Electronic Journal of Statistics
Vol. 18 (2024) 2531–2570
ISSN: 1935-7524
https://doi.org/10.1214/24-EJS2256

Computationally efficient inference for latent position network models

Riccardo Rastelli

School of Mathematics and Statistics, University College Dublin,
e-mail: riccardo.rastelli@ucd.ie

Florian Maire

Department of Mathematics and Statistics, Université de Montréal,
e-mail: florian.maire@umontreal.ca

and

Nial Friel

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

Abstract: Latent position models are widely used for the analysis of networks in a variety of research fields. In fact, these models possess a number of desirable theoretical properties, and are particularly easy to interpret. However, statistical methodologies to fit these models generally incur a computational cost which grows with the square of the number of nodes in the graph. This makes the analysis of large social networks impractical. In this paper, we propose a new method characterised by a much reduced computational complexity, which can be used to fit latent position models on networks of several tens of thousands nodes. Our approach relies on an approximation of the likelihood function, where the amount of noise introduced by the approximation can be arbitrarily reduced at the expense of computational efficiency. We establish several theoretical results that show how the likelihood error propagates to the invariant distribution of the Markov chain Monte Carlo sampler. In particular, we demonstrate that one can achieve a substantial reduction in computing time and still obtain a good estimate of the latent structure. Finally, we propose applications of our method to simulated networks and to a large coauthorships network, highlighting the usefulness of our approach.

Keywords and phrases: Network analysis, latent position models, noisy Markov chain Monte Carlo, Bayesian inference, social networks.

Received March 2023.

1. Introduction

In the last few decades, network data has become extremely common and readily available in a variety of fields, including the social sciences, biology, finance and technology. After the pioneering work of [11], latent position models (hereafter LPMs) have become one of the cornerstones in the statistical analysis of networks. LPMs are flexible models capable of capturing many salient features of realised networks while providing results which can be easily interpreted. However, a crucial aspect in the statistical analyses of networks is scalability: the computational burden required when fitting LPMs generally grows with the square of the number of nodes. This seriously hinders their applicability, since estimation becomes impractical for networks larger than a few hundreds nodes. Here, we precisely address this issue by introducing a new scalable methodology to fit LPMs: we study the new approach by providing theoretical guarantees on its efficiency, and we illustrate its use on simulated and real datasets.

LPMs postulate that the nodes of an observed network are characterised by a unique random position in a latent space: in the most common setup, each node is mapped to a point of $\mathbb{R}^2$. Additionally, the probability of observing an edge between two nodes is determined by the corresponding pairwise latent distance. A common assumption requires that closer nodes are more likely to connect than nodes farther apart, or, equivalently, that the probability of connection $\rho\left(d_{ij}\right)$ is a non-increasing function of the distance d_{ij} between nodes i and j. Evidently, the aforementioned quadratic computing costs originate from the necessity of keeping track of all of the pairwise distances between the nodes.

In our approach, we construct a partition of the latent space, therefore inducing a partition on the nodes of the graph itself. This allows us to cluster together nodes that are expected to have approximately the same behaviour, with regard to their connections. In principle, this is similar to imposing a stochastic block model structure [39], whereby the nodes belonging to the same block are assumed to be *stochastically equivalent* [20]. The crucial advantage of our approach is that, once the partitioning has been set up, we can bypass the calculation of the pairwise distances via an approximation, in fact reducing to a computational complexity (in the number of nodes) that is lower than than of the standard methods.

Similarly to the original paper of [11], our approach also relies on Markov chain Monte Carlo (hereafter MCMC) to obtain a Bayesian posterior sample of the latent positions and other model parameters. However, in contrast to their approach, we replace the likelihood of the LPM with an approximate (hence *noisy*) counterpart that aggregates the latent position of nodes belonging to the same block. By construction, the cost of the calculation of this surrogate likelihood grows linearly in the number of nodes, hence giving a significant computational advantage to our method when compared to the approach of [11] or other subsequent related works.

Since the LPM likelihood is replaced by a proxy, our method broadly fits within the context of *noisy* Markov chain Monte Carlo [1], a topic that has recently generated a noticeable interest within the field of computational statistics and beyond. The theoretical aspect of our paper relies and builds upon the core ideas of noisy MCMC. In particular, our methodology is supported by a collection of theoretical results showing that our approach leads to quantifiable gains in efficiency. More precisely, we show that the error in the MCMC output induced by the likelihood approximation can be arbitrarily bounded by refining the partition in the latent space. Besides, a finer partition also implies higher computational costs. As a consequence, our algorithm allows a trade-off between speed and accuracy that can be set according to the available computational budget, and the level of precision required for inference. We study in detail how this approximation error is affected by the fineness of the partition as the number of nodes increases, hence providing a detailed characterisation of the trade-off. In addition, our theoretical developments include a proposition that can be regarded as an extension of the results of [1] to the widely used Metropolis-Hastings (MH hereafter) algorithm, and which may thus have applications beyond the context of LPMs. The theoretical results are established for a generic LPM framework: the assumptions we use are rather general and encompass most of the commonly used LPMs.

In addition to these results, we propose applications of our method to both simulated and real datasets, whereby we focus on a more specific model which is equivalent to that of [11]. Our simulation study aims at assessing the approximation error bounds from a much more practical perspective, highlighting the validity of the procedure in asymptotic settings, and providing useful indications on how one should set up the partitioning. The study demonstrates that the noisy algorithm succeeds in recovering the latent structure correctly, achieving the same qualitative results obtained with the currently available approaches. Crucially, the computing time required by our proposed approach is only a fraction of that of the non-noisy one.

Finally, we propose an application to a large social network representing coauthorships in the astrophysics category of the repository of electronic preprints, arXiv. This application demonstrates that our approach can be successfully employed on very large[1] networks, to recover the structure of the latent space while using just a small fraction of the actual computational cost. This effectively extends the applicability of latent position models to much larger network datasets.

The structure of the paper is as follows: in Section 2 we give an overview of the literature related to LPMs and noisy Markov chain Monte Carlo. In Section 3, we formally characterise the main features of the original LPM of [11], giving an overview of the MH sampling strategy used to perform inference, highlighting some of its limitations. In Section 4, we lay the foundations for our theoretical results, by defining the general assumptions that our LPMs must satisfy. In Section 5, we formally introduce the partitioning of the latent space and all of the associated notation. Section 6 introduces the novel noisy algorithm, whereas in Section 7 we describe the main theoretical results. Finally, Sections 8 and 9 illustrate the applications of our methodology to simulated and a real dataset, respectively.

2. Review of related literature

The study of the mathematical properties of LPMs dates back at least to [6]. However, the first application of these models in the statistical analysis of social networks is due to [11], who introduced a feasible methodology to fit LPMs to

[1] This is meant in comparison to previous other analyses in the LPM literature.

interaction data. Since the work of [11], LPMs have been intensively studied and widely applied to a variety of contexts, becoming one of the prominent statistical models for network analyses. There are a number of reasons for this success. Most importantly, LPMs are particularly easy to interpret, and offer a clear and intuitive graphical representation of the results. In addition, LPMs are capable of capturing a number of features of interest such as transitivity, clustering, homophily and assortativity, which are often exhibited by observed social networks. An overview of the theoretical properties of realised LPMs is given in [25].

In order to increase the flexibility of these models, a number of extensions of the basic framework have been considered. [10] introduce a more sophisticated prior on the latent point process to represent clustering in the network, that is, the presence of communities. [13] further extends the model to include nodal random effects, i.e. additional latent features on the nodes capable of tuning their in-degrees and out-degrees. Both of these extensions are implemented in the `R` package `latentnet`.

LPMs have also been extended to account for multiple network views [8, 4, 31], binary interactions evolving over time [34, 35, 5, 3], ranking network data [9, 36] and weighted networks [37]. Review papers dealing with LPMs include [33], [16] and [22].

Similarly to our contribution, three other papers address the issue of scalability for the inference on LPMs. In [32], the authors propose a variational approximation (coupled with first order Taylor expansions to deal with various intractabilities) to perform posterior maximisation for the model described by [10]. One drawback of this approach is that it is not possible to assess the magnitude of the error induced by the variational approximation. Also, the modelling assumptions are not flexible, since the variational framework can only be used with a restricted selection of parametric distributions.

In [30], the authors consider the same latent position clustering model, and propose a Gaussian finite mixture prior distribution on the latent point process that allows one to *collapse* the posterior distribution. This means that several model parameters can be analytically integrated out from the posterior distribution of the model, hence simplifying the sampling scheme and achieving better estimators with a smaller computational cost.

Finally, [24] proposes a case-control likelihood approximation for the LPM with nodal random effects. In this paper, the authors argue that the majority of large social networks are sparse, hence, absent edges contribute the most to the LPM likelihood. By analogy with the case-control idea from epidemiology, they estimate the likelihood value using only a subset of the contributions given by the absent edges. We consider this approach similar to ours, since both methods rely on a noisy likelihood. We point out that our algorithm benefits from a series of theoretical results that guarantee its correctness and characterise the error induced by the approximation. In addition, our method may be applied to networks of potentially huge size regardless of the level of sparseness.

Regarding the theoretical analysis of our algorithm, the main reference that we relate to is [1]. These authors argue that the computational problems arising when dealing with intractable likelihoods, or when inferring very large datasets, can often be alleviated by introducing approximations in the MCMC schemes. These approaches are generally referred to as noisy MCMC, since one ends up sampling using a noisy transition kernel, rather than the correct one. In [1], the authors exploit a theoretical result from [17] to characterise the error induced by these approximations on the invariant distribution of the transition kernel. They also propose several applications based on the Metropolis-Hastings algorithm to a number of relevant statistical modelling frameworks. We also point out that, more recently, the noisy Monte Carlo framework has been adopted by [2] and [15], as a means to speed up inference for Gibbs random fields and other general models. Even though the literature on noisy MCMC has been recently enriched by a number of relevant entries [18, 12, 29], the theoretical framework developed in [1] proved sufficient to establish our results, as shown in Section 7.

3. Latent position models

3.1. Definition

A random graph is an object $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{1, \dots, N\}$ is a fixed set of labels for the nodes and $\mathcal{E}$ is a list of the randomly realised edges. In the social sciences, for example, random graphs are used to represent the social interactions within a set of actors. The values appearing on the undirected ties are modeled through the random variables:

$$\mathcal{Y} = \{Y_{ij} : i, j \in \mathcal{V},\ i < j\}. \tag{3.1}$$

In this paper we only deal with undirected binary graphs, hence, the observed realisations are denoted as follows:

$$y_{ij} = \begin{cases} 1, & \text{if an edge between } i \text{ and } j \text{ appears;} \\ 0, & \text{otherwise;} \end{cases} \tag{3.2}$$

for every $i \in \mathcal{V}$ and $j \in \mathcal{V}$ such that $j > i$. Note that, in the framework considered, self-edges are not modelled.

In LPMs the nodes are characterised by a latent position, generically denoted $\mathbf{z} \in \mathbb{R}^m$, which determines their social profile. The choice $m = 2$ is the most common since it usually couples a good fit and a convenient framework to represent the results. Hence, we illustrate our methodology assuming that the number of latent dimensions is two, noting that extensions to other cases may be possible.

In the basic LPM, the probability of an edge appearing is determined by the positions of the nodes at its extremes and by some other global parameters (e.g. an intercept). This may be formally written as follows:

$$p\left(\mathbf{z}_i, \mathbf{z}_j; \boldsymbol{\psi}\right) := \mathbb{P}\left(y_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j, \boldsymbol{\psi}\right) = 1 - \mathbb{P}\left(y_{ij} = 0 | \mathbf{z}_i, \mathbf{z}_j, \boldsymbol{\psi}\right). \tag{3.3}$$

Here $\boldsymbol{\psi}$ is a vector of global parameters with dimensions indexed by the labels $\mathcal{K} = \{1, \dots, K\}$. The parameter $\boldsymbol{\psi}$ is sometimes referred to as the static parameter of the model, as opposed to the latent field $\mathcal{Z} := \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$. A number of possible formulations for the edge probabilities have been proposed. Within the statistical community, the most common choice is the logit link proposed by [11]:

$$\log\left(\frac{p\left(\mathbf{z}_i,\mathbf{z}_j;\psi\right)}{1-p\left(\mathbf{z}_i,\mathbf{z}_j;\psi\right)}\right) := \psi - d\left(\mathbf{z}_i,\mathbf{z}_j\right); \tag{3.4}$$

where $d\left(\mathbf{z}_i,\mathbf{z}_j\right)$ denotes the Euclidean distance between the two nodes, and $\psi \in \mathbb{R}$ may be seen as an intercept parameter ($K = 1$).[2] Alternative formulations are used in [8] and [25]. In physics, a variety of edge probability functions have been proposed. A list of these can be found, for example, in [21] and references therein. One feature that all of these formulations have in common is that the edge probability is a function of the distance between the two nodes, and that its value decreases as the latent distance increases, making long edges less likely to appear.

Since the data observations are conditionally independent given the latent positions, the likelihood of all undirected LPMs may be written as:

$$\mathcal{L}_{\mathcal{Y}}\left(\mathcal{Z},\boldsymbol{\psi}\right) = \prod_{\{i\in\mathcal{V}\}}\prod_{\{j\in\mathcal{V}\setminus i\}}\left\{\left[p\left(\mathbf{z}_i,\mathbf{z}_j;\boldsymbol{\psi}\right)\right]^{y_{ij}}\left[1-p\left(\mathbf{z}_i,\mathbf{z}_j;\boldsymbol{\psi}\right)\right]^{1-y_{ij}}\right\}^{1/2} \tag{3.5}$$

where the square root is introduced to remedy the fact that each edge contributes twice to the likelihood of the undirected network (the motivation behind this particular formulation will be more clear in the following sections). We note that, for a given set of positions $\mathcal{Z}$ and global parameters $\boldsymbol{\psi}$, the computational cost for the likelihood evaluation is $\mathcal{O}\left(N^2\right)$, i.e. it grows with the square of the number of nodes.

