

RESEARCH ARTICLE

Continuous latent position models for instantaneous interactions

Riccardo Rastelli^{1*}  and Marco Corneli^{2*}

¹School of Mathematics and Statistics, University College Dublin, Dublin, Ireland and ²Center of Modeling, Simulation and Interactions, MAASAI Team, Université Côte d'Azur, INRIA, Nice, France

Corresponding author: Riccardo Rastelli; Email: riccardo.rastelli@ucd.ie

Action Editor: Stanley Wasserman

Abstract

We create a framework to analyze the timing and frequency of instantaneous interactions between pairs of entities. This type of interaction data is especially common nowadays and easily available. Examples of instantaneous interactions include email networks, phone call networks, and some common types of technological and transportation networks. Our framework relies on a novel extension of the latent position network model: we assume that the entities are embedded in a latent Euclidean space and that they move along individual trajectories which are continuous over time. These trajectories are used to characterize the timing and frequency of the pairwise interactions. We discuss an inferential framework where we estimate the individual trajectories from the observed interaction data and propose applications on artificial and real data.

Keywords: Latent position models; dynamic networks; non-homogeneous Poisson process; spatial embeddings; statistical network analysis

1. Introduction

The latent position model (LPM, Hoff et al., 2002) is a widely used statistical model that can be used to characterize a network through a latent space representation. The model embeds the nodes of the network as points in the real plane and then uses these latent features to explain the observed interactions between the entities. This provides a neat and easy-to-interpret graphical representation of the observed interaction data, which is able to capture some extremely common empirical features such as transitivity and homophily.

In this paper, we propose a new LPM that can be used to model repeated instantaneous interactions between entities, over an arbitrary time interval. The time dimension is continuous, and an interaction between any two nodes may happen at any point in time. We propose a data generative mechanism which is inspired by the extensive literature on LPMs, and we define an efficient estimation framework to fit our model.

Since the foundational work of Hoff et al. (2002), the literature on LPMs has been developed in many directions, both from the methodological and from the applied point of views. Recent review papers on the topic include Salter-Townshend et al. (2012); Rastelli et al. (2016); Raftery (2017); Sosa and Buitrago (2020).

*The authors contributed equally to this work.

As regards statistical methodology, the original paper of Hoff et al. (2002) defined a framework to infer and interpret a LPM for binary interactions. The authors introduced two types of LPMs: the *projection model* and the *distance model*.

The *projection model* postulates that the probability that an edge appears between any two nodes is determined by the dot product of the latent coordinates of the two respective nodes. As a consequence, a crucial contribution for the edge probability is given by the direction the nodes point toward. By contrast, the *distance model* defines the connection probability as a function of the Euclidean distance between the two nodes. Nodes that are located close to each other are more likely to connect than nodes that are located far apart. Both models provide a clear representation of the interaction data which can be used to study the network's topology or to construct model-based summaries and visualizations or predictions.

We also note the work of Hoff (2008) which introduces a generalization of the projection model, called *eigenmodel*, in which the standard dot product between the latent coordinates is replaced by larger families of inner products, associated with diagonal matrices (possibly) other than the identity. Interestingly, the author shows that the eigenmodel can also generalize the distance model, albeit using a different number of latent dimensions. Since we focus on network visualization, we do not consider the eigenmodel in this paper, although our methods may be extended to that framework.

More recently, the projection model and its variations have been extensively studied and used in a variety of applications (see Hoff (2005, 2018) and references therein). This model has also clear connections to a rich machine learning literature on spatial embeddings, which include Lee and Seung (1999); Halko et al. (2011); Kipf and Welling (2016). Variations of the projection model have been extended to dynamic settings (Durante and Dunson, 2014, 2016) and other types of network frameworks (Durante et al., 2017).

As regards the distance model, this has been extended by Handcock et al. (2007) and Krivitsky et al. (2009) to represent clustering of the nodes and more flexible degree distributions. In the context of networks evolving over time, dynamic extensions of the model have been considered in Sarkar and Moore (2005), and more recently in several works including Sewell and Chen (2015b) and Friel et al. (2016) for binary interactions. The recent review paper of Kim et al. (2018) provides additional references on dynamic network modeling. Other relevant and interesting works that revolve around the distance models in either static or dynamic settings include Gollini and Murphy (2014); Salter-Townshend and McCormick (2017) for multi-view networks, Sewell and Chen (2016) for dynamic weighted networks, and Gormley and Murphy (2007); Sewell and Chen (2015a) for networks of rankings. We also mention Raftery et al. (2012); Fosdick et al. (2019); Rastelli et al. (2018); Tafakori et al. (2019) which introduce original and closely related modeling or computational ideas.

Crucially, we note that almost all existing dynamic LPMs consider a discrete time dimension, whereby the interactions are observed at a number of different points in time. Some works do use stochastic processes to model the latent trajectories over time. For instance in Scharf et al. (2018), the authors employ Gaussian convolution processes to model the node trajectories over time and rely on a dynamic LPM in order to sample binary interactions between nodes. However, for the inference, they approximate the stochastic processes by a set of independent normal random variables anchored at a finite number of knots. Similarly, in Durante and Dunson (2016), the authors consider a set of stochastic differential equations to model the evolution of the nodes over time. By imposing conditions on the first derivative of the processes, the authors obtain time-varying smoothness while keeping a very flexible prior structure. Also in this case, however, the data analyzed by Durante and Dunson (2016) are a collection of adjacency matrices recorded at different points in time. One critical advantage of their approach based on stochastic differential equations is that these adjacency matrices may be observed at arbitrary points in time, which need not be equally spaced. However, all the pairwise interactions need to be observed in any of the network snapshots.

By contrast, a fundamental original aspect of our work is that both the observed data and the latent trajectories of the nodes are defined in a continuous time dimension. This means that any two nodes can interact at any given point in time and that these interaction data inform the latent space by characterizing the fully continuous trajectories that the nodes follow. Our new framework comes at a time when continuous networks are especially common and widely available, as they include email networks (Klimt and Yang, 2004), functional brain networks (Park and Friston, 2013), and other networks of human interactions (see Cattuto et al., 2010; Barrat and Cattuto, 2013 and references therein).

Some of the approaches that have been proposed in the statistics literature to model instantaneous interactions include Corneli et al. (2018) and Matias et al. (2018); however, we note that these approaches rely on extensions of the stochastic blockmodel (Nowicki and Snijders, 2001) and not on the LPM. Another relevant strand of literature focuses instead on modeling this type of data using Hawkes processes (see Junuthula et al., 2019 and references therein).

We propose our new continuous latent position model (CLPM) both for the projection model framework and for the distance model framework. In our approach, each of the nodes is characterized by a latent trajectory on the real plane, which is assumed to be a piece-wise linear curve. The interactions between any two nodes are modeled as events of an inhomogeneous Poisson point process, whose rate is determined by the instantaneous positions of the nodes, at each point in time. The piece-wise linear curve assumption gives sufficient flexibility regarding the possible trajectories, while not affecting the purely continuous nature of the framework, in that the rate of the Poisson process is not piece-wise constant. This is a major difference with respect to other approaches that have been considered (Corneli et al., 2018 and one of the approaches of Matias et al., 2018).

We propose a penalized likelihood approach to perform inference, and we use optimization via stochastic gradient descent (SGD) to obtain optimal estimates of the model parameters. We have created a software that implements our estimation method, which is publicly available from CLPM GitHub repository (2021).

The paper is structured as follows: in Section 2, we introduce our new model and its two variants (i.e., the projection and distance model), and we derive the main equations that are used in the paper; in Section 3, we describe our approach to estimate the model parameters; in Section 5, we illustrate our procedure on three synthetic datasets, whereas in Section 6 we propose real data applications. We give final comments and conclusions in Section 7.

2. The model

2.1 Modeling the interaction times

The data that we observe are stored as a list of interactions (or *edge list*) in the format $\mathcal{E} := \{(\tau_e, i_e, j_e)\}_{e \in \mathbb{N}}$, where $\tau_e \in [0, T]$ for all e is the interaction time between the nodes i_e and j_e , with $i_e, j_e \in \{1, \dots, N\}$. Thus, the integer index e counts the edges in the graph and it ranges from 1 to the total number of edges observed before T , say E . We consider undirected interactions without self-loops, since this is a common setup in network analysis. Extensions to the directed case could be considered; however, we do not pursue this here. We emphasize that all interactions are instantaneous, that is their length is not relevant or not recorded. An interaction between two nodes may occur at any point in time $\tau_e \in [0, T]$. Let us now formally introduce the list of the interaction times between two arbitrary nodes i and j :

$$\mathcal{E}_{ij} = \left\{ \tau_1^{(i,j)}, \dots, \tau_{E_{ij}}^{(i,j)} \right\}, \quad (1)$$

where E_{ij} is the total number of times i interacts with j before T or, equivalently, the number of edges connecting i with j . Also, $E = \sum_{i,j} E_{ij}$. We assume that the interaction times in the above equation are the realization of an inhomogeneous Poisson point process with instantaneous rate

function denoted with $\lambda_{ij}(t) \geq 0, \forall t \in [0, T]$ and nodes i and j . Using a more convenient (but equivalent) characterization, we state that the waiting time for a new interaction event between i and j is exponential with a variable rate that changes over time. Then, if we assume that the inhomogeneous point processes are independent for all pairs i and j , the likelihood function for the rates can be written as:

$$\mathcal{L}(\lambda) = \prod_{i,j: i < j} \left[\left(\prod_{\tau_e \in \mathcal{E}_{ij}} \lambda_{ij}(\tau_e) \right) \exp \left\{ - \int_0^T \lambda_{ij}(t) dt \right\} \right], \tag{2}$$

where, for simplicity, we have removed the superscript (i, j) from $\tau_e^{(i,j)}$. In the sections below, we will specify the conditions that make the processes independent across all node pairs.

2.2 Latent positions

Our goal is to embed the nodes of the network into a latent space, such that the latent positions are the primary driving factor behind the frequency and timing of the interactions between the nodes. Crucially, since the time dimension is continuous and interactions can happen at any point in time, we aim at creating a modeling framework which also evolves continuously over time. Thus, the fundamental assumption of our model is that, at any point in time, the Poisson rate function $\lambda_{ij}(t)$ is determined by the latent positions of the corresponding nodes, which we denote $\mathbf{z}_i(t) \in \mathbb{R}^2$ and $\mathbf{z}_j(t) \in \mathbb{R}^2$.

Remark. We assume that the number of dimensions of the latent space is equal to 2, because the main interest of the proposed approach is in latent space visualization of the network. However, we note that the generative model presented in this section can be easily extended to the case $\mathbf{z}_i(t) \in \mathbb{R}^d$, with $d > 2$.

To facilitate the inference task, the trajectories are assumed to be piece-wise linear curves, characterized by a number of user-defined change points in the time dimension. These change points are in common across the trajectories of all nodes, and they determine the points in time when the linear motions of the nodes change direction and speed. This means that we must define a grid of the time dimension through K change points

$$0 = \eta_1 < \eta_2 < \dots < \eta_{K-1} < \eta_K = T$$

that are common across all trajectories. We stress that this modeling choice is only meant to restrict the variety of continuous trajectories that we may consider, as it allows us to use a tractable parametric structure while keeping a high flexibility regarding the trajectories, as the number of change points increases. Also, we make the assumption that, within any two consecutive critical points, the speed at which any given node moves remains constant. As a consequence, we only need to store the coordinates of the nodes at the change points, since all the intermediate positions can then be obtained with:

$$\mathbf{z}_i \left((1 - t)\eta_k + t\eta_{k+1} \right) = (1 - t)\mathbf{z}_i(\eta_k) + t\mathbf{z}_i(\eta_{k+1}) \quad \forall t \in [0, 1] \tag{3}$$

for any change points η_k and η_{k+1} and node i . Under this parametrization of the trajectories, we can increase the number of change points to allow for more flexible structures, at the expense of computational efficiency since this would also increase the number of model parameters to estimate. The choice of the number of change points is made by the user, who defines directly the level of refinement of the trajectories based on the available computing resources.

2.2.1 Projection model

Similarly to the foundational paper of Hoff et al. (2002), we introduce two possible characterizations of the rates through the latent positions: one is inspired by the projection model and the other is inspired by the distance model. In our projection model, we assume that the rate of interactions is specified by:

$$\log \lambda_{ij}(t) = \beta + \langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle$$

for all $t \in [0, T]$ and for all nodes i and j . Here, $\beta \in \mathbb{R}$ is an intercept parameter regulating the overall interaction rates in a homogeneous fashion, but extensions of the model where it becomes specific to each node can also be considered. As regards the contributions of the latent positions, the further the nodes are positioned from the origin, the more frequent their interactions will be, especially toward other nodes that are aligned in the same direction.¹ Vice versa, we are not expecting frequent interactions for nodes that are located too close to the origin, or between pairs of nodes forming an obtuse angle.

