Poisson degree corrected dynamic stochastic block model

Paul Riverain, Simon Fossier, Mohamed Nadif

To cite this version:

Paul Riverain, Simon Fossier, Mohamed Nadif. Poisson degree corrected dynamic stochastic block model. Advances in Data Analysis and Classification, 2022, 10.1007/s11634-022-00492-9 . hal-03668567

HAL Id: hal-03668567
https://hal.science/hal-03668567v1

Submitted on 15 May 2022

Advances in Data Analysis and Classification manuscript No.
(will be inserted by the editor)

Poisson degree corrected dynamic Stochastic Block Model

Paul Riverain · Simon Fossier ·
Mohamed Nadif

Received: date / Accepted: date

Abstract Stochastic Block Model (SBM) provides a statistical tool for modeling and clustering network data. In this paper, we propose an extension of this model for discrete-time dynamic networks that takes into account the variability in node degrees, allowing us to model a broader class of networks. We develop a probabilistic model that generates temporal graphs with a dynamic cluster structure and time-dependent degree corrections for each node. Thanks to these degree corrections, the nodes can have variable in- and out-degrees, allowing us to model complex cluster structures as well as interactions that decrease or increase over time. We compare the proposed model to a model without degree correction and highlight its advantages in the case of inhomogenous degree distributions in the clusters and in the recovery of unstable cluster dynamics. We propose an inference procedure based on Variational Expectation-Maximization (VEM) that also provides the means to estimate the time-dependent degree corrections. Extensive experiments on simulated and real datasets confirm the benefits of our approach and show the effectiveness of the proposed algorithm.

Keywords Stochastic Block Model · dynamic graph clustering · Expectation Maximization · degree correction

Paul Riverain
Université de Paris, CNRS, Centre Borelli UMR 9010
Thales Research and Technology France
E-mail: paul.riverain@etu.u-paris.fr

Simon Fossier
Thales Research and Technology France
1 Avenue Augustin Fresnel, 91120 Palaiseau, France
E-mail: simon.fossier@thalesgroup.com

Mohamed Nadif
Université de Paris, CNRS, Centre Borelli UMR 9010
45 rue des Saints Pères, 75006 Paris
E-mail: mohamed.nadif@u-paris.fr

Mathematics Subject Classification (2020) 62H30

1 Introduction

The Stochastic Block Model (SBM) (Wang and Wong, 1987) has been extensively studied over the past years for the statistical modeling of relations among objects (Snijders and Nowicki, 1997; Daudin et al., 2008; Airoldi et al., 2008; Mariadassou et al., 2010; Abbe, 2017). It provides a powerful tool for clustering network data with latent variables and can be seen as an extension of the mixture models for relational data. However, real-world networks are often time-dependent (e.g. social networks, transportation networks, citation networks), making of great interest the extension of this model to dynamic data.

Dynamic graphs can be represented in a discrete-time setup as a series of adjacency matrices. We focus on the task of clustering the nodes of the graph at each time step, where we aim at finding meaningful structure in each snapshot of the graph, while preserving the coherence in the dynamics of the structure (Fu et al., 2009; Xu and Hero, 2014; Sewell and Chen, 2016; Matias and Miele, 2017). Classical SBM, applied frame by frame, are here unsatisfactory, since they would miss the temporal structure of the network. Therefore, in this paper, we focus on discrete-time dynamic multigraphs – which can also be seen as temporal graph with non-negative integer weights – and take into account two specificities of dynamic real-world networks.

First, in such networks, the number of interactions can greatly vary over time, and nodes can join or leave the network. For instance, in a transportation network, the number of trips greatly varies from opening to rush hour and some stations are not served by the rest of the network at certain times.

Second, real-world networks exhibit high degree heterogeneity, where the degrees approximately follow a power-law distribution (Barabási and Albert, 1999). Classical SBM tends to cluster together nodes with similar degrees, degree corrections are necessary to model graphs with heterogeneous degrees (Karrer and Newman, 2011).

The main contributions of the paper are as follows:

- We propose a model with dynamic degree corrections, for the clustering of dynamic networks. This allows to model more complex (and arguably more realistic) graph structures and their dynamics, while keeping the interpretation of the clusters simple (see Sect. 3.1).
- We propose a method to estimate the time-dependent degree corrections by making use of their regularity (see Sect. 3.3).
- We present in detail an efficient VEM algorithm for this model, and provide the proofs of the main results (see Sect. 4).
- We illustrate the advantages of applying dynamic degree correction to both simulated and real-world data (see Sect. 5) and demonstrate the efficiency of the proposed algorithm by applying the algorithm to real-world dynamic

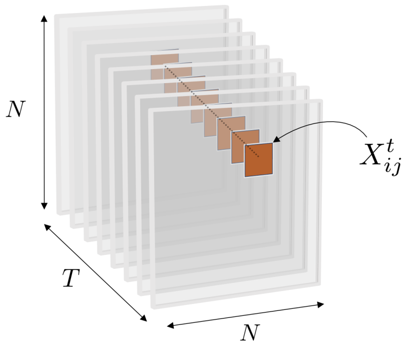


Fig. 1: Matrix representation of the dynamic graph

graphs with more than 700 nodes as well as dynamic graphs with more than 500 snapshots (see Sect. 5.2).

Notations Vectors, matrices and tensors are denoted with boldface letters. We consider a discrete-time temporal network with N nodes in interaction, represented by a series of adjacency matrices $\boldsymbol{X} = (\boldsymbol{X}^t)_{t\in\{1,\ldots,T\}}$, where T corresponds to the number of snapshots. $\boldsymbol{X}^t$ is the weighted adjacency matrix of the graph, where $X^t_{ij} \in \mathbb{N}$ (see Fig. 1). The graph is directed and without self-loops ($X^t_{ii} = 0$). We note $X^t_{.j} = \sum_{i=1}^N X^t_{ij}$ and $X^t_{i.} = \sum_{j=1}^N X^t_{ij}$ the in- and out-degrees. Let $(Z^t_i)_{t\in\{1,\ldots,T\}}$ be the discrete latent process associated with node i, and K denote the number of clusters. If node i belongs to cluster k at time t, we write $Z^t_i = k$, or equivalently $Z^t_{ik} = 1$. The sums and the products relating to time steps, nodes and clusters will be subscripted respectively by the letters t, i, j, k and ℓ, with implicit limits of variations, so $\sum_t$, $\sum_i$, $\sum_j$, $\sum_k$ and $\sum_\ell$ will denote respectively $\sum_{t=1}^T$ $\sum_{i=1}^N$, $\sum_{j=1}^N$, $\sum_{k=1}^K$ and $\sum_{\ell=1}^K$.

2 Related work

In the purpose of modeling real-world dynamic networks with SBM, we first focus on models that account for the degree heterogeneity in the network, then we present existing works on dynamic SBM.

Degree correction in SBM We first consider the following general model for a static SBM with edges distributed according to

$$X_{ij}|(Z_i = k, Z_j = \ell) \sim \mathcal{F}(\lambda_{ijk\ell})$$

where $\mathcal{F}$ is a scalar parametric distribution and $\lambda_{ijk\ell}$ corresponds to the parameter of the edge X_{ij} between clusters k and ℓ. In Karrer and Newman (2011), the authors propose to take into account the variability in nodes' degree in a network by multiplying a node-specific degree correction parameter

$\boldsymbol{\mu}$ to the cluster connectivity matrix $\boldsymbol{\gamma}$ such that $\lambda_{ijk\ell} = \mu_i \mu_j \gamma_{k\ell}$. For identifiability constraints, μ_i is normalized such that $\sum_i \mu_i Z_{ik} = 1$ for each cluster k, and can then be interpreted as the probability that an edge in the cluster of the node i is connected to i. The authors then show that, in contrast to classical SBM where only the block's expected values are preserved, this model also preserves the nodes degrees. The proposed normalization has an intuitive interpretation, but still depends on the latent factors $\boldsymbol{Z}$, which makes it difficult to use in the context of the EM algorithm and thus requires heuristic methods for inference. This idea is developed in Qiao et al. (2017) in order to model networks closer to scale-free networks for Bernoulli-distributed edges, where $\lambda_{ijk\ell} = \gamma_{k\ell}^{1+\mu_i+\mu_j}$, where μ_i is called the degree-decay variable and is chosen to follow an exponential prior. Under the assumption that the node degree is mostly due to the contribution of nodes inside its cluster (assortative mixing), the authors show that the degree of a node converges to a random variable that approximately follows a power law distribution. However, the inference is complex and the authors thus have to rely on MAP estimation and gradient ascent for the estimation of the parameters. In the context of the Latent Block Model (Ailem et al., 2017b,a; Govaert and Nadif, 2018) propose to add a row effect and a column effect $\lambda_{ijk\ell} = \mu_i \nu_j \gamma_{k\ell}$ and normalize μ_i and ν_j such that $\mathbb{E}(X_{i.}) = \mu_i$ and $\mathbb{E}(X_{.j}) = \nu_j$; the model is simply replaced by $\lambda_{ijk\ell} = X_{i.} X_{.j} \gamma_{k\ell}$. However, this normalization is not applicable in the context of the SBM because of a dependency structure stricter than in LBM.

Dynamic extensions of SBM Using a continuous approach, Matias et al. (2018) consider the set of all timestamped interactions between nodes without any aggregation as a realisation of a point process. A non-homogeneous Poisson counting process is associated to the point process, with a SBM latent structure that imposes a common intensity function in each block. The authors propose a VEM algorithm with two different M-steps for the estimation of the infinite-dimensional intensity functions based on an histogram method and a kernel method. Corneli et al. (2018) consider the same model but assume that the intensities are constant on some unobserved intervals that are determined as the changepoints of the intensity functions.

In a discrete-time approach, Corneli et al. (2016) propose to extend SBM to dynamic graphs with discrete time steps by seeking clusters of nodes that do not evolve over time and clusters of time periods. In this approach, the edges of the graph are modeled with non homogeneous Poisson processes and the intensity function for a given block is considered constant on each clustered time interval. For the inference, the authors rely on a greedy maximization of the Integrated Classification Likelihood (ICL) (Biernacki et al., 2000) in a Bayesian context. Another approach is presented in Matias and Miele (2017), in an algorithm referred to as `dynsbm`, where the latent variables can evolve over time, based on independent and identically distributed Markov chains. Thus, the nodes of the graph can change cluster over time, and the initial distribution over the clusters as well as the transition probabilities between clusters of consecutive time steps are estimated. The authors show that, in order to ensure the identifiability of the model: either the clusters are defined by a constant set of nodes and the clusters can have different characteristics over time, or the nodes can change cluster membership over time but the clusters must have some time-independent characteristics. For the inference, the authors follow a variational EM approach. In Rastelli et al. (2018), the authors rely on a model relatively similar to Matias and Miele (2017) but in a Bayesian context. It differs in that the parameters of the blocks' distributions are fixed over time. The authors must then rely on possibly empty clusters at some time steps to provide the model with flexibility. The inference and the model selection are then realized as in Corneli et al. (2016). In the two previous models, the intra-cluster connectivities are time-independent and, since there is no degree correction, the degree distribution inside a cluster is homogeneous. Consequently, if one wants to model decreasing or growing interactions, the model will require a large number of clusters which, for most snapshots, will be empty.

