π:

$$\left(\pi_{g1},\ldots,\pi_{gK}\right) \sim Dir\left(\delta_{g1},\ldots,\delta_{gK}\right), \qquad \forall g \in \mathscr{K} \tag{9}$$

By analytically integrating out the parameters $\{\pi_{gh}\}_{g,h}$, the following compound distribution (marginal prior) for the allocations arises:

$$p\left(\mathscr{Z}|\boldsymbol{\delta}\right) = p\left(\mathbf{Z}^{(1)}\right) \prod_{g\in\mathscr{K}} \left\{ \frac{\prod_{h\in\mathscr{K}} \Gamma\left(\delta_{gh}+R_{gh}\right)}{\Gamma\left(\sum_{h\in\mathscr{K}} \delta_{gh}+\sum_{h\in\mathscr{K}} R_{gh}\right)} \frac{\Gamma\left(\sum_{h\in\mathscr{K}} \delta_{gh}\right)}{\prod_{h\in\mathscr{K}} \Gamma\left(\delta_{gh}\right)} \right\} \tag{10}$$

We assume a uninformative flat prior distribution for the hyperparameters $\{\delta_{gh}\}_{g,h}$ and thus fix all of them to 1.

A separate and independent Multinomial-Dirichlet model may be specified for $\mathbf{Z}^{(1)}$. However, for these starting allocations, we opt for a more pragmatic and parsimonious approach: we use a Multinomial distribution with parameters $\{\alpha_g\}_{g\in\mathscr{K}}$, where

$$\alpha_g \propto \sum_{t\in\mathscr{T}\setminus\{1\}} \sum_{i\in\mathscr{V}} \mathbb{1}_{\left\{Z_i^{(t)}=g\right\}}, \qquad \forall g \in \mathscr{K} \tag{11}$$

The distribution on the initial states so defined approximates the stationary distribution associated to Π.

Regarding the likelihood structure, a number of conjugate models can be considered, encompassing most types of edges. Table 1 provides a list of possible scenarios that may be of interest.

In this paper, we focus on binary edges, and hence specify an independent Beta prior on the likelihood parameters $\Theta = \{\theta_{gh} : g \in \mathscr{K}, h \in \mathscr{K}\}$. In such a case, the compound distribution for $\mathscr{X}$ (marginal likelihood) is

$$p\left(\mathscr{X}|\mathscr{Z},a,b\right) = \prod_{g\in\mathscr{K}} \prod_{h\in\mathscr{K}} \left\{ \frac{\Gamma\left(a+b\right)}{\Gamma\left(a\right)\Gamma\left(b\right)} \frac{\Gamma\left(a+\eta_{gh}\right)\Gamma\left(b+N_{gh}-\eta_{gh}\right)}{\Gamma\left(a+b+N_{gh}\right)} \right\} \tag{12}$$

where a and b are the Beta hyperparameters, η_{gh} counts the number of edges from group g to h:

$$\eta_{gh} = \sum_{t\in\mathscr{T}} \sum_{\left\{i\in\mathscr{V}:\ z_i^{(t)}=g\right\}} \sum_{\left\{j\in\mathscr{V}:\ j\neq i;\ z_j^{(t)}=h\right\}} X_{ij}^{(t)} \tag{13}$$

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

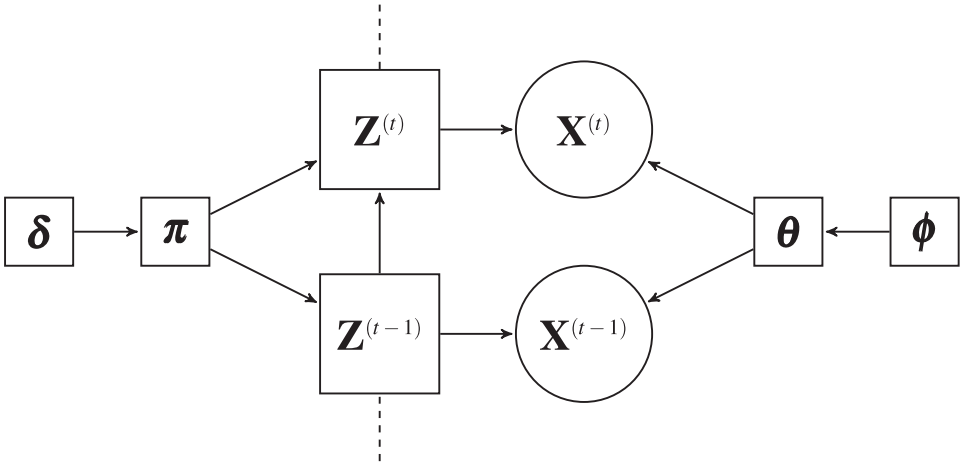

Fig. 1. Graphical model for the Markovian stochastic blockmodel described.

and N_{gh} counts the total number of possible edges from group g to h:

$$N_{gh} = \sum_{t \in \mathscr{T}} \sum_{\left\{i \in \mathscr{V}:\ z_i^{(t)} = g\right\}} \sum_{\left\{j \in \mathscr{V}:\ j \neq i;\ z_j^{(t)} = h\right\}} 1 \tag{14}$$

Note that, similarly to the hyperparameter δ, we make a "symmetric" assumption on the hyperparameters a and b, making them identical over all of the groups. In the applications, these are both fixed to 1 yielding a uninformative flat prior distribution.

Figure 1 shows a graphical model summarizing the hierarchical structure and the dependencies between the variables introduced.

We remark that Yang *et al.* (2011) also focus on the model illustrated so far, additionally proposing a methodology to obtain parameter estimates and predict allocations efficiently. By contrast, in the following sections, we focus our attention on the model choice problem, whereby our goal is to simultaneously learn from the data the optimal allocations and the optimal number of groups. In particular, we emphasize the computational advantages of our proposed method, which is then also compared to other relevant strategies.

4 Exact integrated completed likelihood

In the previous section, we have shown that in a binary Markovian dynamic network, the marginal likelihood $p\left(\mathscr{X}|\mathscr{Z}, a, b\right)$ and the marginal prior $p\left(\mathscr{Z}|\delta\right)$ have an exact analytical form. These terms can be recombined to obtain a particularly meaningful quantity, the so-called exact integrated completed likelihood (ICL), defined as

$$\mathscr{ICL}_{ex} = \log\left[p\left(\mathscr{X}|\mathscr{Z}, a, b\right)\right] + \log\left[p\left(\mathscr{Z}|\delta\right)\right] \tag{15}$$

Such a quantity corresponds to the exact value that the ICL criterion of Biernacki *et al.* (2000) propose to maximize when choosing the number of groups in a finite mixture context. We stress that, although the definition in Equation (15) relates to the specific case of binary dynamic networks, the same quantity can be evaluated analytically for all of the likelihood models listed in Table 1.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

We propose to use $\mathscr{ICL}_{ex}$ as a model-based optimality criterion for the clustering problem on the nodes of our dynamic network: in the space of all possible allocations, we seek a $\hat{\mathscr{Z}}$ maximizing $\mathscr{ICL}_{ex}$. Note that, thanks to the discrete nature of the allocation variables, the optimal total number of groups $\hat{K}$ can be deduced automatically from $\hat{\mathscr{Z}}$, along with the values $\left\{K^{(t)}\right\}_{t\in\mathscr{T}}$. In addition, for any given clustering configuration $\mathscr{Z}$, the following plug-in estimators are available for the *collapsed* model parameters:

$$\hat{\theta}_{gh} = \frac{\eta_{gh}}{N_{gh}}; \qquad \hat{\pi}_{gh} = \frac{R_{gh}}{\sum_{h\in\mathscr{K}} R_{gh}} \tag{16}$$

for all g and h in $\mathscr{K}$.

We also emphasize that, from a Bayesian perspective, $\hat{\mathscr{Z}}$ corresponds to a Maximum A Posteriori (MAP) solution:

$$p\left(\mathscr{X}|\mathscr{Z},a,b\right)p\left(\mathscr{Z}|\delta\right) = p\left(\mathscr{X},\mathscr{Z}|a,b,\delta\right) \propto p\left(\mathscr{Z}|\mathscr{X},a,b,\delta\right) \tag{17}$$

where the proportionality is intended with respect to $\mathscr{Z}$, which is the only non-fixed quantity.

