Multiple change points detection and clustering in dynamic networks

Marco Corneli, Pierre Latouche, Fabrice Rossi

Statistics and computing manuscript No.
(will be inserted by the editor)

Multiple change points detection and clustering in dynamic network

Marco Corneli · Pierre Latouche · Fabrice Rossi

Received: date / Accepted: date

Abstract The increasing amount of data stored in the form of dynamic interactions between actors necessitates the use of methodologies to automatically extract relevant information. The interactions can be represented by dynamic networks in which most existing methods look for clusters of vertices to summarize the data. In this paper, a new framework is proposed in order to cluster the vertices while detecting change points in the intensities of the interactions. These change points are key in the understanding of the temporal interactions. The model used involves non homogeneous Poisson point processes with cluster dependent piecewise constant intensity functions and common discontinuity points. A variational expectation maximization algorithm is derived for inference. We show that the pruned exact linear time method, originally developed for univariate time series, can be considered for the maximization step. This allows the detection of both the number of change points and their location. Experiments on artificial and real datasets are carried out and the proposed approach is compared with related methods.

Keywords Dynamic networks · non homogeneous Poisson point processes · stochastic block model · variational EM · PELT

M. Corneli
90, rue de Tolbiac 75013 Paris, France
Tel: +33 1 44 07 87 06
E-mail: Marco.Corneli@malix.univ-paris1.fr

P. Latouche
90 rue de Tolbiac, 75013 Paris, France Tel: +33 1 44 07 88 26
Fax: +33 1 44 07 89 25
E-mail: pierre.latouche@univ-paris1.fr

F. Rossi
90 rue de Tolbiac, 75013 Paris, France Tel: +33 1 44 07 89 22
Fax: +33 1 44 07 89 22
E-mail: Fabrice.Rossi@apiacoa.org

1 Introduction

The development of communication infrastructures has led to an unprecedented collection of data stored in the form of interactions between units of interest. For instance, interactions can correspond to email exchanges between employees of a company. They can also be characterized by posts between individuals via social media (Twitter, Facebook, Linkedin) or text messaging. In practice, interactions occur at specific time points, which are often recorded. Thus, if actor i interacts with actor j, at time η, the corresponding interaction can be represented by a triple (i, j, η). The set of all triples can then be used to build a dynamic graph where each actor is associated with a node, and an edge between two nodes (i, j) is present at time η if the corresponding interaction (i, j, ν) is recorded in the data set.

Statistical models for dynamic graphs are usually discrete in time, i.e. predefined time intervals are considered and interactions during those time intervals are aggregated to obtain snapshots. In the binary case, two nodes of a graph snapshot are connected if an interaction between them occurred during the corresponding time frame. The number of interactions between the nodes during the time frame can also be recorded. Most of these models consider the stochastic block model (SBM) (Wang and Wong, 1987; Nowicki and Snijders, 2001) and extend it to the dynamic framework. SBM assumes that vertices are spread in latent clusters and that the probability for two nodes to connect depends on their respective clusters. No assumption is made on the connection probabilities such that different types of clusters of nodes can be taken into account. In its original form, SBM cannot deal with dynamic interactions. Yang et al (2011) proposed a dynamic version of SBM by allowing the cluster of each node to switch at time $t + 1$ depending on its current state at time t. The switching probabilities are all characterized by a transition matrix. The alternative approach of Xu and Hero III (2013) characterizes the temporal changes through a state space model and relies on the Kalman filter along with the Rauch-Tung-Striebel smoother, for inference. Contrary to Yang et al (2011), the edge probabilities are seen as time varying parameters in Xu and Hero III (2013). The work of Yang et al (2011) was generalized by Matias and Miele (2016) to deal with other types of edges. In their paper, they also showed that it was not possible to let both the connectivity parameters and cluster memberships vary over time without incurring into identifiability issues. Other static variants of the SBM model have also been adapted to the dynamic context. For instance, Xing et al (2010), Ho et al (2011) and Kim and Leskovec (2013) extended the mixed membership SBM of Airoldi et al (2008) in order to look for overlapping clusters of nodes, through time. Moreover, the dynamic random subgraph model (DRSM) (Zreik et al, 2016) was built upon the RSM model (Jernite et al, 2014) to uncover clusters within subgraphs provided a priori. Note that the popular static latent position model of Hoff et al (2002) was also extended by Sarkar and Moore (2005) and Friel et al (2016) to deal with dynamic interactions.

The scope of application of the models for dynamic graphs we have discussed so far is limited. Indeed, aggregating the data leads to a loss of information and the choice of the time intervals used to build the snapshots has a strong impact on the results (Matias et al, 2015). In order to deal with dynamic interactions on a continuous time frame, a natural choice is to consider point processes. Thus, Matias et al (2015) relied on the so called (doubly stochastic) non homogenous Poisson

point processes (NHPPP). Following a SBM like approach, nodes are assumed to belong to hidden clusters. Each pair of nodes is then associated to a NHPPP whose intensity function depends on the respective clusters. A variational expectation maximization (VEM) algorithm is finally employed to non parametrically estimate these functions and to uncover the clusters. This work is partially related to Dubois et al (2013) who relied on a parametric form for the intensity functions, which depend on the past network history and other predefined statistics. An alternative approach was proposed by Corneli et al (2016b) where the intensity functions of the Poisson point processes are assumed to be piecewise constant on predefined time intervals, each time interval belonging to an hidden time cluster. In their model, the value of each intensity function at time t not only depends on the clusters of nodes, but also on the corresponding time cluster. A greedy search algorithm was proposed to uncover the clusters.

In this paper, we extend the work of Matias et al (2015) to simultaneously uncover clusters of nodes sharing connection profiles, and to look for adjacent time intervals on which the connectivity patterns between pairs of clusters are stationary. In practice, considering dynamic interactions over a continuous time interval, we assume the intensity functions of the NHPPP to depend on the hidden node clusters and to be piecewise constant. Moreover, they are assumed to share $D-1$ *common* discontinuities whose location and number are unknown. These discontinuities induce a segmentation of the entire time interval over which the interactions are observed. In order to perform inference, a VEM algorithm is derived. We show that the V-M step can be tackled relying on a multiple changepoint detection tecnique for univariate time series, the pruned exact linear time (PELT) (Killick et al, 2012) method, which we adapted to our framework. Finally, the number of clusters of vertices is estimated using a Bayesian information criterion (BIC) involving variational approximations.

An alternative model to Poisson processes based ones has been proposed in Guigourès et al (2012, 2015). As the model proposed in the present paper, this approach does not aggregate the temporal network prior analyzing it. It also produces both clusters of nodes and a segmentation of time. It is therefore a natural reference model in our context. By being based on the non parametric MODL approach Boullé (2010), the triclustering technique of Guigourès et al (2012, 2015) can handle non Poisson distributed interaction counts but it is somewhat blind to intensity changes as will be illustrated in Section 4.1.2. In particular, time segments cannot be detected when the type of connectivity structure is persistent through time but subject to parallel shifts in the interaction intensity levels.

The paper is organized as follows. In Section 2, we introduce the model and notations. In Section 3, we derive the VEM algorithm along with the model selection procedure. Finally, in Section 4, some experiments on simulated and real data are carried out to assess the proposed methodology.

2 A generative model for continuous time dynamic graphs

We consider interaction data with time-stamps and assume that a set of N actors $\{1,\dots,N\}$ is provided. Those actors are assumed to interact (possibly repeatedly) at arbitrary times, that are recorded. A flat (i.e. non graphical) representation of those interactions is a finite set $\mathcal{D} = \{(i_m, j_m, \nu_m)\}_{1\leq m\leq M}$, subset of $\{1,\dots,N\}^2\times$

$\mathbb{R}^+$, in which a triple (i, j, ν) represents an interaction between actors i and j, at time ν. The temporal period under study is the interval $[0, T]$. Without loss of generality, we assume that $\mathcal{D}$ is sorted on the time variable. In addition, each interaction time is supposed to be unique (see Section 2.3). Then $0 < \nu_1 < \nu_2 < \ldots < \nu_M < T$[1].

In order to simplify the presentation, self interactions and directed interactions are not considered in this paper. Extensions to those cases are straightforward. The proposed approach makes the implicit assumption that interaction time spans do not play a significant roles in the general behaviour of the actors. Lifting this assumption and modelling explicitly the duration of the interactions is out of the scope of this paper.

2.1 Continuous time dynamic graph

Those interaction data can be seen as a dynamic (or time evolving) graph. The actor set corresponds to the set of nodes of the graph. Then each interaction (i, j, ν) can be seen as an edge in the graph, connecting i and j, and decorated by the time ν. This graph is in general a multiple graph in the sense that two nodes i and j can be connected by multiple edges (with different times). Another way of seeing the interaction data as a graph would be to follow Guigourès et al (2015) and to use the general framework of Casteigts et al (2012). In both cases, the most important aspect is that the proposed model is not based on a time series of graphs but rather on a fixed graph whose edges have a temporal dimension. This latter approach is more general as it allows one to model time as a continuous variable rather than a discrete one.

2.2 Node/actor clusters

Following the Stochastic Block Model (SBM) generative scheme (Wang and Wong, 1987; Nowicki and Snijders, 2001), we assume that the connectivity in a dynamic graph is explained by hidden roles: each node/actor belongs to a cluster of nodes/actors and the way actors interact depends only on their respective clusters. In addition, cluster memberships are assumed to be fixed through time. Conversely, as we shall see in the following section, the connectivity parameters are allowed to change. These two modeling assumptions respect the identifiability principle for dynamic SBMs provided in Matias and Miele (2016) who showed that the connectivity parameters and group memberships should not both be allowed to change over time.

Thus, each node i is associated to a random variable Z_i sampled from a multinomial distribution

$$\mathbb{P}(Z_i = k) = \pi_k, \quad \forall k \in \{1, \ldots, K\},$$

where K is the number of clusters and

$$\sum_{k=1}^{K} \pi_k = 1.$$

[1] Interaction times are assumed to be strictly positive and ν_{M+1} is assumed to fall outside the time interval $[0, T]$: $\nu_{M+1} \geq T$ a.s.

We denote $Z = \{Z_1, \dots, Z_N\}$ the set of all latent variables. Note that the 0-1 notation $Z_i = (Z_{i1}, \dots, Z_{iK})$ with $Z_{ik} = 1$ if node i belongs to the k-th cluster, 0 otherwise, will be used interchangeably with the notation where $Z_i \in \{1, \dots, K\}$, when no confusion occurs. Finally, the vector of cluster proportions is denoted $\pi = (\pi_1, \dots, \pi_K)$.