3.2. Bayesian inference

Inference for LPMs is usually carried out in a Bayesian framework, using MCMC to obtain posterior samples of the model parameters [11, 10, 13, 24]. The posterior density of interest is:

$$\pi\left(\mathcal{Z},\boldsymbol{\psi}|\mathcal{Y}\right) \propto \mathcal{L}_{\mathcal{Y}}\left(\mathcal{Z},\boldsymbol{\psi}\right)\pi\left(\mathcal{Z}\right)\pi\left(\boldsymbol{\psi}\right). \tag{3.6}$$

Assuming that the cost of the evaluation of the priors $\pi\left(\mathcal{Z}\right)$ and $\pi\left(\boldsymbol{\psi}\right)$ is $\mathcal{O}\left(N\right)$ or negligible, the computational cost required to evaluate the posterior value grows with N^2, which corresponds to the bottleneck imposed by the likelihood term. A Markov chain Monte Carlo sampler can be designed to sample each of the model parameters in turn, using the following full-conditional density:

$$\pi\left(\mathbf{z}_i|\mathcal{Z}_{-i},\boldsymbol{\psi},\mathcal{Y}\right) \propto \pi\left(\mathbf{z}_i\right)\prod_{\{j\in\mathcal{V}:\ j\neq i\}}\left[p\left(\mathbf{z}_i,\mathbf{z}_j;\boldsymbol{\psi}\right)\right]^{y_{ij}}\left[1-p\left(\mathbf{z}_i,\mathbf{z}_j;\boldsymbol{\psi}\right)\right]^{1-y_{ij}} \tag{3.7}$$

[2] In the context of a generalised linear model, we can interpret the parameter ψ as an intercept for the model. However it is important to point out that this parameter determines also other aspects of the model. Since the latent distances are lower-bounded by zero, ψ also determines the highest edge probability value that we may obtain (in fact, corresponding to a latent distance of zero. As a consequence, generally speaking, ψ directly regulates the overall density of the network. In this paper we refer to this parameter as the intercept.

$$\pi\left(\psi_k \middle| \boldsymbol{\psi}_{-k}, \mathcal{Z}, \mathcal{Y}\right) \propto \pi\left(\psi_k\right) \mathcal{L}_{\mathcal{Y}}\left(\mathcal{Z}, \boldsymbol{\psi}\right) \tag{3.8}$$

In the previous equations: $i \in \mathcal{V}$, $k \in \mathcal{K}$, whereas $\mathcal{Z}_{-i} = \left\{\mathbf{z}_j\right\}_{j \in \mathcal{V} \setminus \{i\}}$ and $\boldsymbol{\psi}_{-k} = \left\{\psi_{k'}\right\}_{k' \in \mathcal{K} \setminus \{k\}}$. Here we have assumed that the model parameters are all independent a priori: this is indeed very common and it will be formalised in the following sections. Each evaluation of (3.8) clearly requires $\mathcal{O}\left(N^2\right)$. Since each evaluation of (3.7) requires $\mathcal{O}\left(N\right)$ calculations, the overall complexity of the sampler still grows with the square of N.

The full-conditionals (3.7) and (3.8) are generally not in standard form. Hence, new values for the model parameters are sampled through what is usually referred to as a Metropolis-Hastings (MH) type algorithm (see *e.g.* [7]). More precisely, potential new parameters are drawn from proposal densities $q_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right)$ and $q_{\boldsymbol{\psi}}\left(\psi_k \to \psi_k'\right)$ and are then accepted with probability:

$$\alpha_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right) := 1 \wedge \left\{\frac{q_{\mathcal{Z}}\left(\mathbf{z}_i' \to \mathbf{z}_i\right) \pi\left(\mathbf{z}_i' | \mathcal{Z}_{-i}, \boldsymbol{\psi}, \mathcal{Y}\right)}{q_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right) \pi\left(\mathbf{z}_i | \mathcal{Z}_{-i}, \boldsymbol{\psi}, \mathcal{Y}\right)}\right\} \tag{3.9}$$

$$\alpha_{\boldsymbol{\psi}}\left(\psi_k \to \psi_k'\right) := 1 \wedge \left\{\frac{q_{\boldsymbol{\psi}}\left(\psi_k' \to \psi_k\right) \pi\left(\psi_k' \middle| \boldsymbol{\psi}_{-k}, \mathcal{Z}, \mathcal{Y}\right)}{q_{\boldsymbol{\psi}}\left(\psi_k \to \psi_k'\right) \pi\left(\psi_k \middle| \boldsymbol{\psi}_{-k}, \mathcal{Z}, \mathcal{Y}\right)}\right\} \tag{3.10}$$

for the latent positions and global parameters, respectively. In the previous equations, for two real numbers a and b, $a \wedge b$ stands for the minimum between the two numbers. Also, we point out that, as is common practice, the two dimensions of the latent positions are dealt with simultaneously, i.e. they are updated in block.

The MH sampler described above defines a Markov chain whose stationary distribution is the posterior of interest (3.6). As a consequence, provided that the Markov chain is ergodic, the samples obtained at stationarity can be used to fully characterise the posterior distribution of interest. In fact, the MH chain is shown to be geometrically ergodic for a variety of proposal distributions and under some regulatory conditions on the invariant distribution π, see [27, Theorem 5].

3.3. *Non-identifiability of the latent positions*

LPMs are known to be non-identifiable with respect to translations, rotations, and reflections of the latent positions. This issue has no particular effect on the sampling itself, yet it may hinder the interpretation of the posterior samples. For this reason, the latent positions are usually post-processed using the so-called Procrustes' matching. This procedure consists of rotating and translating the configurations of points observed at the end of each iteration, to match a given reference layout. In this way, the trajectory of each node during the sampling may be properly assessed, since the overall rotation and translation effect has been removed. A detailed description of the method is given, for example, in [11] and [38]. In this paper, we adopt exactly this same strategy to solve the non-identifiability problem, using as reference either the true positions (if available) or the maximum a posteriori configuration.

4. Assumptions

The methodology we develop in this paper relies on several assumptions which are described in this section.

Assumption 1. *All of the model parameters are defined on bounded sets, i.e.:*

$$\forall\, k \in \mathcal{K} : \psi_k \in [\psi_k^{\mathcal{L}}, \psi_k^{\mathcal{U}}] =: \mathcal{S}_k, \tag{4.1}$$

$$\forall\, i \in \mathcal{V} : \boldsymbol{z}_i \in [-S, S] \times [-S, S] =: \mathcal{S}_{\mathcal{Z}}, \tag{4.2}$$

for some finite positive constants S, $\psi_k^{\mathcal{L}}$ and $\psi_k^{\mathcal{U}}$.

Remark 1. We note that, as a consequence of Assumption 1, the parameter space:

$$\mathcal{S} = \mathcal{S}_{\mathcal{Z}}^N \times \mathcal{S}_{\{k=1\}} \times \cdots \times \mathcal{S}_{\{k=K\}} \tag{4.3}$$

is a compact set.

Remark 2. Assumption 1 is rather strong and contrasts with the usual LPM frameworks. However, we argue that, from a practical point of view, these imposed conditions do not change the essence of the model. In fact, very large LPM parameters normally lead to degenerate models, and hence to realised networks that are meaningless in this modelling context (e.g. full or empty graphs). In this perspective, there is in fact a necessity to constrain $\boldsymbol{\psi}$ to a bounded space in order to make the model more tractable.

Remark 3. In the applications sections of this paper, a spherical truncated Gaussian distribution is used as prior on the latent positions:

$$\pi\left(\mathbf{z}_i\right) = \prod_{m=1}^{2} \left\{ \frac{\phi\left(\frac{z_{im}}{\gamma}\right)}{\gamma\left[\Phi\left(\frac{S}{\gamma}\right) - \Phi\left(\frac{-S}{\gamma}\right)\right]} \right\}, \qquad \forall i \in \mathcal{V}; \tag{4.4}$$

where z_{im} indicates the m-th coordinate of node i's latent position, $\gamma > 0$, ϕ and Φ are the p.d.f. and c.d.f. of a standard Gaussian distribution, respectively. This prior specification is essential in order to match the grid construction that is described in Section 5, however other priors could be considered.

Assumption 2. *The edge probability function $p : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^K \to [p^{\mathcal{L}}, p^{\mathcal{U}}] \subset (0,1)$ satisfies the following properties:*

a) *p* ***depends on the positions only through the latent distances****. This means that there exists a function $\rho : \mathbb{R}^+ \times \mathbb{R}^K \to [p^{\mathcal{L}}, p^{\mathcal{U}}]$ such that*

$$\forall \boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{R}^2,\ \forall \boldsymbol{\psi} \in \mathbb{R}^K :\ p\left(\boldsymbol{z}_i, \boldsymbol{z}_j; \boldsymbol{\psi}\right) = \rho\left(d\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right), \boldsymbol{\psi}\right).$$

b) *ρ* ***is non-increasing w.r.t. distances****; i.e. for any $\boldsymbol{z}_i \in \mathcal{S}_{\mathcal{Z}}$, $i = 1, 2, 3, 4$:*

$$\text{if } d\left(\boldsymbol{z}_1, \boldsymbol{z}_2\right) \geq d\left(\boldsymbol{z}_3, \boldsymbol{z}_4\right),\ \text{ then } p\left(\boldsymbol{z}_1, \boldsymbol{z}_2; \boldsymbol{\psi}\right) \leq p\left(\boldsymbol{z}_3, \boldsymbol{z}_4; \boldsymbol{\psi}\right).$$

c) ρ ***is Lipschitz w.r.t. distances****; i.e. for any* $\boldsymbol{z}_i \in \mathcal{S}_{\mathcal{Z}}$*,* $i = 1, 2, 3, 4$*:*

$$\left|p\left(\boldsymbol{z}_1, \boldsymbol{z}_2; \boldsymbol{\psi}\right) - p\left(\boldsymbol{z}_3, \boldsymbol{z}_4; \boldsymbol{\psi}\right)\right| \leq \kappa \left|d\left(\boldsymbol{z}_1, \boldsymbol{z}_2\right) - d\left(\boldsymbol{z}_3, \boldsymbol{z}_4\right)\right| \ ;$$

for some finite positive constant κ*.*

Remark 4. Assumption 2 is satisfied by most link functions, including the logit link of Eq. (3.4).

5. Grid approximation of the latent distances

Hereafter, we consider a generic LPM satisfying Assumptions 1 and 2, and we illustrate an estimation procedure based on a grid partitioning of the latent space. Following an approach similar to that of [21], we create a partitioning of the latent positions $\mathcal{Z}$ using a grid in $\mathbb{R}^2$. The grid is made of adjacent squares (called boxes hereafter) of side length $b > 0$, each having both sides aligned to the axes. A generic box $B[g, h]$ has corners located in $(bg - b, bh - b)\,, (bg - b, bh)\,, (bg, bh)$ and $(bg, bh - b)$, where the indices g and h are positive or negative but non-null integers, i.e. $g, h \in \mathbb{Z} \setminus 0$. Figure 5.1 shows the latent space with the partitioning given by these boxes.

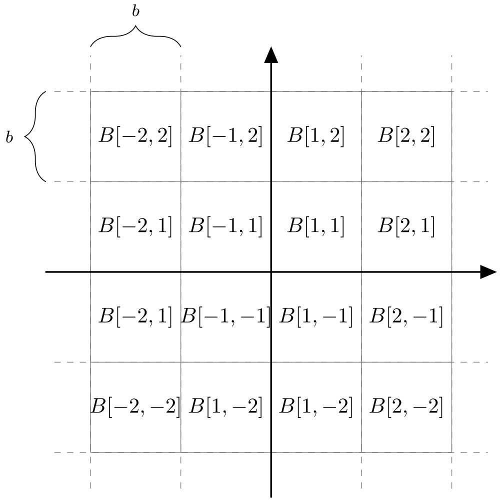

FIG 5.1. *Grid partitioning the latent space.*

For simplicity, we indicate with M the number of intervals in which we divide each axis. This means that the latent space $\mathcal{S}_{\mathcal{Z}}$ is partitioned into M^2 squared boxes, where the value of M satisfies $b = 2S/M$.

We denote with $N[g, h]$ the number of points located in a generic box:

$$N[g, h] = \left|\{i \in \mathcal{V} : \mathbf{z}_i \in B[g, h]\}\right|, \tag{5.1}$$

where $|H|$ denotes the cardinality of the set H.

It is also useful to introduce the centre of a generic box

$$\mathbf{c}[g,h] := (bg - b/2, bh - b/2) \ . \tag{5.2}$$

Given a node $j \in \mathcal{V}$ such that $\mathbf{z}_j \in B[g,h]$, we also indicate the centre of $B[g,h]$ with $\mathbf{c}_j$, representing the centre of the box containing j. An essential aspect of our proposed approach is determined by the fact that the distance $d\left(\mathbf{z}_i, \mathbf{z}_j\right)$ between any two nodes may be approximated by $d\left(\mathbf{z}_i, \mathbf{c}_j\right)$, i.e. the distance between node i and the centre of the box containing j.

Finally, we denote with $\xi_i[g,h]$ the number of edges between node i and the nodes allocated to $B[g,h]$, i.e.:

$$\xi_i[g,h] = \sum_{\{j \in \mathcal{V}:\ \mathbf{z}_j \in B[g,h]\}} y_{ij}; \tag{5.3}$$

and by $\zeta_i[g,h]$ the number of absent edges:

$$\zeta_i[g,h] = N[g,h] - \xi_i[g,h] - \mathbb{1}\left(\{z_i \in B[g,h]\}\right); \tag{5.4}$$

where $\mathbb{1}\left(\mathcal{A}\right)$ is 1 if $\mathcal{A}$ is true or 0 otherwise. Also, the degree of node $i \in \mathcal{V}$, i.e. the number of edges incident to it, is indicated by D_i.