By taking the logarithm of Equation (2) and replacing $\lambda_{ij}(t)$, the log-likelihood for the projection model is:

$$\log \mathcal{L}(\beta, \mathbf{Z}) = \sum_{i,j:i < j} \left[\sum_{\tau_e \in \mathcal{E}_{ij}} (\beta + \langle \mathbf{z}_i(\tau_e), \mathbf{z}_j(\tau_e) \rangle) - \int_0^T e^{\beta + \langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle} dt \right] \quad (4)$$

As we discuss in Appendix A, the integral term appearing in Equation (4) does not generally have a straightforward analytical solution. So, we take advantage of the fact that the integrand function is fairly regular, to efficiently estimate the integral with a composite Simpson's rule (see equation 5.1.16 of Atkinson, 1991).

2.2.2 Distance model

Here, we introduce a version of the LPM that uses the latent Euclidean distances between the nodes, rather than the dot products. The distance model formulation provides easier interpretability than the projection model, and, as we show in the simulation studies (Section 5), it also provides great flexibility, hence generally leading to superior results in this context where the hidden space is in low dimension (two).

In the distance model, we assume that:

$$\log \lambda_{ij}(t) = \beta - \|\mathbf{z}_i(t) - \mathbf{z}_j(t)\|^2 \quad (5)$$

where the last term corresponds to the squared Euclidean distance between nodes i and j at time t . The interpretation of $\beta \in \mathbb{R}$ is analogous to the projection model. By taking the logarithm of Equation (2) and using Equation (5), the log-likelihood of the distance model becomes:

$$\log \mathcal{L}(\beta, \mathbf{Z}) = \sum_{i,j:i < j} \left\{ \left(\sum_{\tau_e \in \mathcal{E}_{ij}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right\} \quad (6)$$

Unlike the projection model, the above log-likelihood has a closed form, since the integral inside the brackets can be calculated analytically (proof in Appendix B).

2.3 Penalized likelihood

Due to the piece-wise linearity assumption in Equation (5), for each node we only need to estimate its positions at times $\{\eta_k\}_{k \in [K]}$. In order to avoid over fitting, and to obtain more interpretable and meaningful results, we introduce likelihood penalizations based on the latent positions parameters

at $\{\eta_k\}_{k \in [K]}$. In particular, a penalization term is included on the right-hand side of Equation (6), whose effect is to disfavor large velocities of the nodes in the latent space.

For both the projection and distance model, as likelihood penalizations,² we define Gaussian random walk priors on the critical points of the latent trajectories:

$$\mathbf{z}_i(\eta_{k+1}) \mid \mathbf{z}_i(\eta_k) \stackrel{\perp}{\sim} \mathcal{N}(\mathbf{z}_i(\eta_k), (\eta_{k+1} - \eta_k)\sigma^2 I_2) \quad \forall k = 1, \dots, K - 1 \tag{7}$$

for every node i where I_2 is the identity matrix of order two. The equation above (with $\sigma^2 = 1$) would correspond to a Brownian motion for the i th latent trajectory, except that we would only observe it at the change points, where the latent positions are estimated. However, as the number of change points increases, the prior that we specify tends to a scaled Brownian motion on the plane. The parameter σ^2 is user-defined; hence, it can be reduced to penalize the movements of the nodes between consecutive change points. In order to obtain sensible penalizations, we choose small values of the variance parameters, as to ensure that the speed of the nodes along the trajectories is not too large. In this way, the nodes are forced to move as little as necessary, making the latent visualization of the network easier to read and interpret, and ensuring that the latent space only captures the critical features that are present in the data.

Remark. The likelihood function of the original latent distance model of Hoff et al. (2002) is not identifiable with respect to translations, rotations, and reflections of the latent positions. This is a challenging issue in a Bayesian setting that relies on sampling from the posterior distribution. In fact, the posterior samples become non-interpretable, since rigid transformations may have occurred during the collection of the sample (Shortreed et al., 2006). These non-identifiabilities are not especially relevant in our optimization setting, since the equivalent configurations of model parameters lead to the same qualitative results and interpretations. However, a case for non-identifiability can be made for dynamic networks, since translations, rotations, and reflections can occur across time, thus affecting results and interpretation. The penalizations that we introduce in this paper ensure that the nodes move as little as necessary, thus disfavoring any rotations, translations, and reflections of the space. As a consequence, the penalizations directly address the identifiability issues and the latent point process remains comparable across time.

3. Inference

In this section, we discuss the inference for the distance model described in Section 2.2.2, but an analogous procedure is considered for the projection model.

Recalling that we work with undirected graphs and in force of Equation (7), the penalized log-likelihood is

$$\log \mathcal{L}(\beta, \mathbf{Z}) = \sum_{i=1}^N \left\{ \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^N \left[\left(\sum_{\substack{\tau_e \in \mathcal{E}_{ij}}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right] - \frac{1}{2\sigma^2} \sum_{k=1}^K \|\mathbf{z}_i(\eta_k) - \mathbf{z}_i(\eta_{k-1})\|^2 \right\} + C, \tag{8}$$

where C is a constant term that does not depend on (β, \mathbf{Z}) and the integral can be explicitly computed as shown in Appendix B. Since the log-likelihood has a closed form, we implement it and rely on automatic differentiation (Griewank, 1989; Baydin et al., 2018) to maximize it numerically, with respect to (β, \mathbf{Z}) , via gradient descent (GD).³ Note that, as pointed out in the previous section, maximizing the above-penalized log-likelihood is equivalent to performing maximum-a-posteriori inference. Moreover, as it can be seen in Equation (8), the log-likelihood is additive in

the number of nodes. This remark allows us to speed up the inference of the model parameters by means of SGD (Bottou, 2010). Indeed, let us introduce ψ_1, \dots, ψ_n such that

$$\psi_i(\beta, \mathbf{Z}) := \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^N \left[\left(\sum_{\substack{\tau_e \in \mathcal{E}_{ij}}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right] - \frac{1}{2\sigma^2} \sum_{k=1}^K \|\mathbf{z}_i(\eta_k) - \mathbf{z}_i(\eta_{k-1})\|^2 \tag{9}$$

and a discrete random variable $\Psi(\beta, \mathbf{Z})$ such that

$$\pi := \pi_i := \mathbb{P}\{\Psi(\beta, \mathbf{Z}) = \psi_i(\beta, \mathbf{Z}) | \mathbf{Z}\} = \frac{1}{N}, \quad \forall i \in \{1, \dots, N\}$$

where we stress that the above probability is conditional to \mathbf{Z} and given the model parameter β . Then, let us denote ∇ the gradient operator with respect to (β, \mathbf{Z}) and \mathbb{E}_π the expectation taken with respect to the probability measure π introduced above (and hence with \mathbf{Z} given). Then, we have the following

Proposition 1. $N\nabla\Psi(\beta, \mathbf{Z})$ is an unbiased estimator of $\nabla \log \mathcal{L}(\beta, \mathbf{Z})$.

Proof.

$$\mathbb{E}_\pi [N\nabla\Psi(\beta, \mathbf{Z})] = \sum_{i=1}^N \nabla\psi_i(\beta, \mathbf{Z}) = \nabla \left(\sum_{i=1}^N \psi_i(\beta, \mathbf{Z}) \right) = \nabla \log \mathcal{L}(\beta, \mathbf{Z}),$$

where the last equality follows from the additivity of the gradient operator. □

The above proposition allows us to sample (subsets of) nodes uniformly at random, with re-injection, and use each sample (a.k.a. mini-batch) to update the model parameters via SGD, as shown in Bottou (2010). In more details, if $\theta := \{\beta, \mathbf{Z}\}$ denotes the set of the model parameters, at the k th iteration of the SGD algorithm, θ is updated as follows

$$\theta := \theta + \rho \frac{N}{|B_k|} \nabla\psi_{B_k}(\theta),$$

where the hyper-parameter ρ is the learning rate, B_k is a set of $|B_k|$ nodes extracted uniformly at random and $\psi_{B_k}(\theta)$ refers to the estimator of the full-batch log-likelihood, based on the data batch B_k , namely:

$$\psi_{B_k}(\beta, \mathbf{Z}) := \frac{1}{2} \sum_{i \in B_k} \left[\sum_{\substack{j=1 \\ j \neq i}}^N \left(\sum_{\substack{\tau_e \in \mathcal{E}_{ij}}} (\beta - \|\mathbf{z}_i(\tau_e) - \mathbf{z}_j(\tau_e)\|^2) \right) - \int_0^T e^{\beta - \|\mathbf{z}_i(s) - \mathbf{z}_j(s)\|^2} ds \right] - \frac{1}{2\sigma^2} \sum_{k=1}^K \|\mathbf{z}_i(\eta_k) - \mathbf{z}_i(\eta_{k-1})\|^2.$$

We stress that, if $|B_k| = 1$, the above equation reduces to Equation (9); conversely if $|B_k| = N$ SGD reduces to full-batch GD. We finally note that the above two equations state that the model parameters can be updated, at each iteration, based on a sub-graph with N nodes whose links are uniquely those connecting the nodes in B_k with their neighbors of order one (a.k.a. *friends*). We have implemented the estimation algorithm and visualization tools in a software repository, called CLPM, which is publicly available (CLPM GitHub repository, 2021).

4. Interpretation and model-based summaries

Both the distance and the projection model provide a visual representation of the latent space as output. Due to the continuous time dimension, the results are most easily shown as a video. For this paper, the code and results (including videos) are publicly available from CLPM GitHub repository (2021).

The evolution of the latent space provides a visualization which can be used to qualitatively assess the connectivity, both at the global level (e.g., contractions and expansions of the latent space) as well as at a local level (e.g., which nodes have more connections and when).

For the projection model, we expect highly connected nodes to be located far from the origin. They would have a higher chance to interact with any other node. On the other hand, nodes that are close to the origin will have lower connectivity, overall. For both types of nodes, the angle in between them will also play a role, favoring interactions between nodes that point in the same direction. For the distance model, we are expecting nodes to take more central positions as they become more active, and, clearly, communities arise when clusters of points are observed.

Clusteredness. In order to capture this particular behavior, we introduce a quantitative measure of clustering, or “clusteredness” of the latent space. The goal of this index is to capture and measure the local contractions of the latent space, whereby nodes tend to aggregate into clusters at a particular point in time. To construct this measure, we choose an arbitrary threshold value φ and consider circles of radius φ around each of the nodes, in the latent space. If we consider an arbitrary node, we want to count how many other nodes fall within its circle, at each point in time. By averaging this measure across all nodes, we obtain our clusteredness index, defined as the average number of nodes that fall within a random node’s circle.

This measure evolves continuously over time, and we can easily calculate it from the algorithm’s output. The relative increases and decreases of the measure over time can permit an appreciation of how the latent space can locally contract, to create communities within the network.

Partition. An additional model-based summary that we consider is a partitioning of the nodes of the network. Based on the latent space representation, we aim at deriving a partitioning of the nodes, whereby nodes in the same group tend to spend more time close to each other. This is achieved by calculating a similarity value for each pair of nodes (i, j) , at each change point η , equal to $\exp\{-\|\mathbf{z}_i(\eta) - \mathbf{z}_j(\eta)\|^2\}$. Then, we can aggregate the pairwise similarities over time by calculating their median and use these node similarities as an input for a spectral clustering algorithm (the number of groups for the algorithm is user-defined).

This approach provides additional information (in the videos and plots, these clusters can be indicated with the different nodes’ colors), and it provides a higher-level visualization and summarization of the model’s results. We note that, ideally, a challenging but interesting idea would be to include these clustering aspects directly into the generative process of the model; however, we do not pursue this here and leave the extension as future work.

Goodness of fit. We consider a basic measure of model fit whereby we calculate the observed number of interactions:

$$u_{ijk} = \sum_{\tau_e \in \mathcal{E}_{ij}} \mathbb{1}_{\{\eta_k \leq \tau_e < \eta_{k+1}\}}$$

Here, $\mathbb{1}_{\{\mathcal{A}\}}$ is equal to one if the event \mathcal{A} is true or zero otherwise. In addition, we calculate the corresponding expectation according to our model:

$$\hat{u}_{ijk} = \int_{\eta_k}^{\eta_{k+1}} \lambda_{ij}(t) dt$$

Then, we calculate the absolute value difference between the two values and average it across all edge pairs and across all change points. This corresponds to a measure of in-sample prediction error for the number of interactions. We emphasize that choosing this particular measure is arbitrary, and in fact, more sophisticated measures may be constructed (e.g., see Yang et al., 2017; Huang et al., 2022) to provide a better assessment of the goodness of fit. Our measure can be used to compare different models (i.e., distance model against projection model or different choices of the penalization parameter σ^2 or the number of the latent dimensions), as long as the change points in the two models are located identically. On the other hand, the measure is sensitive to the change point choice, in that the average number of interactions per time segment directly affects the magnitude of the mean absolute error.