2.3 Time-stamped interactions as point processes

Let us consider two fixed nodes, i and j. We denote $M^{(i,j)}$ the number of edges between i and j, in other words the number of distinct times ν such that $(i, j, \nu) \in \mathcal{D}$. Without loss of generality, those interaction times can be sorted into the following list

$$\mathcal{A}^{(i,j)} := \{\nu_1^{(i,j)}, \dots, \nu_{M^{(i,j)}}^{(i,j)}\}, \tag{1}$$

with $\nu_1^{(i,j)} < \nu_2^{(i,j)} < \cdots < \nu_{M^{(i,j)}}^{(i,j)}$. In probabilistic terms, $\mathcal{A}^{(i,j)}$ can be seen as a point process. As such a point process takes values in $[0, T]$, it is naturally associated to a counting process $\{M^{(i,j)}(t)\}_{t\in[0,T]}$. The random variable $M^{(i,j)}(t)$ counts the number of interactions, between i and j, that happened before (or exactly at) t, i.e.

$$M^{(i,j)}(t) = \left|\mathcal{A}^{(i,j)} \cap]0, t]\right|,$$

where $|S|$ denotes the cardinal of the set S. A simple yet flexible generative model for the interaction times in $\mathcal{A}^{(i,j)}$ is the nonhomogeneous Poisson point process (NHPPP). Such a process is characterized by an intensity function $\kappa^{(i,j)}(\cdot)$, positive and integrable on $[0, T]$. Denoting

$$\bar{\kappa}^{(i,j)}(t) = \int_0^t \kappa^{(i,j)}(s)ds, \qquad t \leq T,$$

assuming that $\mathcal{A}^{(i,j)}$ is generated by a NHPPP with intensity function $\kappa^{(i,j)}$ means that for all $t \in [0, T]$, $M^{(i,j)}(t)$ follows a Poisson distribution with parameter $\bar{\kappa}_{ij}(t)$.

As pointed out in the previous section, the proposed model follows the SBM rationale in which the way actors interact depends only on their hidden role (a.k.a. cluster). Combined with the NHPPP assumption, this leads to the following assumptions:

1. given Z, the $\mathcal{A}^{(i,j)}$ for all $i > j$ are generated by $N(N-1)/2$ independent nonhomogeneous Poisson point processes with intensity functions $\{\kappa^{(i,j)}(\cdot)\}_{i>j}$;
2. there are $K(K+1)/2$ positive integrable functions $\boldsymbol{\lambda} = \{\lambda_{kg}(\cdot)\}_{k,g}$ defined on $[0, T]$ such that $\kappa^{(i,j)}(t) = \lambda_{Z_i Z_j}(t)$ for all $t \in [0, T]$.

With those assumptions, the conditional likelihood of a set of interactions $\mathcal{A}^{(i,j)}$ between two nodes i and j is given by (Daley and Vere-Jones, 2003)

$$p(\mathcal{A}^{(i,j)}|Z_i = k, Z_j = g, \lambda_{kg}) = \prod_{m=1}^{M^{(i,j)}} \lambda_{kg}(\nu_m^{(i,j)}) \exp\left(-\Lambda_{kg}(T)\right), \tag{2}$$

with

$$\Lambda_{kg}(t) := \int_0^t \lambda_{kg}(s)ds.$$

Notice that as the $\mathcal{A}^{(i,j)}$ are assumed to be generated by Poisson point processes, all the interaction times are distinct (almost surely). This justifies the assumption used at the beginning of the present section.

The data set $\mathcal{D}$ can be seen as the union of all the $\mathcal{A}^{(i,j)}$, with the added information of the interacting pairs at each interaction time: the pair (i_m, j_m) corresponds to the nodes actually having an interaction at time ν_m. Combining equation (2) applied to all those processes with the conditional independence assumption between them, we obtain the following complete data likelihood

$$\begin{aligned} p(\mathcal{D}, Z|\boldsymbol{\lambda}, \pi) &= p(\mathcal{D}|Z, \boldsymbol{\lambda}) p(Z|\pi) \\ &= \exp\left(-\sum_{j>i}^{N} \Lambda_{Z_i Z_j}(T)\right) \prod_{m=1}^{M} \lambda_{Z_{i_m} Z_{j_m}}(\nu_m) \prod_{i=1}^{N} \pi_{Z_i}. \end{aligned} \tag{3}$$

2.4 Common change points

The generative model proposed above assumes some structure on the node set but none on the time interval. While this approach can be interesting in some settings (Corneli et al, 2016a), it reduces to some extent interpretation possibilities. In particular, such a flexible model does not emphasize brutal changes in the interaction structure between clusters of nodes.

In order to emphasize those aspects, we assume that the intensity functions of the NHPPPs are piecewise constant and that the D intervals on which they are constant are shared between all functions. In other words, we assume that there are $D-1$ change points

$$0 = \eta_0 < \eta_1 < \cdots < \eta_{D-1} < \eta_D = T, \tag{4}$$

such that for all $0 \leq d \leq D$, $1 \leq k \leq K$ and $1 \leq g \leq K$, $\lambda_{kg}(\cdot)$ is constant on $[\eta_d, \eta_{d+1}[$. Therefore

$$\lambda_{kg}(t) = \sum_{d=1}^{D} \lambda_{kgd} \mathbf{1}_{[\eta_{d-1}\eta_d[}(t), \qquad \forall k, g \in \{1, \ldots, K\}, \tag{5}$$

where $\lambda_{kgd} := \lambda_{kg}(\eta_{d-1})$ and $\mathbf{1}_{\mathcal{G}}(.)$ is the indicator function over a set $\mathcal{G}$[2]. In the following, $\eta = \{\eta_1, \ldots, \eta_{D-1}\}$ denotes the set of change points and $\boldsymbol{\lambda}$ is the $(K \times K \times D)$ tensor[3] with elements λ_{kgd}.

A crucial consequence of the piecewise constant assumption is that on an interval $[\eta_d, \eta_{d+1}[$, all the Poisson point processes are homogeneous and thus do not exhibit any temporal structure. On the contrary, the intensity is allowed to change arbitrarily from one interval to the next one, allowing brutal changes in the interaction patterns between clusters. In addition, taking into account the piecewise constant assumption, allows us to simplify the complete data likelihood as shown in the following proposition.

[2] Since the whole time interval $[0, T]$ is considered, the intensity functions can be assumed (exceptionally) left continuous in $t = T$. In formulas: $\lambda_{kg}(T) = \lambda_{kg}(\eta_{d-1})$.

[3] We use the same notation to denote the set of $K(K+1)/2$ intensity functions and the tensor because under our assumptions they correspond to two different views of the same object. Notice that the frontal slices of $\boldsymbol{\lambda}$ are symmetric $K \times K$ matrices since we are dealing with undirected graphs.

Proposition 1 *Using the constraints in equation* (5)*, the complete data log likelihood becomes*

$$\log p(\mathcal{D}, Z|\theta) = - \sum_{d=1}^{D} \sum_{k,g}^{K} \left[\lambda_{kgd} \Delta_d \left(\sum_{j>i}^{N} Z_{ik} Z_{jg} \right) - \log(\lambda_{kgd}) \left(\sum_{j>i}^{N} Z_{ik} Z_{jg} X_{ij}^{(d)} \right) \right] + \sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} \log \pi_k, \quad (6)$$

where

1. $\theta := \{\eta, \boldsymbol{\lambda}, \pi\}$
2. Δ_d *is the size of the interval* $[\eta_{d-1}, \eta_d[$,
3. $X_{ij}^{(d)} := N^{(i,j)}(\eta_d) - N^{(i,j)}(\eta_{d-1})$ *is the increment of conditional Poisson point process* $\{M^{(i,j)}(t)\}_{t\in[0,T]}$ *over the segment* $[\eta_{d-1}, \eta_d[$, i.e. *the number of interactions that occurred between* i *and* j *during the time interval* $[\eta_{d-1}, \eta_d[$.

Proof See appendix A.1.

3 Estimation

This section focuses on the inference of the model proposed above. This involves estimating the number K of clusters and the number D of adjacent time intervals, as well as the parameter θ. Therefore, we introduce first a penalized likelihood criterion for model selection. A variational approximation for this criterion is then introduced, leading to a variational expectation maximization (VEM) algorithm. It is finally shown how to integrate an efficient change point detection algorithm in the VEM loop to estimate the piecewise constant intensities.

3.1 Penalized likelihood

Given the set of all observed interactions $\mathcal{D}$, our goal is to estimate the clustering structure in the actor/node space and the segmentation structure in the temporal space. In particular, we aim at choosing the number K of clusters and the number of change points (by choosing D) as well as their location η.

A natural quality measure in this context is the observed data (integrated) log-likelihood

$$\log p(\mathcal{D}|K, \eta, D) = \log \left(\int_{\boldsymbol{\lambda}, \pi} p(\mathcal{D}, \boldsymbol{\lambda}, \pi|K, \eta, D) d\boldsymbol{\lambda} d\pi \right). \quad (7)$$

Unfortunately this marginal log likelihood does not have an analytical form so we propose to replace it with a penalized likelihood (BIC like) term

$$\log \tilde{p}(\mathcal{D}|K, \eta, D) = \max_{\boldsymbol{\lambda}, \pi} \log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi) - \frac{1}{2} C(K, D) \log \alpha, \quad (8)$$

where

$$C(K,D) := K - 1 + \frac{K(K+1)D}{2},$$

accounts for the number of model parameters. The term α in equation (8) is related to the number of observations and will be discussed in Section 3.4.2. In other words, we consider the following optimization problem, for inference

$$\max_{K,\eta,D,\boldsymbol{\lambda},\pi} \log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi) - \frac{1}{2}C(K,D)\log\alpha. \tag{9}$$

While this allows to avoid to consider directly the log-likelihood in equation (7), two difficulties remain: $\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi)$ is not directly calculable and the optimization over η and D is a complex non smooth problem. To tackle these issues, two strategies are proposed. A variational approach is used in order to derive a lower bound of $\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi)$ (see Sections 3.2 and 3.3) while a change point detection technique is considered to address the optimization over η and D (see Section 3.4).

3.2 A variational bound

The first difficulty mentioned above is that computing the log likelihood term $\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi)$ is not tractable. Indeed, it involves summing over all the K^N possible sets Z, i.e.

$$\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi) = \log\left(\sum_Z p(\mathcal{D},Z|K,\eta,D,\boldsymbol{\lambda},\pi)\right).$$

Moreover, $p(Z|\mathcal{D},K,\eta,D,\boldsymbol{\lambda},\pi)$ cannot be factorized so the standard expectation maximization (EM) algorithm (A. P. Dempster, 1977) cannot be considered for inference. For more details on these issues, see Daudin et al (2008) in the case of the standard SBM.