3 The proposed model

3.1 Definition of the model

Model Let $(Z_i^t)_i$ be N independent homogeneous Markov chains on the set $\{1, \dots, K\}$ with initial multinomial distribution of parameter $\boldsymbol{\alpha}$, transition matrix $\boldsymbol{\pi}$. The weights of the directed edges of the graph X_{ij}^t are sampled according to a Poisson distribution, whose intensity is determined by 3 factors: the margin term μ_i^t related to the head node i at time t, the margin term ν_j^t of the tail node j at time t and the constant block term $\gamma_{k\ell}$ corresponding to interactions between nodes of cluster k and nodes of cluster ℓ. The model can be written :

$$Z_i^1 \sim \text{Multinomial}(1; \alpha_1, \dots, \alpha_K) \tag{1a}$$

$$Z_i^{t+1} | Z_i^t = k \sim \text{Multinomial}(1; \pi_{k1}, \dots, \pi_{kK}) \tag{1b}$$

$$\text{for } i \neq j,\ X_{ij}^t | \big(Z_i^t = k, Z_j^t = \ell\big) \sim \text{Poisson}(\mu_i^t \nu_j^t \gamma_{k\ell}). \tag{1c}$$

When using this model in a clustering context, we seek a constant cluster structure $\boldsymbol{\gamma}$ and a simple cluster dynamic given by $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$. The use of a constant connectivity matrix $\boldsymbol{\gamma}$ allows for a simple interpretation of the clusters (compared to the dynamic connectivity matrix of `dynsbm`), while the dynamic degree corrections $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ gives the model the flexibility required in a dynamic context by preserving the nodes in- and out-degrees.

These degree corrections can also be related to Banerjee et al. (2007), in the context of co-clustering – i.e. the simultaneous partition of rows and columns of a data matrix; see for instance (Schepers et al., 2017; Bock, 2020; Affeldt et al., 2021). Banerjee et al. (2007) show that there only exists six co-clustering bases, which can be sorted by level of complexity, depending on the number of statistics preserved by the model. The authors experimentally show that more complex bases are required as the clusters separability decreases. Here, clustering using the degree corrections corresponds to using a more complex co-clustering base than without degree correction.

Considering a variable number of nodes As proposed in Matias and Miele (2017), the latent processes can be adapted in order to take into account nodes entering or leaving the network at certain time steps. This allows us to deal with zero-degree nodes as well as nodes with a very low degree at a given time step. For instance, in Sect. 5.2, in order to focus on the main interactions in the network, we consider that nodes with a degree below 5 at a given time step are absent. Let $V = \{1, \dots, N\}$ be the set of all nodes present over the T snapshots, and let $V^t \subset V$ be the set of nodes present at time t. Let $\{0, \dots, K\}$ be the set of clusters, where the cluster 0 at time t contains the nodes considered absent at time t, i.e. $\overline{V^t}$. For $i \in V$, for k, $\ell \in \{1, \dots, K\}$, we consider the transition probabilities (2) from and to cluster 0, where $\mathbb{1}$ equals 1 if its argument is true and 0 otherwise.

$$P(Z_i^t = 0 | Z_i^{t-1} = 0) = \mathbb{1}(i \in \overline{V^t}), \tag{2a}$$
$$P(Z_i^t = 0 | Z_i^{t-1} = k) = \mathbb{1}(i \in \overline{V^t}), \tag{2b}$$
$$P(Z_i^t = k | Z_i^{t-1} = 0) = \alpha_k \mathbb{1}(i \in V^t), \tag{2c}$$
$$P(Z_i^t = \ell | Z_i^{t-1} = k) = \pi_{k\ell} \mathbb{1}(i \in V^t). \tag{2d}$$

Hence $(Z_i^t)_{t \in \{1,\dots,T\}}$ forms an inhomogeneous Markov chain on $\{0, \dots, K\}$, with deterministic transitions to the cluster 0, and transitions from cluster 0 modeled with the same parameter $\boldsymbol{\alpha}$ as the distribution at $t = 0$.

Complete data log-likelihood Let $\phi(.; \lambda)$ be the probability mass function of a Poisson distribution of parameter λ. Let $A^t = \overline{V}^{t-1} \cap V^t$ be the set of nodes appearing at time t and $S^t = V^{t-1} \cap V^t$ be the set of nodes staying between time t and $t+1$. The complete data log-likelihood of the model is:

$$\begin{aligned} \log P(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\theta}) = & \sum_{i \in V^1} \sum_k Z_{ik}^1 \log \alpha_k + \sum_{t \geq 2} \sum_{i \in A^t} \sum_k Z_{ik}^t \log \alpha_k \\ & + \sum_{t \geq 2} \sum_{i \in S^t} \sum_{k\ell} Z_{ik}^{t-1} Z_{i\ell}^t \log \pi_{k\ell} \\ & + \sum_t \sum_{\substack{i,j \in V^t \\ i \neq j}} \sum_{k\ell} Z_{ik}^t Z_{j\ell}^t \log \phi(X_{ij}^t; \mu_i^t \nu_j^t \gamma_{k\ell}). \end{aligned} \tag{3}$$

3.2 Inference

Variational Inference We seek to jointly infer the latent factors $\boldsymbol{Z}$ and obtain a maximum likelihood estimate the model parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\gamma}\}$. As directly maximizing the likelihood of the model with the EM algorithm is not possible, we instead rely on the interpretation of EM proposed in Neal and Hinton (1998) and use, as in Matias and Miele (2017), a structured variational approximation (Ghahramani and Jordan, 1997) of the distribution over the latent variables (Govaert and Nadif, 2005; Daudin et al., 2008). We use a variational distribution Q among the distributions over the latent space that factorize as N independent inhomogeneous Markov chains $Q(\boldsymbol{Z}) = \prod_i Q(Z_i^1) \prod_{t\geq 2} Q(Z_i^t | Z_i^{t-1})$ and define the variational parameters: $q(i,k) = Q(Z_i^1 = k)$ and $q(t,i,k,\ell) = Q(Z_i^t = \ell | Z_i^{t-1} = k)$. We can then recursively compute the marginal probabilities:

$$Q(Z_i^t = k) = q(t,i,k) = \sum_{k'} q(t-1,i,k')q(t,i,k',k). \tag{4}$$

For the Variational EM algorithm, we optimize a lower-bound $F(\boldsymbol{q},\boldsymbol{\theta})$ of the log-likelihood $\ell(\boldsymbol{\theta})$ of the model, where $\boldsymbol{q}$ denotes the vector of variational parameters:

$$\ell(\boldsymbol{\theta}) \geq F(\boldsymbol{q},\boldsymbol{\theta}) = \mathbb{E}_Q\big(\log P(\boldsymbol{X},\boldsymbol{Z};\boldsymbol{\theta}) - \log Q(\boldsymbol{Z})\big). \tag{5}$$

It can be shown (see Appendix A) that:

$$\begin{aligned} F(\boldsymbol{q},\boldsymbol{\theta}) = &\sum_{i\in V^1}\sum_k q(i,k)\log\frac{\alpha_k}{q(i,k)} + \sum_{t\geq 2}\sum_{i\in A^t}\sum_k q(t,i,k)\log\frac{\alpha_k}{q(t,i,k)} \\ &+ \sum_{t\geq 2}\sum_{i\in S^t}\sum_{k\ell} q(t-1,i,k)q(t,i,k,\ell)\log\frac{\pi_{k\ell}}{q(t,i,k,\ell)} \\ &+ \sum_t \sum_{\substack{i,j\in V^t\\ i\neq j}}\sum_{k\ell} q(t,i,k)q(t,j,\ell)\log\phi(X_{ij}^t;\, \mu_i^t\nu_j^t\gamma_{k\ell}). \end{aligned} \tag{6}$$

Expectation step For the expectation step (E-step), we update the variational transition probabilities with (7a, 7c, 7b). These formulas are not obtained by the exact maximization of $F(\boldsymbol{q},\boldsymbol{\theta})$ w.r.t. $\boldsymbol{q}$ because of the heavy computations involved (for this, see (Bartolucci and Pandolfi, 2020)) but are rather obtained by optimizing F for each t with a coordinate ascent algorithm where the transition and marginal probabilities at time step $t' \neq t$ are fixed (see Appendix B).

Let $\phi_{ijk\ell}^t = \phi(X_{ij}^t;\, \mu_i^t\nu_j^t\gamma_{k\ell})$ be the likelihood of edge (i,j) at time t in block (k,ℓ) and let $d_{ik}^t = \sum_\ell q(t,i,k,\ell)\big(\log q(t,i,k,\ell) - \log\pi_{k\ell}\big)$, the Kullback-Leibler divergence between the variational transition probabilities from cluster k and the model transition probabilities from cluster k. Regarding the last time step $t = T$, we also define $d_{ik}^{T+1} = 0$.

$$\forall i \in V^1,\ q(i,k) \propto \alpha_k \exp\left(-d_{ik}^2\right) \prod_{j \neq i} \prod_{\ell} \left(\phi_{ijk\ell}^1 \phi_{ji\ell k}^1\right)^{q(j,\ell)}, \tag{7a}$$

$$\forall i \in A^t,\ q(t,i,k) \propto \alpha_k \exp\left(-d_{ik}^{t+1}\right) \prod_{j \neq i} \prod_{\ell} \left(\phi_{ijk\ell}^t \phi_{ji\ell k}^t\right)^{q(t,j,\ell)}, \tag{7b}$$

$$\forall i \in S^t,\ q(t,i,k,\ell) \propto \pi_{k\ell} \exp\left(-d_{i\ell}^{t+1}\right) \prod_{j \neq i} \prod_{\ell'} \left(\phi_{ij\ell\ell'}^t \phi_{ji\ell'\ell}^t\right)^{q(t,j,\ell')}. \tag{7c}$$

The term $\exp\left(-d_{i\ell}^{t+1}\right)$ in (7c) penalizes the transition of node i to state ℓ at time t if the transition from cluster ℓ at time $t+1$ is not in accordance with the estimated cluster dynamics $\boldsymbol{\pi}$. E-step (7b) is directly expressed in terms of marginal probabilities since nodes that appear at a given time step (i.e. nodes that are present at the current time step and were absent at the previous time step) are necessarily in cluster 0 at the previous time step: for $t \geq 2$ and for $i \in \overline{V}^{t-1} \cap V^t$, we have $q(t,i,k) = q(t,i,0,k)$.