These quantities introduced are exploited in the following sections to illustrate a new way of carrying out Bayesian inference for LPMs, requiring a dramatically reduced computational cost.

6. Noisy MCMC

As explained in the previous section, the distance from node i to the centre of a generic box $\mathbf{c}[g,h]$ can be used as a proxy for the true distances between i and all of the points contained in $B[g,h]$, for all g and h. This in turn allows one to approximate the edge probability $p\left(\mathbf{z}_i, \mathbf{z}_j; \boldsymbol{\psi}\right)$ using $p\left(\mathbf{z}_i, \mathbf{c}_j; \boldsymbol{\psi}\right)$, for all $j \in \mathcal{V}$ such that $\mathbf{z}_j \in B[g,h]$. This fact may be exploited in a number of ways. For example, the likelihood defined in (3.5) may be replaced by the following noisy likelihood:

$$\tilde{\mathcal{L}}_{\mathcal{Y}}\left(\mathcal{Z}, \boldsymbol{\psi}\right) := \left\{ \prod_{i=1}^{N} \prod_{g,h} \left[p\left(\mathbf{z}_i, \mathbf{c}[g,h]; \boldsymbol{\psi}\right)\right]^{\xi_i[g,h]} \left[1 - p\left(\mathbf{z}_i, \mathbf{c}[g,h]; \boldsymbol{\psi}\right)\right]^{\zeta_i[g,h]} \right\}^{1/2}; \tag{6.1}$$

where each edge contribution is essentially replaced by its noisy counterpart. Here, by counting each edge contribution twice and then correcting with the square root, one has the possibility to use the noisy approximation in a symmetric way, with respect to any pair of nodes i and j. We point out that a number of alternative estimators are available for the likelihood value using the grid approximation: the estimator proposed in (6.1) is one that generally works well in practice and that makes our theoretical developments easier to follow.

With `NoisyLPM`, we refer to a MH sampler that relies on the approximate edge probabilities rather than the true ones, or, equivalently, that uses the noisy likelihood $\tilde{\mathcal{L}}_{\mathcal{Y}}$ instead of the true likelihood $\mathcal{L}_{\mathcal{Y}}$. In `NoisyLPM` the full-conditionals introduced in (3.7) and (3.8) can be approximated as follows:

$$\tilde{\pi}\left(\mathbf{z}_i|\mathcal{Z}_{-i},\boldsymbol{\psi},\mathcal{Y}\right) \propto \pi\left(\mathbf{z}_i\right)\prod_{g,h}\left[p\left(\mathbf{z}_i,\mathbf{c}[g,h];\boldsymbol{\psi}\right)\right]^{\xi_i[g,h]}\left[1-p\left(\mathbf{z}_i,\mathbf{c}[g,h];\boldsymbol{\psi}\right)\right]^{\zeta_i[g,h]} \;; \tag{6.2}$$

$$\tilde{\pi}\left(\psi_k\middle|\boldsymbol{\psi}_{-k},\mathcal{Z},\mathcal{Y}\right) \propto \pi\left(\psi_k\right)\tilde{\mathcal{L}}_{\mathcal{Y}}\left(\mathcal{Z},\boldsymbol{\psi}\right) \;. \tag{6.3}$$

It is apparent that the computational cost of one evaluation of the approximate full-conditionals is much smaller than that of the true counterpart. In fact, for a given grid, the complexity of a noisy MH update becomes $\mathcal{O}\left(1\right)$ and $\mathcal{O}\left(N\right)$ for latent positions and global parameters, respectively. Overall, this makes the computational complexity of the `NoisyLPM` procedure of an order smaller than $\mathcal{O}\left(N^2\right)$.

6.1. Algorithmic complexity

The data is given as an undirected edge list, with L entries. An efficient implementation of `NoisyLPM` does not require using an adjacency matrix, however it does require at least an adjacency list. That is, for every node in the network, we store in memory all the edge list locations that involve that particular node. This can be done offline with a storage and computational cost of $\mathcal{O}(L)$.

We continue with the contruction of the grid, for the chosen value of M. First, we identify the thresholds that define the grid using the relation $b = 2/M$. Each node is assigned to a random initial position within the two-units-square centred in the origin of the space. Then, we proceed to allocate each node, based on its position, to the correct grid box. We store this information using a matrix of size $N \times 2$, where the first column indicates the box index on the x-axis, and the second column indicates the box index on the y-axis. In addition, we save the number of nodes that are contained in each box for future reference. These operations can all be performed offline with a storage and computational cost of $\mathcal{O}(N + M^2)$.

The last offline operation that we need to perform to set up the grid structure consists of counting how many edges go from a given node to a given box, i.e. calculating $\xi_i[g,h]$, for all i's, g's, h's. We store this information as an array of size $N \times M \times M$; this requires a memory storage cost of $\mathcal{O}(NM^2)$ and an operational cost of $\mathcal{O}(L)$.

Once the operations above are completed, we can proceed with the MCMC sampling procedure. Let us consider first an update of a non-node-dependent parameter ϕ. From Eq. (6.3) we deduce that the MCMC acceptance probability can be calculated with $\mathcal{O}(NM^2)$. If the chain moves to a new value of the parameter ϕ, then only an additional $\mathcal{O}(1)$ cost is required, since the grid structure does not require any changes.

The situation is slightly different when a latent position is updated. In this case, the acceptance probability can be calculated in $\mathcal{O}(M^2)$ as Eq. (6.2) suggests. However, should the chain accept the proposal and thus move the node to a new position, then two possibilities arise. If the new position is still within the same box, then the move can be performed in a constant time. If the new position falls outside of the node's current box, then we need to update the values of $\xi_i[g,h]$ for the particular node i that we are moving, and for every box that is affected. Since we have stored the memory locations of the edge list where we can find the neighbours of i, then this operation can be performed efficiently with cost in the order of the degree of i. As we scan through the nodes and update the position of each node, we repeat this operation up to N times, thus obtaining an approximate aggregate cost of $\mathcal{O}(L)$ just for moving the nodes.

The overall complexity of the algorithm can be calculated by adding up the offline costs and the costs arising from the sampling procedure. Here, we consider that we have a negligibly small number of global parameters, and that we have N nodes in the network. As a result, we obtain an overall complexity of $\mathcal{O}\left(TNM^2 + TL\right)$, where the term NM^2 is given by the acceptance probabilities calculations, the term L is given by the update of the ξ terms for the nodes that are moved, and T is the total number of MCMC iterations.

We should remark here that, under some scenarios that will be explored in the simulation study, we could have $M^2 > N$ (more grid boxes than nodes). In that case, necessarily, a number of boxes will be empty. As per our code implementation, all empty boxes can be effectively ignored thus reducing the computational term M^2 to N, which, in fact, leads back to the original complexity of the standard algorithm.

6.2. Software and parallelisation

Further speed ups of the algorithm can be achieved using parallel computing, by assigning individual boxes to different calculators. By doing so, we can parallelise the likelihood ratios calculations whenever they are required. The method has been implemented in `C++`, and it uses parallel computing through the library `OpenMPI`. All of the computations described in the paper have been performed on a 8-cores (2.2 GHz) Debian machine. The code for `NoisyLPM` is available from the public GitHub repository [26].

7. Theoretical guarantees

This section provides theoretical results that characterise the error induced by our approximation. Indeed, replacing $\mathcal{L}_{\mathcal{Y}}(\mathcal{Z},\boldsymbol{\psi})$ with $\tilde{\mathcal{L}}_{\mathcal{Y}}(\mathcal{Z},\boldsymbol{\psi})$ in the MH acceptance ratio implies that the stationary distribution of the Markov chain may not coincide anymore with the posterior distribution of interest described in Section 3.2. Here, our main goal is to show that a noisy MH sampler, such as the `NoisyLPM`, generates a sequence of random variables whose distribution can be made arbitrarily close to the true posterior $\pi\left(\,\cdot\,|\mathcal{Y}\right)$.

In fact, one can note that, by construction, our noisy MH sampler admits the approximate posterior $\tilde{\pi}\,(\,\cdot\,|\mathcal{Y})$ as stationary distribution. Hence, the approximation error is directly, and globally, measured by $\|\pi - \tilde{\pi}\|$, i.e. the total variation distance between the two posteriors. However, obtaining an explicit expression or an upper bound of $\|\pi - \tilde{\pi}\|$ is challenging. Our main result (Theorem 2) gives an upper bound of $\|\pi - \tilde{\pi}\|$ obtained by bounding the distance between the exact and noisy Markov chains, following [17]. The core of our work has been to devise a bound, which we believe is tight, on the distance between the two Markov kernels, see Theorem 1 and Corollary 2.

The theoretical framework is the analysis of the perturbation of uniformly ergodic Markov chains, initiated in [17] and refined for the noisy Metropolis-Hastings case in [1]. We first recall the uniform ergodicity assumption.

Assumption 3. *A π-invariant Markov kernel P operating on a state space $\mathcal{S}$ is uniformly ergodic if after $t \in \mathbb{N}$ iterations, the distance between the chain distribution and the stationary distribution is bounded as follows:*

$$\sup_{u \in \mathcal{S}} \|P^t(\boldsymbol{\theta},\,\cdot\,) - \pi\| \leq C\tau^t, \tag{7.1}$$

for some $C < \infty$ and $\tau < 1$.

The section is divided in two parts: in the spirit of [1], we first derive an extension of their theoretical framework to include the analysis of noisy Metropolis-Hastings algorithms in a generic setup, that is, beyond the LPM context. In the second part we give a series of theoretical results that are specific to LPMs, and that aim to characterise the magnitude of the approximation error in the MH acceptance probabilities, in preparation for applying our general result. In particular, we show that the distance between the exact algorithm and the `NoisyLPM` can be arbitrarily reduced by refining the latent grid.

7.1. Noisy MH aggregated errors

This paper deals with an approximation of a MH Markov chain, where the parameters of the model are updated in turn. Perturbations of uniformly ergodic Metropolis-Hastings Markov chains have been studied in [1]. We show, here, that a similar analysis can be carried out in a generic MH sampler framework.

We introduce the following notation. We indicate with r a generic parameter update step of the MH, with R indicating the number of updates performed in a particular algorithmic instance. For example, R may indicate the number of model parameters that are updated in each iteration of the MCMC algorithm.

An arbitrary sigma-algebra on the compact parameter space $\mathcal{S}$ is denoted by $\mathcal{A}$. For any signed measure μ on $(\mathcal{S}, \mathcal{A})$, we denote the total variation distance of μ by $\|\mu\| := \sup_{A \in \mathcal{A}} |\mu(A)|$. For any Markov kernel P taking values in $\mathcal{S} \times \mathcal{A}$, we denote the operator norm of P as:

$$\|P\| := \sup_{\boldsymbol{\theta} \in \mathcal{S}} \|P(\boldsymbol{\theta}, \cdot)\| = \sup_{\boldsymbol{\theta} \in \mathcal{S}} \sup_{A \in \mathcal{S}} |P(\boldsymbol{\theta}, A)|\ . \tag{7.2}$$

Finally, let μP be the measure on $(\mathcal{S}, \mathcal{A})$ defined as $\mu P := \int_{\mathcal{S}} \mu(\mathrm{d}x) P(x, \cdot)$. The kernel P will generally be considered as the *exact* kernel, whereas $\tilde{P}$ will denote its noisy counterpart.

The following proposition shows that the distance between the one step transition of an elementary MH update and its noisy counterpart is uniformly bounded.

Proposition 1. *Let $\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)$ and $\tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)$ be the corresponding exact and noisy acceptance probabilities (respectively), that arise when considering a generic update $\boldsymbol{\theta} \to \boldsymbol{\theta}'$. If there exists some finite constant $\omega > 0$ such that:*

$$\left|\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right| \leq \omega \tag{7.3}$$

then we also have:

$$\|P - \tilde{P}\| \leq \omega \ . \tag{7.4}$$

The proof of this proposition is given in Appendix 11.1. Now, we characterize instead the error that is accumulated over a sweep of the MH sampler over a collection of the model parameters. We denote with $P_{[R]}$ (resp. $\tilde{P}_{[R]}$) the kernel corresponding to a sequential update of a number of model parameters using exact (resp. approximate) acceptance probability:

$$\begin{aligned} P_{[R]}\left(\boldsymbol{\theta}, \cdot\right) &:= P_1 \cdots P_R\left(\boldsymbol{\theta}, \cdot\right) \ , \\ &= \int \cdots \int P_1\left(\boldsymbol{\theta}, d\boldsymbol{\theta}_1\right) \cdots P_{R-1}\left(\boldsymbol{\theta}_{R-2}, d\boldsymbol{\theta}_{R-1}\right) P_R\left(\boldsymbol{\theta}_{R-1}, \cdot\right) \ , \\ \tilde{P}_{[R]}\left(\boldsymbol{\theta}, \cdot\right) &:= \tilde{P}_1 \cdots \tilde{P}_R\left(\boldsymbol{\theta}, \cdot\right) \ . \end{aligned} \tag{7.5}$$

This corresponds to the composition of the R elementary kernels, indicated with P_r or $\tilde{P}_r$, each characterising the update of one model parameter.

Proposition 2. *The error carried by a product of noisy Markov kernels $\tilde{P}_{[R]} = \tilde{P}_1\tilde{P}_2 \cdots \tilde{P}_R$ relative to its exact version $P_{[R]} = P_1 P_2 \cdots P_R$ is subadditive, in the sense that:*

$$\|P_{[R]} - \tilde{P}_{[R]}\| \leq \sum_{r=1}^{R} \|P_r - \tilde{P}_r\| \ . \tag{7.6}$$

The proof of this proposition is provided in Appendix 11.2. We can join the above two results in the following proposition.