Therefore, we introduce an approximate distribution $q(Z)$ for Z and use the standard variational decomposition

$$\log p(\mathcal{D}|K,\eta,D,\boldsymbol{\lambda},\pi) = \mathcal{L}\left(q(Z);K,\eta,D,\boldsymbol{\lambda},\pi\right) + KL\left(q(Z)||p(Z|\mathcal{D},K,\eta,D,\boldsymbol{\lambda},\pi)\right),$$

where

$$\begin{aligned}\mathcal{L}\left(q(Z);K,\eta,D,\boldsymbol{\lambda},\pi\right) &= \sum_Z q(Z)\log\frac{p(\mathcal{D},Z|K,\eta,D,\boldsymbol{\lambda},\pi)}{q(Z)}\\ &= \mathrm{E}_{q(Z)}\left[\log\frac{p(\mathcal{D},Z|K,\eta,D,\boldsymbol{\lambda},\pi)}{q(Z)}\right],\end{aligned}$$

and KL denotes the Kullback-Leibler divergence between the true and approximate posterior distribution $q(Z)$ of Z, given the data and model parameters

$$KL\left(q(Z)||p(Z|\mathcal{D},K,\eta,D,\boldsymbol{\lambda},\pi)\right) = -\mathrm{E}_{q(Z)}\left[\log\frac{p(Z|\mathcal{D},K,\eta,D,\boldsymbol{\lambda},\pi)}{q(Z)}\right].$$

Since $\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi)$ does not depend on the distribution $q(Z)$, maximizing $\mathcal{L}$ with respect to $q(Z)$ is equivalent to minimizing the KL divergence (over $q(Z)$ also).

As the KL divergence is non negative, $\mathcal{L}(q(Z); K, \eta, D, \boldsymbol{\lambda}, \pi)$ is obviously a lower bound of $\log p(\mathcal{D}|K, \eta, D, \boldsymbol{\lambda}, \pi)$ for all $q(Z)$, thus we use the standard variational bounding method. In our case, it consists in replacing the problem (9) by

$$\max_{K,\eta,D,\boldsymbol{\lambda},\pi,q(Z)} f(q(Z), K, \eta, D, \boldsymbol{\lambda}, \pi), \tag{10}$$

where

$$f(q(Z), K, \eta, D, \boldsymbol{\lambda}, \pi) = \mathcal{L}(q(Z); K, \eta, D, \boldsymbol{\lambda}, \pi) - \frac{1}{2} C(K, D) \log \alpha. \tag{11}$$

As in Daudin et al (2008) for the SBM, the Z distribution is approximated by a factorized distribution

$$q(Z) = \prod_{i=1}^{N} q(Z_i) = \prod_{i=1}^{N} \prod_{k=1}^{K} \tau_{ik}^{Z_{ik}}. \tag{12}$$

This leads to the following expression for $\mathcal{L}$.

Proposition 2 *If $q(Z)$ is of the form used in equation* (12)*, then*

$$\begin{aligned}\mathcal{L}(q(Z); K, \eta, D, \boldsymbol{\lambda}, \pi) = \\ - \sum_{d=1}^{D} \sum_{k,g}^{K} \left[\lambda_{kgd} \Delta_d \left(\sum_{j>i}^{N} \tau_{ik} \tau_{jg} \right) - \log(\lambda_{kgd}) \left(\sum_{j>i}^{N} \tau_{ik} \tau_{jg} X_{ij}^{(d)} \right) \right] \\ + \sum_{i=1}^{N} \sum_{k=1}^{K} \tau_{ik} \log \frac{\pi_k}{\tau_{ik}}.\end{aligned} \tag{13}$$

Proof This expression can easily be obtained by taking the expectation with respect to $q(Z)$ of the log likelihood in equation (6) and by adding the following entropy term

$$\mathcal{H}(q) := -E_{q(Z)}[\log(q(Z))].$$

3.3 Variational expectation maximization

Problem (10) is solved relying on a VEM algorithm which optimizes the function $f(q(Z); K, \eta, D, \lambda, \pi)$ which respect to η, D, $\boldsymbol{\lambda}$, π, and with respect to $q(Z)$, alternately. The number K of clusters is handled via a grid search (see Section 3.5) and it is considered fixed in the present section.

Given η, D, $\boldsymbol{\lambda}$ and π, the optimization with respect to $q(Z)$ is straightforward. This corresponds to the E step of the algorithm (see Section 3.3.1). Then, given $q(Z)$, the parameters $\boldsymbol{\lambda}$ as well as π are optimized away, and we use a change point detection procedure to optimize the criterion with respect to η and D. The optimization with respect to $\eta, D, \boldsymbol{\lambda}$ and π corresponds to the M step of the algorithm. The E and M steps are then iterated until convergence.

3.3.1 Maximization with respect to τ (E step)

The E step is based on the following proposition.

Proposition 3 *A first order condition for $f\left(q(Z), K, \eta, D, \boldsymbol{\lambda}, \pi\right)$ to be maximal with respect to $q(Z)$ is*

$$\tau_{ik} = \frac{\pi_k}{C} \exp\left\{ -\sum_{d=1}^{D}\sum_{g=1}^{K}\left[\lambda_{kgd}\Delta_d\left(\sum_{j\neq i}^{N}\tau_{jg}\right) - \log(\lambda_{kgd})\left(\sum_{j\neq i}^{N}\tau_{jg}X_{ij}^{(d)}\right)\right]\right\},$$

where

$$C = \sum_{k=1}^{K}\tau_{ik},$$

$\forall i \in \{1,\ldots,N\}$, $k \in \{1,\ldots,K\}$.

Proof See Appendix A.2.

In the E step of the algorithm, the entries of the set $\tau = \{\tau_1, \ldots, \tau_N\}$ are updated in turn until convergence of τ. This corresponds to a fixed point procedure, as in Daudin et al (2008) for instance. We emphasize that τ_{ik} is the (approximate) posterior probability for node i to be in cluster k, given the data and model parameters. Thus, the clustering structure uncovered by the method is encoded through τ.

3.3.2 Maximization with respect to π

Notice that $f\left(q(Z), K, \eta, D, \boldsymbol{\lambda}, \pi\right)$ can be written

$$f\left(q(Z), K, \eta, D, \boldsymbol{\lambda}, \pi\right) = \sum_{i=1}^{N}\sum_{k=1}^{K}\tau_{ik}\log\frac{\pi_k}{\tau_{ik}} + g(q(Z), K, \eta, D, \boldsymbol{\lambda}),$$

where the function g does not depend on π. Thus, $q(Z)$ and K being fixed, maximizing f with respect to η , D, $\boldsymbol{\lambda}$, π can be done independently on π and on the other parameters. The estimated value for π (under the constraint $\sum_{k=1}^{K}\pi_k = 1$) is then

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N}\tau_{ik}}{N}, \qquad \forall k \in \{1,\ldots,K\}.$$

3.3.3 Maximization with respect to $\boldsymbol{\lambda}$

Maximizing f with respect to $\boldsymbol{\lambda}$ leads to the following estimates

$$\hat{\lambda}_{kgd} = \begin{cases} \dfrac{\sum_{j>i}^{N}\tau_{ik}\tau_{jg}X_{ij}^{d}}{\Delta_d\sum_{j>i}^{N}\tau_{ik}\tau_{jg}} & \text{when } g > k, \\ \dfrac{\sum_{j\neq i}^{N}\tau_{ik}\tau_{jk}X_{ij}^{d}}{\Delta_d\sum_{j\neq i}^{N}\tau_{ik}\tau_{jk}} & \text{when } g = k. \end{cases} \tag{14}$$

Notice that contrary to $\hat{\pi}$, $\widehat{\boldsymbol{\lambda}}$ does depend on η and D, which are also considered in the M optimization step.

3.3.4 Maximization with respect to η and D

Obviously, we have

$$\max_{\eta,D,\boldsymbol{\lambda},\pi} f\left(q(Z),K,\eta,D,\boldsymbol{\lambda},\pi\right) = \max_{\eta,D}\max_{\boldsymbol{\lambda},\pi} f\left(q(Z),K,\eta,D,\boldsymbol{\lambda},\pi\right).$$

Thus, $q(Z)$ and K being fixed, f can be evaluated in $\hat{\boldsymbol{\lambda}}$ and $\hat{\pi}$ obtained in the previous sections. In detail

$$\max_{\boldsymbol{\lambda},\pi} f\left(q(Z),K,\eta,D,\boldsymbol{\lambda},\pi\right) = \sum_{d=1}^{D}\mathcal{G}([\eta_{d-1},\eta_d[) - \frac{1}{2}\frac{K(K+1)D}{2}\log\alpha + const, \quad (15)$$

where all the terms which do not depend on η and/or D have been absorbed into the constant *const* and

$$\mathcal{G}([\eta_{d-1},\eta_d[) := \\ -\sum_{k,g}\left[\hat{\lambda}_{kgd}\Delta_d\left(\sum_{j>i}\tau_{ik}\tau_{jg}\right) - \log(\hat{\lambda}_{kgd})\left(\sum_{j>i}\tau_{ik}\tau_{jg}X_{ij}^{(d)}\right)\right]. \quad (16)$$

Notice that the criterion to maximize (now with respect to η and D) is a sum of independent components: each *gain* function $\mathcal{G}([\eta_{d-1},\eta_d[)$ applies only to interactions that take place in the time interval $[\eta_{d-1},\eta_d[$. Notice in particular that for a given d, the $\hat{\lambda}_{kgd}$ are obtained from the quantities $X_{ij}^{(d)}$ which correspond themselves to interaction counts during the time interval $[\eta_{d-1},\eta_d[$.

3.4 Segmentation

Up to a change of sign[4], the criterion (15), to be maximized with respect to η and D, has the general form used in change point detection problems (see equation (1) in Killick et al, 2012, for instance) and efficient algorithms used in this context can be considered for inference.