5 Greedy optimization

In the Markovian dynamic SBM described, we seek a clustering solution $\hat{\mathscr{Z}}$ maximizing the $\mathscr{ICL}_{ex}$ value defined in Equation (15). A complete exploration of all possible allocations is not feasible, even for small datasets. However, similar combinatorial problems have been recently tackled successfully using heuristic greedy routines, resembling the well-known Iterated Conditional Modes of Besag (1986). Greedy algorithms have been applied in a hierarchical clustering framework for networks by Newman (2004). More recently, they have been adapted to maximize the $\mathscr{ICL}_{ex}$ in a model-based clustering context for networks in Côme & Latouche (2015), Wyse *et al.* (2017), and Corneli *et al.* (2016), and for Gaussian finite mixtures in Bertoletti *et al.* (2015). Here, we propose an extension of the same ideas to our dynamic network context.

The algorithm repeatedly sweeps over the network's nodes and attempts to reallocate them using a greedy behavior. In each step, a random time frame t and a random node i are chosen, and the variation of $\mathscr{ICL}_{ex}$ is evaluated for all the possible reallocations of the corresponding node. For every group g, the value $\ell_{(t,i)\to g}$ is evaluated, corresponding to the $\mathscr{ICL}_{ex}$ value after moving the node identified by (t,i) to g. Eventually, the node is reallocated to the group that yields the best increase in the objective function. This procedure continues until no reallocation can provide a further increase. The corresponding pseudocode is shown in Algorithm 1.

As input, the algorithm requires a starting partition, the hyperparameters' values, and a parameter K_{up} denoting the largest admissible number of groups. During the optimization, groups may be deleted (if they remain empty at all time frames) and created (if a node is reallocated to an empty group). In practice, the former case is very frequent, whereas the latter is rather rare. Hence, the best performance is achieved when the starting partition is composed of K_{up} groups, with K_{up} set to a fairly large value.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

Algorithm 1 `GreedyIcl`

Initialize $Z_i^{(t)}$, $\forall t \in \mathscr{T}$, $\forall i \in \mathscr{V}$.
Evaluate $\mathscr{ICL}_{ex}$ and set $\ell = \mathscr{ICL}_{ex}$ and $\ell_{stop} = \mathscr{ICL}_{ex}$.
Set $stop = false$.
while $!stop$ **do**
  Set $\mathscr{U} = \{(t,i) : t \in \mathscr{T},\ i \in \mathscr{V}\}$.
  Shuffle the elements of $\mathscr{U}$.
  while $\mathscr{U}$ is not empty **do**
    $(t,i) = pop(\mathscr{U})$.
    $\hat{g} = \arg\max_{g=1,2,\ldots,K_{up}} \ell_{(t,i)\to g}$.
    $\ell = \ell_{(t,i)\to \hat{g}}$.
    $Z_i^{(t)} = \hat{g}$.
  end while
  if $\ell \leqslant \ell_{stop}$ **then** $stop = true$ **else** $\ell_{stop} = \ell$.
  end if
end while
Return $\mathscr{Z}$ and ℓ.

Due to its strictly greedy behavior, the algorithm is bound to find only a local optimum for the objective function, and its initialization plays a crucial role. Due to the random update order, it is possible that each run returns a different solution, however we find that good initializations often lead to the same unique maximum.

As concerns the complexity of the algorithm, in the binary case, one iteration can be performed in $\mathscr{O}\left(M + TNK_{up}^2\right)$, where M is the total number of edges appearing in the network. Once a node (t,i) is selected, the number of edges to and from group g are counted, for every $g \in \{1,\ldots,K_{up}\}$. Evidently, this implies a cost of $\mathscr{O}(m)$, where m is the average value for the sum of in-degree and out-degree of a random node. Then, the quantities $\ell_{(t,i)\to g}$ are evaluated for every g. For each of these, the computational bottleneck is given by the calculation of the variation of the marginal likelihood value, which can be performed in $\mathscr{O}\left(K_{up}\right)$. This makes the average cost of updating an allocation $\mathscr{O}\left(m + K_{up}^2\right)$. Since all of the allocations are repeatedly updated, the overall complexity of one iteration is $\mathscr{O}\left(TNm + TNK_{up}^2\right)$ as previously claimed.

5.1 Initialization

We consider several types of initialization methods as follows:

- `aggregated`: an adjacency matrix of size $N \times N$ is obtained by aggregating (summing) the T adjacency matrices of the generated network. Then, the kmeans algorithm is run on such a matrix using as number of centres a random draw from the discrete interval $\lfloor 0.5 * N \rfloor, \ldots, \lfloor 0.75 * N \rfloor$. This associates a cluster membership to each of the N nodes. These allocations are assumed

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

not to change over time, so that the initial partition is given by the output of the kmeans repeated over all the time frames.

- `colbind`: the only difference with the `aggregated` initialization lies in the fact that kmeans is run on a matrix obtained by gathering the adjacency matrices one next to the other, obtaining a $N \times TN$ matrix. Note that this is the same initialization used for the `dynsbm` algorithm of Matias & Miele (2017).
- `rowbind`: in this case, kmeans is instead run on a matrix obtained by stacking up the observed adjacency matrices. Since the size of this matrix is $TN \times N$, the number of groups considered by kmeans is not capped at N as in the other cases: we use a draw from the discrete interval $\lfloor 0.5 * T * N \rfloor, \ldots, \lfloor 0.75 * T * N \rfloor$ as number of centers. Also, in contrast with the previous cases, kmeans returns the allocations for every node at every time frame, hence allocations are not assumed to be unchanged over time.
- `random`: the `GreedyIcl` is initialized using a random partition, with K_{up} chosen as in `GreedyIcl rowbind`.

Once initialized, we typically run the `GreedyIcl` algorithm once for each type of initialization. The additional label

- `all`: is used to indicate the best solution obtained through `GreedyIcl` over all possible types of initializations.

5.2 Final merge step

Once the `GreedyIcl` has converged, we additionally propose a hierarchical clustering on the optimal solution, following the approach of Côme & Latouche (2015). We consider all possible pairs of groups and attempt to merge each pair into a single cluster. Such a move is accepted only if the $\mathscr{ICL}_{ex}$ value increases, and every time a merge move is performed the procedure is restarted. For each try, the computational bottleneck is given by the evaluation of the variation for the marginal prior, which can be performed in $\mathcal{O}(\hat{K}^2)$. Since the number of pairs is $\mathcal{O}(\hat{K}^2)$, and the number of accepted merge moves is capped at $\hat{K}$, the overall complexity of this final step is $\mathcal{O}(\hat{K}^5)$. Here, $\hat{K}$ denotes the number of groups for the optimal solution obtained through the `GreedyIcl`, which is normally much smaller than K_{up}. We find that in practice this final merge step does not impact the overall computational time by much, and the additional computational burden may be neglected.

6 Simulation study

We propose an experiment to validate our methodology and compare it to the existing algorithm of Matias & Miele (2017), using artificial data. The number of time frames, the number of nodes, and the true underlying number of groups are fixed throughout as follows: $T = 4$, $N = 50$, and $K = 4$, respectively. As concerns the transition probability matrices, the following general structure is assumed:

$$\Pi = \begin{pmatrix} \pi & v & v & v \\ v & \pi & v & v \\ v & v & \pi & v \\ v & v & v & \pi \end{pmatrix} \tag{18}$$

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

where $v = (1 - \pi)/(K - 1)$, so that rows sum to one. Choosing a high π value, the clusters tend to be very stable and allocations do not change often over time. By contrast, a small π value represents the opposite scenario, corresponding to a highly instable system. In either case, the stationary distribution, which is used to generate the starting partition, is $\boldsymbol{\alpha} = \left\{\frac{1}{K}, \frac{1}{K}, \frac{1}{K}, \frac{1}{K}\right\}$. In our simulations, we consider two different scenarios: $\pi = 0.7$ and $\pi = 0.9$, which are denoted *low-stability* and *high-stability*, respectively. Although these two cases may offer only a limited view of all the situations encompassed by the model, we believe these to be the most interesting examples and that most realized networks are in fact well captured by this representation.