Proposition 3. *Let α and $\tilde{\alpha}$ be the corresponding exact and noisy acceptance probabilities (respectively), that arise when considering a generic update for any of the model parameters. Assume that there exists some finite constant $\omega > 0$, such that:*

$$|\alpha - \tilde{\alpha}| \leq \omega \ . \tag{7.7}$$

Then, after $R < \infty$ parameter updates, the product kernels satisfy:

$$\|P_{[R]} - \tilde{P}_{[R]}\| \leq R\omega \ . \tag{7.8}$$

Proof. This immediately follows from Propositions 1 and 2. □

Finally, as in [1], we rely on Corollary 3.1 of [17] to give our main result for the `NoisyLPM` algorithm.

Corollary 1. *Let $P_{[R]}$ (resp. $\tilde{P}_{[R]}$) be the transition kernel for the exact MH sampler (resp. noisy) described in Eq. (7.5). Assume that the Markov chain with kernel P is uniformly ergodic (Assumption 3). Then, for any $t > 0$ and for any starting point $\boldsymbol{\theta} \in \mathcal{S}$:*

$$\|\delta_{\boldsymbol{\theta}} P_{[R]}^t - \delta_{\boldsymbol{\theta}} \tilde{P}_{[R]}^t\| \leq \left(\lambda + \frac{C\tau^\lambda}{1-\tau}\right) R\omega \ , \tag{7.9}$$

where $\lambda = \lceil \log{(1/C)} / \log(\tau) \rceil$.

7.2. LPM likelihood errors

We now report theoretical results that are specific to LPMs, in preparation of applying Corollary 1 to this context. In the following theorem, we show that the error on the MH acceptance probabilities is bounded, and that it can be arbitrarily reduced by refining the latent grid partition.

Theorem 1. *Under Assumptions 1 and 2, the error on the acceptance probabilities for a latent position update satisfies for all $i \in \mathcal{V}$:*

$$|\alpha_{\mathcal{Z}}\left(\boldsymbol{z}_i \to \boldsymbol{z}_i'\right) - \tilde{\alpha}_{\mathcal{Z}}\left(\boldsymbol{z}_i \to \boldsymbol{z}_i'\right)| \leq \kappa' b N \ , \tag{7.10}$$

and for any $k \in \mathcal{K}$ identifying a static parameter's update:

$$|\alpha_{\boldsymbol{\psi}}\left(\psi_k \to \psi_k'\right) - \tilde{\alpha}_{\boldsymbol{\psi}}\left(\psi_k \to \psi_k'\right)| \leq \kappa'' b N^2 \ , \tag{7.11}$$

where κ' and κ'' are suitable positive finite constants which do not depend on either b or N.

The proof of Theorem 1 is given in Appendix 11.4.

Corollary 2. *Let $P_{[N+K]}$ and $\tilde{P}_{[N+K]}$ be the exact and noisy composite kernels for the full deterministic scan via MH samplers under Assumptions 1 and 2. These satisfy:*

$$\|P_{[N+K]} - \tilde{P}_{[N+K]}\| \leq \kappa b N^2 \ ; \tag{7.12}$$

for a suitable positive finite constant κ which does not depend on either b or N.

Proof. This is proved using Proposition 3 and Theorem 1. There are N latent position updates, whereby each of the kernels is upper bounded by $\kappa' b N$, for a suitable positive constant κ'. This gives a composite kernel with an upper bound of $\kappa' b N^2$. In addition, there are $K < \infty$ global parameters, whose number does not depend on N or b, each yielding an upper bound of $\kappa'' b N^2$, for a suitable positive constant κ''. So, the upper bound for all combined updates is still $\kappa b N^2$, where $\kappa = \max\left\{\kappa', K\kappa''\right\} < \infty$. □

Now we can state our main result for `NoisyLPM`, which connects Corollary 2 with the theory on noisy MCMC.

Theorem 2. *Let P be the exact MH composite kernel which operates on $\mathcal{S} = \mathcal{S}_\psi \times \mathcal{S}_\mathcal{Z}$ and $\tilde{P}$ be the corresponding kernel of* `NoisyLPM`*. If the LPM satisfies Assumptions 1 and 2, and if P is uniformly ergodic (Assumption 3), then for any starting point $\boldsymbol{\theta} \in \mathcal{S}_\psi \times \mathcal{S}_\mathcal{Z}$ and any $t > 0$:*

$$\|\delta_{\boldsymbol{\theta}} P^t - \delta_{\boldsymbol{\theta}} \tilde{P}^t\| \leq \left(\lambda + \frac{C\tau^\lambda}{1-\tau}\right) \kappa b N^2 \tag{7.13}$$

where $\lambda = \lceil \log(1/C)/\log(\tau) \rceil$ depends on the exact sampler convergence properties, and κ is a positive constant that does not depend on either b or N.

Proof. Assumptions 1 and 2 guarantee that Corollary 2 holds. Then, uniform ergodicity confirms that an LPM version of Corollary 1 exists, hence Eq. (7.13) holds true. □

Remark 5. The significance of Theorem 2 is two-fold. On the one hand, for a fixed N, the error upper bound can clearly be made arbitrarily close to zero by reducing the grid parameter b, denoting the sidelength of the boxes. This confirms that using a finer grid reduces the approximation error. On the other hand, since the constant κ does not depend on N, this result emphasises that, asymptotically, the error grows at most with the squared number of nodes. This suggests that, as a worst case scenario, the error can be kept constant (as N increases) by keeping b on the order of $1/N^2$, or, equivalently, a number of boxes[3] in the order of N^4. However, we show in the simulations (in particular in Section 8.2) that this bound is very conservative.

7.3. Note on the uniform convergence assumption

Assumption 3 is usually strong in the context of MCMC algorithms. However, since the state space is compact (see Assumption 1), it is easy to show that the convergence of the MH kernel $P_{[R]}$ to π is uniform. Even though this result is not surprising, we could not identify a specific entry in the literature providing a rigorous proof of this fact. For completeness, we include Theorem 3 in Appendix 11.5.

8. Experiments

In this section we propose three simulation studies to characterise the bias introduced by our approximation, and to gauge the gain in computing time achieved.

[3] Note that, for a fixed b, we create M^2 boxes by partitioning each axis into M segments of length b. The number of segments M satisfies: $M = 2S/b \approx 2SN^2$, and so $M^2 \sim \mathcal{O}(N^4)$.

We consider an LPM characterised by two global parameters $\boldsymbol{\psi} = (\beta, \theta)$ which determine the edge probabilities as follows:

$$\log\left(\frac{p\left(\mathbf{z}_i, \mathbf{z}_j; \beta, \theta\right)}{1 - p\left(\mathbf{z}_i, \mathbf{z}_j; \beta, \theta\right)}\right) := \beta - e^{\theta} d\left(\mathbf{z}_i, \mathbf{z}_j\right). \tag{8.1}$$

Here, $\beta \in \mathbb{R}$, $\theta \in \mathbb{R}$, and d denotes the Euclidean distance between the two latent positions.

A priori, the latent positions are IID variables distributed according to a truncated Gaussian, as shown in (4.4). The proposal distributions used in the sampling procedures are also truncated Gaussians, defined as random walks over the parameter space, but we note that other proposals may be considered. We fix both the threshold parameter S and the standard deviation γ to 1. This choice does not hinder the flexibility of the model; in fact, the likelihood parameter θ directly regulates the magnitude of the effect of the latent space. In other words, e^{θ} may simply be considered as the standard deviation for the latent positions. The likelihood parameters β and θ are assumed to be independent a priori, and both distributed according to non-informative Gaussian priors with fixed large standard deviations.

We note that the model specification considered does not completely satisfy Assumption 1, since, for example, the supports of β and θ are not bounded. However, we argue that large values of these parameters correspond to degenerate LPMs, which are of little interest in practical situations, and highly unlikely to occur. In other words, the extreme values of the LPM parameters do not play a role and do not affect the MCMC estimation unless the observed graph is degenerate or near-degenerate.

8.1. Study 1: likelihood approximations

In the first study, we focus only on the approximation of the log-likelihood, i.e. we analyse the error introduced when (3.5) is replaced with the noisy counterpart in (6.1). We do not use any MCMC sampling in this first simulation study.

8.1.1. Case A: given and fixed M

First, we generate random LPMs with global parameters set to $\beta = 0.5$ and $\theta = \log(3)$, and with latent positions drawn uniformly in the rectangle $\mathcal{S}_{\mathcal{Z}}$. This combination of parameters yields realised networks where about 10% of the possible edges appear. We simulate 100 networks for each value of N varying in the set $\{100, 250, 500, 1000, 2500, 5000, 10000\}$. For each of these realised networks, we evaluate the exact log-likelihood function derived from (3.5) for the true parameter values.

On each axis, the interval $[-1, 1]$ is segmented in $M = 8$ adjacent intervals of the same length, hence obtaining a grid of 64 squared boxes of side length $1/4$. The noisy log-likelihood derived from (6.1) is thus evaluated using such grid.

In fact, the same procedure is repeated on the same networks using various grids determined by M in the set $\{8, 16, 32, 64\}$. Note that the highest values of N and M are rather extreme: $N = 10000$ gives networks so large that even storing or working with the adjacency matrix is computer intensive (in fact our implementation does not require calculating the adjacency matrix at any stage); $M = 64$ gives a grid which contains 4096 boxes, so, for several N values, we would have more boxes than data points. This scenario is proposed only to provide a more complete assessment, since having so many boxes (compared to nodes) defeats the whole purpose of applying our procedure for a computational advantage.

A preliminary plot is provided in Figure 8.1. This plot shows the proportion of boxes that contain 2 or more nodes. This is derived without using the log-likelihood, but simply by constructing the grid over the randomly generated data. The plot emphasises that, at least for the starting configuration of the noisy algorithms, most boxes would contain more than 1 node. The proportion seems to converge to 1 rather quickly with N increasing, with the only exception being given by the extreme value $M = 64$.

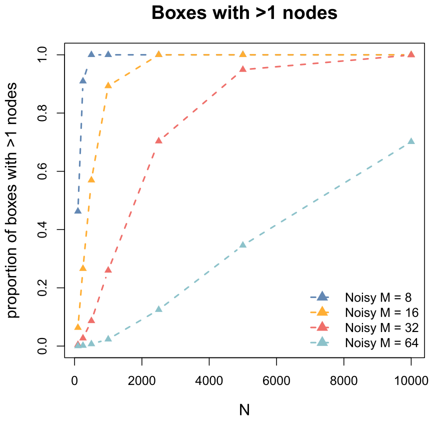

FIG 8.1. ***Simulation study 1A***. *Proportion of grid boxes that contain more than* 1 *node.*

A proportion close to 1 is ideal for `NoisyLPM`, since only these boxes guarantee a gain in computational efficiency. By contrast, boxes containing 0 or 1 nodes would require the algorithm to perform inefficient steps which are less convenient than the corresponding ones under a standard likelihood calculation. The figure thus confirms that, in theory, good computational improvements should be obtained for very small M values.

Figure 8.2 shows the log-likelihood error and the average log-likelihood computing times for all of the combinations of N and M. The left panel of this figure shows the absolute value of the error, divided by N^2. Since we get roughly constant lines for every value of M, this gives evidence that each of the N^2 terms that form the likelihood of Eq. (3.5) is characterised by an error bounded by a

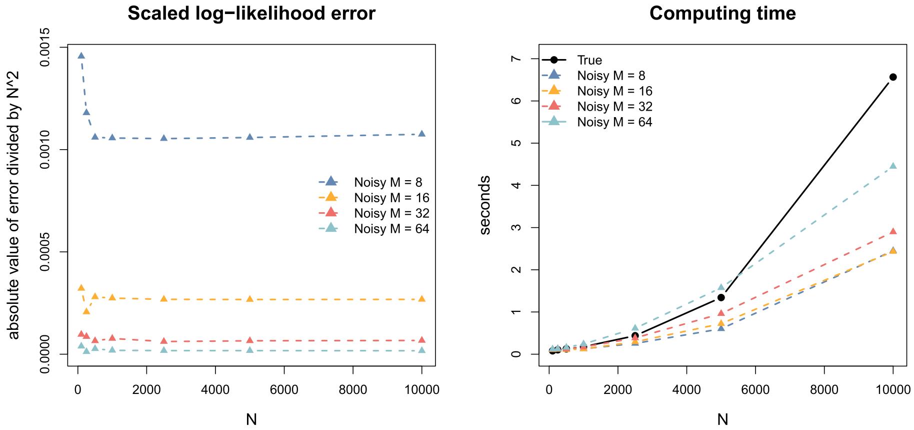

Fig 8.2. ***Simulation study 1A***. *The absolute value of the log-likelihood error, divided by* N^2*, is shown on the left panel. The right panel shows instead the average (across* 100 *networks) computing time for the same log-likelihood evaluations.*

constant. The most accurate algorithm is obtained with $M = 64$, which gives a very minimal average error on each likelihood term.

The right panel shows instead the computing time for one log-likelihood calculation, averaged out across the 100 repetitions. In this plot we can see that the computational complexity is highest for the standard algorithm (labelled as `True`): we know that the complexity of this particular algorithm is quadratic in the number of nodes. The noisy algorithms all exhibits a computational complexity of a lower order, and an overall lower computing time. These algorithms require the construction and maintainment of the grid structure, which involves some roughly constant computing time. As a consequence, for $M = 64$, the noisy algorithm gives a convenient trade-off only when N becomes very large. Again, this is reasonable since this particular value of M is extreme and would not be considered in applications, unless N is also especially large. Figure 8.2 confirms two basic but fundamental asymptotical facts: on the one hand, the log-likelihood errors can be arbitrarily reduced by choosing a finer grid, for a fixed N. On the other, for any grid fineness (fixed M), there exists an N large enough such that the noisy algorithm is faster than the standard one.