3.4.1 Dynamic programming

Let us first recall that those types of additive costs can be solved exactly via a form of dynamic programming (Jackson et al, 2005). Indeed, denoting $F(s,D)$ the maximal value of the criterion (15), when restricted to interactions that happen in $]0,s]$, and for at most D segments, we have, as in Jackson et al (2005); Killick

[4] In the framework of change point detection for time series, a *cost* function to be minimized is associated to each time segment. In contrast, we introduce a *gain* function to be consistent with the problem formulation used so far. However the two definitions are equivalent since the cost function can be thought as a gain function multiplied by -1.

et al (2012)

$$
\begin{aligned}
F(T,D) &= \max_{\eta, D' \leq D} \left[\sum_{d=1}^{D'} \left(\mathcal{G}([\eta_{d-1}, \eta_d[) - \frac{1}{2} \frac{K(K+1)}{2} \log \alpha \right) \right], \\
&= \max_{\zeta} \Bigg\{ \max_{\eta', D' \leq D-1} \left[\sum_{d=1}^{D'} \left(\mathcal{G}([\eta'_{d-1}, \eta'_d[) - \frac{1}{2} \frac{K(K+1)}{2} \log \alpha \right) \right] \\
&\qquad + \mathcal{G}([\zeta, T[) - \frac{1}{2} \frac{K(K+1)}{2} \log \alpha \Bigg\}, \\
&= \max_{\zeta} \left[F(\zeta, D-1) + \mathcal{G}([\zeta, T[) - \frac{1}{2} \frac{K(K+1)}{2} \log \alpha \right]. \qquad (17)
\end{aligned}
$$

Notice that in those equations, η corresponds to at most $D-1$ change points in $[0, T[$, ζ is a unique change point and η' corresponds to at most $D-2$ change points in $[0, \zeta[$.

3.4.2 Restriction on the change point locations

There are two limitations in equation (17): a maximal number of change points has to be specified and the optimization over ζ (i.e. over the position of a given change point) remains an open problem. While in theory this optimization is straightforward, the key point of the recurrence in equation (17) is the possibility of memorizing $F(\zeta, D-1)$ for all values of ζ. Indeed, the dynamic programming algorithm proceeds by computing and memorizing $F(\zeta, 1)$ for all ζ, and then computes and memorizes $F(\zeta, 2)$ using $F(\zeta, 1)$, etc.

Therefore, one has to reduce the search space for the change points to a finite set. In practice, this corresponds to introducing a grid of points which are *a priori* change point candidates:

$$\mathcal{P} = \{(t_0, \ldots, t_U) | 0 = t_0 < t_1 < \cdots < t_U = T, \quad U \in \mathbb{N}^*\}.$$

A natural choice for $\mathcal{P}$ is the set of all interaction times in $\mathcal{D}$, but other choices can be used, such as intermediate times between interactions (e.g., times of the form $\frac{\nu_{m+1}+\nu_m}{2}$) or arbitrary regular grids. Notice that choosing a grid immediately solves the problem of choosing the maximal value for D: it is exactly U.

The choice of $\mathcal{P}$ has several consequences. Firstly, the computational cost of the dynamic programming (with or without pruning) is directly linked to U (see below for details). Secondly, $\mathcal{P}$ acts as a minimal time resolution constraint. Thus, a high value of U allows to pinpoint change points very precisely but at a high computational cost, and vice versa. Therefore, both computational and expert considerations should be taken into account for choosing $\mathcal{P}$. If the computational load is acceptable, using the set of all interaction times offers a maximal resolution, but that might emphasize unneeded details.

The last consequence of the choice of $\mathcal{P}$ concerns the value of α in the penalized log likelihood (see equation (8)). According to the hypotheses on the generative model, we observe $N(N-1)/2$ conditionally independent NHPPP. Each of those processes is observed on the intervals $[t_u, t_{u+1}[$ via interaction counts. Those interaction counts are independent per the general hypothesis on NHPPP. Thus, we have $\alpha = UN(N-1)/2$ independent observations.

3.4.3 Pruned exact linear time

In this paper, we consider the pruned exact linear time (PELT) method of Killick et al (2012) originally developed for change point detection in univariate time series. Using the recursive decomposition in (17) for maximization has a cost of $\mathcal{O}(K^2U^2)$. For more details, we refer to Killick et al (2012). The key observation is that for each point t_u in $\mathcal{P}$ the gain of setting $t_{u'}$ as the last change point before t_u, has to be computed, for all $t_{u'} < t_u$. Fortunately, some points t_u can be pruned through the optimization routine. This is the principle of PELT and in practice it allows to speed up the exploration of the segmentation space. We use the following result whose general statement and formal proof are given in Killick et al (2012).

Again, consider $t_{u'}$ and t_u such that $t_{u'} < t_u$ and $t_{u'}$ is *not* the last change point before t_u. Then

$$\left(F(t_{u'}, U-1) + \mathcal{G}([t_{u'}, t_u[) - \frac{1}{2}\frac{K(K+1)}{2}\log\alpha\right) < F(t_u, U).$$

Moreover if $t_{u'}$ fulfils the condition

$$F(t_{u'}, U-1) + \mathcal{G}([t_{u'}, t_u[) < F(t_u, U),$$

then $t_{u'}$ can never be the optimal last change point prior to $t_{u''}$, for all $t_{u''} > t_u$. This statement is true for all gain functions satisfying the following condition

$$\mathcal{G}([t_{u'}, t_u[) + \mathcal{G}([t_u, t_{u''}[) \geq \mathcal{G}([t_{u'}, t_{u''}[), \qquad t_{u'} < t_u < t_{u''}. \tag{18}$$

Proposition 4 *The condition* (18) *is fulfilled by the gain function* $\mathcal{G}(\cdot)$ *we defined.*

Proof See appendix A.3.

The pseudocode Algorithm 1 illustrates how the PELT algorithm works.

3.5 Selection of K and initialization clusters

As any EM like approach, the algorithm proposed for inference depends on some initializations. Notice, however, that once initial values for τ and K are provided, the other model parameters $(D, \theta) = (D, \eta, \boldsymbol{\lambda}, \pi)$ are estimated in the maximization (M) step and those estimates are employed in the E step to obtain a better estimate of τ and so on until the criterion $f(q(Z), K, \eta, D, \lambda, \pi)$ (11) no longer increases (i.e. convergence). For a fixed value of K the initialization of τ can be obtained in several ways. For example a $N \times N$ adjacency matrix can be built by aggregating all the interactions over the time interval $[0, T]$, hence the entry (i, j) of this adjacency matrix corresponds to $M^{(i,j)}$, with the previous notations. Then, clustering algorithms like k-means, hierarchical clustering or spectral clustering can be used to get an estimate of Z. Finally the initial matrix τ is built such that τ_{ik} is one if $Z_i = k$, zero otherwise. Another method to initialize τ consists in applying a k-means clustering on the rows of the $N \times UN$ matrix corresponding

[5] Notice that in case the minimal partition is used, $X_{ij}^{(u)}$ is trivially equal to one for the pair (i, j) interacting at t_{u-1} and zero for all the other pairs.

Algorithm 1 PELT for dynamic SBM

Input:

A grid $0 = t_0 < t_1 < \cdots < t_U = T$.

An $(N \times N \times U)$ tensor X whose entry[5] (i, j, u) is $X_{ij}^{(u)}$

A matrix τ of variational probabilities.

The penalty $\alpha = UN(N-1)/2$.

A fixed positive number of clusters K.

The gain function $\mathcal{G}(.)$.

Initializations: $F(0) = \frac{1}{4}K(K+1)\log\alpha$, $cp(0) = NULL$, $R_1 = 0$.

for η^* in $1, \ldots, U$ **do**

Calculate $F(\eta^*) = \max_{\eta \in R_{\eta^*}} [F(\eta) + \mathcal{G}([t_{\eta+1}, t_{\eta^*}[) - F(0)]$.

Let $\overline{\eta}$ =argmax$_{\eta \in R_{\eta^*}} [F(\eta) + \mathcal{G}([t_{\eta+1}, t_{\eta^*}[) - F(0)]$.

Set $cp(\eta^*) = [cp(\overline{\eta}), \overline{\eta}]$.

Set $R_{\eta^*+1} = \{\eta \in R_{\eta^*} \cup \{\eta^*\} \quad | \quad F(\eta) + \mathcal{G}([t_{\eta+1}, t_{\eta^*}[) \geq F(\eta^*)\}$.

end for

Output: The change points stored in $cp(U)$.

to the mode-1 unfolding of the tensor X (see Kolda and Bader, 2009, for more details). In each experiment in Section 4, all the mentioned initialization techniques are attempted. The one leading to the highest finale value of the lower bound is finally retained.

So far, we have assumed that the number of clusters was fixed. However, in practice K is unknown and has to be inferred from the data. Again, we rely on the criterion defined in equation (11) which involves a penalization term over K. Recalling that the optimal number of segments is selected by the PELT procedure (see Section 3.3), the VEM algorithm described in this section is run for different values of K in $\{1, \ldots, K_{\max}\}$, for some fixed $K_{\max}$, and the value K maximizing the criterion is retained. The pseudocode Algorithm 2 summarizes the whole estimation routine.

4 Experiments

4.1 Simulated datasets

Some experiments on simulated data are carried out to test the proposed approach. Our model (called hereafter PELT-Dynamic SBM) is compared with the triclustering approach proposed in Guigourès et al (2012, 2015). The approach is referred to as MODL (while MODL is a more generic technique Boullé (2010)). As pointed out in the introduction, this method is non parametric and search for clusters of nodes and time segments. It is based on a combinatorial generative

Algorithm 2 VEM algorithm

Input:

A maximum number of clusters $K_{\max}$.

A set $\mathcal{D}$ of M interaction times.

The criterion $f(\cdot)$ in equation (11).

Initializations: Store $\leftarrow$ vector($K_{\max}$), Pmts $\leftarrow$ list($K_{\max}$)

for K in $1, \ldots, K_{\max}$ **do**

$\tau \leftarrow$ Some clustering algorithm

$\{D, \theta\} \leftarrow$ Maximization($\tau, \mathcal{D}$)

while $f(\cdot)$ increases **do**

$\tau \leftarrow$ Expectation($\mathcal{D}, D, \theta$)

$\{D, \theta\} \leftarrow$ Maximization($\tau, \mathcal{D}$)

end while

Store[K] $\leftarrow f(K, D, \tau, \theta)$

Pmts[K] $\leftarrow \{\tau, \theta\}$

end for

$K^* \leftarrow$ argmax(Store).

Output: The estimated parameters in Pmts[K^*].

model estimated via a maximum a posteriori approach. As our model, it has no user tunable parameter and is therefore fully automated.