As concerns the connection probability matrices, an affiliation structure is assumed:

$$\begin{pmatrix} \theta_0 & \epsilon_0 & \epsilon_0 & \epsilon_0 \\ \epsilon_0 & \theta_0 & \epsilon_0 & \epsilon_0 \\ \epsilon_0 & \epsilon_0 & \theta_0 & \epsilon_0 \\ \epsilon_0 & \epsilon_0 & \epsilon_0 & \theta_0 \end{pmatrix} \tag{19}$$

A small perturbation is added independently to each of the entries of such matrix:

$$\theta = \theta_0 + 0.1u \qquad\qquad \epsilon = \epsilon_0 + 0.1u \tag{20}$$

where u is drawn from a Uniform distribution in the interval $[-1, 1]$ for each entry. While ϵ_0 is always fixed to 0.1, θ_0 is different for each matrix and determines the difficulty level in recovering the underlying latent clustering. We consider nine different scenarios, each corresponding to a different choice of θ_0, ranging from 0.1 to 0.9. The perturbation created by the random variable u is necessary to avoid an affiliation structure and ensure identifiability of the model, as explained in Matias & Miele (2017).

The experiment is executed as follows: for every choice of π and θ_0, 100 random dynamic undirected networks are generated, each corresponding to a different random realization of the connection probability matrix. In each network, the variational algorithm of Matias & Miele (2017), implemented in the `R` package `dynsbm` (version `0.3`), and our implementation of the greedy algorithm are executed. As concerns the variational approach, `dynsbm` is run four times for every value of K in the set $\{1, 2, \ldots, 6\}$, and the best run is retained as optimal solution according to the approximate ICL criterion advocated in Matias & Miele (2017). Higher values of K make `dynsbm` particularly time consuming and hence are not considered, however, we note that in all of the runs the approximate ICL criterion never favors the model with six groups. We also run our `GreedyIcl` once for each type of initialization on each dataset.

The optimal clusterings obtained using each method are compared to the true allocations using the *Normalized Mutual Information* (NMI) index (Strehl & Ghosh, 2003). This index takes values in the interval $[0, 1]$ and describes how similar two partitions are. These partitions must be vectors, hence the optimal clusterings obtained are vectorized by concatenating the partitions timewise.

Figure 2 shows the results obtained regarding the clustering performance. It appears that the greedy optimization coupled with kmeans initializations and `dynsbm`

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

Table 2. **Simulation study**. Average computing time required by each of the algorithms to be run on a generated dynamic network. Note that the variational expectation-maximization of `dynsbm` and the greedy updates of our algorithm are both run exactly four times for each dataset. The impact of the initialization step may vary across methods, but we find that this does not impact the computing time by much.

Algorithm	Average computing time (seconds)
`dynsbm`	35.46
`GreedyIcl all`	4.58
`GreedyIcl aggregated`	1.18
`GreedyIcl colbind`	1.19
`GreedyIcl rowbind`	1.14
`GreedyIcl random`	1.07

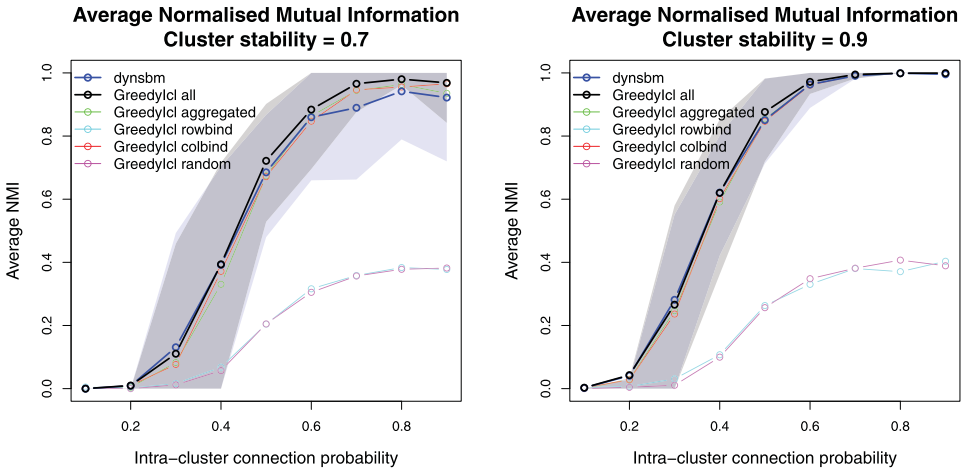

Fig. 2. (Color online) **Simulation study**. Average Normalized Mutual Information index between the true clustering and the optimal clustering solutions obtained through `dynsbm` (version `0.3`) and several versions of `GreedyIcl`. Shaded regions represent the 90% quantile region associated to each scenario. These regions are plotted only for `GreedyIcl all` (black color) and `dynsbm` (blue color). It appears that the greedy algorithm with a good initialization performs as well as the existing algorithm `dynsbm`.

achieve the best results, whereas the greedy optimization with random intializations perform poorly. The average computing times are provided in Table 2.

As concerns model selection, our methodology outperforms the existing algorithm of Matias & Miele (2017), as shown in Figure 3.

7 Enron dataset

The data. The Enron Corporation filed for bankruptcy in late 2001, leading to an unprecedented scandal and to dire consequences for the US stock market. The data we use in this paper consist of all the emails exchanged from January 2000 to March 2002 between the Enron members. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

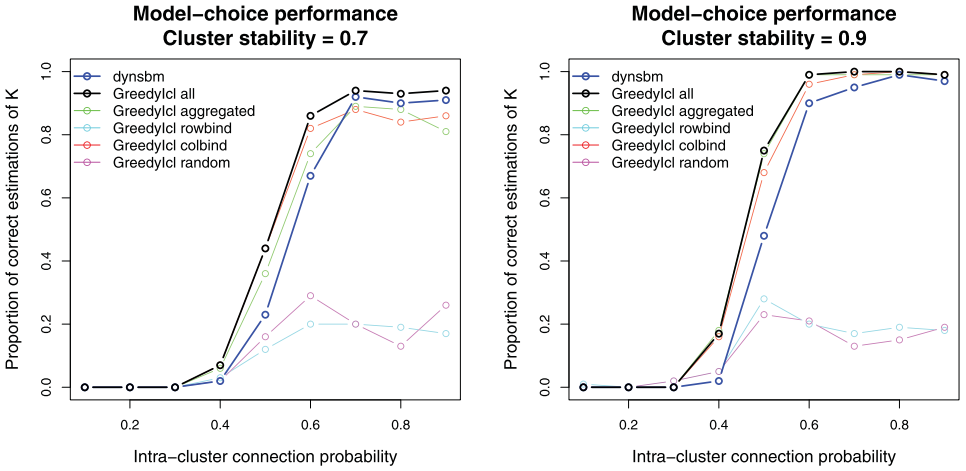

Fig. 3. (Color online) **Simulation study**. For each combination of π and θ_0, the proportion of networks where K is properly estimated is shown, for all of the algorithms considered.

Statistical network analyses for this data have been proposed in a number of papers, including Xing *et al.* (2010), Ishiguro *et al.* (2010), Ho *et al.* (2011), Xu & Hero (2014), and Matias *et al.* (2018). Similar to some of these works, we construct a binary directed dynamic network of emails transforming the data into an adjacency cube $\mathscr{X}$, such that

$$x_{ij}^{(t)} = \begin{cases} 1, & \text{at least an email is sent from member } i \text{ to member } j \text{ at time frame } t \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

with each time frame t corresponding to a one month period. The number of nodes is $N = 148$ and the number of time frames is $T = 27$. The number of edges observed at each time is shown in Figure 4. Along with the email data, the status of each member within the company is known, and is one of the following: *CEO*, *Director*, *Employee*, *In House Lawyer*, *Manager*, *Managing Director*, *N/A*, *President*, *Trader*, and *Vice President*. To simplify the exposition of the results, we gather every status other than *Employee* and *N/A* into a single class named *Manager*, so that only three status classes are considered.

We run our `GreedyIcl all` algorithm as in the simulation study, with each single run of the algorithm taking on average about 400 seconds. The overall best solution has 17 groups. An analysis of this clustering solution follows.