8.1.2. *Case B: asymptotics for M growing with N*

While the study above confirms the correctness of our approach for fixed N and M values, it does not address an asymptotical setting where N and M are determined using the results of Section 7. In fact, Theorem 2 indicates that we can control the approximation error as long as M grows at most with N^2, up to a proportionality constant. We show in this simulation study some numerical results that are consistent with these theoretical findings. In particular, we illustrate that the bounds of Theorem 2 are rather conservative, and, in practice, we can achieve an asymptotically decreasing error and convienent computational costs for smaller order values of M.

In terms of study setup, we use a similar framework to Section 8.1.1. We generate random LPMs with global parameters set to $\beta = 0.5$ and $\theta = \log(3)$, and with latent positions drawn uniformly in the rectangle $\mathcal{S}_{\mathcal{Z}}$. We simulate 200 networks for each value of N varying from 400 to 800. As concerns M, for each network we define $M \propto N^P$, where P can be one of the values in the set $\{1, 1.5, 2\}$. The case $P = 2$ corresponds to the result of Theorem 2, whereby we expect the errors to be bounded essentially by a constant. By contrast, the other two values represent two more ambitious setups, where we hope to get a more convenient computational cost while still controlling the errors with a constant bound.

As Figure 8.3 shows, it turns out that the errors are asymptotically decreasing for all three values of P. The trends are not exactly monotonous due to the transformation from N to M, which is not guaranteed to return an integer value for M. As a consequence, we always round the value of M to the closest integer, and this creates groups of points in the plot that are obtained from the same value of M, even while N is increasing along the x-axis. This result is in agreement with our theoretical findings, and, in fact, it highlights that the given theoretical bounds are very conservative.

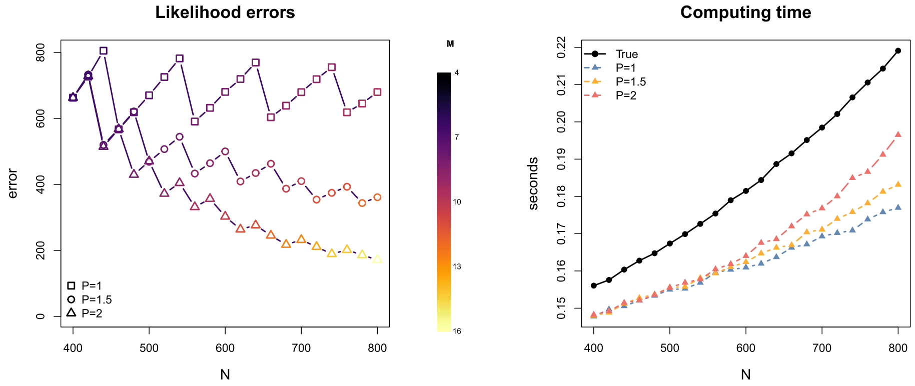

Fig 8.3. ***Simulation study 1B***. *The absolute value of the log-likelihood error is shown on the left panel, for the three cases $M \propto N$, $M \propto N^{1.5}$, and $M \propto N^2$. The value of M corresponding to each point is indicated by the color shift. The overlap between some of the lines is due to the same value of M being used under the three cases, which leads to similar errors. The right panel shows instead the average computing time for the same log-likelihood evaluations.* `True` *refers to the standard algorithm, which has a quadratic computing time in N.*

As regards the computing costs, we start by saying that, in principle, the complexity of the standard algorithm corresponds to a worst case scenario. This is because, even in the case of an extraordinarily high number of boxes, we would only be using the non-empty boxes of which there are up to N. The right panel of Figure 8.3 indicates that we get a clear advantage in the case $P = 1$. The case $P = 1.5$ remains unclear. As per the case $P = 2$, which corresponds to the bounds of Theorem 2, this leads to an algorithm that for a large N might be comparable or slower than the standard algorithm. These results highlight how the choice $M \propto N$ is ideal as a compromise between accuracy and computational efficiency, since for any network size N, we can achieve a relatively small error but with a much faster algorithm compared to the standard procedures.

8.2. Study 2: Metropolis-Hastings asymptotics

In the second simulation study, we aim at characterising the estimation errors using the complete MCMC sampling procedure. This means that we run the MH sampler on a number of networks, for both the non-noisy (which we take as ground truth) and noisy procedures. Then, we compare noisy and exact posterior distributions using their means. In particular, we quantify the discrepancy by calculating the Mean Squared Error (MSE) from 100 replications of both chains, for each parameter setting.

Since running the complete sampling procedure requires a much higher computational cost, we only provide these results for relatively smaller networks and for fixed values of M. However, in our third simulation study, we provide further evidence that the results hold also for larger networks.

The setup of this study is as follows. We consider networks of N nodes where N varies in the set $\{20, 40, 60, 80, 100\}$. For each value N we generate 100 networks using $\beta = 2.5$ and $\theta = \log(3.5)$, and run the exact MH sampler for 20,000 iterations. The first 10,000 iterations are discarded as burn-in, and only one draw every 10-th is stored to be kept in the final sample. We consider grids defined by M in the set $\{4, 8, 12\}$, and run the three corresponding samplers using the same number of iterations. Although we do not check convergence diagnostics for each individual generated network and posterior sample, we argue that convergence is satisfactory across all experiments. Besides, this aspect is not critical since we can study the errors even if convergence is not reached; or, in other words, we are studying the errors after 20,000 MCMC iterations.

Figure 8.4 shows the results that we obtained. Even though Theorem 2 suggests that M should scale with N^2, the evidence from the simulations highlight a much more positive outcome, where, in fact, the errors do not increase with N for $M = 8$ and $M = 12$. The approximation with $M = 4$ ends up being too rough, as the posterior mean MSEs slightly increases between $N = 80$ and $N = 100$. The bound of Theorem 2 offers a scaling which is, as expected, too conservative. Recall that Theorem 2 was essentially obtained by bounding the distance between both Markov kernels. Thus, since more data points propagate a larger discretisation error, the approximation offered by the noisy Markov kernel and the exact one appears to get rougher too.

One fundamental aspect that is not taken into account by our analysis is that more data points also bring more information available for inference. Our reasonably regular models imply that both noisy and exact posterior distributions see their probability mass concentrate towards the Maximum Likelihood Estimator (MLE) as N increases, in an asymptotic regime typically described by

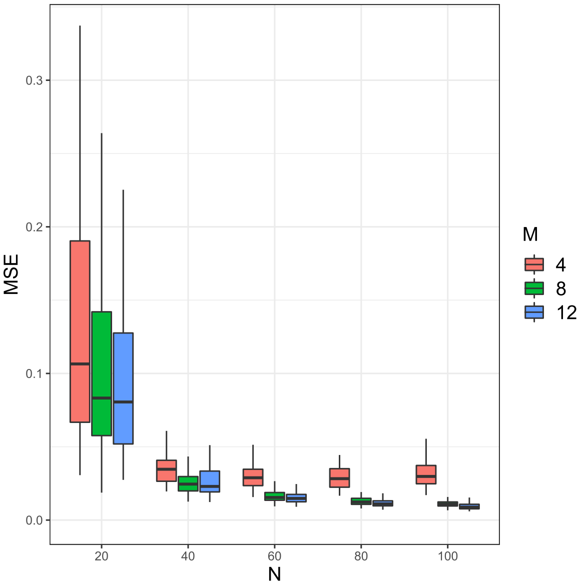

Fig 8.4. ***Simulation study 2.*** *The Mean Squared Error, where the mean is calculated across the N nodes. The boxplot represents the variability across the* 100 *repetitions, for each combination of N and M.*

a Bernstein-von Mises concentration result. The fact that M needs to increase with N is thus necessary since, both MLEs being different, the total variation distance between both posterior distributions will eventually (as N gets larger) increase. What this study shows is that before entering this asymptotic regime, both posteriors actually get first closer. In fact, the discretisation approximation is clearly overcompensated by that concentration phenomenon, since keeping M constant one still observes an improvement as N increases. We suspect that for $M = 4$, the Bernstein-von Mises regime kicks in after $N = 80$. We also suspect that for a large enough N, a similar observation could be done for $M = 8$ and $M = 12$.

For the noisy posterior to keep track of the exact one even in the asymptotic regime, it could be possible that a (much) less aggressive scaling than $M = \mathcal{O}(N^2)$ is sufficient. One way to answer that question could be to search for tighter bounds between both posterior distributions using their Bernstein-von Mises approximation, hence by totally bypassing the analysis of both Markov kernels.

A second aspect that transpires from this simulation study is a confirmation that finer grids will give more accurate results, with major improvements happening with relatively coarse grids. We expand more on this aspect in the next simulation study, where we show that we can choose very small values of M (relative to N), and still obtain very accurate inference results, with only a fraction of the computing time.

8.3. Study 3: Metropolis-Hastings error

As data, we use three artificial networks, which are generated so that each of them has network density close to 10%. In this simulation study we do not use any repetitions, so we analyse exactly three networks. Another difference with the previous setup is that, in this study, node 1 is assumed to be located exactly at the origin of the space, for comparison purposes. The number of nodes N of the networks is set to 200, 400 and 600, respectively. For the `NoisyLPM`, we consider three different grid structures: the number of intervals M in each axis varies in the set $\{8, 12, 16\}$ (the results are shown for $M = 8$ and $M = 16$, but all results are available from the authors upon request).

The non-noisy MH sampler and the `NoisyLPM` are run on each dataset for a total of 200,000 iterations. The first 100,000 iterations are discarded as burn-in, and only one draw every 10-th is stored to be kept in the final sample. Eventually, all of the algorithms are bound to return a collection of 10,000 draws for each model parameter. In this simulation study, we checked and confirmed MCMC convergence using trace plots, individually for each of the posterior samples. In addition, we used Raftery and Lewis's diagnostic [23] to estimate the required sample size to find a $q = 0.025$ quantile with a $r < 0.01$ error, for each MCMC run. This returned the required sample sizes (averaged across all model parameters) as reported in Table 1. This table highlights that the convergence properties of the chains are not likely not affected by the grid approximations.

Table 1

***Simulation study 3**. Required sample size to estimate a $q = 0.025$ quantile with a $r < 0.01$ error, according to Raftery and Lewis's diagnostic [23].*

N	Ground truth	`NoisyLPM`		
		$M = 8$	$M = 12$	$M = 16$
200	1,181	1,157	1,135	1,181
400	1,076	1,066	1,083	1,074
600	1,050	1,063	1,042	1,049

Figure 8.5 shows the posterior densities for the node located in the centre of the space. The two `NoisyLPM` posterior densities shown are extremely similar to the ground truth, proving that the uncertainty in the positioning is not necessarily amplified by the approximation.

Figure 8.6 focuses instead the (posterior) average position as a point estimator, and compares the estimated positions of all nodes in the ground truth and noisy case. Again, the approximation appears to have very limited consequences on the correctness of the results. In particular, the estimation error is almost non-existent when $M = 16$.

Figures 8.7 and 8.8 illustrate the posterior densities for the global parameters β and θ. Note that, in both figures, the horizontal axes of the plots are on different scales. In fact, these plots confirm that the uncertainty on global parameters tends to vanish as N increases, for both non-noisy and noisy algorithms. As expected, a larger M gives results closer to the ground truth.

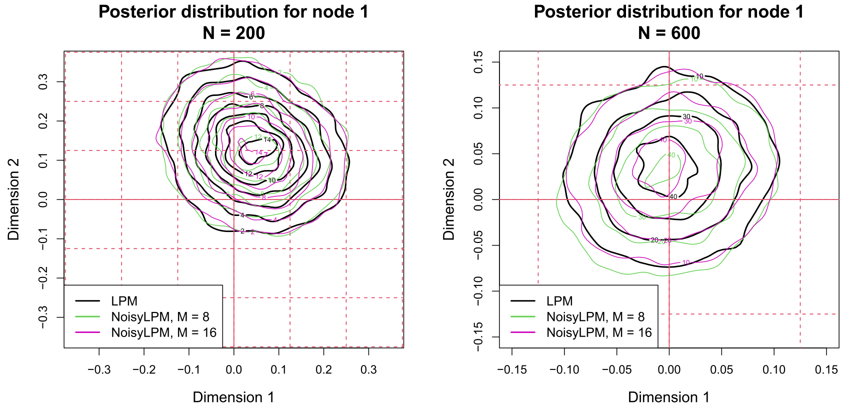

FIG 8.5. ***Simulation study 3***. *Posterior densities for the node in the centre.*

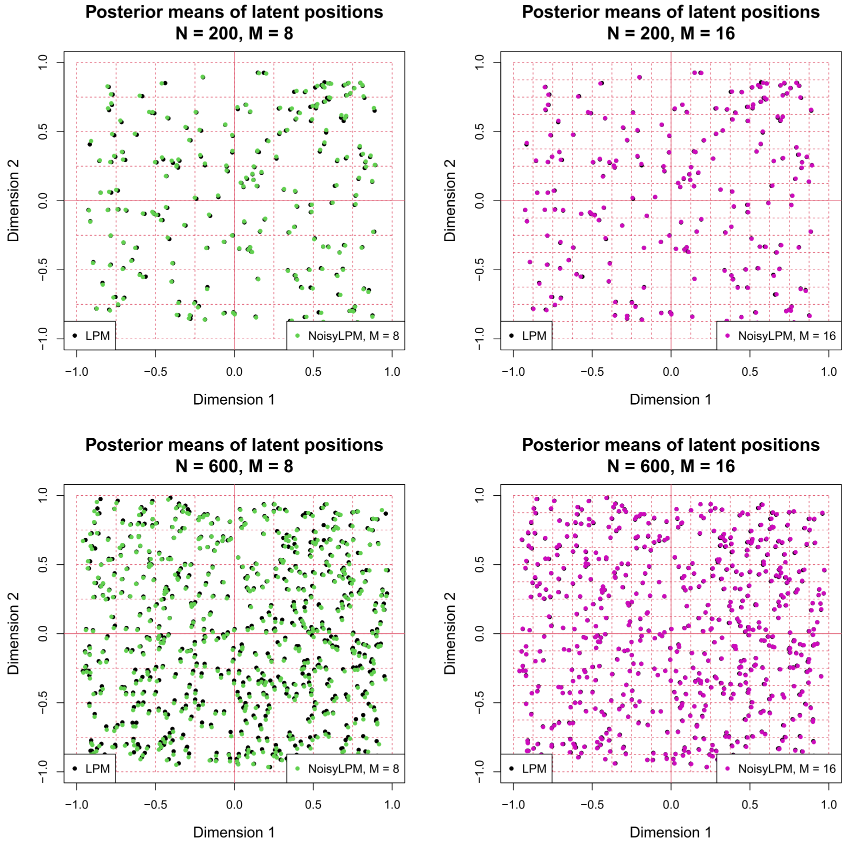

FIG 8.6. ***Simulation study 3***. *Comparison between ground truth and noisy estimates of the positions. The black circles correspond to the posterior means of the positions in the ground truth configuration, whereas the green and pink nodes correspond to the noisy counterparts.*