4.1.1 First scenario

The experiments considered in this section are related to the simulation setup in Section III.A of Guigourès et al (2012). Each network generated is made of 40 nodes, grouped in four clusters: 5 vertices are in clusters 1 and 2, 10 vertices in cluster 3 and 20 vertices in cluster 4. The time interval $[0, 100]$ is split into four segments ($I_1 = [0, 20[$, $I_2 = [20, 30[$, $I_3 = [30, 60[$ and $I_4 = [60, 100[$) and each segment is associated with a specific interaction pattern between clusters, as illustrated in Figure 1. For each number of edges varying from 50 to 10000, 50

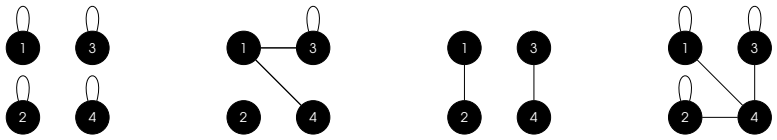

Fig. 1: Image graph for each time segment

dynamic graphs are sampled according to the following procedure:

1. A node and a random interaction time are sampled uniformly in $\{1, \dots, 40\}$ and $[0, 100]$, respectively. The vertex is then assigned to its cluster and the interaction time to its segment.
2. If the cluster of the selected vertex is connected to one or more clusters over the considered time segment (see Figure 1), a second vertex is sampled uniformly at random in the union of these clusters, and an edge is generated.
3. The first two steps are repeated until the desired number of edges is reached.
4. Finally, 30% of edges are rewired uniformly at random.

The only difference between the current setup and the one used in Guigourès et al (2012), is that the networks in the present paper are undirected. Notice that the generative process for those data is neither Poisson based model nor the combinatorial model used by MODL.

For estimation purposes, a regular grid $\mathcal{P}$ (Section 3.4.2) with unitary length time intervals is used for PELT-Dynamic SBM. Both algorithms (PELT-Dynamic SBM and MODL) are applied to the generated interaction data and results are assessed at an aggregated level (cluster numbers) and at a more refined level relying on the adjusted Rand indexes (Rand, 1971).

In Figure 2 the mean number of clusters K (respectively time segments, D)

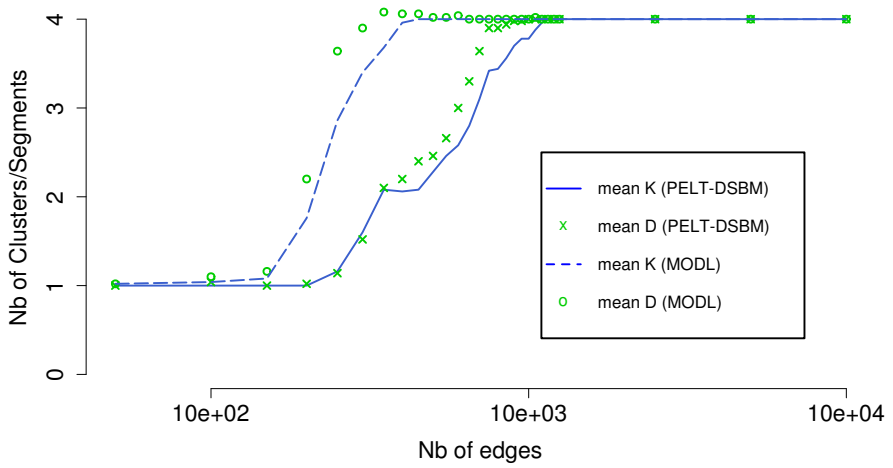

Fig. 2: The average number of clusters and time segments detected by MODL and PELT-Dynamic SBM versus number of sampled edges

found by the two methods is plotted in blue (green) as a function of the number of edges.

As it can be seen, MODL provides more accurate estimates of both K and D for a small number of edges, while PELT-Dynamic SBM needs denser networks to recover the true number of clusters and time segments. These results are confirmed in Figures 3 and 4.

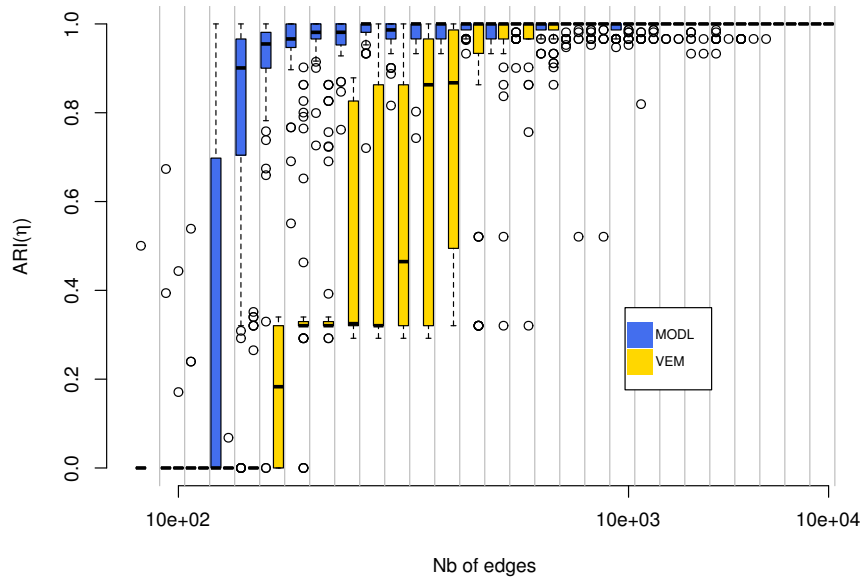

Fig. 3: ARIs for the change points (η)

For each number of edges, adjusted Rand indexes (ARI) are computed to asses the quality of the estimates provided for Z and η by the two models. For what concerning the change point locations, when no change point is detected the ARI is zero, conversely when $\hat{\eta}_1 = 20$, $\hat{\eta}_2 = 30$ and $\hat{\eta}_3 = 60$ the ARI is one. For each number of edges, two box-and-whiskers plots are produced, one for MODL (on the left hand side) and the other for our method (right hand side).

As pointed out above, the data are not generated according to our model or to MODL combinatorial one. However, our model is still parametric which might explain the less accurate estimates provided here as compared to MODL. We show in the following sections that when the data are generated with a model closer to our model, the results are quite different.

4.1.2 Second scenario

The graphs generated in the second simulation scenario are made of 75 nodes, grouped into two clusters and undirected interactions are simulated over the time interval $[0, 10]$. This interval is split into three segments $I_1 = [0, \eta_1[$, $I_2 = [\eta_1, \eta_2[$, $I_3 = [\eta_2, 10[$, where the change points η_1 as well as η_2 are set to 2.1 and 6.9, respectively. Interactions are sampled by thinning (Lewis and Shedler, 1979) according to the model we introduced in Section 2, based on the following intensity

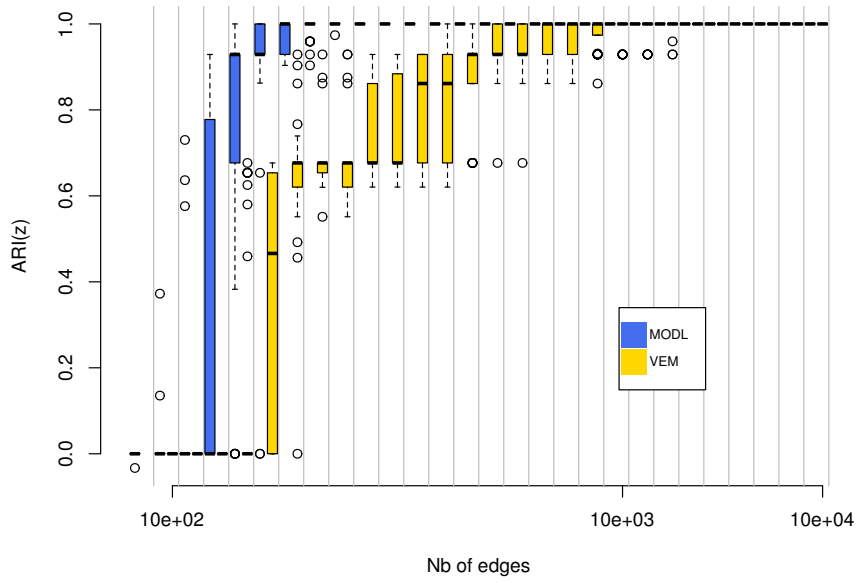

Fig. 4: ARIs for the cluster memberships (Z)

functions (IFs)

$$\lambda_{Z_i Z_j}(t) = \begin{cases} 0.1\mathbf{1}_{I_1}(t) + 0.2\mathbf{1}_{I_2}(t) + 0.05\mathbf{1}_{I_3}(t) & \text{if} \quad Z_i = Z_j \\ 0.05\mathbf{1}_{I_1}(t) + 0.1\mathbf{1}_{I_2}(t) + 0.025\mathbf{1}_{I_3}(t) & \text{if} \quad Z_i \neq Z_j, \end{cases}$$

for all $j > i$. Thus, the IFs define a persistent community structure (see Fortunato, 2010, for more details) through time in which the intensity of the interactions within clusters is twice the intensity of the interactions between clusters. The following sampling procedure is used to sampled 50 dynamic graphs:

1. Each vertex is assigned to one of the two clusters with probability $1/2$.
2. Interactions between each pair of nodes (i, j) are sampled according to the NHPPP with IF $\lambda_{Z_i, Z_j}(t)$.

Again, to introduce some noise in the data, 10% of edges are rewired. Considering this new setting, we aim at evaluating the clusters uncovered by our methodology along with the estimates of the change point locations η_1 and η_2. The grid $\mathcal{P}$ of all observed interactions in $\mathcal{D}$ was considered for inference. As mentioned in Section 3.4.2, this allows to pinpoint change points more accurately. The use of such a grid is made possible here because of the limited number of interactions generated.

Regarding the clustering task, we present in Figure 5 the results for MODL and a k-means algorithm applied on the adjacency matrix where all interactions over the time interval $[0, 10]$ are aggregated. Note that contrary to MODL and PELT-Dynamic SBM, k-means was provided with the true number of clusters.