Maximization step To update the parameters in the maximization step (M-step), we use an ECM algorithm (Meng and Rubin, 1993), replacing the maximization of $F(\boldsymbol{q}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ by a series of conditional maximizations. The CM-steps are given in (8). We first update the mixture proportions $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$, and $\boldsymbol{\gamma}$ since they only depend on $\boldsymbol{q}$. Next, we update $\boldsymbol{\mu}$ given $\boldsymbol{\gamma}$ and $\boldsymbol{\nu}$, and finally $\boldsymbol{\nu}$ given $\boldsymbol{\gamma}$ and $\boldsymbol{\mu}$ (see Algorithm 1).

$$\widehat{\alpha}_k \propto \sum_i q(i,k) + \sum_{t \geq 2} \sum_{i \in A^t} q(t,i,k), \tag{8a}$$

$$\widehat{\pi}_{k\ell} \propto \sum_{t \geq 2} \sum_{i \in S^t} q(t-1,i,k) q(t,i,k,\ell), \tag{8b}$$

$$\widehat{\gamma}_{k\ell} = \frac{\sum_t \sum_{ij|i \neq j} q(t,i,k) q(t,j,\ell) X_{ij}^t}{\sum_t \sum_{ij|i \neq j} q(t,i,k) q(t,j,\ell) \mu_i^t \nu_j^t}, \tag{8c}$$

$$\widehat{\mu}_i^t = \frac{X_{i.}^t}{\sum_{j \neq i} \sum_{k\ell} q(t,i,k) q(t,j,\ell) \nu_j^t \gamma_{k\ell}}, \tag{8d}$$

$$\widehat{\nu}_j^t = \frac{X_{.j}^t}{\sum_{i \neq j} \sum_{k\ell} q(t,i,k) q(t,j,\ell) \mu_i^t \gamma_{k\ell}}. \tag{8e}$$

We can notice that M-step for μ_i^t corresponds to setting the expected in-degree (with current parameters) to the observed in-degree. In fact, for a Poisson distribution, we have:

$$\mathbb{E}_Q\Big(\mathbb{E}\big(X_{i.}^t | \boldsymbol{Z}; \boldsymbol{\theta}\big)\Big) = \mu_i^t \sum_{j \neq i} \sum_{k\ell} q(t,i,k) q(t,j,\ell) \nu_j^t \gamma_{k\ell}.$$

3.3 Temporal smoothing for continuous margins

The margins provide the model with great flexibility (see Sect. 5), but makes the inference more dependent on the initial parameters. However, in many applications, the degrees of a node are correlated between two consecutive time steps. Consequently, the degree correction parameters should exhibit some regularity. We thus propose to use a method we call *temporal smoothing* to better estimate the margins using their regularity. This method consists in keeping the margins constant over time in a first phase of the VEM algorithm ($\mu_i^t = \mu_i$ and $\nu_j^t = \nu_j$), and then to progressively release this constraint during the next iterations of the algorithm. This is equivalent to using the estimated parameters of a model with constant margins to initialize a model with less constrained margins. This procedure is iterated until the model without constrained margins (i.e. μ_i^t and ν_j^t are estimated as in (8d, 8e)) is initialized.

In order to release the constraint of constant margins progressively, it should first be noticed that, if in the model μ_i^t is constant over time, then its M-step resolves to $\sum_t n_{\mu_i}^t / \sum_t d_{\mu_i}^t$, where $n_{\mu_i}^t$ and $d_{\mu_i}^t$ are respectively the numerator and denominator of the unconstrained M-step (8d). We consider $\boldsymbol{n}_{\mu_i} = (n_{\mu_i}^1, \dots, n_{\mu_i}^T)^\intercal$ and $\boldsymbol{d}_{\mu_i} = (d_{\mu_i}^1, \dots, d_{\mu_i}^T)^\intercal$ and propose a temporal filtering S_{W^τ} for the numerators and denominators $\boldsymbol{n}_{\mu_i}$ and $\boldsymbol{d}_{\mu_i}$, where $\tau \in [0, 1]$ controls the level of smoothing. The filtering is written $S_{W^\tau}(\boldsymbol{n}_{\mu_i})_t = \sum_{t'} W_{tt'}^\tau n_{\mu_i}^{t'}$, where $W_{tt'}^\tau$ is given by (9)(see Fig. 2).

$$W_{tt'}^\tau \propto \exp\left(-\frac{\tau}{1-\tau}(t'-t)^2\right), \ \sum_{t'} W_{tt'} = 1. \tag{9}$$

This expression of the weights has the following properties:

$$W_{tt'}^\tau \xrightarrow[\tau \to 0]{} \frac{1}{T}, \ W_{tt'}^\tau \xrightarrow[\tau \to 1]{} \mathbb{1}(t' = t).$$

We then estimate μ_i^t and ν_j^t with smoothing parameter τ according to (10).

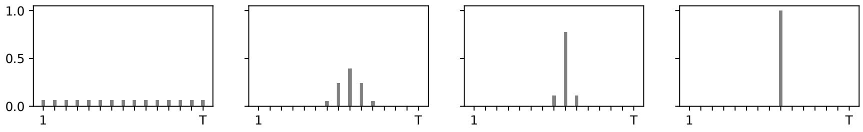


Fig. 2: Weights $W_{tt'}^\tau$ of the temporal smoothing as a function of t' for $t = 8$, for increasing amount of smoothing $\tau \in \{0, 1/3, 2/3, 1\}$.

$$^\tau\widehat{\mu}_i^t = \frac{S_{W^\tau}(\boldsymbol{n}_{\mu_i})_t}{S_{W^\tau}(\boldsymbol{d}_{\mu_i})_t}, \ ^\tau\widehat{\nu}_j^t = \frac{S_{W^\tau}(\boldsymbol{n}_{\nu_j})_t}{S_{W^\tau}(\boldsymbol{d}_{\nu_j})_t}. \tag{10}$$

The same principle can also be applied successfully to the time-varying connectivity matrix of `dynsbm` (data not shown). This approach also benefits the estimation of the mixture parameters. In fact, we start the inference with a model with a reduced capacity that will have to first focus on the estimation of $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ before reaching its full capacity.

Here are some guidelines for the choice of smoothing schedule. The key is to aim for the objective criterion $F(\boldsymbol{q}^{(c)}, \widehat{\boldsymbol{\theta}}^{(c)})$ to increase smoothly between two consecutive values of the smoothing parameter τ. The length of the schedule should also be increased according to the difficulty of the clustering problem and the number of snapshots considered. After experimentation, we opted for a sigmoidal schedule with 10 steps for our experiments.

4 Algorithm

4.1 Initialization

As the proposed model relies on the principles of EM for inference, it finds local optima of the likelihood function, thus making it dependent on its initialization. In static SBM, a natural idea to initialize the parameters is to rely on `k-means` applied to the rows of the adjacency matrix to obtain a partition of the nodes from which we apply M-step. In the dynamic case, in order to avoid local label-switching issues, we follow the method of Matias and Miele (2017) and Rastelli et al. (2018) that consists in applying a clustering algorithm on the $N \times TN$ matrix formed by the concatenation of the rows of each adjacency matrix and setting each node to the same cluster over time. We initialize $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ with a first M-step and set $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ to 1 for all nodes and time steps. Since at initialization each node is in the same cluster over time, initializing $\boldsymbol{\pi}$ with M-step would result in $\boldsymbol{\pi} = \boldsymbol{I}_K$, leading the algorithm to favor stable partitions (i.e. without inter-cluster transitions) during its first iterations, instead of exploring more complex cluster dynamics. Thus, we propose to initialize $\boldsymbol{\pi}$ with $\pi_{kk} = \pi_0 \in [0,1]$ and for $k \neq \ell$, $\pi_{k\ell} = 1 - \frac{\pi_0}{K-1}$, where π_0 can be chosen according to our prior knowledge on the cluster dynamic (in our experiments, we set $\pi_0 = 0.7$). The algorithm is initialized several times for a given initial partition, and, at each new initialization, a fraction of the nodes of each snapshot is reassigned to random clusters. We then select the partition of the nodes and estimated parameters that produces the highest objective function F.

4.2 Intermediate variables for the E and M steps

To reduce the computing time, note that the E-M steps can be simplified by considering the expression of the Poisson density function. For this, we rely on the intermediate variables (11) before E-step.

$$\begin{aligned}
A^t_{i\ell'} &= \sum_{j\neq i} q(t,j,\ell')X^t_{ij}, & B^t_{i\ell'} &= \sum_{j\neq i} q(t,j,\ell')X^t_{ji},\\
M^t_k &= \sum_i q(t,i,k)\mu^t_i, & N^t_\ell &= \sum_j q(t,j,\ell)\nu^t_j, \qquad (11)\\
M^t_{jk} &= M^t_k - q(t,j,k)\mu^t_j, & N^t_{i\ell} &= N^t_\ell - q(t,i,\ell)\nu^t_i.
\end{aligned}$$

Using (11), E-step (7c) can be written, up to a constant:

$$\begin{aligned}
\forall i \in S^t,\ \log q(t,i,k,\ell) = \log \pi_{k\ell} - d^{t+1}_{i\ell} &+ \sum_{\ell'} \left(\gamma_{\ell\ell'} A^t_{i\ell'} + \gamma_{\ell'\ell} B^t_{i\ell'}\right)\\
&- \mu^t_i \sum_{\ell'} \gamma_{\ell'\ell} N^t_{i\ell'} - \nu^t_i \sum_{\ell'} \gamma_{\ell\ell'} M^t_{i\ell'}. \qquad (12)
\end{aligned}$$

The M-steps for $\boldsymbol{\gamma}$, $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ can be written:

$$\begin{aligned}
\widehat{\gamma}_{k\ell} &= \frac{\sum_{ti} q(t,i,k)A^t_{i\ell}}{\sum_t M^t_k N^t_\ell},\\
\widehat{\mu}^t_i = \frac{X^t_{i.}}{\sum_{k\ell} q(t,i,k) N^t_{i\ell}\gamma_{k\ell}}&,\ \widehat{\nu}^t_j = \frac{X^t_{.j}}{\sum_{k\ell} q(t,j,\ell) M^t_{jk}\gamma_{k\ell}}. \qquad (13)
\end{aligned}$$

4.3 Algorithm and computational complexity

The proposed algorithm, which we refer to as `pdc-dsbm`, is presented in Algorithm 1. The computational bottleneck of the algorithm is the computation of the intermediate variables and the expectation step of EM (12). Let I be the total number of iterations of the VEM algorithm, the time complexity is $O(ITN^2K^2)$. Regarding the space complexity of the algorithm, the use of the intermediate quantities $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{M}$ and $\boldsymbol{N}$ allows E-step to require only $O(TNK)$ space. The space complexity of the algorithm is thus determined by $\boldsymbol{X}$ and $\boldsymbol{q}$, which makes it $O(TN^2 + TNK^2)$.

5 Experimental results

In this section, we first evaluate `pdc-dsbm` on synthetic data, where we control the complexity of the clustering task w.r.t. the class transitions and the class overlap. We show experimentally the benefits of the margins and compare the performances of the proposed algorithm with different underlying models (with and without margins). We then apply `pdc-dsbm` to two real-world datasets corresponding to two very different transportation networks and show the coherence of the obtained results.