5. Experiments: Synthetic data

In this section, we illustrate applications of our methodology on artificial data. We propose two types of frameworks: in the first one, we consider dynamic block structures (which involve the presence of communities, hubs, and isolated points). In this case, our aim is to inspect how the network dynamics are captured by CLPM. In the second framework, we generate data using the distance CLPM and we aim at recovering the simulated trajectories for each node.

5.1 Dynamic block structures

Simulation study 1. In this first experiment, we use a data generative mechanism that relies on a dynamic blockmodel structure for instantaneous interactions (Corneli et al., 2018). We specifically focus on a special case of a dynamic stochastic blockmodel where we can have community structure, but we cannot have disassortative mixing, that is, the rate of interactions within a community cannot be smaller than the rate of interactions between communities. In this framework, the dynamic stochastic blockmodel approximately corresponds to a special case of our distance CLPM, whereby the nodes clustered together essentially are located nearby.

In the generative framework that we consider the only node-specific information is the cluster label, hence, this structure is not as flexible as the CLPM as regards modeling node's individual behaviors. So, our goal here is to obtain a latent space visualization for these data and to ensure that CLPM can accurately capture and highlight the presence of communities. An aspect of particular importance is how CLPM reacts to the creation and dissolution of communities over time: for this purpose, our generated data include changes in the community structure over time.

For this setup, we consider the time interval $[0, 40]$ (for simplicity, we use seconds as a unit measure of time) and divide this into four consecutive time segments of 10 s each. In each of the four time segments, 60 nodes are arranged into different community structures. Thus, any changes in community structure are synchronous for all nodes and they happen at the endpoints of a time segment. The rate of interactions between any two nodes is determined by their group allocations in that specific time segment. The rate remains constant in each time segment, so that we effectively have a piece-wise homogeneous Poisson process over time, for each dyad.

We denote with $X^{(s)} \in \mathbb{N}^{N \times N}$ a simulated weighted interaction matrix which counts how many interactions occur in the s th time segment for each dyad:

$$X_{ij}^{(s)} | \mathbf{C} \sim \mathcal{P} \left(\theta_{\mathbf{c}_i \mathbf{c}_j}^{(s)} \right),$$

where $\mathcal{P}(\cdot)$ indicates the Poisson probability mass function, and \mathbf{C} is a latent vector of length N indicating the cluster labels of each of the nodes. Once we know the number of interactions for each dyad and each segment, the timing of these interactions can be sampled from a uniform distribution in the respective time segment. More in detail, the rate parameters are characterized as follows:

- (i) in the time segment $[0, 10[$, the expected number of interactions is the same for every pair of nodes: $\theta_{c_i c_j}^{(1)} = 1$, for all i and j ;
- (ii) in the time segment $[10, 20[$, three communities emerge, in particular $\theta_{11}^{(2)} = 10$, $\theta_{22}^{(2)} = 5$ and $\theta_{33}^{(2)} = 1$, whereas the rate for any two nodes in different communities is 1;
- (iii) in the time segment $[20, 30[$, the first community splits and each half joins a different existing community. The two remaining communities are characterized by $\theta_{11}^{(3)} = \theta_{22}^{(3)} = 5$. Again, any two nodes in different communities interact with rate 1;
- (iv) in the time segment $[30, 40]$, we are back to the same structure as in (i).

Throughout the simulation, node 1 always behaves as a hub, and node 60 is always isolated. This means that node 1 interacts with rate 10 at all times with any other node, whereas node 60 interacts with rate 0.01 at all times with any other node, regardless of any cluster label.

In Figure 1, we show a collection of snapshots at some critical time points, for the projection model.

The full videos of the results are provided in the code repository. The main observation is that the communities are clearly captured at all times, and they are clearly visually separated. In the two cluster formation, we see that the clusters are almost aligned to the axes; hence, they point in perpendicular directions. In the three cluster formation, the non-community third cluster, which has low interaction rate, is instead positioned more centrally between the two, but still separated from the others. This is perhaps surprising since this group should be expected to have fewer interactions and a weaker community structure. The hub is always located very far from the origin and from other points, since this guarantees a large dot product value with respect to all other nodes, at all times. By contrast, the isolated node is always located toward the opposite direction, which is very reasonable.

Figure 2 shows instead the snapshots for the distance model. In this case, the clusters are clearly separated at all times. The cluster with a strong community structure is less dispersed than the clusters with a weaker community structure. The hub is constantly positioned in the center of the space, as to minimize the distance from all of the nodes at the same time. The isolated node is instead wandering in the outskirts of the latent social space. The creation and dissolution of communities only happens right at the proximity of start/end of each time segment. For this simulation study, the mean absolute error arising from the goodness of fit procedure is 0.63 for the projection model against 0.73 for the distance model, thus preferring the projection model.

Technical details regarding the simulation's parameters, including the penalization terms and number of change points, can be consulted on the CLPM code repository.

Simulation study 2. In the second simulation study, we use again a blockmodel structure; however, in this case we approximate a continuous time framework by defining very short time segments and letting the communities change from one time segment to the next. Since creations and dissolutions of communities would be unlikely in such a short period of time, we keep the community memberships unchanged, and we progressively increase the cohesiveness of the communities. This means that we progressively increase the rates of interactions between any pairs of nodes that belong to the same community, while keeping any other rate constant. The rate of interactions within each community starts at value 1 and increases in a step-wise fashion over 40 segments, up to the value 5. The time interval is $[0, 40]$, and we consider two communities. Halfway through the simulation, a special node moves from one community to the other.

For the projection CLPM, we show the results in Figure 3, whereas Figure 4 shows the results for the distance model. Both approaches clearly capture the reinforcement of the communities over time by aggregating the nodes of each group. We observe this behavior both for the projection model and for the distance model. The projection model also exhibits nodes getting farther from the center of the space, since this would give them higher interaction rates, overall. As concerns the special node moving from one community to the other, this is well captured in that

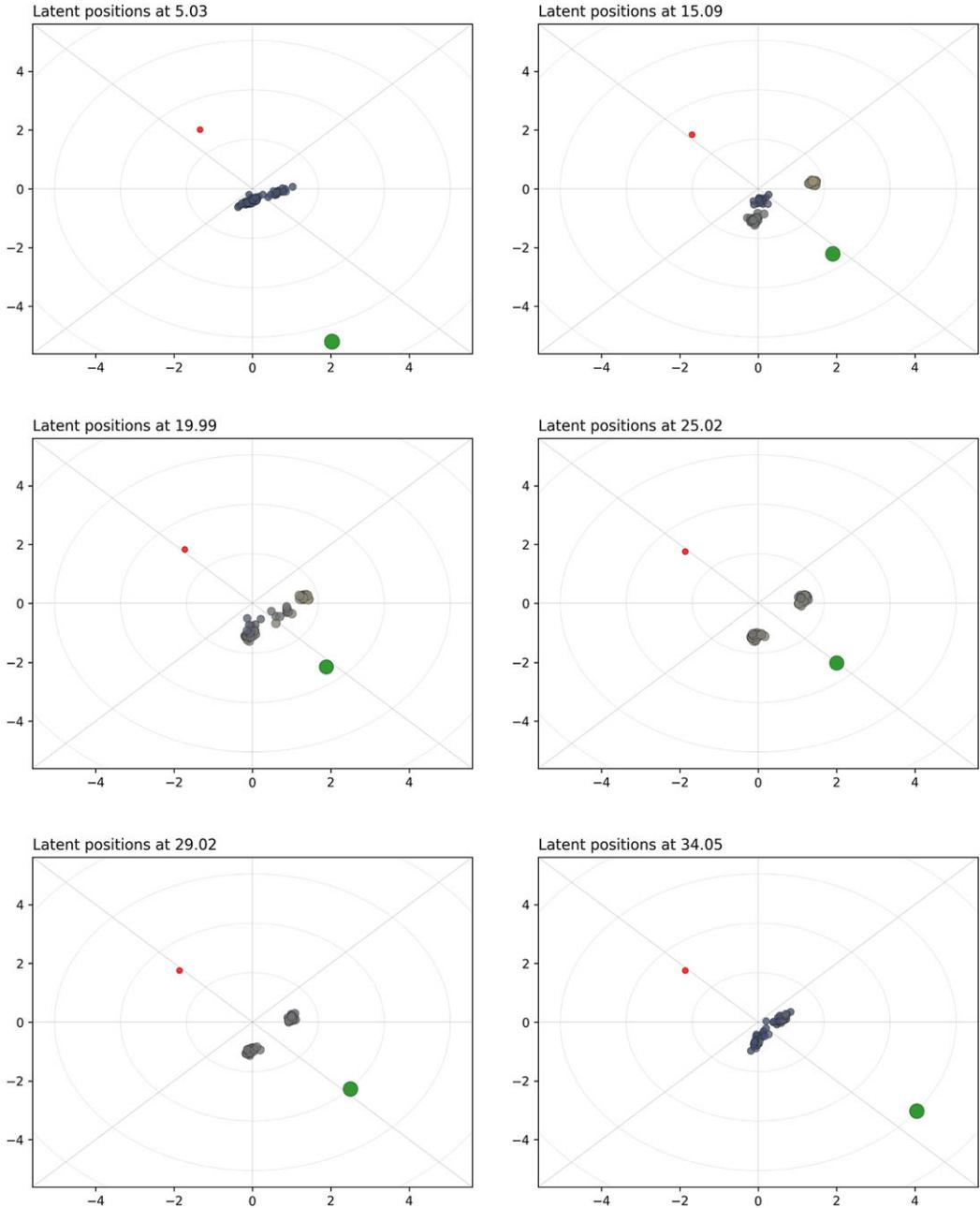


Figure 1. Simulation study 1: snapshots for the projection model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction. The hub and the isolated node are colored in green and red, respectively.

the node transitions smoothly after approximately 20 s, in both models. As concerns model fit and model choice, the mean absolute error is 5.48 for the projection model against 3.97 for the distance model, thus preferring the distance model.

5.2 Comparison with the static LPM

For simulation study 2, we propose a comparison of our results with a static LPM, as per Hoff et al. (2002). We use an implementation of the static LPM available from the R package `latentnet`.

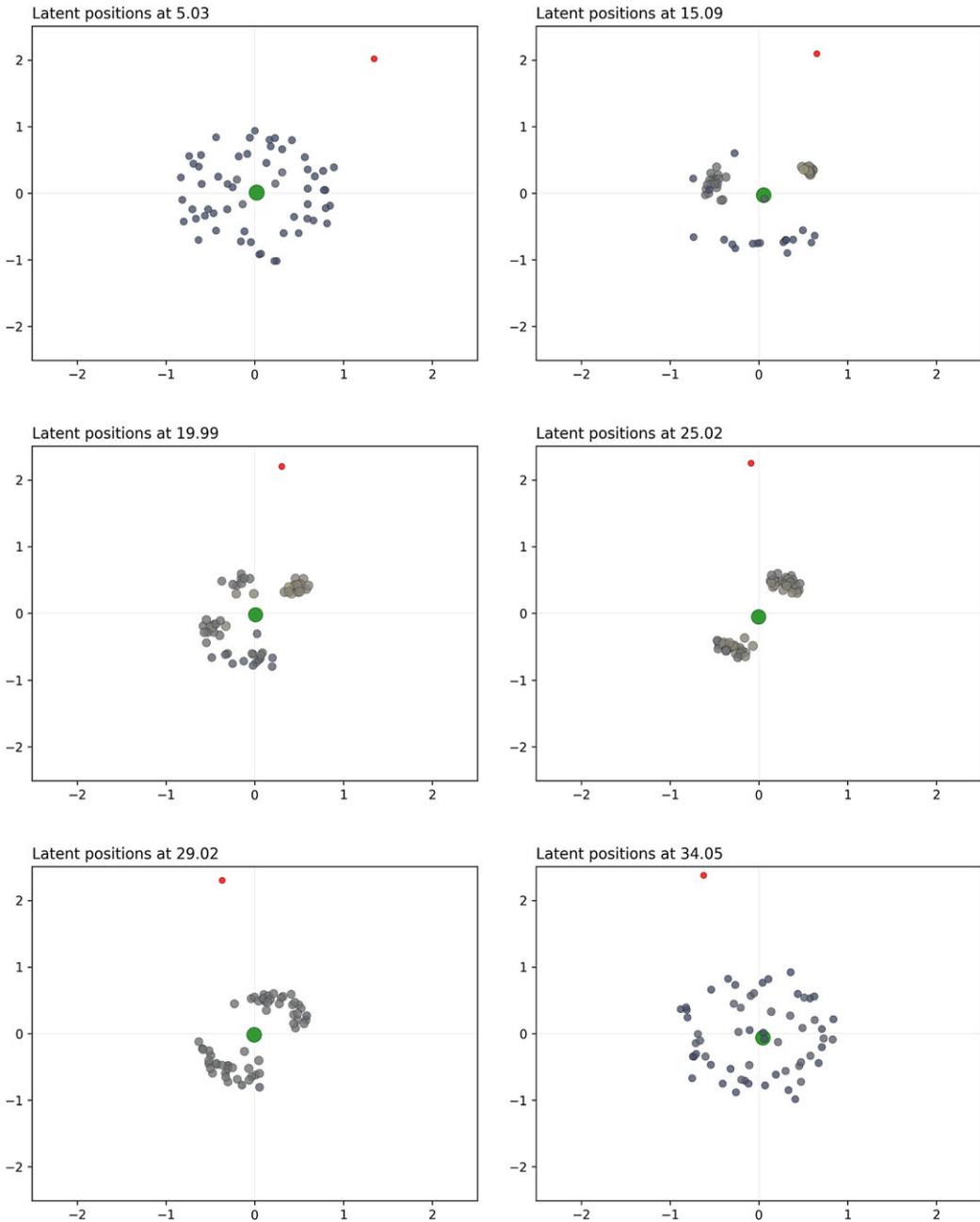


Figure 2. Simulation study 1: snapshots for the distance model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction. The hub and the isolated node are colored in green and red, respectively.