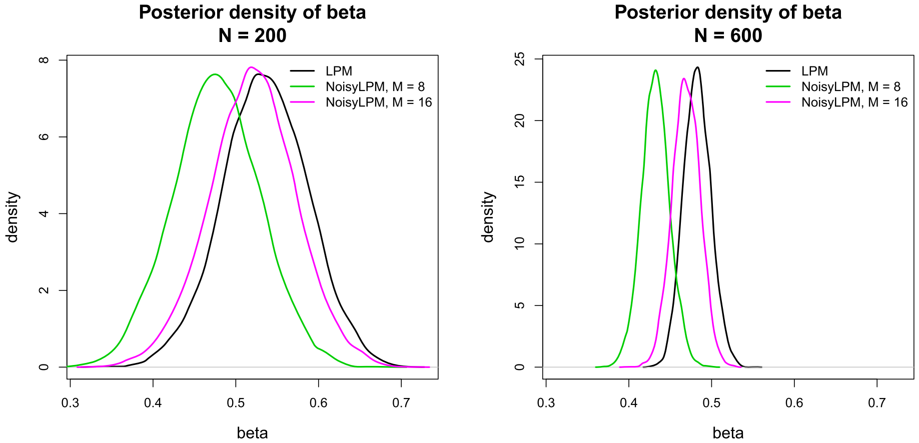

FIG 8.7. ***Simulation study 3.*** *Posterior densities for β. Note the different scaling in the horizontal axis.*

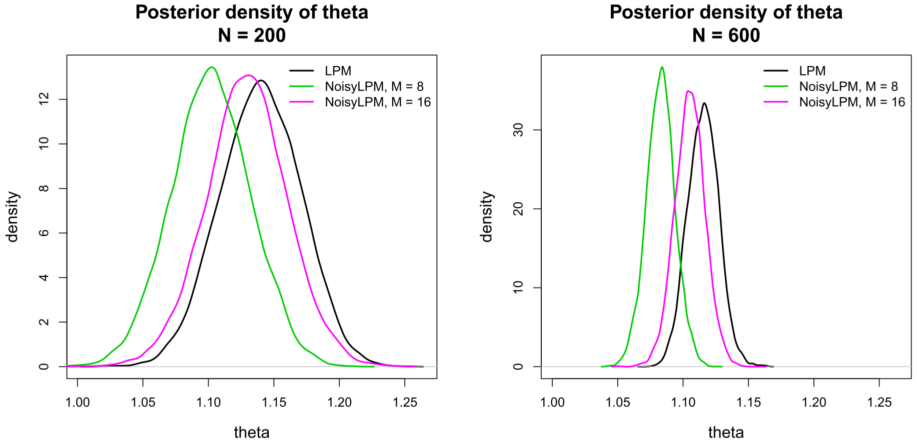

FIG 8.8. ***Simulation study 3.*** *Posterior densities for θ. Note the different scaling in the horizontal axis.*

We further analyse the results by comparing the estimated edge probabilities in Figure 8.9. These plots also confirm the correctness of the noisy procedure, and the limited effects of the approximation on the results.

Finally, in Table 2 we show the computing time required for each sampler. The highest gain is achieved for $M = 8$ and $N = 600$, where the `NoisyLPM` is roughly three times faster than the benchmark. As we will show in the next section, the gain can become substantial when larger networks are considered.

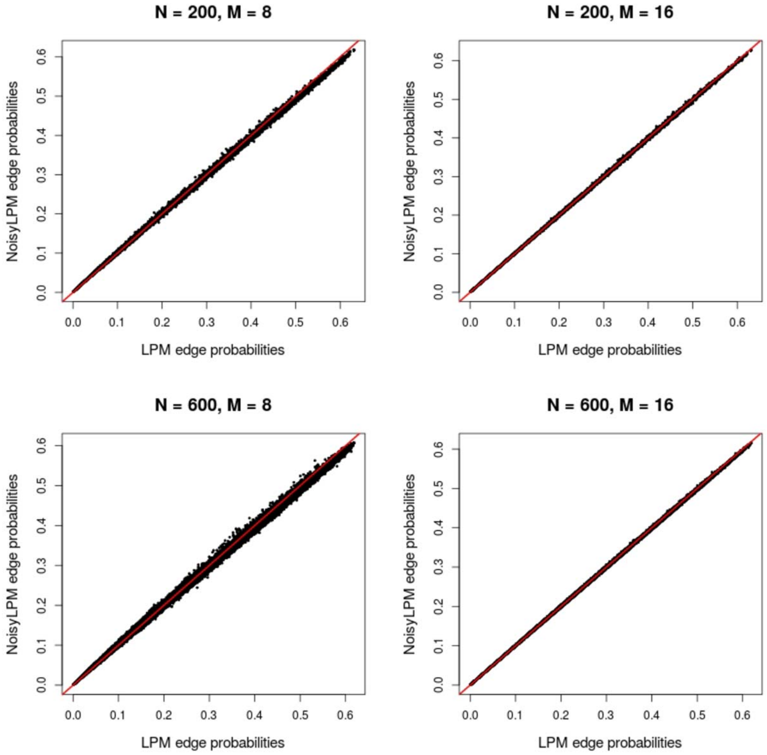

FIG 8.9. ***Simulation study 3***. *Comparison between ground truth and noisy estimates of the edge probabilities. These estimates are obtained by pluggin-in the posterior mean estimates of the model parameters in* (8.1).

TABLE 2
Simulation study 3. *Seconds (rounded value) required to obtain* 200,000 *iterations from each of the networks, for both algorithms.*

N	Ground truth	`NoisyLPM`		
		$M = 8$	$M = 12$	$M = 16$
200	2,310	1,669	2,252	2,767
400	7,242	3,515	5,458	7,673
600	14,347	4,718	7,501	11,825

9. Coauthorship in astrophysics

9.1. Binary case

The coauthorship network studied in this section was first analysed by [14]. The nodes correspond to authors, whereas the presence of an edge between two nodes means that the two researchers appear as coauthors on a paper submitted to arXiv, in the astrophysics category. The network is by construction undirected and without self-edges. The number of nodes is 18,872, whereas the number of edges is 198,110, corresponding to an average degree of about 21.

We fit the LPM of Section 8 to this data using the `NoisyLPM` with $M = 16$. First, we let the algorithm run for a large number of iterations. We use this phase as burn-in, and to tune the proposal variances individually for each parameter until the corresponding acceptance probability lies between 20% and 50%. Then, we run the `NoisyLPM` for 50,000 iterations, storing only one draw every 10-th. Trace plots and other standard convergence diagnostics suggest good mixing and good convergence of the chain to its stationary distribution. In summary, for each latent position and global parameter, we obtain 5,000 random draws that can be used to characterise the distribution of interest. In order to check MCMC convergence, we used the trace plots for the global parameters and for a selection of latent positions. We also used Raftery and Lewis's diagnostic [23] to estimate the required sample size to find a $q = 0.025$ quantile with a $r < 0.01$ error. This returned an average sample size of 2,198 across all parameters, which is well below our choice of 5,000.

As regards the results, Figure 9.1 shows the average latent positions for all of the nodes in the network. We point out that the nodes have a tendency to be distributed close to the centre of each box. Quite reasonably, this is a natural consequence of our construction, since the centre of the boxes is used as a proxy to calculate the latent distances. For example, if a node with a low degree is connected only to nodes allocated to the same box, it will tend to move towards the centre of the same box, since that would maximise the likelihood of those edges appearing. More generally, we argue that, while the overall macro-structure of the latent space (i.e. the association of nodes to boxes, or the association of nodes to sub-regions of the space) is properly recovered, the micro-structure, given by the relative positions of the nodes within each box, may not necessarily be accurate.

Figure 9.2 shows instead the posterior densities for the global parameters β and θ. We find the parameter θ to be rather large, signalling that the heterogeneity of the graph is well captured by the latent space.

The computing time required to obtain the sample was about 46 hours (3.3 seconds per iteration). After convergence of the Markov chain, we also ran the non-noisy MH sampler for 50 additional iterations, to compare the computational efficiency of the two methods. The non-noisy MH sampler required an average of 453 seconds per iteration, corresponding to a theoretical 262 days of computations for the full sample. The vast difference between the two computing times highlights the scalability of our method, which extends the applicability of LPMs to networks of much larger sizes.

9.2. Poisson case

The coauthorship network data was in fact collected as a weighted graph, where the non-negative weights of edges represent the strength of the collaboration

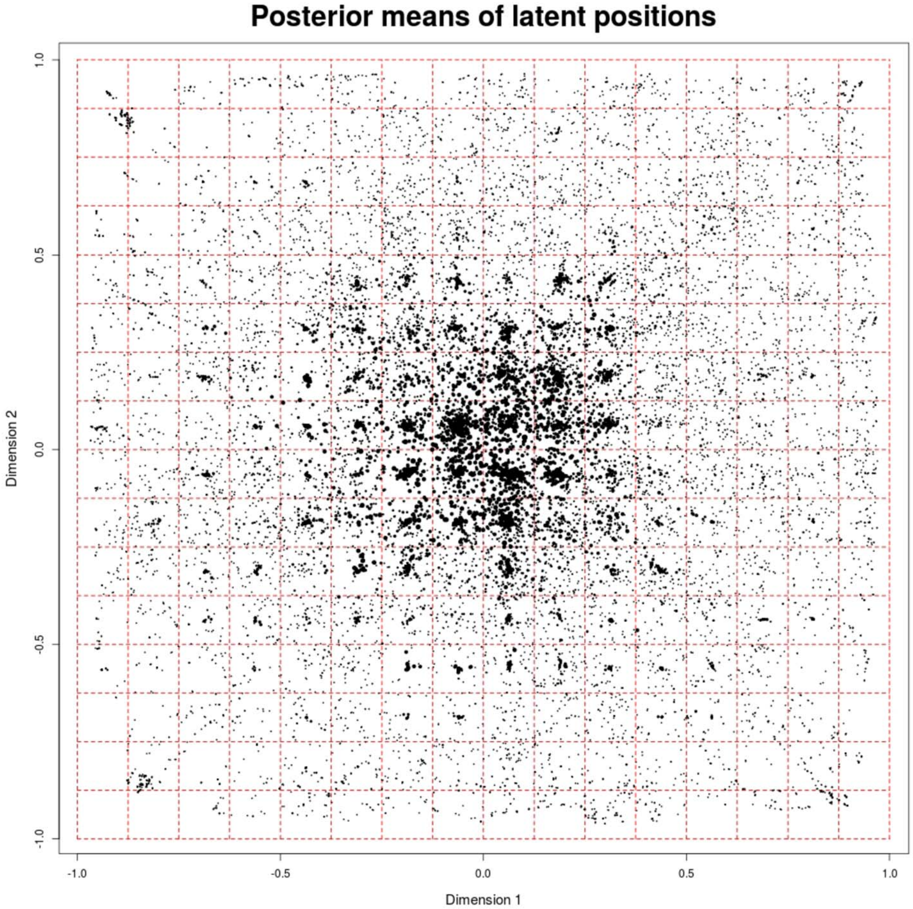

FIG 9.1. ***Astrophysics***. *Average latent positions of the nodes with circle size proportional to the node's degree. The grid in dashed red line corresponds to the partitioning imposed.*

relations[4]. This dataset offers then a nice setup to extend our methodology to a discrete Poisson setup, where edge weights are non-negative integers. The extension of our framework turns out to be rather straightforward, since the model can be defined as:

$$\begin{aligned} Y_{ij} &\sim Pois(\lambda_{ij}) \\ \log \lambda_{ij} &= \beta - e^{\theta} d\left(\mathbf{z}_i, \mathbf{z}_j\right) \end{aligned} \tag{9.1}$$

As regards `NoisyLPM`, the implementation of the algorithm requires some modifications to take into account the fact that the grid does not partition the

[4] Note that the edge weights are not exactly equal to the number of coauthored papers, however, they are constructed from this information through some rescaling. Details can be found in the paper by [19]. In our paper, we use these weights in a discretised way to fit our Poisson model.