Activity levels and connectivity. As shown in the left panel of Figure 5, the clustering solution has very high stability over time, in that nodes do not change allocations frequently. The connection probability matrix, shown on the right panel, exhibits instead a much more complex situation. Evidently, a strong community structure is present, since most of the diagonal elements of this matrix are fairly large.

Similarly to Xing *et al.* (2010), we propose a brief characterization of the role played by each group, using the information provided by this connection probability matrix, in conjunction with Tables 3–5.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

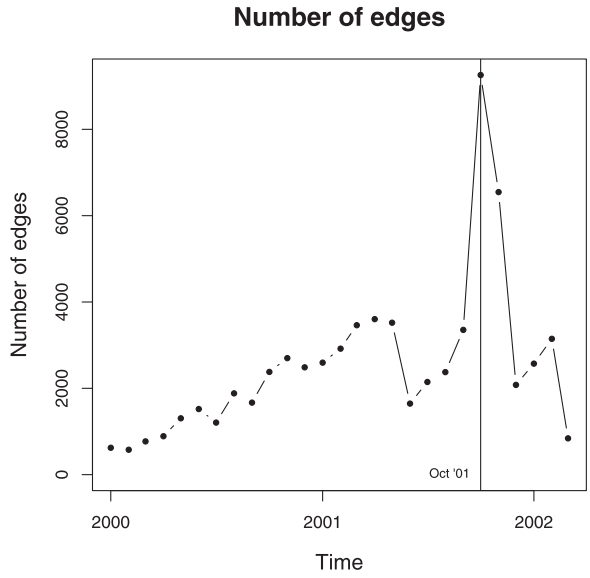

Fig. 4. **Enron dataset**. Number of edges at each time frame for the Enron email dataset. The peak in October 2001 corresponds to the disclosure of bankruptcy.

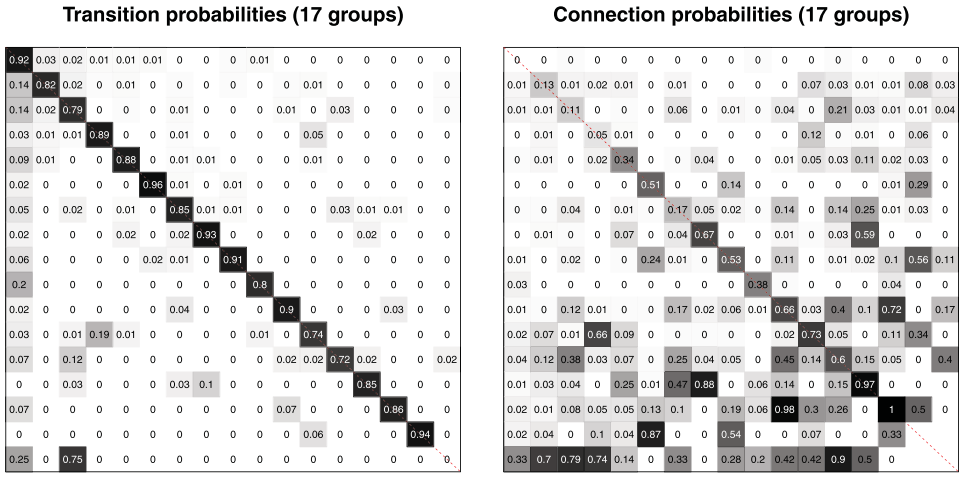

Fig. 5. (Color online) **Enron dataset**. The plots show the estimated transition probabilities (left panel) and connection probabilities (right panel) evaluated through the formulas in 16. It appears that the network is very stable over time, and nodes tend not to change their allocations often. Most groups have high within cluster connectivity, although a number of off-diagonal entries are also relevant. In both plots, the groups are ordered according to their aggregated size (i.e. the sum of their sizes over all times), in descending order. The entries left blank are those that cannot be properly estimated due to a small cluster size.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

Table 3. **Enron dataset**. Rounded average degrees for the Enron dataset, for each of the 17 clusters found.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Avg. out-degree	0	3	3	1	3	4	3	5	5	3	7	10	13	12	11	11	57
Avg. in-degree	1	3	3	3	4	5	4	5	4	2	6	5	7	7	5	7	2

Table 4. **Enron dataset**. Cluster sizes separated by status for the Enron dataset. For this table, the same node at two different time frames is considered as two separate entities, hence the sum of all entries is TN.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Employee	576	90	36	45	92	50	61	22	7	46	1	18	0	17	18	0	1
Manager	1,027	231	191	102	34	83	53	16	89	40	87	45	41	6	5	0	2
N/A	475	57	27	50	59	40	12	84	2	9	4	5	2	16	5	16	1

- **Group 1**: This group contains inactive nodes only. These members do not send emails, but they may receive a few newsletters.
- **Groups 2 to 8**: These contain a massive portion of the nodes, and hence describe the most common connectivity profiles that may arise.
  The nodes in these groups are not particularly active (the connection probabilities are typically not large) and tend to receive more emails than they send. The sizes of these groups swell increasingly by acquiring nodes from group 1 until the bankruptcy, signaling increased activity in the months before the default.
- **Groups 9, 11, 12, 13**: These groups are mainly composed of managers, and correspond to active nodes, who send more than they receive. These groups represent the different profiles of the executives of the company. Also, note that these groups have a high within cluster connectivity, hence it is reasonable to find the corresponding conversations particularly meaningful in terms of intelligence and company directives.
- **Groups 10, 14, 15**: three groups of very active nodes, although these are mainly composed of employees. Note that groups 10 and 15 only become relevant during the year 2001.
- **Group 16**: this group actually only ever contains one person (with a non-specified role). This person apparently has some special position and hence may be considered as an outlier.
- **Group 17**: the smallest of groups, containing 4 unique members, each for 1 time-frame only. Since this group has high connection probabilities towards many groups, it is reasonable to believe that nodes join this group whenever they send out newsletters to all members.

Although the number of groups considered is clearly different, our characterization of the different profiles appears to have similarities with that of Xing *et al.* (2010).

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

Table 5. **Enron dataset**. Group sizes at each time step for the Enron dataset.

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2000-01-01	115	8	3	2	1	2	5	4	2	0	2	0	1	2	0	1	0
2000-02-01	114	5	6	2	1	2	5	4	2	0	3	1	0	2	0	1	0
2000-03-01	112	5	7	3	2	2	4	4	2	0	3	1	0	2	0	1	0
2000-04-01	107	5	8	4	4	3	4	3	2	0	3	1	0	3	0	1	0
2000-05-01	103	7	6	6	5	3	4	2	2	0	3	2	1	3	0	1	0
2000-06-01	96	10	7	6	6	3	5	3	2	1	3	2	1	2	0	1	0
2000-07-01	93	10	6	5	5	4	5	3	3	1	5	3	1	2	1	1	0
2000-08-01	87	14	4	5	6	5	4	4	3	2	6	3	2	2	0	1	0
2000-09-01	86	13	5	6	7	4	4	5	2	2	5	3	2	2	1	1	0
2000-10-01	79	16	7	5	8	5	5	5	3	0	5	4	2	2	1	1	0
2000-11-01	71	15	7	5	8	5	5	5	6	6	4	4	2	2	2	1	0
2000-12-01	71	14	6	7	9	5	6	5	6	5	5	3	1	2	2	1	0
2001-01-01	72	14	8	7	9	4	7	5	3	4	3	4	2	2	3	1	0
2001-02-01	68	17	7	8	10	5	7	5	3	4	3	3	2	2	3	1	0
2001-03-01	69	15	7	9	11	5	5	6	3	4	3	3	3	1	3	1	0
2001-04-01	61	18	10	8	10	6	4	6	4	6	3	4	4	1	2	1	0
2001-05-01	52	21	15	8	11	7	4	7	3	6	3	4	4	0	2	0	1
2001-06-01	58	17	19	10	9	7	4	7	3	6	4	1	1	1	1	0	0
2001-07-01	67	13	13	9	6	8	4	5	5	7	4	3	2	1	1	0	0
2001-08-01	58	15	13	10	9	9	4	5	5	7	4	3	3	1	1	0	1
2001-09-01	50	19	16	10	10	11	5	5	5	7	4	4	1	0	1	0	0
2001-10-01	42	25	16	11	10	11	5	5	5	7	4	4	1	0	1	0	1
2001-11-01	41	26	17	11	10	11	5	5	6	6	3	4	1	1	1	0	0
2001-12-01	56	20	17	12	6	11	4	4	6	4	3	2	1	1	1	0	0
2002-01-01	64	20	14	10	3	11	3	5	5	5	2	1	3	1	1	0	0
2002-02-01	78	13	9	9	5	11	5	2	5	5	1	1	2	1	0	0	1
2002-03-01	108	3	1	9	4	13	4	3	2	0	1	0	0	0	0	0	0

We note that the behavior described by group 17 was also captured and pointed out in Ishiguro *et al.* (2010).