In order to make the results comparable, we divide the time interval of 40 s into 80 sub-intervals of 0.5 s each. Then, within each sub-interval, we aggregate the interaction data by creating an edge between all those nodes that have at least one interaction. By doing so, we obtain a sequence of 80 binary undirected networks, on which we fit the distance model of Hoff et al. (2002).

We propose the visual results for four sub-intervals in Figure 5. From these results, we make the following observations:

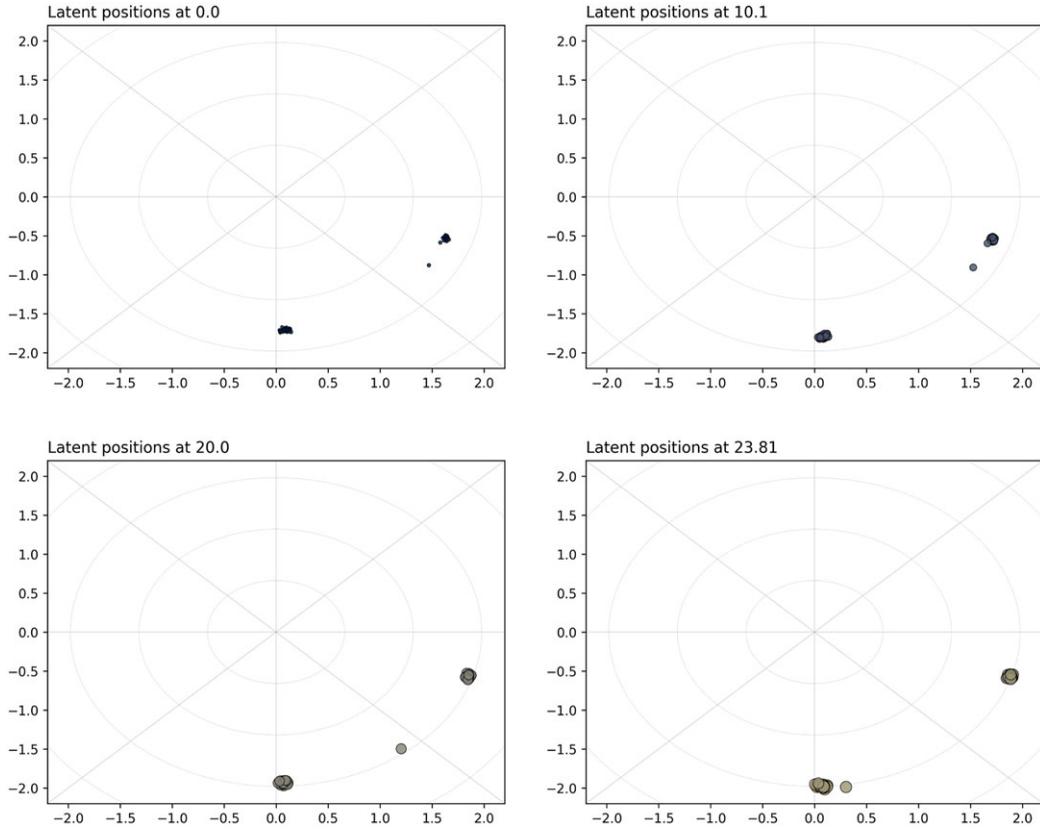


Figure 3. Simulation study 2: snapshots for the projection model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction.

- The static LPM does not capture the transition of the special node from one community to the other.
- The static LPM does not provide a model-based framework to make the snapshots comparable across time frames, due to rotations, translations, and reflections.
- The static LPM does not provide an initial strong separation of the communities, due to the adaptation that is made in discretizing the data over time.

On the other hand, the CLPM can address these issues directly by providing a continuous time evolution and thus a more accurate representation of the trajectories, without using any ad hoc data transformation.

5.3 Distance model

Simulation study 3. In this simulation study, we generate data from the latent distance model itself (Section 2.2.2). In this case, our goal can be more ambitious, and thus, we aim at reconstructing the individual trajectory of each of the nodes, at every point in time, as accurately as possible. To make the reading of the results easier, we assume that the nodes move along some pre-determined trajectories that are easy to visualize. The $N = 20$ nodes start on a ring which is centered at the origin of the space and has radius equal to 1. The nodes are located consecutively and in line along the ring, with equal space in between any two consecutive nodes. Then, they

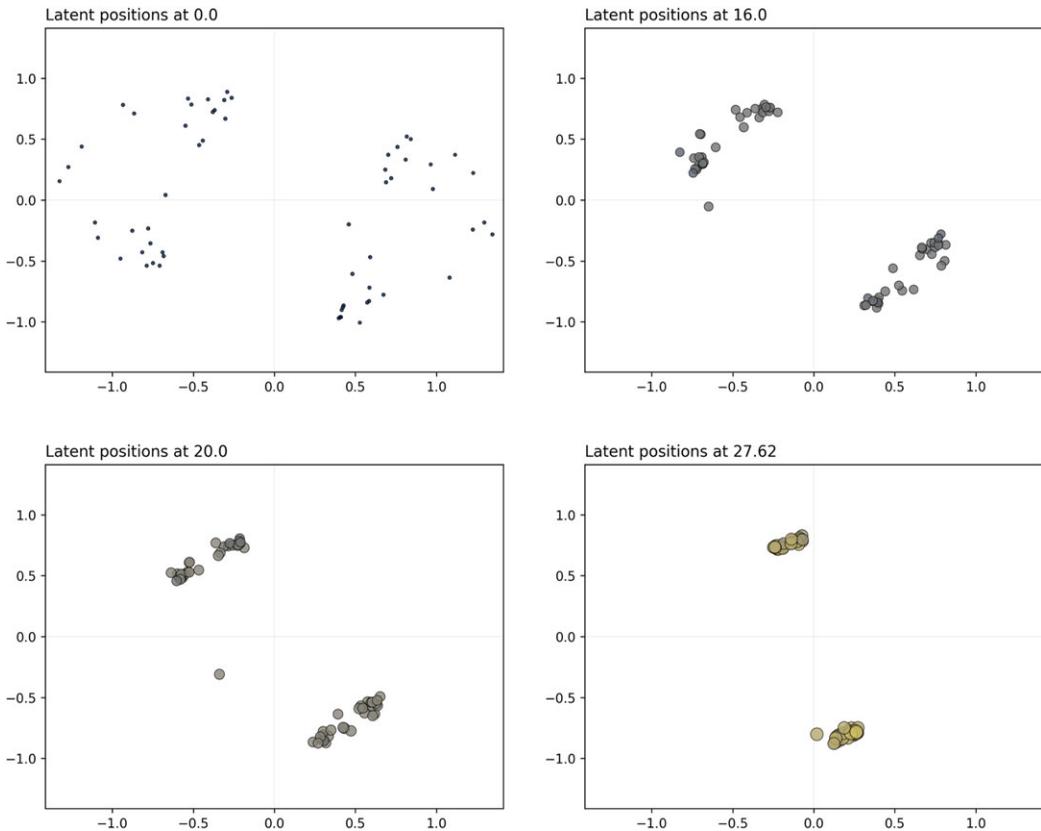


Figure 4. Simulation study 2: snapshots for the distance model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction.

start to move at constant speed toward the center of the space, which they reach after 5 s. After reaching the center, they perform the same motion backward, and they are back at their initial positions after 5 more seconds. The trajectories of the nodes make it so that, when the nodes are along the largest ring, their rate of interaction is essentially zero; however, the rate increases as they are closer and closer to the center of the space.

Figure 6 shows a collection of snapshots for the projection model. The nodes are approximately equally spaced along a line, and they progress outwards from the center of the space. As they get far apart from the center and from each other, their dot products increase and so do their interaction rates. The projection model, which is not the same model that has generated the data, tends to spread out the nodes on the space, which is ideal and expected from these data. However, this means that some of the nodes almost point in perpendicular directions, which is at odds with the fact that, halfway through the study, all nodes should interact with all others.

As concerns the results for the distance model, these are shown in Figure 7, and they highlight that the true trajectories are essentially accurately recovered. The model can capture really well the contraction and expansion of the latent space, and the individual trajectories of the nodes are closely following the theoretical counterparts. The scale of the latent space is also correctly estimated since the largest ring has approximately radius 1. In addition, the goodness of fit criterion is equal to 43.9 for the projection model and 33.6 for the distance model, thus preferring the distance model. This is an expected result since the data are in fact generated using the distance model itself.

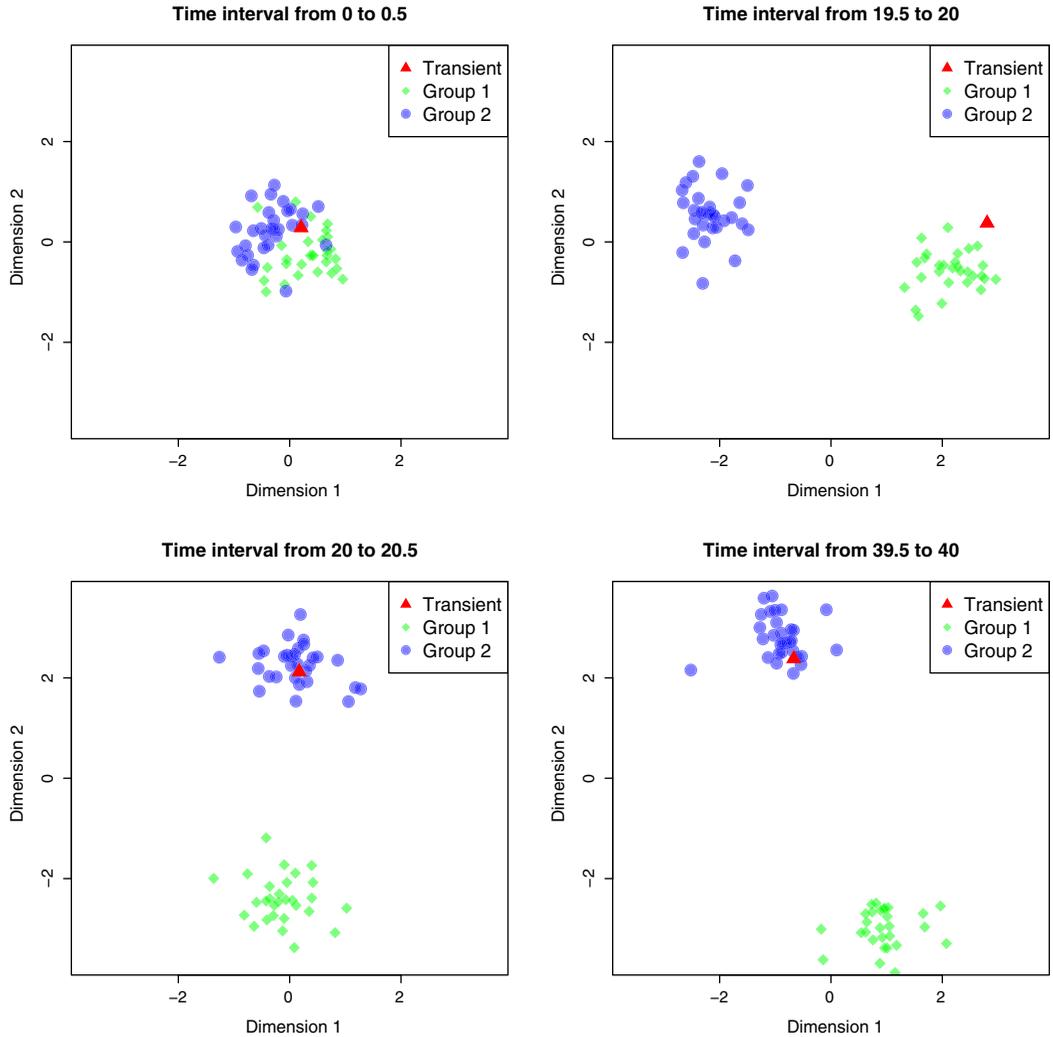


Figure 5. Simulation study 2: fitted static LPM on four sub-intervals. Colors indicate the cluster membership, with one node in red being the transient node that changes community.