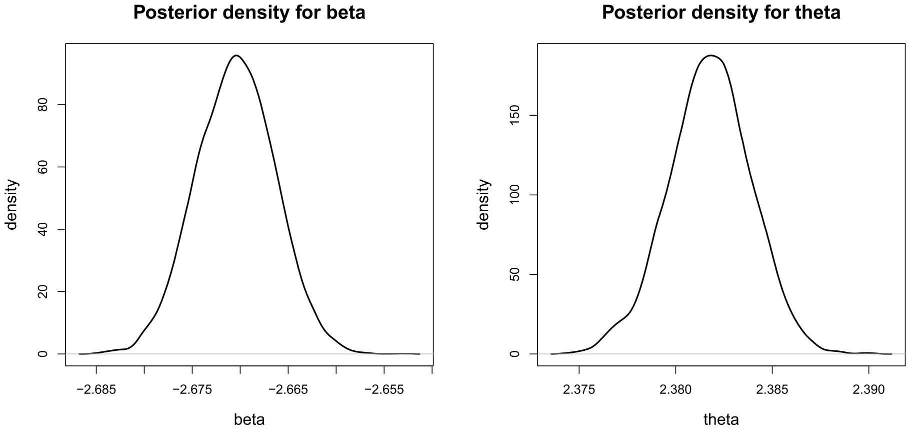

FIG 9.2. ***Astrophysics***. *Posterior densities for the global parameters β and θ.*

nodes into homogeneous boxes: the boxes now include different nodes which give different likelihood contributions based on their Poisson weight. To overcome this, we add an new dimension to our grid, whereby we split up each box into subgroups of nodes that carry the same Poisson weight. In other words, we further break down the partition to recover the homogeneity within each of the partition sets.

As a demonstration, we fit the model to the same data, this time using a grid obtained through $M = 8$: the latent space is shown in Figure 9.3. The latent space for the weighted case appears to be equivalent to that obtained in the binary case, with a stronger clustering effect around the centres of the boxes, as a consequence of using a coarser grid. The posterior average of β and θ are -2.83 and 2.34, respectively. This again signals a relatively strong latent space effect in the model, and thus a good fit to the data.

10. Conclusions

In this paper, we have introduced a new methodology to perform inference on latent position models. Our approach specifically addresses a crucial issue: the scalability of the method with respect to the size of the network. By taking advantage of a discretisation of the latent space, our proposed approach is characterised by a reduced computational complexity compared to the state-of-the-art procedures.

The framework introduced heavily relies on several important results introduced in the context of noisy MCMC. We have followed the core ideas of such strand of literature, and adapted the main results to the latent position model context, thereby giving theoretical guarantees for our proposed approximate method. In particular, our results underline the existence of a trade-off between the speed and the bias of the noisy algorithm, whereby the user can arbitrarily increase the accuracy at the expense of speed.

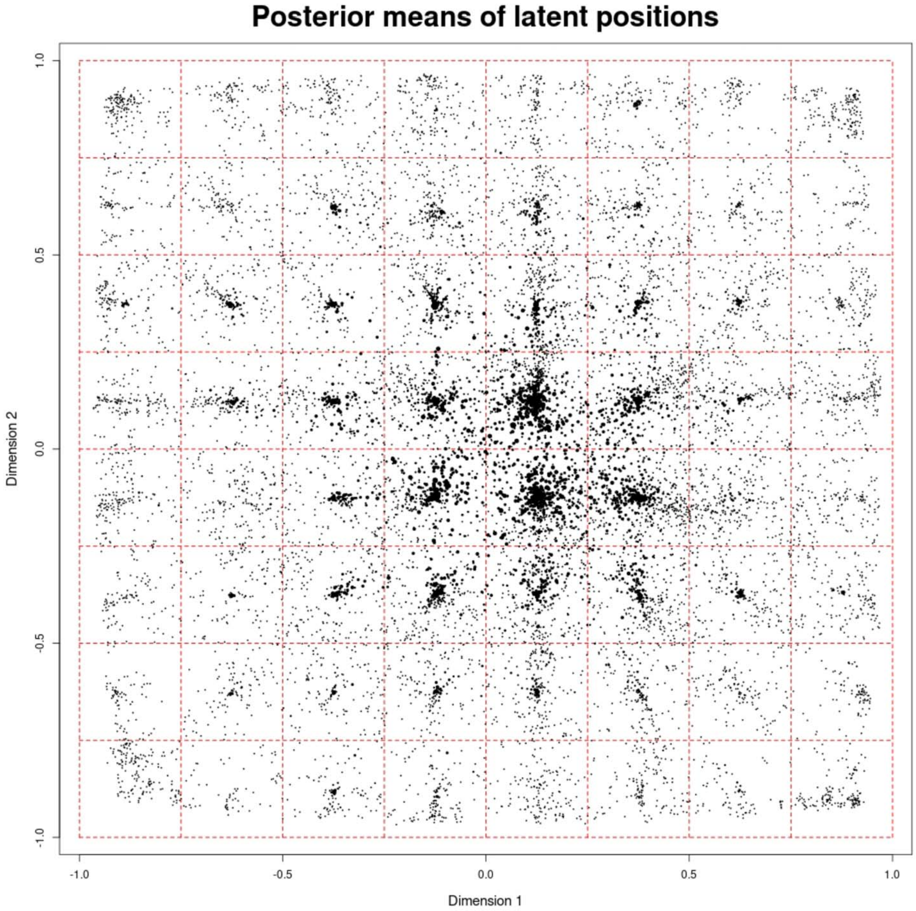

FIG 9.3. ***Astrophysics, weighted****. Average latent positions of the nodes with circle size proportional to the node's weighted degree. The grid in dashed red line corresponds to the partitioning imposed.*

Additionally, we have proposed applications to both simulated and real datasets. When compared to the non-noisy algorithm, the noisy results did not show any relevant qualitative difference, yet they were obtained with a substantially smaller computing time. In an asymptotic context for networks of large size, we show that our theoretical result can provide some guidance to determine a suitable approximation framework. In our simulations we show that our theoretical analysis leads to results that are conservative in the asymptotic setting, both in terms of likelihood errors and in terms of algorithmic complexity. This opens up the possibility of future research that may explore whether our theoretical bounds can be refined, and whether there may be other procedures to optimise the latent space discretisation.

In terms of applicability, while we describe and illustrate our method in the case of binary networks, we also provide an application to Poisson weighted networks, showing that the method could be extended to other common network structures. The code for `NoisyLPM` is available from the public GitHub repository [26].

A limitation of our work is that it might not cope well with an increasing number of latent dimensions. This would imply an increase in the dimensionality of the grid, and, in turn, a much larger number of boxes. Similarly, introducing nodal random effects or covariates would create additional heterogeneity within any given box, thus requiring further dimensions in the latent grid. This would make our approach impractical. For these reasons, our approach, in its current version, is best suited for a setting where covariates are not available or not included, and where we are interested in a two dimensional latent space. We stress the importance of future research in this direction, which could address this limitation and thus make our method more widely applicable and useful in applied fields.

Finally, our work can be easily extended to include different distributions on the latent space, such as Gaussian mixture models [10] or different types of edge probabilities, or, more generally, networks factor models, such as the projection models of [11].

11. Appendix

Before proceeding with two lemmas that will be useful in the proof of Proposition 2, we start with some elements of context. Consider a state-space $(\mathcal{S},\mathcal{A})$ and define $\mathcal{M}_1$ the set of probability measures on $(\mathcal{S},\mathcal{A})$. A Markov kernel P operates (from the left) on $\mathcal{M}_1$ by $\mu \mapsto \mu P := \int \mu(\mathrm{d}\boldsymbol{\theta})P(\boldsymbol{\theta},\,\cdot\,) \in \mathcal{M}_1$. Define the total variation of P as the number $\|P\| := \sup_{\mu\in\mathcal{M}_1}\|\mu P\| = \sup_{\boldsymbol{\theta}\in\mathcal{S}}\|P(\boldsymbol{\theta},\,\cdot\,)\|$ where for any measure μ defined on $(\mathcal{S},\mathcal{A})$, the positive number $\|\mu\| := \sup_{A\in\mathcal{A}}|\mu(A)|$ denotes its total variation.

Lemma 1. *The total variation norm of a Markov kernel is* 1*.*

Proof. On the one hand, we have

$$\|P\| = \sup_{\boldsymbol{\theta}\in\mathcal{S}}\|P(\boldsymbol{\theta},\,\cdot\,)\| = \sup_{\boldsymbol{\theta}\in\mathcal{S}}\sup_{A\in\mathcal{A}}|P(\boldsymbol{\theta},A)| \leq 1$$

but, since $\mathcal{S}\in\mathcal{A}$, $1 = P(\boldsymbol{\theta},\mathcal{S}) \leq \sup_{A\in\mathcal{A}}P(\boldsymbol{\theta},A)$, for all $\boldsymbol{\theta}\in\mathcal{S}$, and we thus also have that

$$1 = \sup_{\boldsymbol{\theta}\in\mathcal{S}}P(\boldsymbol{\theta},\mathcal{S}) \leq \sup_{\boldsymbol{\theta}\in\mathcal{S}}\sup_{A\in\mathcal{A}}|P(\boldsymbol{\theta},A)| = \|P\|\ .$$

□

We now consider a signed Markov kernel K which is the difference between two Markov kernels P_1 and P_2, that is $K = P_1 - P_2$. Define by $\mathcal{M}_0$ the set of signed measures on $(\mathcal{S},\mathcal{A})$ such that for all $\mu\in\mathcal{M}_0$, $\mu(\mathcal{S}) = 0$. It is easy to see that K operates (from the left) on $\mathcal{M}_0$ by $\mu\mapsto\mu K := \int\mu(\mathrm{d}\boldsymbol{\theta})K(\boldsymbol{\theta},\,\cdot\,)\in\mathcal{M}_0$. Hence, we define the total variation of such a signed Markov kernel K by $\|K\| = \sup_{\mu\in\mathcal{M}_0}\|\mu K\| = \sup_{\mu\in\mathcal{M}_0}\sup_{A\in\mathcal{A}}|\mu K(A)|$.

Lemma 2. *The total variation norm for signed kernels is submultiplicative: for two signed kernels K_1 and K_2 wih $K_1(\boldsymbol{\theta}, \mathcal{S}) = K_2(\boldsymbol{\theta}, \mathcal{S}) = 0$, we have $\|K_1K_2\| \leq \|K_1\|\|K_2\|$.*

Proof. First, note that

$$\begin{aligned}\sup_{\nu\in\mathcal{M}_0} \|\nu K_1K_2\| &= \max\left[\sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|>0} \|\nu K_1K_2\|, \sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|=0} \|\nu K_1K_2\|\right] \\ &= \sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|>0} \|\nu K_1K_2\|\,,\end{aligned} \tag{11.1}$$

since for all $\nu \in \mathcal{M}_0$, $\|\nu K_1\| = 0$ implies that νK_1 is the null measure, which implies that νK_1K_2 is also the null measure and thus that $\|\nu K_1K_2\| = 0$. We have that

$$\begin{aligned}\|K_1K_2\| = \sup_{\nu\in\mathcal{M}_0} \|\nu K_1K_2\| &= \sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|>0} \|\nu K_1K_2\| \\ &= \sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|>0} \frac{\|\nu K_1K_2\|}{\|\nu K_1\|}\|\nu K_1\| \\ &\leq \sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|>0} \frac{\|\nu K_1K_2\|}{\|\nu K_1\|} \sup_{\nu\in\mathcal{M}_0} \|\nu K_1\| \\ &= \sup_{\nu\in\mathcal{M}_0\,:\,\|\nu K_1\|>0} \frac{\|\nu K_1K_2\|}{\|\nu K_1\|}\|K_1\|\,.\end{aligned} \tag{11.2}$$

But note that for all $\nu \in \mathcal{M}_0$ with $\|\nu K_1\| > 0$, we have for all $A \in \mathcal{A}$

$$\frac{\nu K_1K_2}{\|\nu K_1\|}(A) = \iint \frac{\nu(\mathrm{d}\boldsymbol{\theta})K_1(\boldsymbol{\theta}, \mathrm{d}\boldsymbol{\theta}')}{\|\nu K_1\|}K_2(\boldsymbol{\theta}', A) = \int \nu'(\mathrm{d}\boldsymbol{\theta}')K_2(\boldsymbol{\theta}', A) = \nu' K_2(A)\,,$$

where $\nu' = \int \nu(\mathrm{d}\boldsymbol{\theta})\frac{K_1(\boldsymbol{\theta},\,\cdot\,)}{\|\nu K_1\|}$. This implies that $\frac{\|\nu K_1K_2\|}{\|\nu K_1\|} = \|\nu' K_2\|$. But since $\nu' \in \mathcal{M}_0$,

$$\frac{\|\nu K_1K_2\|}{\|\nu K_2\|} \leq \sup_{\nu'\in\mathcal{M}_0} \|\nu' K_2\| = \|K_2\|\,,$$

which does not depend on ν. The proof is completed by taking the supremum over all $\nu \in \mathcal{M}_0$ in the last inequality. □

11.1. Proof of Proposition 1

Proof. By definition: $\|P - \tilde{P}\| = \sup_{\boldsymbol{\theta}\in\mathcal{S}} \|P(\boldsymbol{\theta}, \cdot) - \tilde{P}(\boldsymbol{\theta}, \cdot)\|$. Now, $P(\boldsymbol{\theta}, \cdot)$ and $\tilde{P}(\boldsymbol{\theta}, \cdot)$ are measures that admit a similar decomposition, and, in particular for any $\boldsymbol{\theta} \in \mathcal{S}$ and $A \in \mathcal{A}$,

$$\begin{aligned}P(\boldsymbol{\theta}, A) &= \int_A Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) + \delta_{\boldsymbol{\theta}}(A)\, r(\boldsymbol{\theta})\,, \\ \tilde{P}(\boldsymbol{\theta}, A) &= \int_A Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) + \delta_{\boldsymbol{\theta}}(A)\, \tilde{r}(\boldsymbol{\theta})\end{aligned} \tag{11.3}$$

so that the signed measure $\mu_{\boldsymbol{\theta}} := P(\boldsymbol{\theta}, \cdot) - \tilde{P}(\boldsymbol{\theta}, \cdot)$ decomposes as:

$$\mu_{\boldsymbol{\theta}}(A) = \int_A Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\left(\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right) + \delta_{\boldsymbol{\theta}}(A)\left(r(\boldsymbol{\theta}) - \tilde{r}(\boldsymbol{\theta})\right) \ . \tag{11.4}$$