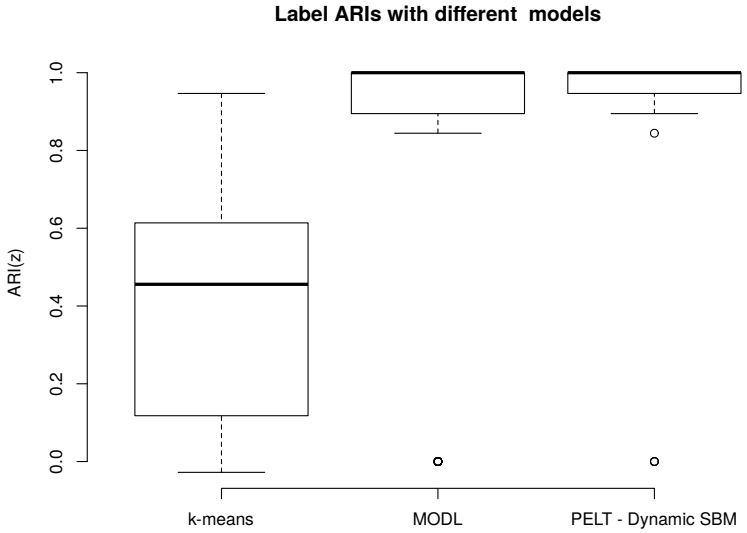

Fig. 5: Boxplots over 50 simulations of the ARIs for the clustering structures obtained by k-means, MODL and PELT-Dynamic SBM for scenario 2

Clearly, MODL and PELT-Dynamic SBM recover the clusters more accurately than k-means with PELT-Dynamic SBM slightly outperforming MODL: 3 null ARIs versus 8 and 33 unitary ARIs versus 26.

In terms of change point detection, due to the particular generative structure, MODL cannot recover any time cluster and considers the dynamic graph as stationary. Indeed in order to avoid parametric assumptions, MODL uses rank based modeling for numerical values. In the triclustering context of Guigourès et al (2012, 2015) this means that interaction times are replaced interaction orders. This explains why MODL is blind to the time structure in the present scenario.

Conversely PELT-Dynamic SBM always retrieves the right number of change points in the data. The change point estimates can be observed in Figure 6 and Kernel density estimates are plotted along with the true change points as red vertical lines. This illustrates the accuracy of the proposed estimation procedure and its superiority to MODL in this situation.

4.1.3 Third scenario

This section aims at illustrating that aggregating interactions can lead to an important loss of information. Thus, each graph generated is made of 100 nodes clustered in two groups, with 50 nodes each. Moreover, the time interval $[0, 12]$ is split into four segments of equal size delimited by the change points $\eta_1 = 3$, $\eta_2 = 6$ and $\eta_3 = 9$. Finally, interactions are sampled by thinning according to the

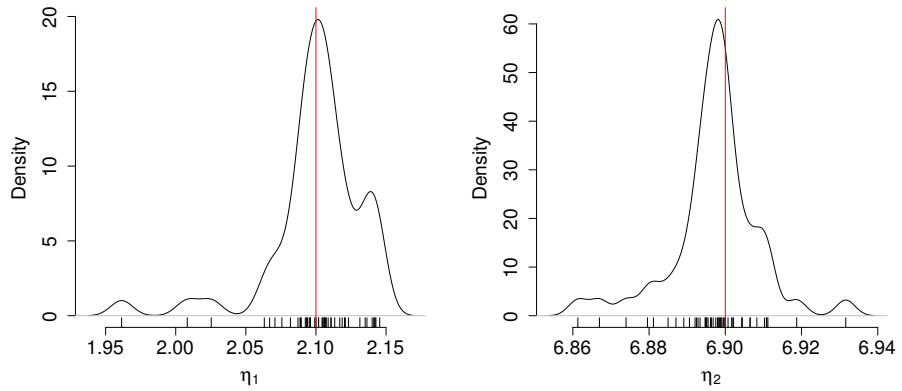

Fig. 6: Kernel density estimates over 50 simulations of the change points estimated by PELT-Dynamic SBM, for scenario 2. The true values of η_1 and η_2 are given by the red vertical lines

following IFs

$$\lambda_{Z_i Z_j}(t) = \begin{cases} 0.05\mathbf{1}_{I_1}(t) + 0.1\mathbf{1}_{I_2}(t) + 0.05\mathbf{1}_{I_3}(t) + 0.1\mathbf{1}_{I_4} & \text{if} \quad Z_i = Z_j \\ 0.1\mathbf{1}_{I_1}(t) + 0.05\mathbf{1}_{I_2}(t) + 0.1\mathbf{1}_{I_3}(t) + 0.05\mathbf{1}_{I_4} & \text{if} \quad Z_i \neq Z_j, \end{cases}$$

for all $j > i$ and I_d denotes the d-th segment. By construction, integrating the IFs over $[0, 12]$ leads to

$$\Lambda_{11}(T) = \Lambda_{12}(T) = \Lambda_{21}(T) = \Lambda_{22}(T) = 3.8.$$

Thus, the average number of interactions is the same for all pairs of clusters which makes clusters indistinguishable when aggregating the interactions over the time interval. As for the previous simulation scenarios, 50 dynamic graphs are generated and 10% of edges of each graph are rewired uniformly at random. MODL as well as PELT-Dynamic SBM are then used to uncover clusters of vertices and to segment the time interval. A regular grid $\mathcal{P}$ with unitary length time intervals is considered. The results are presented in Figure 7 as ARIs for both the change points and the cluster memberships. In this context, PELT-Dynamic SBM provides more reliable estimates than MODL, which fails in retrieving any cluster or temporal structures, in most cases. It's a form of extreme blindness to the whole structure of the data induced by the blindness to the temporal structure. Note that the results for PELT-Dynamic SBM are similar when relying on a grid with a higher time resolution.

4.1.4 Summary

The scenarios studied in this section show that PELT-Dynamic SBM is able to recover both the cluster and the temporal structure of dynamic graphs without the need for a strong prior aggregation of the interactions (a minimal aggregation is used for computational reasons in some of the experiments). Contrarily to MODL,

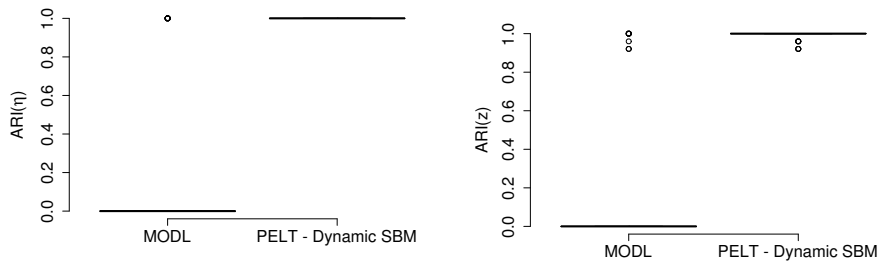

Fig. 7: Adjusted Rand indexes for the change points η (left hand side) and the cluster memberships Z (right hand side)

PELT-Dynamic SBM discovers structural changes that are only based on a modification of the intensity of the interaction. Thus both approaches have different use cases. In particular, the temporal structure of the data is more easily captured by our model than by MODL.

4.2 Real data

We now focus on a cycle hire usage dataset, collected in London and publicly available at `http://api-portal.tfl-gov.uk/docs`. It characterizes the interactions that occurred on September 9, 2015, between the Santander stations. The dynamic network considered is made of 735 nodes and 64514 undirected edges (with no self loops), collected with a minute precision over the day. One edge connecting nodes i and j at a given time corresponds to a cycle hire from station i to station j or, conversely from station j to station i. To limit the computational burden of the segmentation step, we relied on a regular grid $\mathcal{P}$ corresponding to 96 time intervals of 15 minutes. The PELT-Dynamic SBM was then applied several times, for different values of K from 0 to 20. The highest value of the criterion f (equation 11) was attained for $K = 11$ clusters and $D = 5$ time segments. MODL does not find any temporal structure in those data despite obvious changes in the aggregated intensities (see Figure 9). Results are presented in Figure 8. The Santander stations are plotted on a London map[6] different symbols/colors corresponds to different clusters identified by the model. Interestingly (and as expected), generally nearby stations are placed in the same cluster and the geographical distance between them plays a key role. In Figure 9, an histogram of the interaction times in the whole network is provided. Two peaks are visible around 8.30 and 18.30. The five segments detected by the methodology are delimited by the vertical red lines in the figure. We observe a strong alignment between the histogram and the segments estimated. In particular, the two observed picks are clearly associated with segment 2 and 3, respectively. We now concentrate on analyzing the results

[6] Map data are available from `http://www.openstreetmap.org` and copyrighted OpenStreetMap contributors.

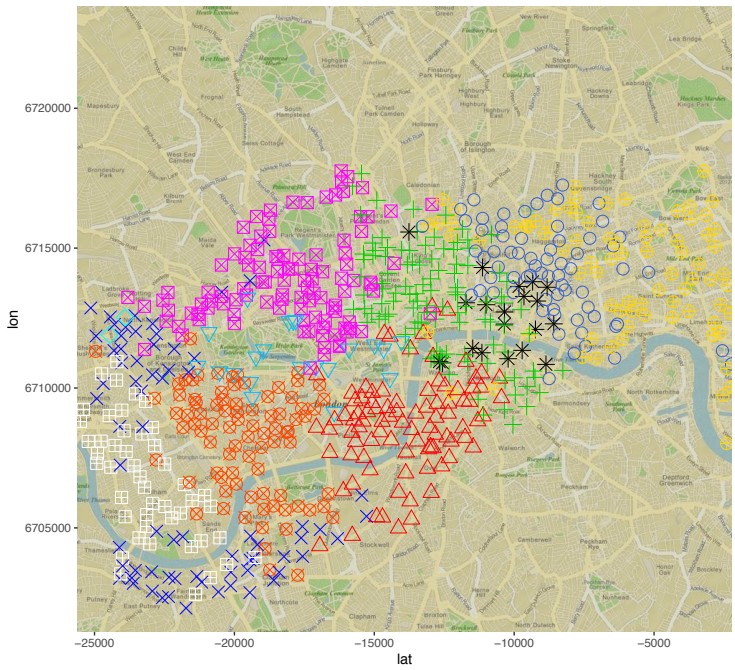

Fig. 8: The 11 clusters found by PELT-Dynamic SBM represented here with 11 different symbols/colors on the left hand side

for cluster 3 (identified by the symbol + in Figure 8) which is made of stations from central London. This cluster actually corresponds to a community with higher values for the IF within the cluster than for all the IFs describing the levels of interactions with other clusters. Figure 10 gives some examples of intra and inter IFs related to cluster 3. The results are presented for clusters 1 (⊠) and 7 (◯) which are geographically adjacent to cluster 3, and for clusters 4 (×) and 10 (⊗), which are not. Overall, as mentioned already, the intra IF is higher. However, this figure also highlight a temporal pattern. Indeed, it appears the inter IFs for the adjacent clusters are higher in the morning and in the evening than for the rest of the day. A somehow similar pattern is observed for clusters 4 and 10 but with much lower values in general. This is coherent with cycles being hired more often to go to a station close geographically.