There are some important remarks to make. First, after 5 s, that is, when all nodes are located close to the center, it is understandable that a rotation or reflection (with respect to the origin of the space) may happen. This is inevitable since the solution can only be recovered up to a rotation/reflection of all the latent trajectories, but also because the first 5 s and the last 5 s can technically be seen as two independent problems. The collapse to zero can be seen as a reset in terms of orientation of the latent space. That is because the penalization terms only work with two consecutive change points, so, if we view them as identifiability constraints, they would lose their effectiveness when all the nodes collapse to zero for some time. A second fundamental remark is that the estimation procedure can lead to good results only if we observe an appropriate number of interactions. This is a specific trait of LPMs in general, since we can only guess the position of one node accurately when we know to whom it connects (or, in this context, how frequently), as we would tend to locate it close to its neighbors. In our simulated setting, there are few to no interactions when nodes are along the largest ring, so it makes sense that the results seem a bit more noisy in those instants.

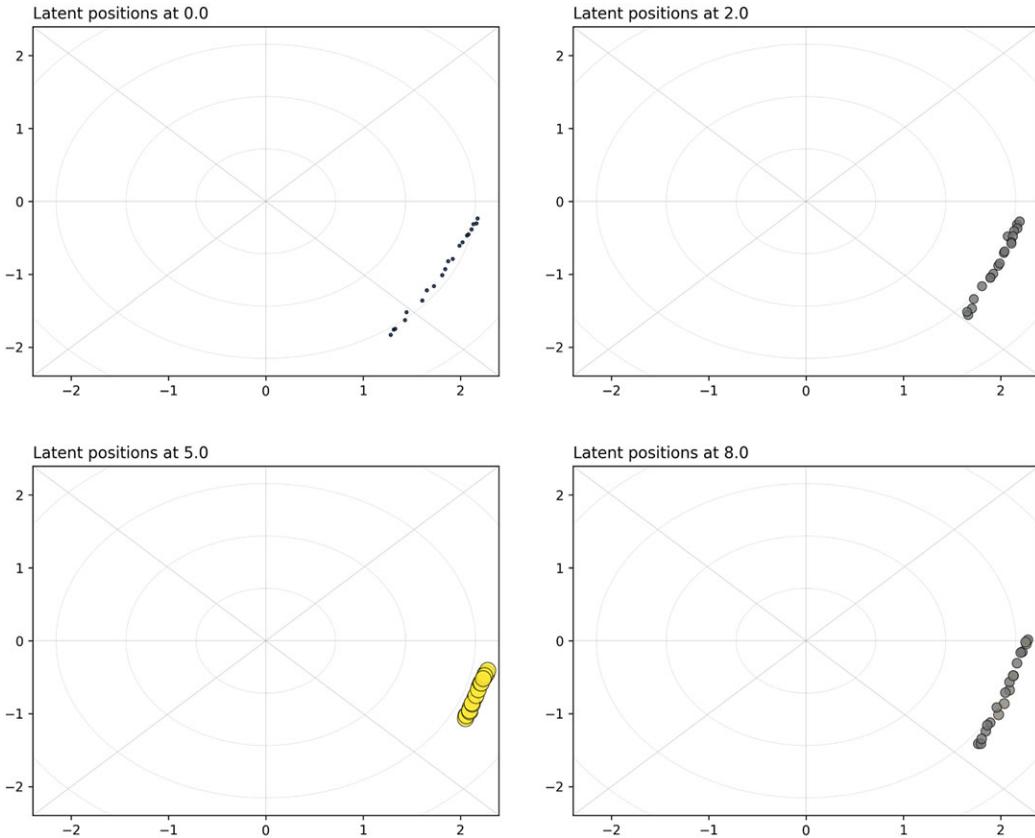


Figure 6. Simulation study 3: snapshots for the projection model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction.

5.4 Running times

We report some results aiming at quantifying the time gain due to the use of the SGD algorithm detailed in Section 3. We simulated dynamic networks according to the setup of Simulation study 2 (Section 5.1) with number of nodes varying in the range $\{30, 60, \dots, 180\}$. For each number of nodes, the distance model was first fit to the data with (mini) batch size $N/10$ and 25 epochs.⁴ These settings were checked to be sufficient to numerically reach a stationary point (\mathbf{Z}, β) stabilizing the log-likelihood. The average log-likelihood (say l) on the last epoch was computed and a full-batch GD algorithm was independently run on the same data and stopped either once reaching l or after 250 epochs. The same initial (random) values for \mathbf{Z} and β were adopted for both SGD and GD, with a learning rate of 1.0^{-4} for \mathbf{Z} and 1.0^{-7} for β . The running times (seconds) needed to reach l for each number of nodes are reported in Figure 8. As it can be seen, with $N = 180$ nodes, the full-batch GD algorithm roughly needs 16 min to converge versus 4 min required by the mini-batch SGD counterpart. The experiment was run on a DELL server PowerEdge T640, equipped with an Intel Xeon processor, 28 cores (12 cores and 27 of RAM memory available on a dedicated virtual machine), and a NVIDIA GeForce RTX 2080 graphic card. The optimization algorithms (GD and SGD) were coded in PyTorch and exploited GPU acceleration.

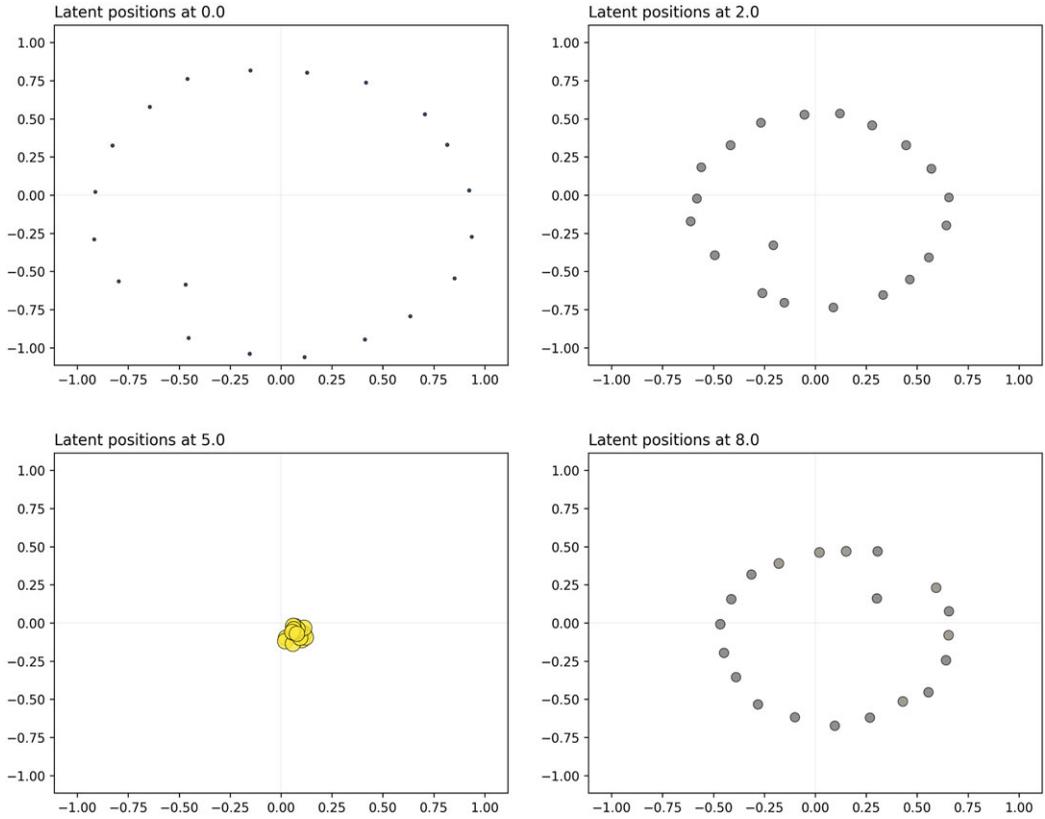


Figure 7. Simulation study 3: snapshots for the distance model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction.

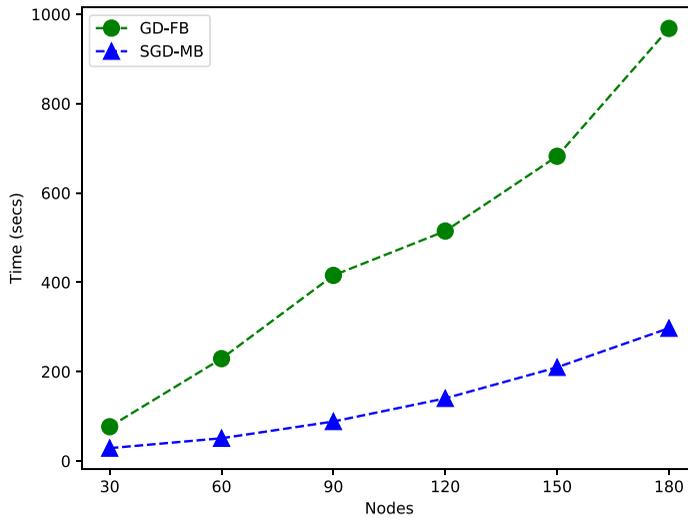


Figure 8. Time (in seconds) needed to maximize the penalized log-likelihood with full-batch GD (green) and mini-batch SGD (blue).

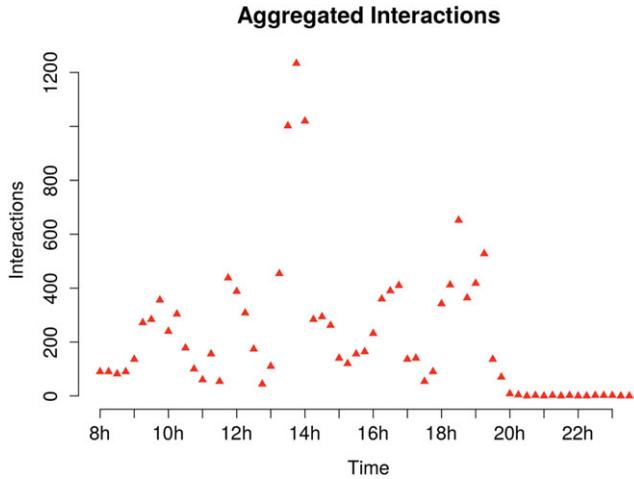


Figure 9. ACM application: cumulative number of interactions for each quarter hour (first day).

6. Applications

In this section, we illustrate our approach over three real datasets, highlighting how we can characterize the trajectories of individual nodes, the formation and dissolution of communities, and other types of connectivity patterns. From the simulation studies, we have pointed out that the distance model generally provides a more convenient and appropriate framework to study these aspects of the data. In addition, the distance model is also easier to interpret. So, since our main focus here is on model-based visualization and interpretation, we only show the results for the distance model and redirect the reader to the associated code repository where the complete results can be found also for the projection model.

6.1 ACM hypertext conference dataset

The ACM Hypertext 2009 conference was held over 3 days in Turin, Italy, from June 29 to July 1. At the conference, 113 attendees wore special badges which recorded an interaction whenever two badges were facing each other at a distance of 1.5 m or less, for at least 20 s. For each of these interactions, a timestamp was recorded as well as the identifiers of the two personal badges.

This interaction dataset was first analyzed by Isella et al. (2011) and is publicly available from [Hypertext 2009 network dataset - KONECT \(2017\)](#). Similarly to Corneli et al. (2016), we focus our analysis on the first day of the conference. On the first day, the main events that took place included a poster session in the morning (starting from 8 a.m.), a lunch break around 1 p.m., and a cheese and wine reception in the evening between 6 and 7 p.m. (Figure 9).

We use our distance CLPM to provide a graphical representation of these data and to note how the model responds to the various gatherings that happened during the day. We use 20 change points, which we found provided a sufficient level of flexibility for this application. Figures 10 and 11 show a number of snapshots highlighting some of the relevant moments of the day. The complete results, shown as a video, can be found in the code repository that accompanies this paper.