Algorithm 1 `pdc-dsbm`: Poisson degree corrected dynamic SBM

```
Input:
  – Adjacency matrices $\boldsymbol{X} \in \mathbb{N}^{T\times N\times N}$,
  – Number of clusters $K$,
  – Schedule for the smoothing parameter $\tau$,
    starting from 0 and increasing to 1,
  – Stopping criterion $\epsilon > 0$,
  – Intra-cluster initial transition probability $\pi_0$,
  – Initial partition of the nodes ${}^0\boldsymbol{Z}$ obtained following Sect. 3.2.
Initialization:
    Set the loop counter $c = 0$ and $\Delta^{(0)} > \epsilon$
    Set the smoothing parameter $\tau_{(0)} = 0$
    Initialize $\boldsymbol{q}$ with the initial partition: $\forall k$, $q^{(0)}(t,i,k,\ell) = 1$ if ${}^0Z^t_{i\ell} = 1$
    Update the marginal probabilities with (4)
    Set $\widehat{\boldsymbol{\mu}}^{(0)}$ and $\widehat{\boldsymbol{\nu}}^{(0)}$ to $\mathbf{1}$
    Initialize $\widehat{\boldsymbol{\alpha}}^{(0)}$, $\widehat{\boldsymbol{\gamma}}^{(0)}$ with M-step (8a, 8c)
    Initialize $\widehat{\boldsymbol{\pi}}^{(0)}$ with $\pi_0$ following Sect. 3.2,
    Initialize $\widehat{\boldsymbol{\mu}}^{(0)}$ and $\widehat{\boldsymbol{\nu}}^{(0)}$ with M-step (10)
repeat
    repeat
        increment $c$
        Compute the intermediate variables (11)
        E-step:
            for $t = 1, \dots, T$ do
                Update $q^{(c)}(i,k)$ with (7a)
                Update $q^{(c)}(t,i,k,\ell)$ with (7c) if $i \in S^t$
                Update $q^{(c)}(t,i,k)$ with (7b) if $i \in A^t$
                Update marginal $q^{(c)}(t,i,k)$ with (4)
            end for
        M-step:
            Update $\widehat{\boldsymbol{\alpha}}^{(c)}$ and $\widehat{\boldsymbol{\pi}}^{(c)}$ with (8a) and (8b)
            Update $\widehat{\boldsymbol{\gamma}}^{(c)}$ with (8c) using ${}^{\tau_{(c-1)}}\widehat{\boldsymbol{\mu}}^{(c-1)}$ and ${}^{\tau_{(c-1)}}\widehat{\boldsymbol{\nu}}^{(c-1)}$
            Update ${}^{\tau_{(c)}}\widehat{\boldsymbol{\mu}}^{(c)}$ with (10) using $\widehat{\boldsymbol{\gamma}}^{(c)}$ and ${}^{\tau_{(c-1)}}\widehat{\boldsymbol{\nu}}^{(c-1)}$
            Update ${}^{\tau_{(c)}}\widehat{\boldsymbol{\nu}}^{(c)}$ with (10) using $\widehat{\boldsymbol{\gamma}}^{(c)}$ and ${}^{\tau_{(c)}}\widehat{\boldsymbol{\mu}}^{(c)}$
        Compute $\Delta^{(c)} = |F(\boldsymbol{q}^{(c)}, \widehat{\boldsymbol{\theta}}^{(c)}) - F(\boldsymbol{q}^{(c-1)}, \widehat{\boldsymbol{\theta}}^{(c-1)})|$
    until $\Delta^{(c)} < \epsilon$
    Update $\tau_{(c)}$ according to the smoothing schedule
until $\tau_{(c)} = 1$ and $\Delta^{(c)} < \epsilon$
Return: $\boldsymbol{q}^{(c)}, \widehat{\boldsymbol{\theta}}^{(c)}$
```

Compared algorithms In Matias and Miele (2017), the authors showed through intensive experiments – in the context of Bernoulli distributions – that `dynsbm` outperforms the algorithm presented in Yang et al. (2011) (in its offline version), which gives similar performances compared to the algorithm of Xu and Hero (2014) but with slower inference. In Yang et al. (2011), the authors showed that their algorithm outperforms the ones of Lin et al. (2009) and Chi et al. (2007). Consequently, we choose to compare our algorithm — `pdc-dsbm` — to `dynsbm` for Poisson distributions[1]. The algorithm `dynsbm` for complete

[1] We could not compare `pdc-dsbm` directly to the authors' algorithm because their R package `dynsbm V0.7` only implements Bernoulli, Multinomial and Gaussian distributions, so we had to re-implement it for Poisson distributions.

multigraphs corresponds to a model $X_{ij}^t | \left(Z_i^t = k, Z_j^t = \ell\right) \sim \mathcal{P}(\gamma_{k\ell}^t)$ with $\gamma_{kk}^t = \gamma_{kk}$ and with a VEM algorithm, without temporal smoothing.

Comparing partitions with global and local metrics In a dynamic clustering context, in addition to the global label-switching problem – i.e. we can only recover the dynamic clusters up to a permutation – we also face the more challenging but essential problem of local label-switching, that consists in correctly matching clusters of nodes over time. In order to evaluate the partitions obtained by the model when one has a reference partition, we consider the Adjusted Rand Score (ARI) (Hubert and Arabie, 1985). This metric can be computed for each snapshot of the graph, however this is not informative of the matching of the clusters over time. Consequently, we use two different versions of these metrics: the average of the metric on each *local* snapshot, i.e. T partitions of N points, and a *global* metric, considering one partition of TN points.

5.1 Experiments on synthetic data

Sampling data from the model In the experiments, we consider directed graphs with self-loops sampled with $T = 15$ time steps and $N = 200$ nodes. In the following, we will denote the model without margins – used in `dynsbm` – as M_- and the model we propose as M_+. For a given parameter $\boldsymbol{\theta}$ we sample the complete data $(\boldsymbol{X}, \boldsymbol{Z})$ either with M_- or M_+. For each set of parameters, we sample 20 samples of complete data. Then, for each complete data, we apply 20 times the two algorithms we compare, each time with the same initialization not to favor any model. When generating the margins, their values at $t = 0$ μ_i^0 and ν_j^0 are sampled from $\{1, \ldots, 100\}$ with a power law $P(k) \propto k^{-\lambda}$ with $\lambda = 1.5$, resulting in skewed margins. The dynamic margins are sampled using a first order auto-regressive process with Gaussian noise, where $\mu_i^t \sim \mathcal{N}(a\mu_i^{t-1} + b; \sigma_{\text{AR}})$, where $a = 1.15$, $b = \mu_i^0(1 - a)$ and $\sigma_{\text{AR}} = 0.05$. We add a zero-mean Gaussian noise to each edge sampled from the model, and we control the class overlap with the standard deviation of the noise (σ_{M_-} and σ_{M_+}). Edges weights are rounded down and edges with negative weights are clipped to zero. For a given set of complete data, the compared algorithms are initialized with the same initial partition, and, at each new initialization, a fraction $f = 0.1$ of the nodes of each snapshot is reassigned to random clusters. The experiments are carried out with $K = 3$ clusters and $\boldsymbol{\alpha} = (0.5, 0.3, 0.2)^\intercal$, for 4 transition matrices: $\boldsymbol{\pi}^0$ (diagonal) and $\boldsymbol{\pi}^+$, $\boldsymbol{\pi}^{++}$, $\boldsymbol{\pi}^{+++}$ with respectively 10 %, 25 % and 40 % of inter-cluster transitions between two time steps an equal off-diagonal terms, for $\sigma_{M_-} \in \{4, 5, 6\}$ and[2] $\sigma_{M_+} \in \{40, 50, 60\}$ respectively corresponding to overlaps σ_+, σ_{++} and σ_{+++}, and for $\boldsymbol{\gamma} = \left(\begin{smallmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{smallmatrix}\right)$. In order to measure select the noise levels to obtain different degrees of overlap

[2] More noise needs to be added for M_+ since margins greater than one spread the classes apart

of the clusters, we first project the rows and columns of the adjacency matrix with Correspondence Analysis (CA) (Benzecri, 1973; Greenacre, 2007) onto $\mathbb{R}^N$, then we measure the linear separability of the clusters with Linear Discriminant Analysis. As described in Govaert and Nadif (2013, 2018), there is a link between a Poisson model and mutual information on the one hand, and between the mutual information and the χ^2 criterion on the other. This justifies the choice of CA, which uses the χ^2 metric (where PCA relies on the variance). To measure the linear separability of the clusters, we compute the ratio between the inter-cluster variance and the total variance of the data projected onto each of the factorial axes. Each of the ratios is in [0, 1], a ratio of 1 meaning that the intra-cluster variance on the factorial axis is null (i.e. the clusters are linearly separable), and a ratio of 0 meaning that the centers of gravity of each cluster are projected onto the same point on the factorial axis. These values, reported on Table 1, show that the three connectivity matrices lead to clusters that are decreasingly linearly separable. The projection of the data onto the factorial axes of order greater than 3 cannot separate the classes since most inertia (measured by χ^2) is explained by the first two axes.

Table 1: Ratios of the inter-cluster variance and the total variance of the data projected by CA onto each of the factorial axes for the rows and the columns of the adjacency matrix sampled using M_+, with different noise levels and equal mixing proportions. The results are averaged over 100 matrices.

		Factorial axis	
Class separability	Noise level	1	2
Rows	σ_+	0.85 ± 0.01	0.45 ± 0.04
	σ_{++}	0.79 ± 0.02	0.33 ± 0.04
	σ_{+++}	0.71 ± 0.02	0.27 ± 0.04
Columns	σ_+	0.85 ± 0.01	0.47 ± 0.04
	σ_{++}	0.77 ± 0.02	0.32 ± 0.04
	σ_{+++}	0.69 ± 0.02	0.26 ± 0.03

Benefits of the temporal smoothing We sample complete data with a model M_+ and compare the results obtained when also testing with a model M_+, with and without the proposed temporal smoothing for the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ on Fig. 3. We observe that the algorithm with temporal smoothing performs better in terms of global metrics than without temporal smoothing, with the exception of the smoothing of the margins for a diagonal transition matrix. This difference in performance is not very consequent in terms of local metrics, which indicates that the partitions obtained with an algorithm with or without smoothing mainly differ in that they correctly match the clusters over time. This can be explained by the fact that the model capacity is reduced at the beginning of the EM algorithm in order to focus on the mixture proportions. Hence, the proposed temporal smoothing allows to avoid some of the local optima that correspond to partitions with local label-switchings. It should

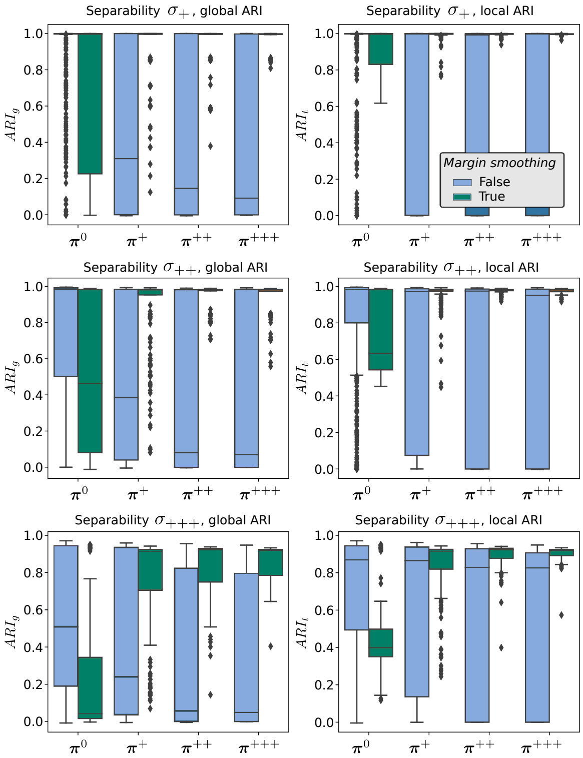


Fig. 3: Comparison of the clustering performances in terms of local and global ARI of model M_+ with and without smoothing of both $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, for increasing levels of class separability and class transitions. Data sampled using M_+.

nevertheless be noted that this is necessarily done at the expense of a slower convergence.