Temporal dynamics. Our model allows for groups to become *inactive*, whenever their sizes decrease to zero. This means that the connection profile associated to a particular group is not exhibited by any of the nodes in the time frame considered. Hence, the number of non-empty groups can be used as a measure of heterogeneity in the network, and the temporal dynamics of this quantity can be used to assess how heterogeneity evolves over time. In the left panel of Figure 6, the evolution of the number of non-empty groups is shown. It appears that the network is particularly heterogeneous in the year before the collapse ($K^{(t)} = 15$ or 16), whereas it eventually becomes rather homogeneous ($K^{(t)} = 10$) after the default. A similar message is conveyed by the plot on the right panel of Figure 6. Here, the number of inactive nodes (i.e. nodes allocated to group 1) is shown to decrease convincingly for both employees and managers. Eventually, after the collapse, the first group is repopulated as members quit their jobs and activities.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

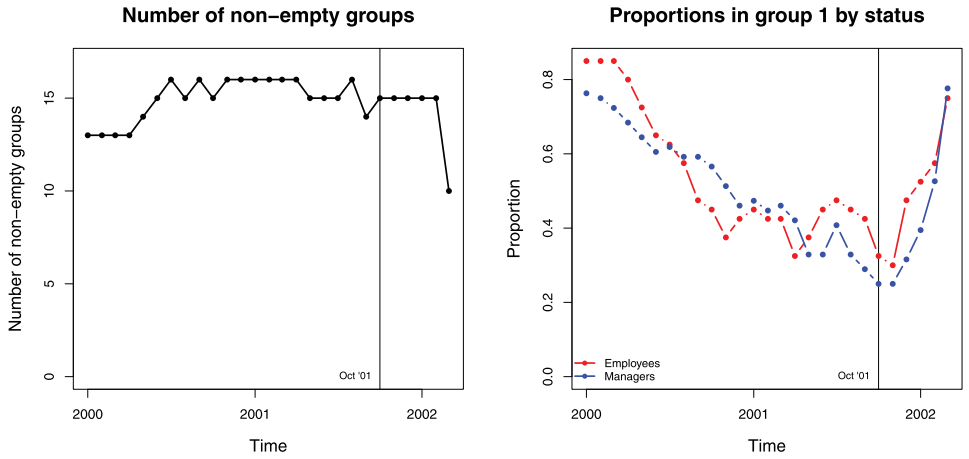

Fig. 6. (Color online) **Enron dataset**. On the left panel, the evolution of the number of non-empty groups is shown. This quantity appears to increase until the collapse, signaling an increase in heterogeneity in the network. On the right panel, the proportions of managers and employees allocated to the first group are shown. These proportions correspond to the nodes that are not particularly active.

8 London bike sharing

Data obtained from bike sharing systems is particularly suited to perform network analyses. Statistical analyses of the flows of bikes may provide important information regarding the management of the system and could help increase the efficiency of the service. Cycle hires data can be easily rearranged and visualized as a dynamic network structure, where edges correspond to hires and nodes to stations. Similar approaches have been recently proposed in a number of works, including Guigourès *et al.* (2015), Randriamanamihaga *et al.* (2014), Matias *et al.* (2018), and Corneli *et al.* (2017). Here, in the same fashion, we propose an application of our methodology to a dataset of cycle hires in London.

The data. The cycle hire data for the London bike sharing system is publicly available from Transport for London (2016). We use here the data from Wednesday, 5 June 2013 to Wednesday, 12 June 2013 (included). The discrete time frames correspond to blocks of 3 hours. For each time frame, we create a network adjacency matrix by transmuting bike trips into directed edges: an edge from station i to station j appears at time frame t if at least one bike is hired at station i during the corresponding 3 hours, and the same bike is then returned to station j. Bikes may be returned to the same station where they were hired: we consider this information not important for our analysis and hence discard all of the self-edges. The dynamic network so obtained is made of $N = 566$ nodes and $T = 64$ time frames. The data exhibits a very strong temporal heterogeneity, mainly due to the day–night cycle and the presence of the weekend days. In Figure 7, the observed number of edges is shown at every time frame. Weekdays typically exhibit two peaks in the activity level, corresponding to the start and the end of office hours. This suggests that commuters are the main users during working days. By contrast, low activity is

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

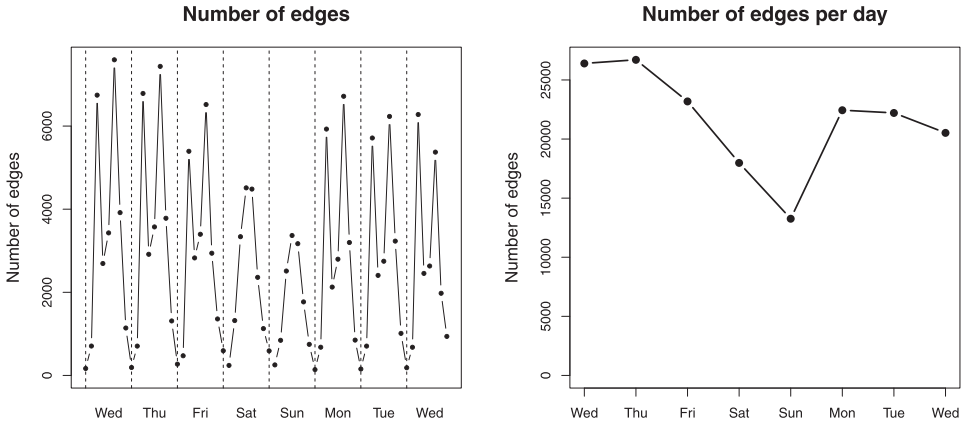

Fig. 7. **London bikes data**. The number of edges in each time frame for the London bikes dataset are shown on the left panel. The right panel shows instead the total number of edges in each day.

observed throughout the weekend, with a single activity peak appearing in the late mornings. Although Randriamanamihaga *et al.* (2014) focus their analysis on the bike sharing system of Paris, they observe and point out exactly the same type of temporal dynamics.

Results. We run our algorithm `GreedyIcl` once for each of the initialization methods, with each run taking about 6 hours long. The optimal clustering found has a total of $K = 43$ groups.

Heterogeneity. The temporal dynamic of the activity level is also exhibited in Figure 8 by the number of active groups at each time frame: as in the Enron dataset, we use this as a measure for the level of heterogeneity in the network. It appears that, as activity peaks in weekdays, the number of active groups doubles, signaling a heavy increase in heterogeneity within the network. Surprisingly, weekend days do not exhibit a markedly different heterogeneity level with respect to weekdays, even though the overall activity level is lower. This suggests that fewer nodes become active in the weekend, but their connection patterns are not particularly different than those seen in weekdays.

Characterization of the groups. The estimated transition probabilities and connection probabilities are shown in Appendix A. The groups can be roughly divided in two categories: the first 20 groups have a large aggregated size, mostly high stability, low connection probabilities and a good balance between expected out-degree and in-degree. The rest of the groups have smaller sizes, high instability, higher connection probabilities, and in some cases very unbalanced out-degrees and in-degrees.