We can see that, in the morning, there is a high level of mixing between the attendees. The visitors tend to merge and split into different communities that change very frequently and very randomly. These communities reach a high level of clusteredness, which signals that the participants of the study are mixing into different groups. This is perfectly in agreement with the idea that the participants are moving from one location to another, as it usually happens during poster

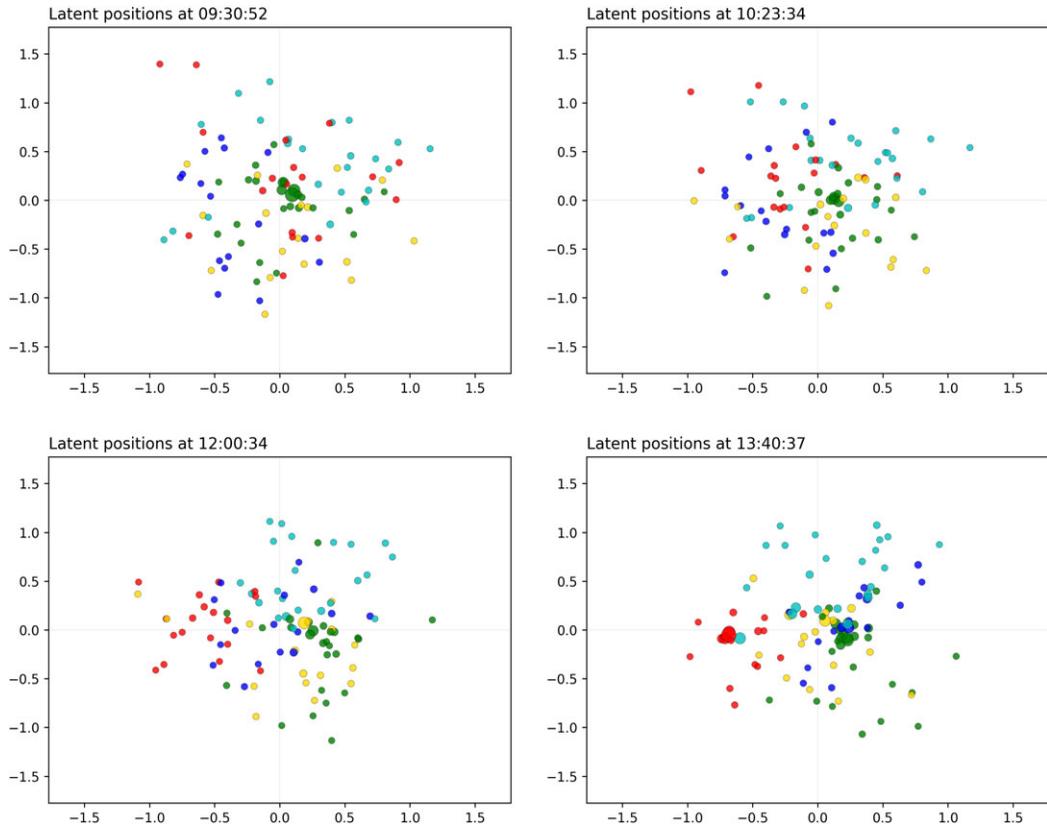


Figure 10. ACM application: snapshots for the distance model (morning hours). The sizes of the nodes reflect their current level of interaction. The colors are obtained with the spectral procedure of Section 4, with five groups.

sessions and parallel talk sessions. The nodes exhibit different types of patterns and behaviors, in that some nodes are central and tend to join many communities, whereas others have lower levels of participation and remain at the outskirts of the space.

In the late morning, we see a clear close gathering around 12 p.m., whereby almost all nodes move toward the center of the space. This is emphasized even more at 1.40 p.m., which corresponds to the lunch break. It is especially interesting that, even though the space becomes more contracted at this time, we can still clearly see a strong clustering structure.

In the afternoon, we go back to the same patterns as in the morning, whereby the participants mix in different groups and move around the space. The wine reception is also clearly captured around 6 p.m. where we see again some level of contraction of the space, to signal a large gathering of the participants.

After this event, the overall rate of interactions diminishes sharply, and as a consequence we see the nodes spreading out in the space.

In terms of clustering, we use the index introduced in Section 4 on the results for the distance model, for various threshold levels. The results are shown in Figure 12, where we can appreciate strong time-dynamic patterns. All threshold values highlight several peaks for the clustering measure, confirming the aggregations happening throughout the morning, at lunchtime, and in the evening. In addition, we can also highlight a number of smaller cycles, which are congruent with the creation and dissolution of small communities as may be observed during parallel sessions or poster presentations.

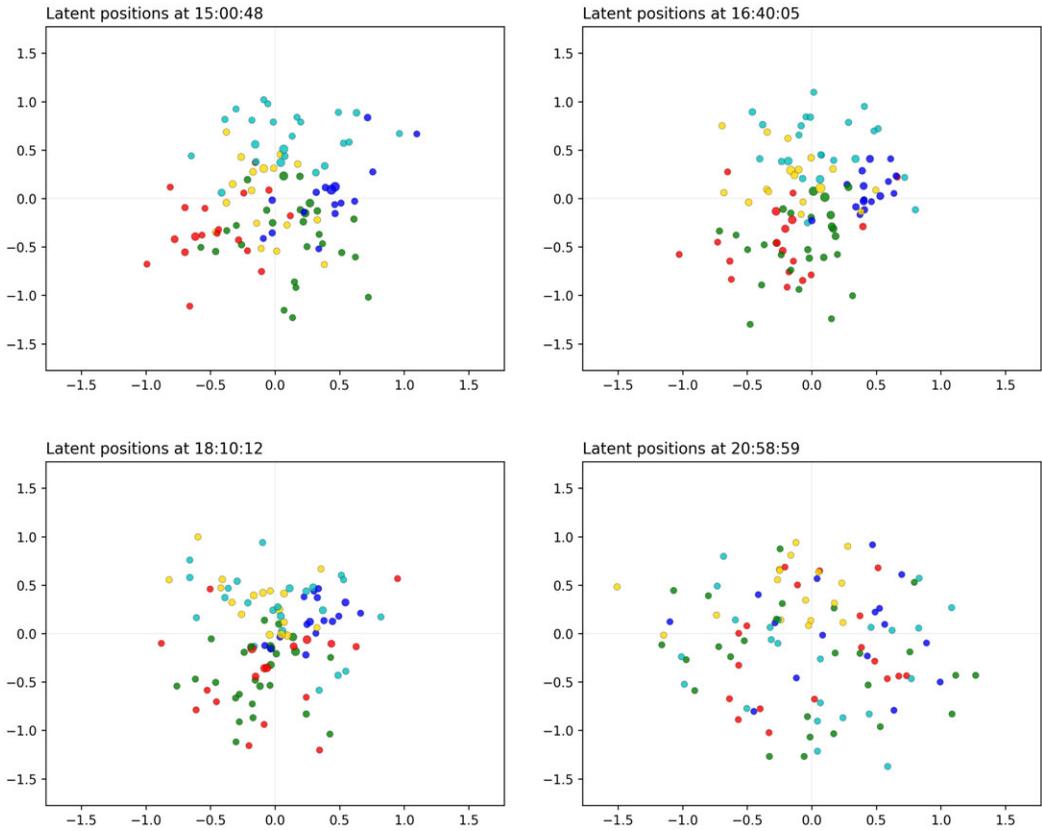


Figure 11. ACM application: snapshots for the distance model (afternoon hours). The sizes of the nodes reflect their current level of interaction. The colors are obtained with the spectral procedure of Section 4, with five groups.

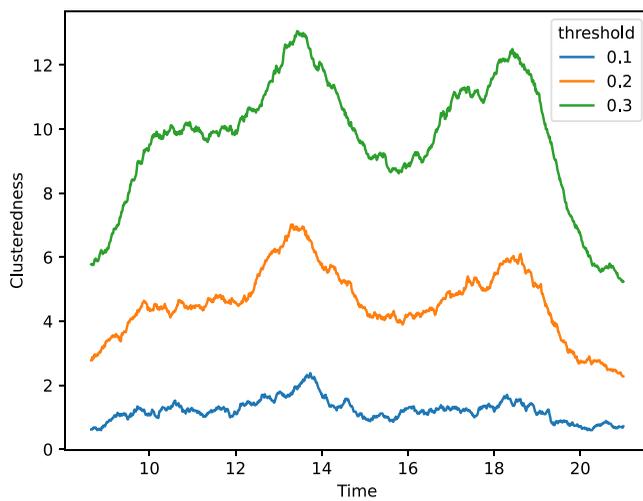


Figure 12. ACM application: clusteredness measure for various threshold values. The x -axis shows the hour of the day, whereas the y -axis shows the average number of nodes that a random node would have within the threshold distance.

6.2 Reality mining

The reality mining dataset (Eagle and Pentland, 2006) is derived from the Reality Commons project, which was run at the Massachusetts Institute of Technology from 14 September 2004 to 5 May 2005. The dataset describes proximity interactions in a group of 96 students, collected primarily through Bluetooth devices. An overview of this network dataset is also given by Rastelli (2019).

In the context of this paper, the proximity interactions can be reasonably considered as instantaneous interactions, due to the study being 9 months long. With our latent space representation, we aim at highlighting the patterns of connections of the students during the study, and any social communities that arise and how these change over time.

Figure 13 shows a few snapshots of our fitted distance CLPM. For this application, we also used 20 change points; however, all the implementation details, along with the complete results shown as a video, can be found in the code repository that accompanies this paper.

We observe that, in general, the students are quite separated and few communities arise. This does not necessarily mean that the nodes do not interact, but it is a sign that there are no sub-groups of students with an uncommonly high interaction rate. Over time, the students tend to mix in different social groups, thus quickly forming and undoing communities. This could be explained by the interactions that the students have due to college activities or other daily activities. Near the end of the study, a large cluster appears, signaling a large gathering to which the students participated. This may correspond to the period before a deadline, as outlined in Eagle and Pentland (2006).

6.3 London bikes

Infrastructure networks provide an excellent example of instantaneous interaction data. In this section, we consider a network of bike hires which is collected and publicly distributed by Transport for London: Cycle hire usage data 2012–2015 (n.d.). We focus on a specific weekend day (Sunday 6 September 2015) and study the patterns of interactions between all bike hire stations in London over 24 h, from midnight to midnight. The bike hire stations correspond to the nodes of our network, whereas an instantaneous interaction between two nodes at time t simply means that a bike started a journey from one station toward the other, at that time (we consider undirected connections). For the change points η_1, \dots, η_K , we set a change point every 15 min, for a total $K = 96$ change points over the day.

In Figure 14, we show a collection of snapshots at some critical time points during the day, for the distance model. Figure 15 shows instead the clusteredness index. The complete results, shown as a video, can be found in the code repository that accompanies this paper. Although there are a total of 818 stations that are active in this dataset, we provide a visualization for the 60 most active stations only. However, we emphasize that the results were obtained using the whole dataset. As said, we highlight with a different color the three stations with the highest number of interactions, overall. These stations are:

- Belgrove Street, King's Cross, situated next to King's Cross Square (shown in red);
- Finsbury Circus, Liverpool Street, situated next to Liverpool Street station (shown in green);
- Newgate Street, St. Paul's, situated next to St. Paul's Cathedral (shown in yellow).

The first aspect that we notice is that the latent space expands during inactive times, and it contracts during busy hours. The contractions and expansions are not homogeneous; rather they highlight the presence of dense and less dense clusters of stations.

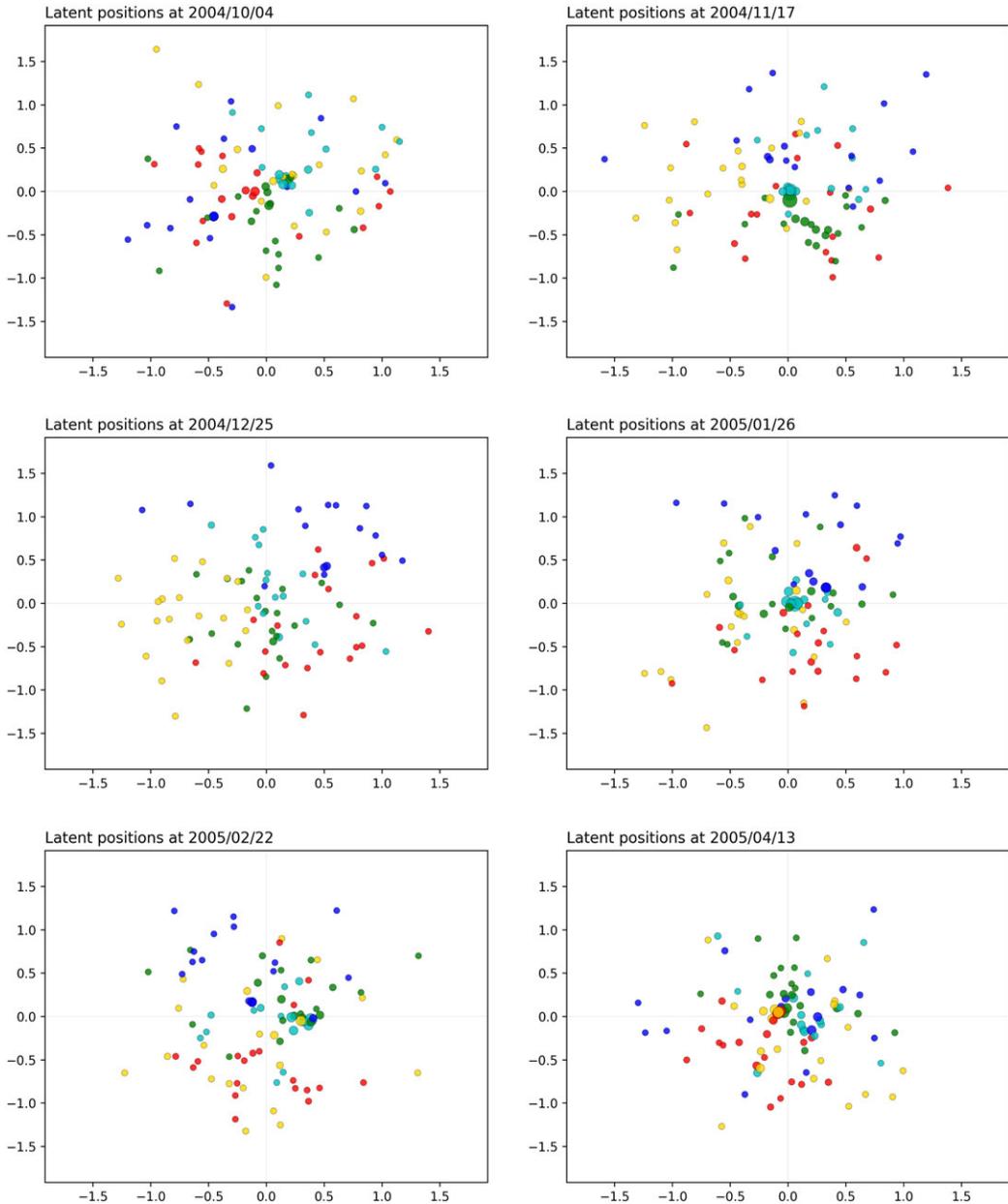


Figure 13. MIT application: snapshots for the distance model. The sizes of the nodes reflect their current level of interaction. The colors are obtained with the spectral procedure of Section 4, with five groups.