Comparison with a model without margins First, we observe that the addition of margins to the initial model gives qualitatively different results, as illustrated in Fig. 4. The clusters found with M_+ are not limited to nodes of similar degrees as with M_-, but consist of nodes with similar connection profiles, i.e. nodes that connect in the same proportions to the nodes of other clusters, which tends to be of greater interest for real-world networks. In order to assess quantitatively the performances of `pdc-dsbm`, we set up the following experiment: the complete data are sampled from the model M_+ and M_- as described above. The two algorithms compared use temporal smoothing, as we showed it results in better performances. We do not report the results when the data is sampled using M_+ since in that case M_- always returns partitions with metrics close to zero (local ARIs below 0.01), whereas M_+ correctly recovers the clusters, making M_+ clearly superior in this setup. We observe on Fig. 5 that, even in an unfavorable setup, the model M_+ remains competitive

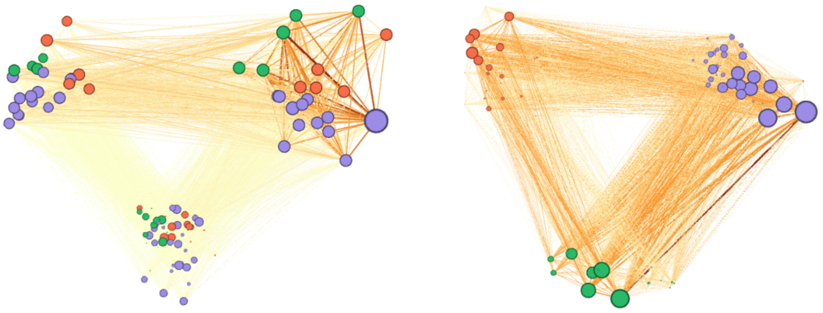


Fig. 4: Clusters obtained respectively with M_- (*left*) and M_+ (*right*). The initial graph is undirected and sampled using M_+. The color of the nodes represents their true cluster and their position on the plane corresponds to the clusters obtained with the algorithm. The weights of the edges are given by their color (yellow and red correspond respectively to low and high weight).

with M_-, especially in terms of global metrics. In fact, the differences in global metrics between M_- and M_+ are relatively small but the differences in local metrics are greater, indicating that the margins provide more flexibility to M_+ that deals better with local label-switching. Another important point to be noticed is that when the separability of the classes is very low, M_+ seeks slightly different clusters than M_-, as can be seen in the local metrics of $(\sigma^{+++}, \boldsymbol{\pi}^0)$, where class transitions do not come into play.

5.2 Experiments on real-world data

Ridership in the Bay Area Rapid Transit BART is a rapid transit public transportation system serving the San Francisco Bay Area in California. This dataset[3] represents BART ridership by origin and destination pairs for each of the $N = 45$ stations of the network, each hour from 7 a.m. to 1 a.m, from January 5 to February 1, 2015, over $T = 528$ snapshots. The number of edges in the multigraph varies over time, with increasing and decreasing interactions everyday. We chose the number of clusters $K = 7$ using the elbow method on the likelihood of the complete data. We first observe a strong periodicity in the clusters sizes over time, with a special behavior during the week-end (Fig. 6), and that most snapshots of the graph can be described by 2 or 3 non-empty clusters. However, on Monday, January 19, we observe a different behavior compared to other Mondays: there is no time period when all the stations are in clusters $\{2, 3, 4\}$ as during the peak times in the morning and in the evening. In fact, this day corresponds to the Martin Luther King Jr. holiday. Moreover, we observe 4 main settings over time: morning peak time (7 a.m. to 11 a.m., with clusters 2, 3 and 4), daytime (11 a.m. to 3 p.m, with cluster

[3] http://64.111.127.166/origin-destination/

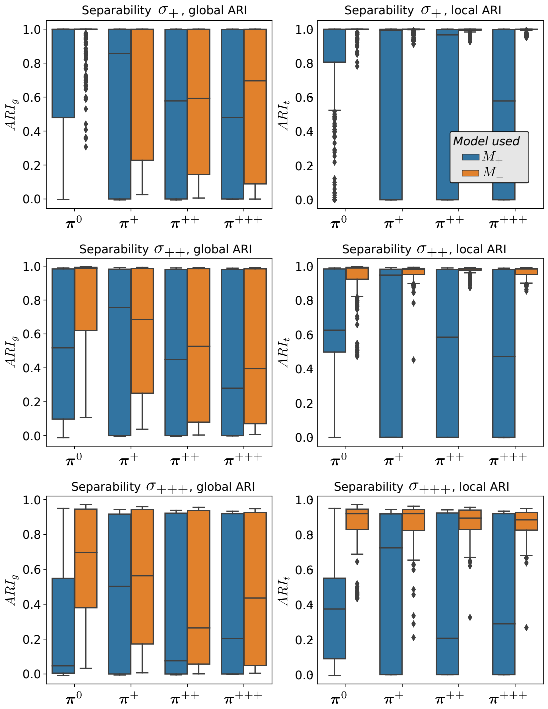


Fig. 5: Comparison of the clustering performances in terms of local and global ARI of model M_- and model M_+, for increasing levels of class separability and class transitions. The algorithms use temporal smoothing and the data is sampled using M_-, which makes the setup unfavorable for model M_+. The same experiment with data sampled from M_+ always returns partitions with local ARIs below 0.01 for model M_-, while M_+ correctly recovers the clusters.

1, 5 and 6), evening peak time (3 p.m. to 7 p.m., clusters 2, 3 and 4) and night (7 p.m. to 12 p.m, with clusters 1 and 7). Note that the morning and the evening peak time are composed of the same clusters. The obtained clusters exhibited both assortative and disassortative connectivities (not shown here). The periodicity of the cluster sizes over time can then be exploited to consider the most frequent cluster assignment of the stations at a given period of the day.

Transport for London cycle data The dataset[4] is a record of all the trips on the bike-sharing network of London between August 15 and August 28, 2019 over $N = 778$ stations. We considered a dynamic network with one snapshot every day (i.e. $T = 14$). As the sparsity of the graphs is important (95% on average), a station can be classified based on a very small number of trips, which can lead to very noisy partitions. Thus, we propose to set a threshold of 5 for the

[4] https://cycling.data.tfl.gov.uk/

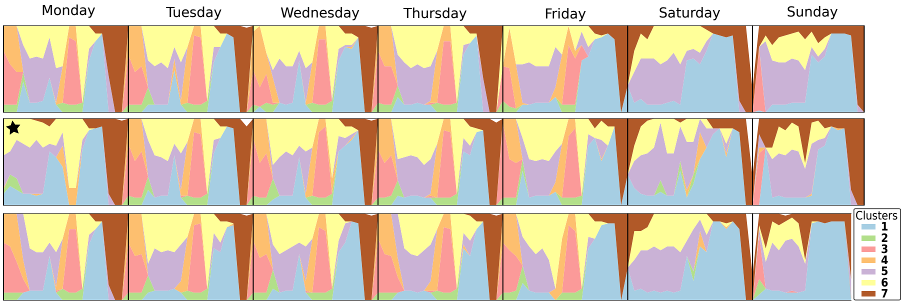


Fig. 6: Cluster sizes during the three last weeks of the dataset, from January 12 to February 1, 2015. Each sub-block corresponds to a day of operations, from 7 a.m. to 2 a.m, with January 19 (Martin Luther King Jr. holiday) marked with a star. The size of each cluster at a given time is represented by its vertical span, ranging from 0 (absent) to 45 stations (full height).

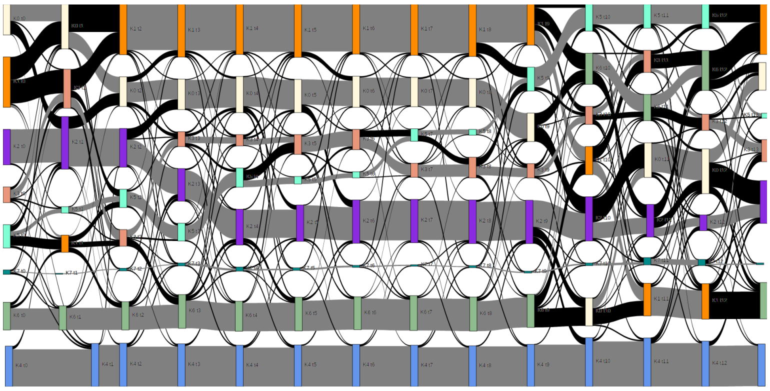
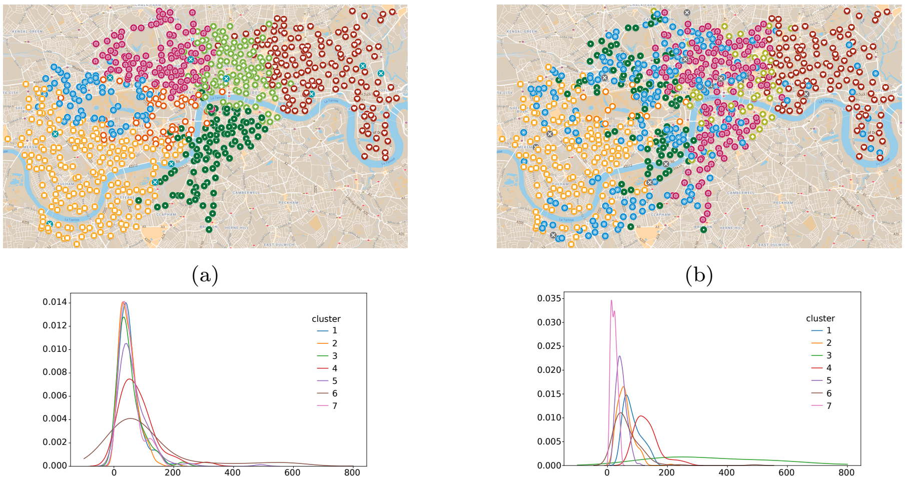


Fig. 7: Class transitions between August 15 and 28, 2019 represented by an alluvial diagram. Each cluster is characterized by a color, its size is proportional to the number of nodes it contains. Time is represented from left to right, and cluster transitions between consecutive time steps are represented by a colored flow (grey for intra-cluster and black for inter-cluster transitions). The cluster of absent nodes is represented in third position, starting from the bottom.

minimum degree of a node: below this threshold, the node is considered absent. The number of cluster $K = 7$ is chosen as previously, and, unlike BART, the obtained clusters are highly assortative (not shown here). The clusters size and transitions are represented on Fig. 7. We observe relatively stable clusters from the 16th to the 23th and many cluster transitions the 24th and 25th. These transitions can probably be explained by the fact that the Notting Hill Carnival (2.5 million attendees) took place on the 25th and 26th and that the 25th is a bank holiday. The clusters on Sunday, August 18, 2019 are presented on Fig. 8a and 8b. It can be seen on Fig. 8a that the clusters are



(a) (b)

(c) (d)

Fig. 8: Clusters on Sunday, August 18, 2019 and kernel density estimates of the stations degrees for each cluster. *Left*: model with margins, *right*: model without margins

geographically correlated, which is coherent with bike trips since the obtained cluster structure is assortative. However, some stations of the cluster 3 (red points with white circle on) are surrounded by stations of other clusters. This can be justified by the stations' direct proximity to Hyde Park, which has designated cycle routes, making trips across the park very likely. The clusters are of comparable geographical sizes, with larger clusters on the periphery and smaller ones in the city center, around touristic areas. When comparing the partitions obtained with and without margins, we observe that M_{-} provides a partition which is less geographically coherent than M_{+}. Moreover, Fig. 8c and 8d shows that the degree distribution inside each cluster is homogeneous in the case without margins: the degree of a nodes determines in part the cluster it can belong to. In fact, for a model without margins, a node with a degree of 180 is more likely to be in cluster 5 than in cluster 7, independently of the node's connectivity profile. In the case with margins (*left*), we observe that the clusters have similar degree distributions. Thus, the margins allow our model to be independent from the nodes' degrees and to focus on the nodes connection profiles.