In order to highlight another feature of the proposed methodology, we now give some results regarding cluster 8. More specifically, Table 1 provides the aggregated interactions between clusters 7 (◯) and 8 (∗), over the segments uncovered. In the first segment, 16 interactions occurred between vertices of cluster 8 and 47 between vertices of cluster 8 and vertices of cluster 1. Thus, cluster 8 has a disassortative connectivity pattern with less intra edges. Conversely, in segment 2, they are more intra edges (502) within cluster 8 than between clusters 8 and 1. This corresponds

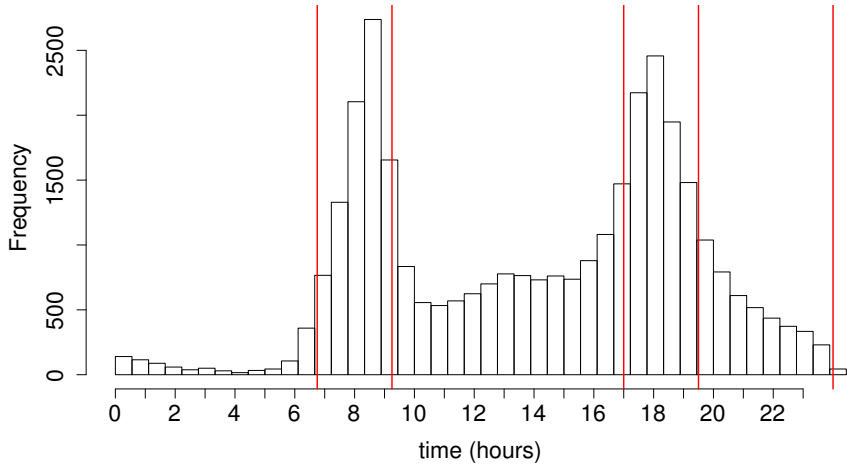

Fig. 9: An histogram shows how frequent interactions (cycle hires) are during the day. The vertical lines correspond to the estimated change points

to a community pattern. Looking through all the segments, we can observe that the community and disassortative pattern for cluster 8 alternates through time. Thus, clustering the vertices while detecting change points in the intensity of the interactions is mandatory here because the connectivity patterns of the data set keep changing. Therefore, any method aggregating the data would miss important information present in the data. In conclusion, the uncovered clusters as well as

Table 1: Aggregated interactions for clusters 1 (◯) and 8 (∗) during the five segments uncovered. On the main diagonal of each table, the numbers of interactions within clusters are reported: for cluster 1 on the left/top, for cluster 8 on the right/bottom. Interactions between clusters are outside the main diagonal. Community structure for cluster 8 is indicated in blue. Disassortative structure for cluster 8 is indicated in red.

104	47
47	16

0.00-6.45

742	441
441	502

6.45-9.45

1106	368
368	338

9.45-17

912	419
419	984

17-19.45

572	128
128	108

19.45-0.00

the change point locations seems to be meaningful on a ground truth basis and the PELT-Dynamic SBM proved to be fit to uncover interaction patterns that could not easily be detected by other static or dynamic clustering algorithms.

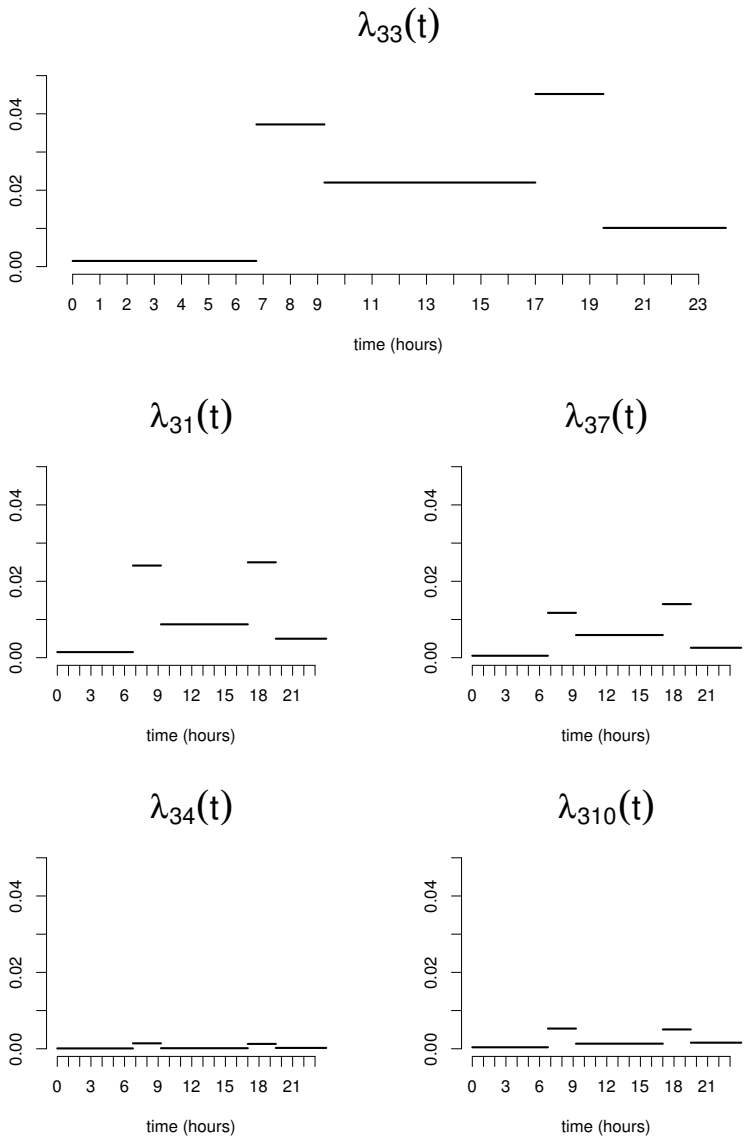

Fig. 10: Estimated IFs for groups $(3, \cdot)$. Groups 1 and 7 are geographically adjacent to cluster 3 whereas 4 and 10 are not

5 Conclusion

In this paper, we proposed a new model for dynamic networks, based on conditional non homogeneous Poisson point processes. The model assumes that vertices belong to unknown clusters whose number and composition are fixed in time but unknown. The intensity functions are then dependent on the clusters of nodes

and assumed to be stepwise, with common discontinuity points. In order to rely on the exact pruned linear time method to infer the location and number of the discontinuity points, we considered a specific inference framework. Thus, we approximated a marginal log likelihood through a penalized BIC like term, for which we derived a variational lower bound. In this context, we used a variational expectation maximization algorithm for the inference. The methodology we proposed allows to do the segmentation of the intensity functions and the clustering of the nodes simultaneously. Experiments on toy data sets and real data were used to illustrate the relevance of the method.

A Proofs

A.1 Proof of Proposition 1

Proof Notice, first, that the central factor on the r.h.s. of the equality in equation (3) can be written as

$$\prod_{m=1}^{M} \lambda_{Z_{i_m} Z_{j_m}}(\nu_m) = \prod_{m=1}^{M} \prod_{j>i}^{N} \left(\lambda_{Z_i Z_j}(\nu_m)\right)^{\mathbf{1}_{\mathcal{A}^{(i,j)}}(\nu_m)},$$

where $\mathbf{1}_{\mathcal{G}}(\cdot)$ is the indicator function on a set $\mathcal{G}$ and $\mathcal{A}^{(ij)}$ has been defined in Equation (1). By inverting the product on the right hand side, because of the indicator function we get

$$\begin{aligned}\prod_{j>i}^{N} \prod_{m=1}^{M} \left(\lambda_{Z_i Z_j}(\nu_m)\right)^{\mathbf{1}_{\mathcal{A}^{(i,j)}}(\nu_m)} &= \prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \lambda_{Z_i Z_j}(\nu_m^{(i,j)}) \\ &= \prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \left[\prod_{k,g} (\lambda_{kg}(\nu_m^{(i,j)}))^{Z_{ik} Z_{jg}}\right],\end{aligned}$$

where $\nu_m^{(i,j)}$ are the interaction times in the set $\mathcal{A}^{(i,j)}$, whose cardinality is $M^{(i,j)}$. Thanks to the equation (5), the following holds

$$\begin{aligned}\prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \left[\prod_{k,g} (\lambda_{kg}(\nu_m^{(i,j)}))^{Z_{ik} Z_{jg}}\right] &= \prod_{j>i}^{N} \prod_{m=1}^{M^{(i,j)}} \left[\prod_{k,g} \prod_{d=1}^{D} \lambda_{kgd}^{Z_{ik} Z_{jg} \mathbf{1}_{[\eta_{d-1}, \eta_d[}(\nu_m^{(i,j)})}\right] \\ &= \prod_{k,g} \prod_{d=1}^{D} \lambda_{kgd}^{\sum_{j>i}^{N} Z_{ik} Z_{jg} (N^{(i,j)}(\eta_d) - N^{(i,j)}(\eta_{d-1}))}. \quad (19)\end{aligned}$$

Note that the last equality employs the definition of counting process

$$M^{(i,j)}(t) = \sum_{m=1}^{M^{(i,j)}} \mathbf{1}_{]0,t]}(\nu_m^{(i,j)}).$$

By replacing (19) into the equation (3) and using that

$$\Lambda_{Z_i Z_j}(T) = \sum_{k,g}^{K} \Lambda_{kg}(T) Z_{ik} Z_{jg} = \sum_{d=1}^{D} \sum_{k,g}^{K} \lambda_{kgd} \Delta_d Z_{ik} Z_{jg},$$

it suffices to take the logarithm of the likelihood and the proposition is proven.