The estimate for the intercept parameter is -4.95 , and the dispersion of the points in the latent space is not particularly large. This indicates that the latent space characterization is not having a very strong effect on the rate of interactions, and the model does not capture much variability in the rates of interactions. This highlights that these connectivity data follow patterns that cannot be completely explained by the purely geometrical nature of our model. That is, the connections are determined by a variety of factors that cannot be framed into this latent position's context, and the geographical information on bike hiring accounts for only a part of the problem.

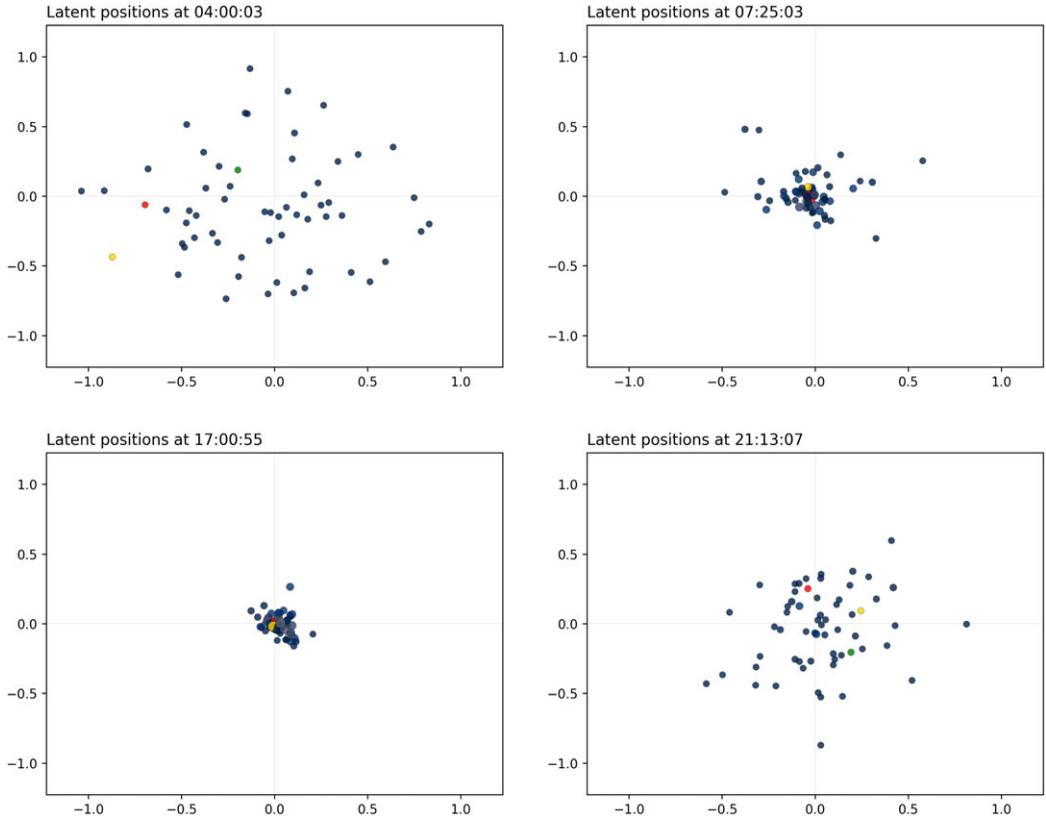


Figure 14. London bikes application: snapshots for the distance model. The sizes and colors (fading from blue to yellow) of the nodes reflect their current level of interaction. Three most active stations are shown in red, green, and yellow.

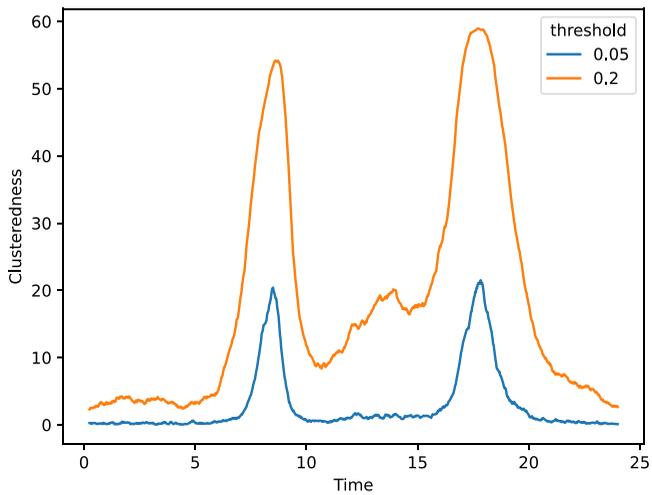


Figure 15. London bikes application: clusteredness measure for various threshold values. The x-axis shows the hour of the day, whereas the y-axis shows the average number of nodes that a random node would have within the threshold distance.

7. Conclusions and perspectives

We have introduced a new time-continuous version of the well known and widely used LPM, as an extension which can model instantaneous interactions between entities. We have proposed a new methodology which provides good flexibility while also allowing for an efficient inferential framework. The methodology is implemented in our software CLPM which accompanies this paper and is publicly available. This provides an essential additional tool for practitioners that are interested in deriving latent space visualizations from observed instantaneous interaction data.

The framework that we propose is highly inspired by the work of Hoff et al. (2002) and by the vast literature that has followed in this direction. Our work combines some crucial theoretical and statistical aspects of latent position modeling, with a pragmatic approach to inference and visualization of the results. Crucially, we provide simulation studies and real data applications to demonstrate how our method leads to sensible and accurate results, with low computational demands.

As regards extensions and future work, our research opens up several new directions, to address and potentially change some crucial parts of our procedure. Being able to assess the uncertainty around the estimated latent positions would be crucial, but it is not trivial. Indeed, as the observed data log-likelihood is invariant with respect to isometries in the latent space, sampling from the posterior distribution could lead to non-interpretable samples. For the very same reason, parametric bootstrap cannot be easily exploited. Stochastic variational inference seems to be the most valuable way to quantify the uncertainty around Z . Another fundamental challenge is related to the geometric nature of the latent space. In this paper, and in the literature cited here, affine latent spaces are considered, endowed with the standard dot product, which, in turn, induces the Euclidean distance. However, some important works in the literature of the *static* LPM consider the latent space to be spherical (McCormick and Zheng, 2015) or hyperbolic (Krioukov et al., 2010; Asta and Shalizi, 2015). As expected, since LPMs are generative models, the geometry of the latent space has crucial consequences on the properties of the simulated network (Smith et al., 2019). Indeed, recently, Lubold et al. (2020) introduced a method to consistently estimate the manifold type, dimension, and curvature from a class of latent spaces. Addressing these topics in the context of dynamic LPMs is a promising avenue of research that can extend our work. Another challenging aspect of our methodology regards inference: in this paper, we propose an optimization approach to maximize a penalized likelihood. An interesting alternative would be to consider a different approach that could allow one to also quantify uncertainty around the parameter estimates. Goodness of fit and model selection is also another critical aspect of our proposed method. In this work, we have primarily focused on data visualization and interpretation as a goal. However, many modeling decisions that we have arbitrarily taken (e.g., number of latent dimensions, penalization parameter values) could be replaced by more formal procedures that rely on model choice criteria and goodness of fit. This opens up a number of interesting research directions that we leave as potential future work. Finally, in terms of modeling, we use piecewise continuous trajectories due to their flexibility and easy tractability. However, the assumption of fixing the change points arbitrarily is a limitation of our framework, and so alternatives to our parametrization may be considered to allow the change points to be estimated from the data.

Acknowledgments The authors would like to thank Prof. Charles Bouveyron for useful feedback in the early stages of this work and also for supporting and coordinating a research visit which fostered this project.

Funding This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interests None.

Notes

1 Indeed, we could say equivalently

$$\langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle = \cos(\alpha_{ij}) \|\mathbf{z}_i(t)\| \|\mathbf{z}_j(t)\|$$

where $\alpha_{ij} \in [0, \frac{\pi}{2}]$ is the angle between $\mathbf{z}_i(t)$ and $\mathbf{z}_j(t)$ and $\|\cdot\|$ is the Euclidean norm.

2 Since our inferential approach relies on penalized likelihood optimization, it is also equivalent to a maximum-a-posteriori Bayesian framework, whereby the likelihood penalization corresponds to a prior for the latent positions. For this reason, we introduce the penalizations as prior distributions over the latent positions, noting that our inferential approach is not necessarily Bayesian since we only obtain point estimates.

3 Since we maximize a log-likelihood, gradient *ascent* is performed instead of gradient *descent*. However, since maximizing the log-likelihood is equivalent to minimize its opposite and in order to adopt the same convention as in machine/statistical learning literature we keep using the expression “gradient descent.”

4 In more details, groups of $N/10$ nodes were selected uniformly at random in the graph. The k th epoch is concluded once all nodes were selected at least k times.

References

- Asta, D. M., & Shalizi, C. R. (2015). Geometric network comparisons. In *Proceedings of the thirty-first conference on uncertainty in artificial intelligence* (pp. 102–110). Corvallis, OR: AUAI Press.
- Atkinson, K. (1991). *An introduction to numerical analysis*. New Jersey: John Wiley & Sons.
- Barrat, A., & Cattuto, C. (2013). Temporal networks of face-to-face human interactions. In P. Holme & J. Saramäki (Eds.), *Temporal networks* (pp. 191–216). Berlin: Springer.
- Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153), 1–43.
- Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, Springer (pp. 177–186).
- Cattuto, C., Van den Broeck, W., Barrat, A., Colizza, V., Pinton, J.-F., & Vespignani, A. (2010). Dynamics of person-to-person interactions from distributed RFID sensor networks. *PLoS ONE*, 5(7), e11596.
- CLPM GitHub repository (2021). <https://github.com/marcogenni/CLPM>.
- Corneli, M., Latouche, P., & Rossi, F. (2016). Exact ICL maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. *Neurocomputing*, 192, 81–91.
- Corneli, M., Latouche, P., & Rossi, F. (2018). Multiple change points detection and clustering in dynamic networks. *Statistics and Computing*, 28(5), 989–1007.
- Durante, D., & Dunson, D. B. (2014). Nonparametric bayes dynamic modelling of relational data. *Biometrika*, 101(4), 883–898.
- Durante, D., & Dunson, D. B. (2016). Locally adaptive dynamic networks. *The Annals of Applied Statistics*, 10(4), 203–2232.
- Durante, D., Dunson, D. B., & Vogelstein, J. T. (2017). Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112(520), 1–15.
- Eagle, N., & Pentland, A. S. (2006). Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4), 255–268.
- Fosdick, B. K., McCormick, T. H., Murphy, T. B., Ng, T. L. J., & Westling, T. (2019). Multiresolution network models. *Journal of Computational and Graphical Statistics*, 28(1), 185–196.
- Friel, N., Rastelli, R., Wyse, J., & Raftery, A. E. (2016). Interlocking directorates in Irish companies using a latent space model for bipartite networks. *Proceedings of the National Academy of Sciences*, 113(24), 6629–6634.
- Gollini, I., & Murphy, T. B. (2014). Joint modelling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1), 246–265.
- Gormley, I. C., & Murphy, T. B. (2007). A latent space model for rank data. In Edoardo Airoldi, David M. Blei, Stephen E. Fienberg, Anna Goldenberg, Eric P. Xing, & Alice X. Zheng (Eds.), *Statistical network analysis: Models, issues, and new directions*. Lecture Notes in Computer Science 4503 (pp. 90–102). Berlin: Springer.
- Griewank, A. (1989). On automatic differentiation. *Mathematical Programming: Recent Developments and Applications*, 6(6), 83–107.
- Halko, N., Martinsson, P.-G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2), 217–288.
- Handcock, M. S., Raftery, A. E., & Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2), 301–354.
- Hoff, P. (2008). Modeling homophily and stochastic equivalence in symmetric relational data. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in Neural Information Processing Systems* (pp. 657–664). Cambridge, MA: MIT Press.