Figure 9 shows the temporal evolution of the size of a selection of groups. It appears that the trends in the activity level have a huge effect on the migrations of stations between groups. As nodes become more active, they leave the large groups (first category) and move to smaller groups, which are better suited to capture the details of their new connection profile in the network. During the high-activity

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

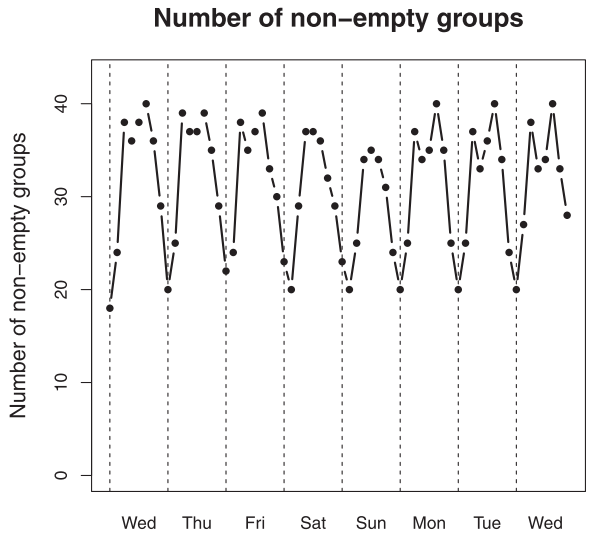

Fig. 8. **London bikes data**. This plot shows the number of non-empty groups at each time frame for the London bikes dataset. The total number of unique groups found is 43. The level of heterogeneity in the network has a noticeable temporal dynamic, which mimics the evolution of the number of links shown in Figure 7.

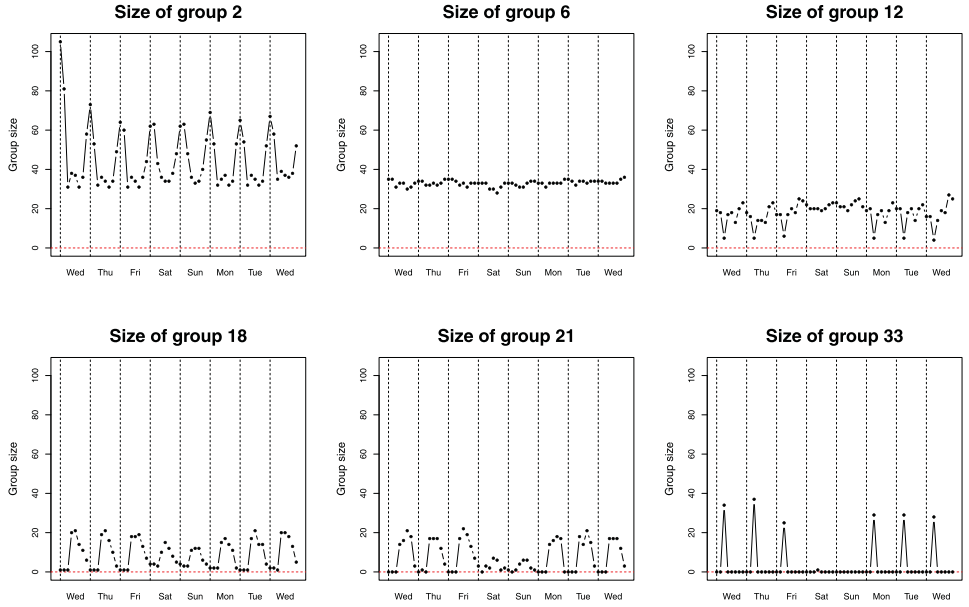

Fig. 9. (Color online) **London bikes data**. Temporal dynamics for the sizes of some of the groups found. The three groups on the upper row belong to the first category (groups containing mostly inactive nodes): group 2 swells consistently every night; group 6 is stably present throughout; whereas group 12 is almost emptied in every weekday at the start of the office hours. The groups on the lower row contain instead very active stations: group 18 is emptied every night; group 21 is also emptied throughout the weekends; whereas group 33 activates every weekday at the start of office hours.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

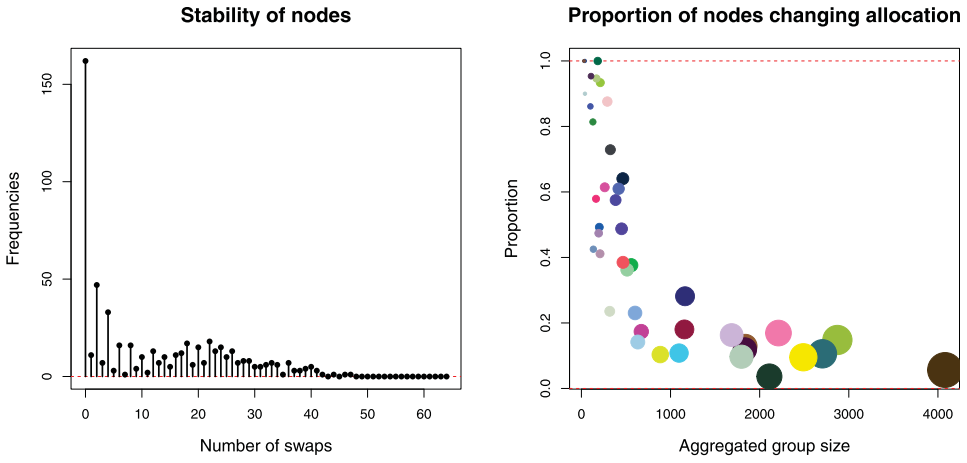

Fig. 10. (Color online) **London bikes data**. Stability level in the London bikes dataset. The left panel shows that although a good portion of nodes are very stable, there are some stations changing their allocations in more than 45 out of 64 time frames. The plot on the right-hand side studies instead the stability of groups. It appears that larger groups are very stable, and vice versa smaller groups are not. The size of the circles is proportional to the size of the group aggregated over time.

regime, these nodes may end up visiting several groups based on the evolution of their connectivity patterns. Nonetheless, a good portion of nodes is not affected by the variation of activity level and simply remains stably in the larger and inactive groups.

These arguments are very well supported by the plots in Figure 10. It appears that the stability of nodes is very heterogeneous in that there are many very stable nodes but also some very unstable ones (left panel). At the same time, the plot on the right-hand side reinforces the idea that larger groups are also very stable, whereas smaller groups exhibits high instability.

As shown in Figure 11, the stability does not seem to be particularly related to the geographical position of stations. Furthermore, the number of groups containing stable stations (nodes that never change allocation) is relatively high.

One additional feature that distinguishes the two categories of groups found can be observed in Figure 12. Here, for every group, the expected out-degrees versus the expected in-degrees are plotted. While stable groups (first category) tend to exhibit a good balance between the two degrees, smaller and unstable groups can also have very unbalanced degrees. This means, evidently, that whenever a node joins one of this unbalanced groups, it may send many more edges than what it receives (or vice versa). In this bike sharing context, the corresponding stations may require special attention due to temporary excess of hiring demands (or excess of arrivals).

We finally point out that our results on the stability of this network appear to be qualitatively different from those of Matias *et al.* (2018) and Corneli *et al.* (2017). The main reason for this is that these works aim at capturing the temporal dynamics through the evolutions of the block connection probabilities (i.e. the frequencies of interactions), whereas in our model these quantities are fixed and the allocation variables are instead allowed to change over time.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

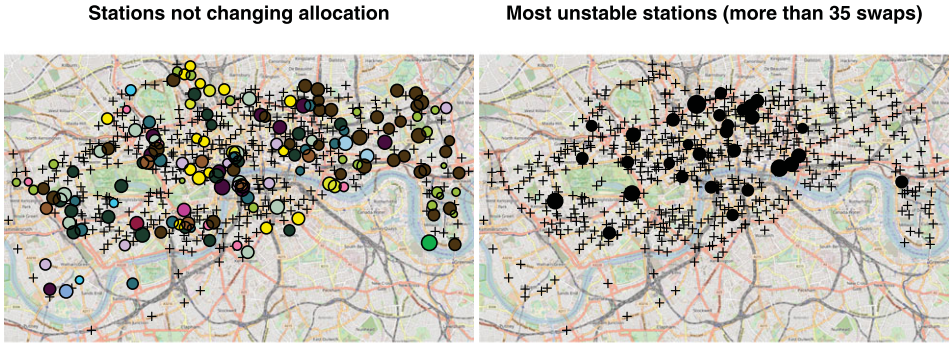

Fig. 11. (Color online) **London bikes data**. On the left panel, the spatial distribution of the stations not changing allocations is shown for the London bikes dataset. The color of the circles correspond to the group they are allocated to at every time frame. On the right panel, instead, the stations performing more than 35 swaps are shown. In both plots, the size of nodes is proportional to the sum of the out-degree and in-degree, whereas the crosses correspond to the remaining stations.