6 Discussion and future work

In this paper, we introduced a Poisson dynamic Stochastic Block Model with degree corrections that allowed us to model dynamic graphs with a high variability in the nodes' degrees. We highlighted experimentally that these degree corrections allow us to deal with more complex cluster structures and that the degree corrected model seems more adapted to some real-world data.

The degree corrections also allows to model growing or decreasing interactions, which is not possible with a reasonable number of clusters in the models of Yang et al. (2011); Matias and Miele (2017) or Rastelli et al. (2018). In fact, even if the model of `dynsbm` has a time-dependent connectivity matrix, the constraint $\gamma_{kk}^t = \gamma_{kk}$ is set for identifiability reasons. Note that a weaker constraint can also be set, but it requires a more complex a priori modeling of the data (Matias and Miele, 2017), which makes the clustering of assortative networks with growing interactions difficult. We also proposed a method to estimate the time-dependent degree corrections by making use of their regularity, presented in detail an efficient VEM algorithm for this model called `pdc-dsbm` and provided the proofs of the main results. We applied this algorithm to real-world dynamic graphs and highlighted the coherence of the results.

The introduction of the margins in our model is done at the expense of the identifiability of the model. There is fact no normalization of the mixture distribution w.r.t. the margins. However, we argue that keeping the connectivity matrix constant, in addition to facilitating the interpretation of the clusters, strongly constrains the proposed model. As our experiments suggest, the model proves useful for clustering and provides partitions with good stability in their dynamics. Moreover, the smoothing of the margins acts as a regularizer that favors continuous margins and thus constrains the parameter space. In future work, the model can be eventually modified to assert the identifiability constraint of Karrer and Newman (2011) in an EM algorithm based on the work of Razaee et al. (2019) for static Poisson SBM, and also use the normalization of the time-varying parameters presented in Liu et al. (2014).

During our experiments, we tried to select the number of clusters and the appropriate model (with or without margins) using the ICL criterion (see Appendix D), as proposed in previous works (Daudin et al., 2008; Matias and Miele, 2017; Rastelli et al., 2018). However, we observed that the penalty that the criterion applies to the log-likelihood of the complete data appeared often negligible when compared to the latter in the context of Poisson distributions. This is reminiscent of the works of Salah and Nadif (2019) for von-Mises Fisher distributions, where ICL also showed strong limitations. We thus suggest further investigations on the model selection criteria in dynamic SBM for Poisson distributions.

Acknowledgements We thank the three anonymous reviewers for their detailed comments that have helped us a lot to improve this manuscript.

Competing Interests

The authors declare that they have no conflict of interest.

A Derivation of the objective criterion (6)

We derive the criterion (6) in the case of a constant number of nodes ($\forall t$, $V^t = V$), the other case easily follows. Let Q be a probability over the space of complete data $\mathcal{Z}$, i.e. the set of all possible latent trajectories for N nodes, over K possible states and T time steps. From Neal and Hinton (1998), we have:

$$\begin{aligned}\ell(\boldsymbol{\theta}) \geq F(\boldsymbol{q}, \boldsymbol{\theta}) &= \ell(\boldsymbol{\theta}) - KL(Q||P(.|\boldsymbol{X}, \boldsymbol{\theta})) \\ &= \mathbb{E}_Q(\log P(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\theta})) + \mathbb{H}(Q) \\ &= \mathbb{E}_Q(\log P(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\theta}) - \log Q(\boldsymbol{Z}; \boldsymbol{q})).\end{aligned}$$

Let Q factorize as N independent inhomogeneous Markov models:

$$\begin{aligned}Q(\boldsymbol{Z}; \boldsymbol{q}) &= \prod_i Q(Z_i^1; \boldsymbol{q}) \prod_{t\geq 2} Q(Z_i^t | Z_i^{t-1}; \boldsymbol{q}) \\ &= \prod_{ik} q(i,k)^{Z_{ik}^1} \prod_{t\geq 2} \prod_{\ell} q(t,i,k,\ell)^{Z_{ik}^{t-1} Z_{i\ell}^t}\end{aligned}$$

where Q is parameterized by $\boldsymbol{q} = \Big(\big(q(i,k)\big)_{ik}, \big(q(t,i,k,\ell)\big)_{tikl} \Big)$, with $q(i,k) = Q(Z_{ik}^1 = 1)$, $q(t,i,k,\ell) = Q(Z_{i\ell}^t = 1 | Z_{ik}^{t-1} = 1)$.

First, from the variational distributions, we have:

$$\begin{aligned}\mathbb{E}_Q(\log Q(\boldsymbol{Z}; \boldsymbol{q})) = &\sum_{ik} \mathbb{E}_Q(Z_{ik}^1) \log q(i,k) \\ &+ \sum_{t\geq 2} \sum_{ik\ell} \mathbb{E}_Q(Z_{ik}^{t-1} Z_{i\ell}^t) \log q(t,i,k,\ell).\end{aligned} \tag{14}$$

Secondly, from the model, we have:

$$\begin{aligned}\mathbb{E}_Q(\log P(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\theta})) = &\sum_{ik} \mathbb{E}_Q(Z_{ik}^1) \log \alpha_k + \sum_{t\geq 2} \sum_{ik\ell} \mathbb{E}_Q(Z_{ik}^{t-1} Z_{i\ell}^t) \log \pi_{k\ell} \\ &+ \sum_t \sum_{i\neq j} \sum_{k\ell} \mathbb{E}_Q(Z_{ik}^t Z_{j\ell}^t) \log \phi(X_{ij}^t; \mu_i^t \nu_j^t \gamma_{k\ell}).\end{aligned} \tag{15}$$

To develop $F(\boldsymbol{q}, \boldsymbol{\theta})$ we rely on the following Lemma.

Lemma 1 *We have the following equalities:*

$$\mathbb{E}_Q(Z_{ik}^1) = q(i,k), \tag{16a}$$

$$\mathbb{E}_Q(Z_{ik}^{t-1} Z_{i\ell}^t) = q(t-1,i,k) q(t,i,k,\ell), \tag{16b}$$

$$\forall i \neq j,\ \mathbb{E}_Q(Z_{ik}^t Z_{j\ell}^t) = q(t,i,k) q(t,j,\ell). \tag{16c}$$

Thereby, from the expressions of (14, 15) and Lemma 1, the variational lower-bound of the log-likelihood of the model is given by:

$$\begin{aligned}F(\boldsymbol{q}, \boldsymbol{\theta}) &= \mathbb{E}_Q(\log P(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\theta}) - \log Q(\boldsymbol{Z}; \boldsymbol{q})) \\ &= \sum_{ik} q(i,k) \log \alpha_k + \sum_{t\geq 2} \sum_{ik\ell} q(t-1,i,k) q(t,i,k,\ell) \log \pi_{k\ell} \\ &+ \sum_t \sum_{i\neq j} \sum_{k\ell} q(t,i,k) q(t,j,\ell) \log \phi(X_{ij}^t; \mu_i^t \nu_j^t \gamma_{k\ell}) \\ &- \sum_{ik} q(i,k) \log q(i,k) - \sum_{t\geq 2} \sum_{ik\ell} q(t-1,i,k) q(t,i,k,\ell) \log q(t,i,k,\ell).\end{aligned}$$

Proof (of Lemma 1)
As the Markov chains $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$ are independent for the distribution Q, we have to prove that $\mathbb{E}_Q(Z_{ik}^t) = q(t,i,k)$. The proof for (16a) and (16b) are analogous.

In the paper, the latent processes are defined over the index set $\{1,\dots,T\}$. In the following, we consider a virtual source cluster k_s at virtual time step $t = 0$ from which every node starts. Let $\mathcal{Z}_i^{(t,t')}(k)$ be all the possible latent trajectories for node i over $t'-t$ time steps, starting at cluster k at time t:

$$\mathcal{Z}_i^{(t,t')}(k) = \{\boldsymbol{Z}_i \in \{0,1\}^{(t'-t+1)K} | \boldsymbol{Z}_i = (\boldsymbol{Z}_i^t, \dots, \boldsymbol{Z}_i^{t'})^\intercal, Z_{ik}^t = 1 \wedge \forall \tau, \sum_k Z_{ik}^\tau = 1\}.$$

For $t' \leq t$ and $k' \in \{1,\dots,K\}$, we define $\mathcal{Z}_i^{(t,t')}(k,\tau,k')$, the set of all paths of $\mathcal{Z}_i^{(t,t')}(k)$, that pass through cluster k' at time step τ:

$$\mathcal{Z}_i^{(t,t')}(k,\tau,k') = \{\boldsymbol{Z}_i \in \mathcal{Z}_i^{(t,t')}(k) | Z_{ik'}^\tau = 1\}.$$

Let Q^i be the distribution for node i. As the N chains are independent:

$$\mathbb{E}_Q(Z_{ik}^t) = \mathbb{E}_{Q^i}(Z_{ik}^t) = Q^i(Z_{ik}^t = 1) = \sum_{\boldsymbol{Z} \in \mathcal{Z}_i^{(0,T)}(k_s,t,k)} Q^i(\boldsymbol{Z}).$$

As $\mathcal{Z}_i^{(0,T)}(k_s,t,k)$ decomposes as $\mathcal{Z}_i^{(0,t-1)}(k_s) \times \mathcal{Z}_i^{(t,T)}(k)$. In the following, we identify the elements of the sets with their index ($(k_0^{(c)},\dots,k_T^{(c)})$ is the cth element of $\mathcal{Z}_i^{(0,T)}(k)$). For consistency with the notations, we define $q(1,i,k_s,k') = q(i,k')$ the transition probability from virtual cluster k_s at $t = 0$. We can then write:

$$\begin{aligned}
\mathbb{E}_Q(Z_{ik}^t) &= \sum_{\boldsymbol{Z} \in \mathcal{Z}_i^{(T)}(s,t,k)} Q^i(\boldsymbol{Z}) \\
&= \sum_{c \in \mathcal{Z}_i^{(T)}(s,t,k)} q(1,i,k_0^{(c)},k_1^{(c)})q(2,i,k_1^{(c)},k_2^{(c)}) \dots q(T,i,k_{T-1}^{(c)},k_T^{(c)}) \\
&= \sum_{c' \in \mathcal{Z}_i^{(0,t-1)}(k_s)} \sum_{c'' \in \mathcal{Z}_i^{(t,T)}(k)} \Big(q(1,i,k_0^{(c')},k_1^{(c')}) \dots q(t,i,k_{t-1}^{(c')},k) \\
&\times q(t+1,i,k,k_1^{(c'')}) \dots q(T,i,k_{T-t-1}^{(c'')},k_{T-t}^{(c'')}) \Big) \\
&= \Big(\sum_{c' \in \mathcal{Z}_i^{(0,t-1)}(k_s)} q(1,i,k_0^{(c')},k_1^{(c')}) \dots q(t,i,k_{t-1}^{(c')},k) \Big) \\
&\times \Big(\sum_{c'' \in \mathcal{Z}_i^{(t,T)}(k)} q(t+1,i,k,k_1^{(c'')}) \dots q(T,i,k_{T-t-1}^{(c'')},k_{T-t}^{(c'')}) \Big).
\end{aligned}$$

The second sum in the last equation corresponds to summing over all possible paths in a chain of length $T-t$ starting at cluster k, so it equals one. Now, recall that:

$$\begin{aligned}
q(t,i,k) &= \sum_{k'} q(t-1,i,k')q(t,i,k',k) \\
&= \sum_{k_1,\dots,k_{t-1}} q(i,k_1)q(2,i,k_1,k_2) \dots q(t,i,k_{t-1},k) \\
&= \sum_{c' \in \mathcal{Z}_i^{(0,t-1)}(k_s)} q(1,i,k_0^{(c')},k_1^{(c')})q(2,i,k_1^{(c')},k_2^{(c')}) \dots q(t,i,k_{t-1}^{(c')},k).
\end{aligned}$$

This concludes the proof.