- Hoff, P. D. (2005). Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association*, 100(469), 286–295.
- Hoff, P. D. (2018). Additive and multiplicative effects network models, *arXiv preprint arXiv:1807.08038*.
- Hoff, P. D., Raftery, A. E., & Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460), 1090–1098.
- Huang, Z., Soliman, H., Paul, S., & Xu, K. S. (2022). A mutually exciting latent space Hawkes process model for continuous-time networks. In *Uncertainty in artificial intelligence* (pp. 863–873). PMLR.
- Hypertext 2009 network dataset - KONECT (2017). <http://konect.cc/networks/sociopatterns-hypertext>.
- Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F., & den Broeck, W. V. (2011). What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1), 166–180.
- Junuthula, R., Haghdan, M., Xu, K. S., & Devabhaktuni, V. (2019). The block point process model for continuous-time event-based dynamic networks. In *The world wide web conference* (pp. 829–839). New York, NY: Association for Computing Machinery.
- Kim, B., Lee, K. H., Xue, L., & Niu, X. (2018). A review of dynamic network models with latent variables. *Statistics Surveys*, 12, 105.
- Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders, *arXiv preprint arXiv:1611.07308*.
- Klimt, B., & Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, Springer (pp. 217–226).
- Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., & Boguná, M. (2010). Hyperbolic geometry of complex networks. *Physical Review E*, 82(3), 036106.
- Krivitsky, P. N., Handcock, M. S., Raftery, A. E., & Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3), 204–213.
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791.
- Lubold, S., Chandrasekhar, A. G., & McCormick, T. H. (2020). *Identifying the latent space geometry of network models through analysis of curvature*. Technical Report, National Bureau of Economic Research.
- Matias, C., Rebafka, T., & Villers, F. (2018). A semiparametric extension of the stochastic block model for longitudinal networks. *Biometrika*, 105(3), 665–680.
- McCormick, T. H., & Zheng, T. (2015). Latent surface models for networks using aggregated relational data. *Journal of the American Statistical Association*, 110(512), 1684–1695.
- Nowicki, K., & Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455), 1077–1087.
- Park, H.-J., & Friston, K. (2013). Structural and functional brain networks: From connections to cognition. *Science*, 342(6158), 1238411.
- Raftery, A. E. (2017). Comment: Extending the latent position model for networks. *Journal of the American Statistical Association*, 112(520), 1531–1534.
- Raftery, A. E., Niu, X., Hoff, P. D., & Yeung, K. Y. (2012). Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, 21(4), 901–919.
- Rastelli, R. (2019). Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks. *Journal of the French Statistical Society*, 160(1), 35–56.
- Rastelli, R., Friel, N., & Raftery, A. E. (2016). Properties of latent variable network models. *Network Science*, 4(4), 407–432. doi: [10.1017/nws.2016.23](https://doi.org/10.1017/nws.2016.23).
- Rastelli, R., Maire, F., & Friel, N. (2018). Computationally efficient inference for latent position network models, *arXiv preprint arXiv:1804.02274*.
- Salter-Townshend, M., & McCormick, T. H. (2017). Latent space models for multiview network data. *The Annals of Applied Statistics*, 11(3), 1217–1244.
- Salter-Townshend, M., White, A., Gollini, I., & Murphy, T. B. (2012). Review of statistical network analysis: Models, algorithms, and software. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4), 243–264.
- Sarkar, P., & Moore, A. W. (2005). Dynamic social network analysis using latent space models. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems* (pp. 1145–1152) Cambridge, MA: MIT Press.
- Scharf, H. R., Hooten, M. B., Johnson, D. S., & Durban, J. W. (2018). Process convolution approaches for modeling interacting trajectories. *Environmetrics*, 29(3), e2487.
- Sewell, D. K., & Chen, Y. (2015a). Analysis of the formation of the structure of social networks by using latent space models for ranked dynamic networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(4), 611–633.
- Sewell, D. K., & Chen, Y. (2015b). Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512), 1646–1657.
- Sewell, D. K., & Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, 44, 105–116.
- Shortreed, S., Handcock, M. S., & Hoff, P. (2006). Positional estimation within a latent space model for networks. *Methodology*, 2(1), 24–33.

Smith, A. L., Asta, D. M., & Calder, C. A. (2019). The geometry of continuous latent space models for network data. *Statistical Science: A Review Journal of the Institute of Mathematical Statistics*, 34(3), 428.

Sosa, J., & Buitrago, L. (2020). A review of latent space models for social networks, *arXiv preprint arXiv:2012.02307*.

Tafakori, L., Pourkhanali, A., & Rastelli, R. (2019). Measuring systemic risk and contagion in the European financial network, *arXiv preprint arXiv:1911*.

Transport for London: Cycle hire usage data 2012-2015. <http://cycling.data.tfl.gov.uk/>. Accessed 6 September 2015.

Yang, J., Rao, V. A., & Neville, J. (2017). Decoupling homophily and reciprocity with latent space network models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. <https://www.auai.org/uai2017/accepted.php>.

A. Log-likelihood for the projection model

To make the forthcoming results clearer, we introduce the following notation for the dot products:

$$S_{ij}^{gh} := \langle \mathbf{z}_i(\eta_g), \mathbf{z}_j(\eta_h) \rangle \tag{A1}$$

for some nodes i and j , and for any $g, h \in \{1, \dots, K\}$. Most commonly, g and h will correspond to the labels of two consecutive change points; thus, we will use them to identify two breakpoints for the trajectories and to reconstruct the positions in between. Also, we denote:

$$S_{ij}(t) := \langle \mathbf{z}_i(t), \mathbf{z}_j(t) \rangle$$

for the dot product at a generic time $t \in [0, T]$.

Proposition 2. *Under the projection model, the log-likelihood is exactly equal to:*

$$\log \mathcal{L}(\beta, \mathbf{Z}) = \sum_{i,j: i < j} \left\{ \sum_{\tau_e \in \mathcal{O}_{ij}} (\beta + S_{ij}(\tau_e)) - e^\beta \sum_{g=1}^{K-1} (\eta_h - \eta_g) \int_0^1 \exp \left\{ \left[(1-t)^2 S_{ij}^{gg} + t(1-t)(S_{ij}^{gh} + S_{ij}^{hg}) + t^2 S_{ij}^{hh} \right] \right\} dt \right\} \tag{A2}$$

where $h = g + 1$.

Proof. The only non-trivial part in formula (4) regards the integral:

$$\int_0^T \exp\{\beta + \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle\} ds$$

where we have that, for $s \in [\eta_g, \eta_{g+1}]$:

$$\mathbf{z}_i(s) = (1-t)\mathbf{z}_i(\eta_g) + t\mathbf{z}_i(\eta_{g+1})$$

and, thanks to Equation (3), $t \in [0, 1]$ is such that $s = (1-t)\eta_g + t\eta_{g+1}$. Back to the calculation of the integral, we can decompose this into $K - 1$ integrals using the K change points as follows:

$$\int_0^T \exp\{\beta + \langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle\} ds = e^\beta \sum_{g=1}^{K-1} \int_{\eta_g}^{\eta_{g+1}} \exp\{\langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle\} ds$$

and then, in each integral, we apply the transformation $s = (1-t)\eta_g + t\eta_{g+1}$ and set $h = g + 1$ to obtain:

$$\begin{aligned} \int_{\eta_g}^{\eta_h} \exp\{\langle \mathbf{z}_i(s), \mathbf{z}_j(s) \rangle\} ds &= (\eta_h - \eta_g) \int_0^1 \exp \left\{ \langle \mathbf{z}_i((1-t)\eta_g + t\eta_h), \mathbf{z}_j((1-t)\eta_g + t\eta_h) \rangle \right\} dt \\ &= (\eta_h - \eta_g) \int_0^1 \exp \left\{ \left[(1-t)^2 S_{ij}^{gg} + t(1-t)(S_{ij}^{gh} + S_{ij}^{hg}) + t^2 S_{ij}^{hh} \right] \right\} dt \end{aligned}$$

where S_{ij}^{gh} is defined in Equation (A1). Since the S terms do not depend on t , the integral can be efficiently approximated using a composite Simpson’s rule (see equation 5.1.16 of Atkinson (1991)). \square

B. Log-likelihood for the distance model

We focus on the integral in Equation (6) and prove that it can be explicitly solved.

Proof. In force of Equation (3), it reads

$$\begin{aligned}
 & e^\beta \left(\sum_{g=1}^{K-1} \int_0^1 \exp \left\{ - \left\| (1-t)(\mathbf{z}_i(\eta_g) - \mathbf{z}_j(\eta_g)) + t(\mathbf{z}_i(\eta_{g+1}) - \mathbf{z}_j(\eta_{g+1})) \right\|^2 \right\} (\eta_{g+1} - \eta_g) dt \right) \\
 &= e^\beta \left(\sum_{g=1}^{K-1} (\eta_{g+1} - \eta_g) \int_0^1 \exp \left\{ - \left\| t(\Delta_i^g - \Delta_j^g) + (\mathbf{z}_i^g - \mathbf{z}_j^g) \right\|_2^2 \right\} dt \right)
 \end{aligned} \tag{B1}$$

where the variable change $t = \frac{s-\eta_g}{\eta_{g+1}-\eta_g}$ was performed and the following notations were adopted to simplify the exposition

$$\begin{aligned}
 \Delta_i^g &:= \mathbf{z}_i(\eta_{g+1}) - \mathbf{z}_i(\eta_g) \\
 \Delta_j^g &:= \mathbf{z}_j(\eta_{g+1}) - \mathbf{z}_j(\eta_g) \\
 \mathbf{z}_i^g &:= \mathbf{z}_i(\eta_g) \\
 \mathbf{z}_j^g &:= \mathbf{z}_j(\eta_g)
 \end{aligned}$$

By denoting $f(t) := - \left\| t(\Delta_i^g - \Delta_j^g) + (\mathbf{z}_i^g - \mathbf{z}_j^g) \right\|_2^2$ the exponent inside the integral, we can “complete the square” as follows

$$\begin{aligned}
 f(t) &= - \left\| \Delta_i^g - \Delta_j^g \right\|^2 \left(t^2 + 2t \left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \frac{\mathbf{z}_i^g - \mathbf{z}_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|} \right\rangle + \frac{\left\| \mathbf{z}_i^g - \mathbf{z}_j^g \right\|^2}{\left\| \Delta_i^g - \Delta_j^g \right\|^2} \right) \\
 &= - \left\| \Delta_i^g - \Delta_j^g \right\|^2 \left(t - \left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \frac{\mathbf{z}_j^g - \mathbf{z}_i^g}{\left\| \Delta_i^g - \Delta_j^g \right\|} \right\rangle \right)^2 \\
 &\quad - \left(\left\| \mathbf{z}_i^g - \mathbf{z}_j^g \right\|^2 - \left(\left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \mathbf{z}_j^g - \mathbf{z}_i^g \right\rangle \right)^2 \right) \\
 &= - \frac{1}{2(\sigma^2)_{ij}^g} (t - \mu_{ij})^2 - \left(\left\| \mathbf{z}_i^g - \mathbf{z}_j^g \right\|^2 - (\left\| \Delta_i^g - \Delta_j^g \right\| \mu_{ij}^g)^2 \right)
 \end{aligned}$$

where

$$\begin{aligned}
 \mu_{ij}^g &:= \left\langle \frac{\Delta_i^g - \Delta_j^g}{\left\| \Delta_i^g - \Delta_j^g \right\|}, \frac{\mathbf{z}_j^g - \mathbf{z}_i^g}{\left\| \Delta_i^g - \Delta_j^g \right\|} \right\rangle, \\
 (\sigma^2)_{ij}^g &:= \frac{1}{2 \left\| \Delta_i^g - \Delta_j^g \right\|^2}.
 \end{aligned} \tag{B2}$$

By plugging all this into Equation (B1), it follows that

$$\int_0^T e^{\beta - \|\mathbf{z}_i(t) - \mathbf{z}_j(t)\|_2^2} dt =$$

$$= \sqrt{2\pi} e^{\beta} \left\{ \sum_{g=1}^{K-1} e^{-(\|\mathbf{z}_i^g - \mathbf{z}_j^g\|^2 - (\|\Delta_i^g - \Delta_j^g\| \mu_{ij}^g)^2)} \sigma_{ij}^g (\eta_{g+1} - \eta_g) \left[\Phi \left(\frac{1 - \mu_{ij}^g}{\sigma_{ij}^g} \right) - \Phi \left(\frac{0 - \mu_{ij}^g}{\sigma_{ij}^g} \right) \right] \right\}$$

(B3)

□