9 Conclusions

The statistical analysis of dynamic networks is particularly challenging, both from a modeling point of view (due to the temporal dynamics) and from an estimation point of view, due to the inherent computational difficulties. In this paper, we have focused on an extension of the SBM to dynamic networks, where the temporal evolution of the nodal information is captured by a Markovian process. Our formulation allows one to integrate out all of the model parameters from both the likelihood and the prior, thereby obtaining an analytical formula for the marginal posterior of the allocation variables. In a model-based clustering context, such a posterior is equivalent to the exact ICL, which is widely used as an optimality criterion for partitions.

Taking advantage of these results, we have proposed a greedy algorithm to estimate the optimal clustering of the nodes. Our algorithm resembles other tools proposed in the recent literature, and scales particularly well with the size of the data. However, only convergence to a local optimum is guaranteed, hence several restarts and a careful initialization are particularly useful in order to find the global optimum.

One appealing feature of our approach is that model-choice is carried out automatically, since the number of groups can be deduced at each time frame from the optimal clustering solution.

Through a simulation study, we have validated both our optimality criterion and the greedy algorithm used to estimate the optimal partition. Also, we have compared our method to one based on a variational Expectation-Maximization algorithm, showing that with careful initialization our approach achieves better results when the underlying number of groups is unknown.

We have applied our methodology to the Enron email dataset, and found 17 underlying groups. Each of the groups found appears to have a specific role within the network and the original status of the members seem to be particularly related

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

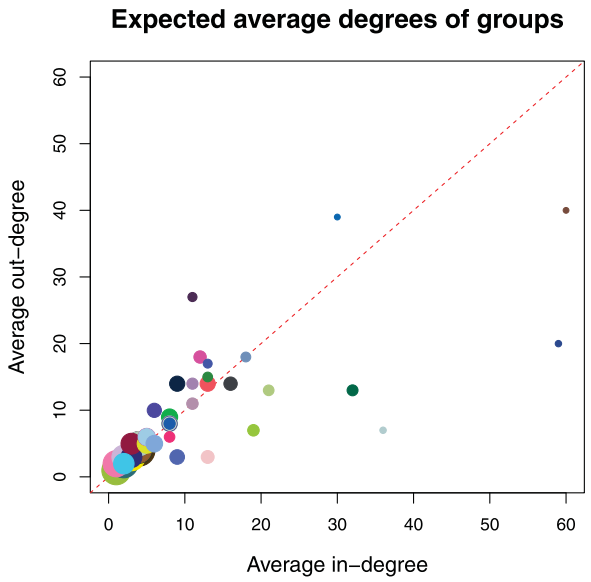

Fig. 12. (Color online) **London bikes data**. Expected out-degree versus expected in-degree for the groups found in the London bikes dataset. The size of circles is proportional to the size of groups aggregated over time. All of the large groups of inactive nodes have balanced degrees, in that they tend to send and receive the same number of edges. Smaller groups, instead, can exhibit very unbalanced degrees, either in favor of the out-degree or in-degree.

with the nodes' allocations. Also, we have analyzed some relevant summaries from our optimal clustering underlining the dynamic evolution of the heterogeneity within the network and of the overall activity level.

We have also applied our methodology to a London Bikes dataset, to study the flows and connections between bike stations. Our method has returned an optimal clustering made of 43 groups. The analysis of such a complex structure has been particularly challenging but we have managed to extract some interesting information regarding the dynamics captured by the model.

We note that, while the practice of integrating out the likelihood parameters has been exploited in a number of recent papers, the collapsing of the transition probabilities in the Markov process is a rather uncommon technique. Not only this creates an ideal setup for our method, but it may also be generalized and exploited in arbitrary discrete hidden Markov models and their extensions.

Additionally, our work may be extended in a number of ways. We have used our algorithm only on binary networks, but this approach generalizes to networks with other edge types. Incomplete weighted graphs may be handled using two likelihood parts: one corresponding to a binary network indicating only the presence or absence of edges, and another modeling the actual values appearing on the present edges. The parameters of this model may be collapsed leading to the same $\mathscr{ICL}_{ex}$ optimization problem.

The initialization plays an important role in our method, yet finding a good starting partition is a great challenge. In fact, there is a lack of scalable methods that can handle complex dynamic objects such as networks. Furthermore, clustering

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

cannot be attempted for each time frame independently since this results in label-switching problems. Here, we have proposed two initialization methods which may in some cases improve the results. Another relevant approach that may be employed is spectral clustering, which has been shown to perform well in a wide range of scenarios (see, for example, Von Luxburg, 2007 and references therein), and may be used to initialize efficiently model-based methods such as ours.

Extensions to the supervised classification case are also straightforward to consider: if some of the allocations are known, these need not be updated during the greedy optimization. From a Bayesian perspective, the optimal clustering solution obtained will then maximize the posterior predictive distribution, rather than the posterior. This strategy would be useful when dealing with nodes that join or leave the study dynamically, since, as suggested by Matias & Miele (2017), a fixed group of inactive nodes may be created.

Acknowledgments

The authors would like to thank the three reviewers and the editor for providing important suggestions that helped to improve the paper. The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289. Riccardo Rastelli and Nial Friel's research was supported by a Science Foundation Ireland grant: 12/IP/1424. Riccardo Rastelli's research was also funded through the Vienna Science and Technology Fund (WWTF) Project MA14-031.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, **9**(Sep), 1981–2014.

Bertoletti, M., Friel, N., & Rastelli, R. (2015). Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion. *Metron*, **73**(2), 177–199.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, **48**(3), 259–302.

Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(7), 719–725.

Côme, E., & Latouche, P. (2015). Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Statistical Modelling*, **15**(6), 564–589.

Corneli, M., Latouche, P., & Rossi, F. (2016). Exact ICL maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. *Neurocomputing*, **192**, 81–91.

Corneli, M., Latouche, P., & Rossi, F. (2017). Multiple change points detection and clustering in dynamic network. In press.

Daudin, J. J., Picard, F., & Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, **18**(2), 173–183.

Farajtabar, M., Wang, Y., Rodriguez, M. G., Li, S., Zha, H., & Song, L. (2015). Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in neural information processing systems*. NIPS, pp. 1954–1962.

Friel, N., Rastelli, R., Wyse, J., & Raftery, A. E. (2016). Interlocking directorates in irish companies using a latent space model for bipartite networks. *Proceedings of the National Academy of Sciences*, **113**(24), 6629–6634.

Guigourès, R., Boullé, M., & Rossi, F. (2015). Discovering patterns in time-varying graphs: A triclustering approach. *Advances in Data Analysis and Classification*, 1–28. Retrieved from https://link.springer.com/article/10.1007/s11634-015-0218-6.

Ho, Q., Song, L., & Xing, E. P. (2011). Evolving cluster mixed-membership blockmodel for time-evolving networks. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, **15**, 342–350.

Hoff, P. D., Raftery, A. E., & Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, **97**(460), 1090–1098.

Ishiguro, K., Iwata, T., Ueda, N., & Tenenbaum, J. B. (2010). Dynamic infinite relational model for time-varying relational data analysis. In *Advances in neural information processing systems*. NIPS, 919–927.

Kim, M., & Leskovec, J. (2013). Nonparametric multi-group membership model for dynamic networks. In *Advances in neural information processing systems (25)*. NIPS, pp. 1385–1393.

Matias, C., & Miele, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **79**(4), 1119–1141.

Matias, C., Rebafka, T., & Villers, F. (2018). A semiparametric extension of the stochastic block model for longitudinal networks. *Biometrika*, **105**(3), 665–680.

McDaid, A. F., Murphy, T. B., Friel, N., & Hurley, N. J. (2013). Improved bayesian inference for the stochastic block model with application to large networks. *Computational Statistics & Data Analysis*, **60**, 12–31.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, **69**(6), 066133.

Nobile, A., & Fearnside, A. T. (2007). Bayesian finite mixtures with an unknown number of components: the allocation sampler. *Statistics and Computing*, **17**(2), 147–162.

Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, **96**(455), 1077–1087.

Randriamanamihaga, A. N., Côme, E., Oukhellou, L., & Govaert, G. (2014). Clustering the velib dynamic origin/destination flows using a family of poisson mixture models. *Neurocomputing*, **141**, 124–138.

Sarkar, P., & Moore, A. W. (2005). Dynamic social network analysis using latent space models. *Sigkdd Explorations: Special Edition on Link Mining*, **7**, 31–40.

Strehl, A., & Ghosh, J. (2003). Cluster ensembles – A knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, **3**(Dec), 583–617.

Tran, L., Farajtabar, M., Song, L., & Zha, H. (2015). Netcodec: Community detection from individual activities. In *Proceedings of the 2015 SIAM international conference on data mining*. SIAM, pp. 91–99.

Transport for London. (2016). Retrieved from http://cycling.data.tfl.gov.uk/.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, **17**(4), 395–416.

Wang, Y. J., & Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, **82**(397), 8–19.

Wyse, J., Friel, N., & Latouche, P. (2017). Inferring structure in bipartite networks using the latent blockmodel and exact icl. *Network Science*, **5**(1), 45–69.

Xing, E. P., Fu, W., & Song, L. (2010). A state-space mixed membership blockmodel for dynamic network tomography. *Annals of Applied Statistics*, **4**(2), 535–566.

Xu, K. (2015). Stochastic block transition models for dynamic networks. In *Artificial intelligence and statistics*. AISTATS, pp. 1079–1087.

Xu, K. S., & Hero, A. O. (2014). Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, **8**(4), 552–562.

Yang, T., Chi, Y., Zhu, S., Gong, Y., & Jin, R. (2011). Detecting communities and their evolutions in dynamic social networks – A Bayesian approach. *Machine Learning*, **82**(2), 157–189.

Zhou, K., Zha, H., & Song, L. (2013). Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial intelligence and statistics*. AISTATS, pp. 641–649.

A Appendix: Additional plots

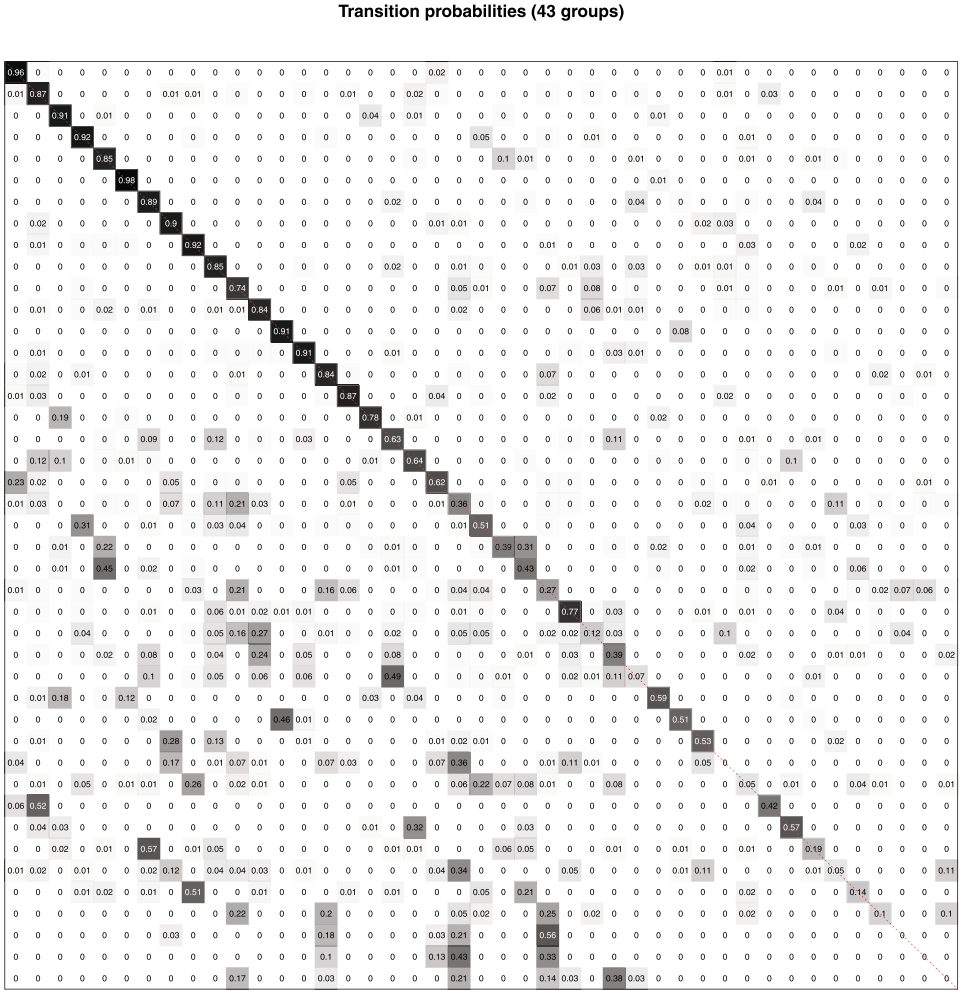

Fig. A 1. (Color online) **London bikes dataset**. Transition probabilities for all groups, ordered decreasingly by aggregated size.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press

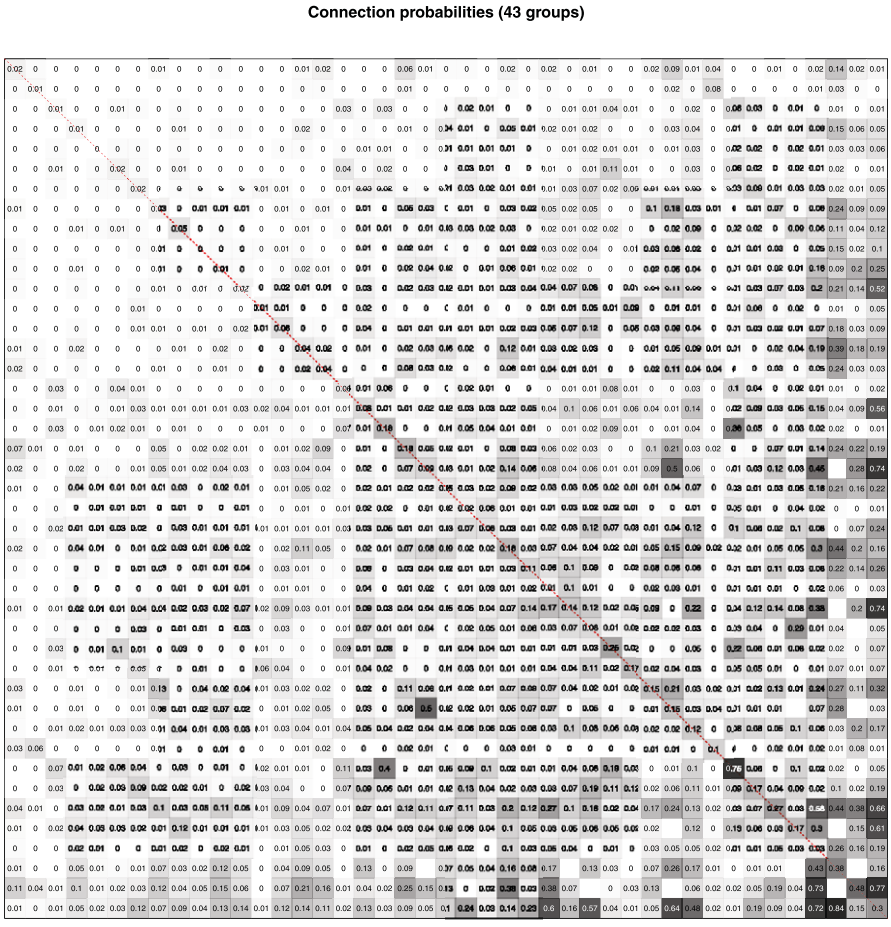

Fig. A 2. (Color online) **London bikes dataset**. Connection probabilities for all groups, ordered decreasingly by aggregated size.

https://doi.org/10.1017/nws.2018.19 Published online by Cambridge University Press