B Derivation of the Expectation step

Here, we present a way to derive the proposed formulae in E-step for a fixed set of nodes (i.e. $\forall t,\ V^t = V$). The results when considering a variable number of nodes easily follows.

As proposed in Bartolucci and Pandolfi (2020), the true VE step can be realized but is computationally heavy. In fact, in order to optimize $F(\boldsymbol{q},\boldsymbol{\theta})$ w.r.t. $q(t,i,k,\ell)$, we notice that every $q(t',i,k')$ with $t' \geq t$ depends on $q(t,i,k,\ell)$. Here, we instead propose a VE step that increases $F(\boldsymbol{q},\boldsymbol{\theta})$ w.r.t. the variational parameters $\boldsymbol{q}$.

We consider the variational parameters $\boldsymbol{q}(i) = \Big(\big(q(i,k)\big)_k, \big(q(t,i,k,\ell)\big)_{tkl}\Big)$ as well as auxiliary variables $\boldsymbol{q}_m^t(i) = \big(q(t,i,k)\big)_{ik}$ for the marginal probabilities, where $q(t,i,k) = Q(Z_{ik}^t = 1)$.

We first note that F can be decomposed over each node and cluster thanks to the variational approximation: $F(\boldsymbol{q},\boldsymbol{\theta}) = \sum_{i\ell} F_{i\ell}(\boldsymbol{q}(i), \boldsymbol{q}_m(-i), \boldsymbol{\theta})$ where $\boldsymbol{q}_m(-i) = \big(\boldsymbol{q}_m^1(j), \dots, \boldsymbol{q}_m^T(j)\big)_{j\neq i}$ and

$$\begin{aligned}
&F_{i\ell}(\boldsymbol{q}(i), \boldsymbol{q}_m(-i), \boldsymbol{\theta}) = q(i,\ell)\log\frac{\alpha_\ell}{q(i,\ell)} \\
&+ \sum_{t\geq 2}\sum_k q(t-1,i,k)q(t,i,k,\ell)\log\frac{\pi_{k\ell}}{q(t,i,k,\ell)} \\
&+ \sum_t \left(q(t,i,\ell)\sum_{j\neq i}\sum_k q(t,j,k)\log\phi_{ij\ell k}^t + q(t,i,\ell)\sum_{j\neq i}\sum_k q(t,j,k)\log\phi_{jik\ell}^t \right) \\
&= q(i,\ell)\log\frac{\alpha_\ell}{q(i,\ell)} \\
&+ \sum_{t\geq 2}\sum_k q(t-1,i,k)q(t,i,k,\ell)\log\frac{\pi_{k\ell}}{q(t,i,k,\ell)} \\
&+ \sum_t q(t,i,\ell)\sum_{j\neq i}\sum_k q(t,j,k)\log\Phi_{ij\ell k}^t
\end{aligned}$$

where we note $\Phi_{ijk\ell}^t = \phi_{ijk\ell}^t\phi_{ji\ell k}^t$.

For constant marginal probabilities $\boldsymbol{q}_m(-i)$, we optimize

$$F_{i\ell}((\boldsymbol{q}^1(i), \boldsymbol{q}_m^1(i))\dots, (\boldsymbol{q}_m^T(i), \boldsymbol{q}^T(i))|\boldsymbol{q}_m^t(-i), \boldsymbol{\theta})$$

by applying a single step of coordinate ascent on each coordinate $(\boldsymbol{q}^t(i), \boldsymbol{q}_m^t(i))$. When applying this procedure, the other coordinates $((\boldsymbol{q}^{-t}(i), \boldsymbol{q}_m^{-t}(i)))$ are considered constant. We apply this procedure sequentially, for t in $\{1,\dots,T\}$, and update the marginal probabilities $q(t,i,k)$ with the obtained transition probabilities $q(t,i,k,\ell)$ at each time step.

The formula for E-step can be obtained as follows. Since $q(t,i,k) = \sum_{k'} q(t-1,i,k')q(t,i,k',k)$, $(\boldsymbol{q}^t(i), \boldsymbol{q}_m^t(i))$ only depends on $\boldsymbol{q}^t(i)$. For $t \geq 2$, we can write:

$$\begin{aligned}
&F_{i\ell}\big(q(t,i,1,\ell),\dots,q(t,i,K,\ell)|\boldsymbol{q}^{-t}(i), \boldsymbol{q}_m^{-t}(i), \boldsymbol{q}_m(-i), \boldsymbol{\theta}\big) \\
&= \sum_k q(t-1,i,k)q(t,i,k,\ell)\log\frac{\pi_{k\ell}}{q(t,i,k,\ell)} \\
&+ \sum_k q(t,i,k)q(t+1,i,k,\ell)\log\frac{\pi_{k\ell}}{q(t+1,i,k,\ell)} \\
&+ q(t,i,\ell)\sum_{j\neq i}\sum_k q(t,j,k)\log\Phi_{ij\ell k}^t.
\end{aligned}$$

Let $\mathcal{L}(\boldsymbol{q}^t(i), \lambda)$ be the Lagrangian of the constrained optimization problem:

$$\begin{aligned}
&\mathcal{L}(q(t,i,1,\ell),\dots,q(t,i,K,\ell),\lambda) \\
&= F_{i\ell}\big(q(t,i,1,\ell),\dots,q(t,i,K,\ell)|\boldsymbol{q}^{-t}(i), \boldsymbol{q}_m^{-t}(i), \boldsymbol{q}_m(-i), \boldsymbol{\theta}\big) + \lambda\big(1 - \sum_{\ell'} q(t,i,k,\ell')\big)
\end{aligned}$$

For $(q(t',i,k))_{k\in\{1,\dots,K\},t'\neq t}$ constant and $s \in \{1,\dots,T\}$, we have:

$$\begin{aligned}\frac{\partial q(s,i,k')}{\partial q(t,i,k,\ell)} &= \mathbb{1}(s=t)\frac{\partial}{\partial q(t,i,k,\ell)}\sum_{\ell'} q(t-1,i,\ell')q(t,i,\ell',k') \\ &= \mathbb{1}(s=t)\mathbb{1}(k'=\ell)q(t-1,i,k)\end{aligned}$$

and

$$\begin{aligned}&\frac{\partial}{\partial q(t,i,k,\ell)}\sum_{k} q(t,i,k)q(t+1,i,k,\ell)\log\frac{\pi_{k\ell}}{q(t+1,i,k,\ell)} \\ &= -q(t-1,i,k)\sum_{\ell'} q(t+1,i,\ell,\ell')\big(\log q(t+1,i,\ell,\ell') - \log(\pi_{\ell\ell'}\big) \\ &= -q(t-1,i,k)D_{\mathrm{KL}}(\boldsymbol{q}(t+1,i,\ell,:)||\boldsymbol{\pi}_{\ell,:})\end{aligned}$$

where $\boldsymbol{q}(t+1,i,\ell,:) = (q(t+1,i,\ell,1),\dots,q(t+1,i,\ell,K))^\intercal$ and $\boldsymbol{\pi}_{\ell,:} = \big(\pi_{1\ell},\dots,\pi_{K\ell}\big)^\intercal$. Let $d^t_{ik} = D_{\mathrm{KL}}(\boldsymbol{q}(t,i,k,:)||\boldsymbol{\pi}_{k,:})$. We then have:

$$\begin{aligned}&\frac{\partial}{\partial q(t,i,k,\ell)}F_{i\ell}\big(q(t,i,1,\ell),\dots,q(t,i,K,\ell)|\boldsymbol{q}^{-t}(i),\boldsymbol{q}_m^{-t}(i),\boldsymbol{q}_m(-i),\boldsymbol{\theta}\big) \\ &= q(t-1,i,k)\big(\log\pi_{k\ell} - d^{t+1}_{i\ell} - 1 - \log q(t,i,k,\ell)\big) \\ &+ \frac{\partial}{\partial q(t,i,k,\ell)}\sum_{s} q(s,i,\ell)\sum_{j\neq i}\sum_{\ell'} q(s,j,\ell')\log\Phi^s_{ij\ell\ell'} \\ &= q(t-1,i,k)\big(\log\pi_{k\ell} - d^{t+1}_{i\ell} - 1 - \log q(t,i,k,\ell) + \sum_{j\neq i}\sum_{\ell'} q(t,j,\ell')\log\Phi^t_{ij\ell\ell'}\big).\end{aligned}$$

Setting the derivative of the Lagrangian to zero, we have:

$$\log q(t,i,k,\ell) = -\frac{\lambda}{q(t-1,i,k)} - 1 + \log\pi_{k\ell} - d^{t+1}_{i\ell} + \sum_{j\neq i}\sum_{\ell'} q(t,j,\ell')\log\Phi^t_{ij\ell\ell'}.$$

Thus, $q(t,i,k,\ell) \propto \pi_{k\ell}\exp(-d^{t+1}_{i\ell})\prod_{j\neq i}\prod_{\ell'}{\Phi^t_{ij\ell\ell'}}^{q(t,j,\ell')}$. This justifies the proposed formula. We can note that contrary to Matias and Miele (2017), this formula includes a penalty term $\exp(-d^{t+1}_{i\ell})$ to the mixture proportions. In Matias and Miele (2017), the formula for E-step seems to be an approximation of this formula. In our experiments, we observed that our formula gives better clustering results when the data has many cluster transitions ($\boldsymbol{\pi}$ has low trace) without smoothing the margins, but comparable results when smoothing the margins.