A.2 Proof of Proposition 3

Proof The following objective function is taken into account

$$\mathcal{L}\left(q(Z); K, \eta, D, \boldsymbol{\lambda}, \pi\right) + \sum_{i=1}^{N} l_i \left(\sum_{k=1}^{K} \tau_{ik} - 1 \right).$$

This function has to be maximized with respect to both τ and the N Lagrange multipliers $l_1, \ldots, l_N$, introduced to take into account the normality of the lines of τ. The most difficult step consists in taking the partial derivative of the objective function with respect to $\tau_{i_0 k_0}$. We first focus on those terms of $\mathcal{L}(\cdot)$ depending on d

$$Q(\tau, \theta) := -\sum_{d=1}^{D} \sum_{k,g}^{K} \left(\lambda_{kgd} \Delta_d \left(\sum_{i=1}^{N} \sum_{j>i}^{N} \tau_{ik} \tau_{jg} \right) - \log(\lambda_{kgd}) \left(\sum_{i=1}^{N} \sum_{j>i}^{N} \tau_{ik} \tau_{jg} X_{ij}^{(d)} \right) \right).$$

Hence

$$\begin{aligned}
\frac{\partial Q(\tau, \theta)}{\partial \tau_{i_0 k_0}} &= \sum_{d=1}^{D} \frac{\partial}{\partial \tau_{i_0 k_0}} \left(-\sum_{i=1}^{N} \sum_{j>i}^{N} \sum_{k,g}^{K} \tau_{ik} \tau_{jg} \lambda_{kgd} \Delta_d \right) \\
&+ \sum_{d=1}^{D} \frac{\partial}{\partial \tau_{i_0 k_0}} \left(\sum_{i=1}^{N} \sum_{j>i}^{N} \sum_{k,g}^{K} \tau_{ik} \tau_{jg} X_{ij}^{(d)} \log(\lambda_{kgd}) \right) \\
&= -\sum_{d=1}^{D} \left[\sum_{j>i_0}^{N} \sum_{g=1}^{K} \tau_{jg} \Delta_d \lambda_{k_0 g d} + \sum_{j<i_0}^{N} \sum_{g=1}^{K} \tau_{jg} \Delta_d \lambda_{g k_0 d} \right] \\
&+ \sum_{d=1}^{D} \left[\sum_{j>i_0}^{N} \sum_{g=1}^{K} \tau_{jg} X_{i_0 j}^{(d)} \log(\lambda_{k_0 g d}) + \sum_{j<i_0}^{N} \sum_{g=1}^{K} \tau_{jg} X_{j i_0}^{(d)} \log(\lambda_{g k_0 d}) \right] \\
&= -\sum_{d=1}^{D} \left[\sum_{j \neq i_0}^{N} \sum_{g=1}^{K} \tau_{jg} \Delta_d \lambda_{k_0 g d} - \sum_{j \neq i_0}^{N} \sum_{g=1}^{K} \tau_{jg} X_{i_0 j}^{(d)} \log(\lambda_{k_0 g d}) \right],
\end{aligned}$$

where the last equality comes from the symmetry (i.e. interactions are undirected) of the frontal slices of tensors X and $\boldsymbol{\lambda}$. Notice that the last term in the above equation is the function inside the exponential in Proposition 3. The remaining terms of $\mathcal{L}(\cdot)$, not involving d, can be differentiated straightforward. Imposing the partial derivatives of $\mathcal{L}(\cdot)$ equal to zero and using the above equation leads to the following system

$$\begin{cases} \log(\tau_{i_0 k_0}) = \log(\pi_{k_0}) - \frac{\partial Q(\tau, \theta)}{\partial \tau_{i_0 k_0}} + l_{i_0} - 1 \\ \sum_{k_0=1}^{K} \tau_{i_0 k_0} = 1 \end{cases} \qquad \forall (i_0, k_0).$$

The solution is obtained straightforward after some manipulations and this concludes the proof.

A.3 Proof of Proposition 4

Proof The following definitions are introduced to keep the notation uncluttered

$$\begin{aligned}
S_{kg} &:= \sum_{j>i}^{N} \tau_{ik} \tau_{jg} \\
Y_{kg}^{[s,t[} &:= \sum_{j>i}^{N} \tau_{ik} \tau_{jg} (M^{(i,j)}(t) - M^{(i,j)}(s)) \qquad \forall s < t.
\end{aligned}$$

Moreover, for every $t_{u_e} < t_{u_f} < t_{u_g}$, the following short hand notation is used

$$\Delta^{e,f} := t_{u_f} - t_{u_e}$$

and similarly for $\Delta^{f,g}$ and $\Delta^{e,g}$. Hence, we get

$$\begin{aligned}
\mathcal{G}([t_{u_e}, t_{u_g}[) &= \sum_{k,g} \left[\max_{\lambda_{kg}^{e,g} \in]0,+\infty[} \left(-\lambda_{kg}^{e,g} \Delta^{e,g} S_{kg} + \log(\lambda_{kg}^{e,g}) Y_{kg}^{[t_{u_e}, t_{u_g}[} \right) \right] \\
&= \sum_{k,g} \left[\max_{\lambda_{kg}^{e,g} \in]0,+\infty[} \left(-\lambda_{kg}^{e,g} \Delta^{e,f} S_{kg} + \log(\lambda_{kg}^{e,g}) Y_{kg}^{[t_{u_e}, t_{u_f}[} \right) \right] \\
&+ \sum_{k,g} \left[\max_{\lambda_{kg}^{e,g} \in]0,+\infty[} \left(-\lambda_{kg}^{e,g} \Delta^{f,g} S_{kg} + \log(\lambda_{kg}^{e,g}) Y_{kg}^{[t_{u_f}, t_{u_g}[} \right) \right] \\
&\leq \sum_{k,g} \left[\max_{\lambda_{kg}^{e,f} \in]0,+\infty[} \left(-\lambda_{kg}^{e,f} \Delta^{e,f} S_{kg} + \log(\lambda_{kg}^{e,f}) Y_{kg}^{[t_{u_e}, t_{u_f}[} \right) \right] \\
&+ \sum_{k,g} \left[\max_{\lambda_{kg}^{f,g} \in]0,+\infty[} \left(-\lambda_{kg}^{f,g} \Delta^{f,g} S_{kg} + \log(\lambda_{k,g}^{f,g}) Y_{kg}^{[t_{u_f}, t_{u_g}[} \right) \right] \\
&= \mathcal{G}([t_{u_e}, t_{u_f}[) + \mathcal{G}([t_{u_f}, t_{u_g}[),
\end{aligned}$$

where the first and the last equalities come from the definition of $\mathcal{G}(\cdot)$. This concludes the proof.

References

A P Dempster DBR N M Laird (1977) Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society Series B (Methodological) 39(1):1–38, URL http://www.jstor.org/stable/2984875

Airoldi E, Blei D, Fienberg S, Xing E (2008) Mixed membership stochastic blockmodels. The Journal of Machine Learning Research 9:1981–2014

Boullé M (2010) Data grid models for preparation and modeling in supervised learning, Microtome

Casteigts A, Flocchini P, Quattrociocchi W, Santoro N (2012) Time-varying graphs and dynamic networks. International Journal of Parallel, Emergent and Distributed Systems 27(5):387–408, DOI 10.1080/17445760.2012.668546

Corneli M, Latouche P, Rossi F (2016a) Block modelling in dynamic networks with non-homogeneous poisson processes and exact ICL. Social Network Analysis and Mining 6(1):1–14, DOI 10.1007/s13278-016-0368-3, URL http://dx.doi.org/10.1007/s13278-016-0368-3

Corneli M, Latouche P, Rossi F (2016b) Exact ICL maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. Neurocomputing 192:81 – 91, DOI 10.1016/j.neucom.2016.02.031

Daley DJ, Vere-Jones D (2003) An introduction to the theory of point processes: volume I: Elementary Theory and Methods. Springer

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

Dubois C, Butts C, Smyth P (2013) Stochastic blockmodelling of relational event dynamics. In: International Conference on Artificial Intelligence and Statistics, vol 31 of the Journal of Machine Learning Research Proceedings, pp 238–246

Fortunato S (2010) Community detection in graphs. Physics Reports 486(3-5):75 – 174

Friel N, Rastelli R, Wyse J, Raftery AE (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, DOI 10.1073/pnas.1606295113, http://www.pnas.org/content/113/24/6629.full.pdf

Guigourès R, Boullé M, Rossi F (2012) A triclustering approach for time evolving graphs. In: Co-clustering and Applications, IEEE 12th International Conference on Data Mining Workshops (ICDMW 2012), Brussels, Belgium, pp 115–122, DOI 10.1109/ICDMW.2012.61

Guigourès R, Boullé M, Rossi F (2015) Discovering patterns in time-varying graphs: a triclustering approach. Advances in Data Analysis and Classification pp 1–28, DOI 10.1007/s11634-015-0218-6, URL http://dx.doi.org/10.1007/s11634-015-0218-6

Ho Q, Song L, Xing EP (2011) Evolving cluster mixed-membership blockmodel for time-evolving networks. In: International Conference on Artificial Intelligence and Statistics, pp 342–350

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

Jackson B, Sargle J, Barnes D, Arabhi S, Alt A, Giomousis P, Gwin E, Sangtrakulcharoen P, Tan L, Tsai T (2005) An algorithm for optimal partitioning of data on an interval. Signal Processing Letters pp 105–108

Jernite Y, Latouche P, Bouveyron C, Rivera P, Jegou L, Lamassé S (2014) The random subgraph model for the analysis of an acclesiastical network in merovingian gaul. Annals of Applied Statistics 8(1):55–74

Killick R, Fearnhead P, Eckley IA (2012) Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association 107(500):1590–1598, DOI 10.1080/01621459.2012.737745, URL http://dx.doi.org/10.1080/01621459.2012.737745, http://dx.doi.org/10.1080/01621459.2012.737745

Kim M, Leskovec J (2013) Nonparametric multi-group membership model for dynamic networks. In: Advances in Neural Information Processing Systems (25), pp 1385–1393

Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM review 51(3):455–500

Lewis P, Shedler G (1979) Simulation of nonhomogeneous poison processes by thinning. Naval Res Logist Quart 26(3):403–413

Matias C, Miele V (2016) Statistical clustering of temporal networks through a dynamic stochastic block model. The Journal of the Royal Statistical Society: Series B, to appear

Matias C, Rebafka T, Villers F (2015) Estimation and clustering in a semiparametric Poisson process stochastic block model for longitudinal networks. ArXiv e-prints 1512.07075

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

Rand WM (1971) Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association 66(336):846–850

Sarkar P, Moore AW (2005) Dynamic social network analysis using latent space models. ACM SIGKDD Explorations Newsletter 7(2):31–40

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

Xing EP, Fu W, Song L (2010) A state-space mixed membership blockmodel for dynamic network tomography. Ann Appl Stat 4(2):535–566, DOI 10.1214/09-AOAS311

Xu KS, Hero III AO (2013) Dynamic stochastic blockmodels: Statistical models for time-evolving networks. In: Social Computing, Behavioral-Cultural Modeling and Prediction, Springer, pp 201–210

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

Zreik R, Latouche P, Bouveyron C (2016) The dynamic random subgraph model for the clustering of evolving networks. Computational Statistics pp 1–33, DOI 10.1007/s00180-016-0655-5, URL http://dx.doi.org/10.1007/s00180-016-0655-5