C Derivation of the M-step

To update the parameters in the maximization step, we increase $F(\boldsymbol{q},\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ by maximizing F for each parameter, conditionally on the others. We first update the mixture proportions $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$, since they only depend on $\boldsymbol{q}$. Next, we update $\boldsymbol{\gamma}$, then $\boldsymbol{\mu}$ and finally $\boldsymbol{\nu}$. The updates (8a, 8b) with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ are direct. Concerning $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{\gamma}$, the lower-bound on the log-likelihood of the model is:

$$\begin{aligned}F(\boldsymbol{q},\boldsymbol{\theta}) &= \sum_{t}\sum_{\substack{ij\\ i\neq j}}\sum_{k\ell} q(t,i,k)q(t,j,\ell)\log\phi^t_{ijk\ell} + \mathrm{const} \qquad (17) \\ &= \sum_{t}\sum_{\substack{ij\\ i\neq j}}\sum_{k\ell} q(t,i,k)q(t,j,\ell)\big(\mu^t_i\nu^t_j\gamma_{k\ell} - X^t_{ij}\log\phi(X^t_{ij};\mu^t_i\nu^t_j\gamma_{k\ell})\big) + \mathrm{const}.\end{aligned}$$

By computing the derivative of (17) w.r.t. μ^t_i, ν^t_j and $\gamma_{k\ell}$ and setting it to zero we obtain the maximization step in (8d, 8e, 8c).

D Model selection with the ICL criterion

In order to choose the appropriate number of clusters K we considered the Integrated Classification Likelihood Biernacki et al. (2000), as proposed in Daudin et al. (2008) for the static SBM and in Corneli et al. (2016); Matias and Miele (2017); Rastelli et al. (2018) for dynamic models based on the SBM. The ICL criterion for a model M_K with K clusters is defined as:

$$ICL(M_K) = \log P(\boldsymbol{X}, \boldsymbol{Z}|M_K) = \int_{\boldsymbol{\Theta}} P(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}, M_K) g(\boldsymbol{\theta}|M_K)\, d\boldsymbol{\theta}, \tag{18}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\nu}, \gamma) \in \boldsymbol{\Theta}$, $\boldsymbol{\Theta} = A_K \times A_K^K \times \mathbb{R}^{+TN} \times \mathbb{R}^{+TN} \times \mathbb{R}^{+TK^2}$, A_K is the K-dimensional simplex and g is the density of the prior distribution on $\boldsymbol{\Theta}$.

Let $g_{\pi_k}(\boldsymbol{\pi}_k|M_K) = \frac{1}{B(\delta,\ldots,\delta)} \prod_{k'} \pi_{kk'}^{\delta-1}$ be a prior on $\boldsymbol{\pi}_k$, the kth row of $\boldsymbol{\pi}$.

$$\begin{aligned}\log P(\boldsymbol{Z}|M_K) =& \log \int_{A_K} \frac{1}{B(\delta,\ldots,\delta)} \alpha_1^{Z^1_{.1}+\delta-1} \ldots \alpha_K^{Z_{.K}+\delta-1}\, d\boldsymbol{\alpha} \\ &+ \log \int_{A_K^K} \prod_k \frac{1}{B(\delta,\ldots,\delta)} \pi_{k1}^{n_{k1}+\delta-1} \ldots \pi_{kK}^{n_{kK}+\delta-1}\, d\boldsymbol{\pi} \\ =& I_\alpha + I_\pi\end{aligned}$$

Where $n^z_{kk'} = \sum_{t\geq 2} \sum_i Z^{t-1}_{ik} Z^t_{ik'}$ and I_α is computed as in Daudin et al. (2008).

$$\begin{aligned}I_\pi &= \log \int_{A_g^g} \prod_k \frac{1}{B(\delta,\ldots,\delta)} \pi_{k1}^{n^z_{k1}+\delta-1} \ldots \pi_{kg}^{n^z_{kg}+\delta-1}\, d\boldsymbol{\pi} \\ &= \log \prod_k \frac{1}{B(\delta,\ldots,\delta)} \int_{A_K} \pi_{k1}^{n^z_{k1}+\delta-1} \ldots \pi_{kg}^{n^z_{kK}+\delta-1}\, d\boldsymbol{\pi}_k \\ &= \sum_k \log \Big(\frac{B(n^z_{k1}+\delta, \ldots, n^z_{kK}+\delta)}{B(\delta,\ldots,\delta)} \Big) \\ &= g \log \Gamma(\delta g) - g^2 \log \Gamma(\delta) - \sum_k \log \Gamma(n^z_{k.} + K\delta) + \sum_{kk'} \log \Gamma(n^z_{kk'} + \delta)\end{aligned}$$

We use Stirling's formula $\log \Gamma(x) \approx (x - \frac{1}{2}) \log(x-1) - (x-1) + \frac{1}{2} \log \pi$, which is even valid for small values of x. Thus, Stirling's formula for $\log \Gamma(n^z_{kk'} + \delta)$ remains valid with small values of $n^z_{kk'}$. Following Biernacki et al. (2000), it can be shown that, assuming $K = o(N)$ and removing terms in $O(1)$ (since the error term of the BIC is $O(1)$):

$$\begin{aligned}\log P(\boldsymbol{Z}|M_K) =& -\frac{K-1}{2} \log N + \sum_k Z^1_{.k} \log \frac{Z^1_{.k}}{N} \\ & -\frac{K-1}{2} \sum_k \log n^z_{k.} + \sum_{kk'} n^z_{kk'} \log \frac{n^z_{kk'}}{n^z_{k.}},\end{aligned}$$

where $n^z_{k.} = \sum_{k'} n^z_{kk'}$ and $Z^1_{.k} = \sum_i Z^1_{ik}$. Using the hypothesis $n^z_{k.} = \frac{N(T-1)}{K}$, we have $\sum_k \log n^z_{k.} = K \log N(T-1) + o(N)$. Replacing $\boldsymbol{Z}$ by $\widehat{\boldsymbol{Z}}$, the estimated partition, we obtain:

$$\begin{aligned}ICL(K) \approx& \max_{\boldsymbol{\theta}} \log P(\boldsymbol{X}, \widehat{\boldsymbol{Z}}|\boldsymbol{\theta}, M_K) \\ & -\frac{K-1}{2} \log N - \frac{K(K-1)}{2} \log N(T-1) - \frac{K^2+2TN}{2} \log(TN(N-1)).\end{aligned}$$

The term $\frac{K-1}{2} \log N$ is due to the estimated parameter $\boldsymbol{\alpha}$. In Matias and Miele (2017), the parameter $\boldsymbol{\alpha}$ is not estimated and is considered to be equal to the stationary distribution of $\boldsymbol{\pi}$. Omitting the term due to $\boldsymbol{\alpha}$ in the proposed ICL results in the same ICL as proposed in Matias and Miele (2017).

We note that we have no guarantee that the assumption of Dirichlet priors for each row of $\boldsymbol{\pi}$ with Jeffrey's uninformative priors is a good choice. In fact, with this dynamic model, we are interested in partitions that are relatively stable through time, which implies that $\boldsymbol{\pi}$ should be diagonally dominant. Thus, contrary to mixture proportions in mixture models, some dimensions of the simplex should be preferred by the prior for the rows of $\boldsymbol{\pi}$, such that $\boldsymbol{\pi}_k$, the kth row of $\boldsymbol{\pi}$, could have a prior in the form $\text{Dir}(\boldsymbol{\delta}_k)$, with $\delta_{k\ell} = \delta_0$ if $k \neq \ell$ and $\delta_{kk} = \delta_{\text{diag}}$, where $\delta_{\text{diag}} > \delta_0$.

References

Abbe, E. (2017). Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531.

Affeldt, S., Labiod, L., and Nadif, M. (2021). Regularized bi-directional co-clustering. *Statistics and Computing*, 31(3):1–17.

Ailem, M., Role, F., and Nadif, M. (2017a). Model-based co-clustering for the effective handling of sparse data. *Pattern Recognition*, 72:108–122.

Ailem, M., Role, F., and Nadif, M. (2017b). Sparse Poisson Latent Block Model for Document Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1563–1576.

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

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., and Modha, D. S. (2007). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *Journal of Machine Learning Research*, 8(67):1919–1986.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Bartolucci, F. and Pandolfi, S. (2020). An exact algorithm for time-dependent variational inference for the dynamic stochastic block model. *Pattern Recognition Letters*, 138:362 – 369.

Benzecri, J.-P. (1973). *L'analyse des données, tome 2 : l'analyse des correspondances*. Dunod, Paris.

Biernacki, C., Celeux, G., and 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.

Bock, H.-H. (2020). Co-clustering for object by variable data matrices. In Imaizumi, T., Nakayama, A., and Yokoyama, S., editors, *Advanced Studies in Behaviormetrics and Data Science: Essays in Honor of Akinori Okada*, pages 3–17, Singapore. Springer Singapore.

Chi, Y., Song, X., Zhou, D., Hino, K., and Tseng, B. L. (2007). Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*, pages 153–162. Association for Computing Machinery.

Corneli, M., Latouche, P., and 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., and Rossi, F. (2018). Multiple change points detection and clustering in dynamic networks. *Statistics and Computing*, 28(5):989–1007.

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

Fu, W., Song, L., and Xing, E. P. (2009). Dynamic mixed membership blockmodel for evolving networks. In *ICML*, pages 329–336.

Ghahramani, Z. and Jordan, M. I. (1997). Factorial Hidden Markov Models. *Machine Learning*, 29(2–3):245–273.

Govaert, G. and Nadif, M. (2005). An EM Algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4).

Govaert, G. and Nadif, M. (2013). *Co-clustering: models, algorithms and applications*. John Wiley & Sons.

Govaert, G. and Nadif, M. (2018). Mutual information, phi-squared and model-based co-clustering for contingency tables. *Advances in Data Analysis and Classification*, 12(3):455–488.

Greenacre, M. (2007). *Correspondence analysis in practice*. Chapman & Hall/CRC.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 83(1).

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., and Tseng, B. L. (2009). Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data*, 3(2):1–31.

Liu, S., Wang, S., and Krishnan, R. (2014). Persistent community detection in dynamic social networks. In Tseng, V. S., Ho, T. B., Zhou, Z.-H., Chen, A. L. P., and Kao, H.-Y., editors, *Advances in Knowledge Discovery and Data Mining*, pages 78–89. Springer International Publishing.

Mariadassou, M., Robin, S., and Vacher, C. (2010). Uncovering latent structure in valued graphs: a variational approach. *Ann. Appl. Stat.*, 4(2):715–742.

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

Meng, X.-L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278.

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer.

Qiao, M., Yu, J., Bian, W., Li, Q., and Tao, D. (2017). Improving stochastic block models by incorporating power-law degree characteristic. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 2620–2626.

Rastelli, R., Latouche, P., and Friel, N. (2018). Choosing the number of groups in a latent stochastic blockmodel for dynamic networks. *Network Science*, 6(4):469–493.

Razaee, Z., Amini, A., and Li, J. J. (2019). Matched bipartite block model with covariates. *Journal of Machine Learning Research*, 20:1–44.

Salah, A. and Nadif, M. (2019). Directional co-clustering. *Advances in Data Analysis and Classification*, 13(3):591–620.

Schepers, J., Bock, H.-H., and Van Mechelen, I. (2017). Maximal interaction two-mode clustering. *Journal of Classification*, 34(1):49–75.

Sewell, D. K. and Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, 44:105–116.

Snijders, T. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14:75–100.

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

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

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