It is well known that for this type of measure with atoms, the total variation verifies:

$$\|\mu_{\boldsymbol{\theta}}\| = \frac{1}{2}\int_A Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\left|\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right| + \frac{1}{2}\left|r(\boldsymbol{\theta}) - \tilde{r}(\boldsymbol{\theta})\right| \ . \tag{11.5}$$

But since $r(\boldsymbol{\theta}) = \int Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\left(1 - \alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right)$, with a similar result for $\tilde{r}(\boldsymbol{\theta})$, we have that:

$$\begin{aligned} \left|r(\boldsymbol{\theta}) - \tilde{r}(\boldsymbol{\theta})\right| &= \left|\int Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\left(\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right)\right| \\ &\leq \int Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\left|\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right| \ . \end{aligned} \tag{11.6}$$

Since by assumption $\left|\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right| \leq \omega$, we have that:

$$\|\mu_{\boldsymbol{\theta}}\| \leq \int Q\left(\boldsymbol{\theta}, d\boldsymbol{\theta}'\right)\left|\alpha\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right) - \tilde{\alpha}\left(\boldsymbol{\theta} \to \boldsymbol{\theta}'\right)\right| \leq \omega \ , \tag{11.7}$$

which is independent of $\boldsymbol{\theta}$ and thus concludes the proof. □

11.2. Proof of Proposition 2

Proof. If $R = 1$ then $\|P_{[1]} - \tilde{P}_{[1]}\| = \|P_1 - \tilde{P}_1\|$. If $R = 2$, then, using the fact that the signed kernel $P_r - \tilde{P}_r$ satisfies conditions of Lemma 2, we have

$$\begin{aligned} \|P_{[2]} - \tilde{P}_{[2]}\| &= \|P_1P_2 - \tilde{P}_1\tilde{P}_2\| \\ &= \|P_1(P_2 - \tilde{P}_2) + \tilde{P}_2(P_1 - \tilde{P}_1)\| \\ &\leq \|P_1(P_2 - \tilde{P}_2)\| + \|\tilde{P}_2(P_1 - \tilde{P}_1)\| \\ &\leq \|P_1\|\|P_2 - \tilde{P}_2\| + \|\tilde{P}_2\|\|P_1 - \tilde{P}_1\| \\ &= \|P_2 - \tilde{P}_2\| + \|P_1 - \tilde{P}_1\| \, , \end{aligned} \tag{11.8}$$

where last inequality follows from Lemma 1. Now, we assume that (7.6) is valid for every $r \leq R-1$, and prove the statement for $r = R$. We note that $P_{[R-1]}$ and $\tilde{P}_{[R-1]}$ are both Markov kernels. Hence:

$$\begin{aligned} \|P_{[R]} - \tilde{P}_{[R]}\| &= \|P_RP_{[R-1]} - \tilde{P}_R\tilde{P}_{[R-1]}\| \\ &\leq \|P_R(P_{[R-1]} - \tilde{P}_{[R-1]})\| + \|\tilde{P}_{[R-1]}(P_R - \tilde{P}_R)\| \\ &= \|P_R - \tilde{P}_R\| + \|P_{[R-1]} - \tilde{P}_{[R-1]}\| \end{aligned} \tag{11.9}$$

proving the proposition by mathematical induction. □

11.3. Preliminary results

We define here two independent results which are needed to prove the main theorem of our paper.

Lemma 3. *Let $x,\ y \in \mathbb{R}^+$, then:*

$$|1 \wedge x - 1 \wedge y| \leq |\log x - \log y| \ . \tag{11.10}$$

Proof. We note that:

$$|1 \wedge x - 1 \wedge y| \leq |\log(1 \wedge x) - \log(1 \wedge y)| \ , \tag{11.11}$$

because the logarithm function acts as a expansive mapping locally in the set $[0,1]$, or, equivalently, the exponential function is a contraction in the set $[0,1]$.

Then, we can proceed with:

$$|\log(1 \wedge x) - \log(1 \wedge y)| \leq |1 \wedge \log x - 1 \wedge \log y| \ . \tag{11.12}$$

The following cases are now possible:

1. If $x \geq 1$ and $y \geq 1$, then $|1 \wedge \log x - 1 \wedge \log y| \leq 0$.
2. If $x < 1$ and $y < 1$, then $|1 \wedge \log x - 1 \wedge \log y| = |\log x - \log y|$.
3. If $x < 1$ and $y \geq 1$ (or, analogously, $x \geq 1$ and $y < 1$), then

$$|1 \wedge \log x - 1 \wedge \log y| \leq |\log x - 0| \leq |\log x - \log y| \ ,$$

because $\log x$ is a negative number.

As a worst case scenario across the above cases, we have the required statement

$$|1 \wedge x - 1 \wedge y| \leq |\log x - \log y| \ . \qquad \square$$

Lemma 4. *Let x, y and z be real numbers such that $x \wedge y \geq z > 0$. Then:*

$$|\log x - \log y| \leq \frac{1}{z}|x - y| \ . \tag{11.13}$$

Proof. We assume, without loss of generality, that $x \geq y$. We have:

$$\begin{aligned} |\log x - \log y| &= \left|\log \frac{x}{y}\right| = \left|\log \frac{y + x - y}{y}\right| \\ &= \left|\log\left(1 + \frac{x-y}{y}\right)\right| \leq \left|\frac{x-y}{y}\right| \leq \frac{1}{y}|x - y| \ ; \end{aligned} \tag{11.14}$$

where we have used the fact that $\log(1 + u) \leq u$ for any $u \geq 0$.

In the case of $y \geq x$, the inequality becomes

$$|\log x - \log y| \leq \frac{1}{x}|x - y| \ ; \tag{11.15}$$

so, we can combine the two inequalities to obtain the statement of the lemma, for any $z \leq x \wedge y$. $\square$

Proposition 4. *If the edge probability function $\rho\left(d, \boldsymbol{\psi}\right)$ defined in Assumption 2 satisfies:*

$$\left|\rho\left(d, \boldsymbol{\psi}\right) - \rho\left(\tilde{d}, \boldsymbol{\psi}\right)\right| \leq \kappa_0 |d - \tilde{d}| \ ,$$

for some distances d and $\tilde{d}$, model parameters $\boldsymbol{\psi}$, and positive constant κ_0, then it also satisfies:

$$\left|\log\rho\left(d, \boldsymbol{\psi}\right) - \log\rho\left(\tilde{d}, \boldsymbol{\psi}\right)\right| \leq \kappa_1 |d - \tilde{d}| \ ;$$

for a suitable positive constant κ_1.

Proof. Thanks to Lemma 4, we have:

$$\begin{aligned}\left|\log\rho\left(d, \boldsymbol{\psi}\right) - \log\rho\left(\tilde{d}, \boldsymbol{\psi}\right)\right| &\leq \frac{1}{\rho\left(d, \boldsymbol{\psi}\right) \wedge \rho\left(\tilde{d}, \boldsymbol{\psi}\right)} \left|\rho\left(d, \boldsymbol{\psi}\right) - \rho\left(\tilde{d}, \boldsymbol{\psi}\right)\right| \\ &\leq \frac{1}{\rho^{\mathcal{L}}} \left|\rho\left(d, \boldsymbol{\psi}\right) - \rho\left(\tilde{d}, \boldsymbol{\psi}\right)\right| \ ;\end{aligned} \tag{11.16}$$

where $\rho^{\mathcal{L}} = \inf_{d \geq 0} \{\rho\left(d, \boldsymbol{\psi}\right)\}$. We can now use the Proposition's assumption to obtain the result. □

Corollary 3. *Analogously to the previous proposition, we have that*

$$\left|\log\left[1 - \rho\left(d, \boldsymbol{\psi}\right)\right] - \log\left[1 - \rho\left(\tilde{d}, \boldsymbol{\psi}\right)\right]\right| \leq \kappa_2 |d - \tilde{d}| \ ,$$

for a suitable positive constant κ_2.

11.4. Proof of Theorem 1

Proof. First we use Lemma 3 to obtain:

$$\left|\alpha_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right) - \tilde{\alpha}_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right)\right| \leq \left|\log\frac{\pi\left(\mathbf{z}_i' | \mathcal{Z}_{-i}, \boldsymbol{\psi}, \mathcal{Y}\right)}{\pi\left(\mathbf{z}_i | \mathcal{Z}_{-i}, \boldsymbol{\psi}, \mathcal{Y}\right)} - \log\frac{\tilde{\pi}\left(\mathbf{z}_i' | \mathcal{Z}_{-i}, \boldsymbol{\psi}, \mathcal{Y}\right)}{\tilde{\pi}\left(\mathbf{z}_i | \mathcal{Z}_{-i}, \boldsymbol{\psi}, \mathcal{Y}\right)}\right| \ . \tag{11.17}$$

Simplify the notation with:

$$\begin{aligned} p_{ij} &= p\left(\mathbf{z}_i, \mathbf{z}_j; \boldsymbol{\psi}\right) \\ p_{ij}' &= p\left(\mathbf{z}_i', \mathbf{z}_j; \boldsymbol{\psi}\right) \\ q_{ij} &= 1 - p\left(\mathbf{z}_i, \mathbf{z}_j; \boldsymbol{\psi}\right) \\ q_{ij}' &= 1 - p\left(\mathbf{z}_i', \mathbf{z}_j; \boldsymbol{\psi}\right) \end{aligned}$$

with approximate counterparts indicated with a tilde, respectively.

We note that in the right hand side of (11.17), all the terms referring to the prior and proposal distributions cancel each other out. The only remaning terms are:

$$
\begin{aligned}
&\left|\alpha_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right) - \tilde{\alpha}_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right)\right| \leq \\
&\qquad \leq \left| \sum_{j\in\mathcal{V}\setminus\{i\}} y_{ij}\left[\log p_{ij}' - \log \tilde{p}_{ij}' - \log p_{ij} + \log \tilde{p}_{ij}\right] + \right. \\
&\qquad\qquad \left. + \sum_{j\in\mathcal{V}\setminus\{i\}} (1-y_{ij})\left[\log q_{ij}' - \log \tilde{q}_{ij}' - \log q_{ij} + \log \tilde{q}_{ij}\right] \right| \\
&\qquad \leq \sum_{j\in\mathcal{V}\setminus\{i\}} y_{ij}\left[\left|\log p_{ij}' - \log \tilde{p}_{ij}'\right| + \left|\log p_{ij} - \log \tilde{p}_{ij}\right|\right] + \\
&\qquad\qquad + \sum_{j\in\mathcal{V}\setminus\{i\}} (1-y_{ij})\left[\left|\log q_{ij}' - \log \tilde{q}_{ij}'\right| + \left|\log q_{ij} - \log \tilde{q}_{ij}\right|\right] \ .
\end{aligned}
\tag{11.18}
$$

We note that each of the terms in absolute values are upper bounded by κb, for some positive constant κ. This follows from Proposition 4, whereby $|d - \tilde{d}| \leq b\sqrt{2}$ by construction. Hence, we can conclude with

$$
\left|\alpha_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right) - \tilde{\alpha}_{\mathcal{Z}}\left(\mathbf{z}_i \to \mathbf{z}_i'\right)\right| \leq \kappa' b N \ . \tag{11.19}
$$

The proof for the global parameters is done analogously. □

11.5. Uniform convergence of Metropolis-Hastings kernels operating on a compact state space

Theorem 3. *Let $\mathcal{S}$ be a bounded state space with $\mathcal{S} \subset \mathbb{R}^d$ (for some $d > 0$) and $\mathcal{A}$ be a sigma-algebra on $\mathcal{S}$. Let P be a MH kernel operating on $\mathcal{S} \times \mathcal{A}$ with invariant distribution π defined on $(\mathcal{S}, \mathcal{A})$. Then the function $u \mapsto \|P(u,,\cdot)^t - \pi\|$ converges uniformly to 0 as $t \to \infty$, at a geometric rate.*

Proof. For simplicity, we take the case $d = 3$, but generalizing the following reasoning for all $d > 0$ is straightforward. Denoting with P_i the MH kernel that keeps $x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ fixed, we have for all $x \in \mathcal{S}$

$$
P_i(x, \mathrm{d}x') = \left\{Q_i(x, \mathrm{d}x_i')\alpha_i(x, x') + \delta_{x_i}(\mathrm{d}x_i')\rho_i(x)\right\}\delta_{x_{-i}}(\mathrm{d}x_{-i}') \,, \tag{11.20}
$$

where Q_i is the proposal kernel of the i-th dimension, $\alpha_i(x, x') = 1 \wedge \pi(x')Q_i(x'x)/\pi(x)Q_i(x, x')$ and $\rho_i(x) = 1 - \int Q(x, \mathrm{d}x')\alpha_i(x, x')$. With regulatory conditions on the proposal kernels $Q_1, Q_2, \ldots$ and since the state space is compact, we have for all $i \in \{1, \ldots, d\}$:

$$
\overline{Q}_i := \sup_{(x,y)\in\mathcal{S}^2} Q_i(x, y) < \infty \,, \qquad \underline{Q}_i := \inf_{(x,y)\in\mathcal{S}^2} Q_i(x, y) > 0 \,. \tag{11.21}
$$

Moreover, since the pdf of π is a continuous function and $\mathcal{S}$ is bounded, we have:

$$
0 < \underline{\pi} \leq \pi(x) \leq \overline{\pi} < \infty \,. \tag{11.22}
$$

Assuming that, for all i, Q_i is absolutely dominated by a common dominating measure, we have that $Q_i(x, \mathrm{d}x'_i) = Q(x, x'_i)\mathrm{d}x'_i$ which combined with Eqs. (11.20), (11.21) and (11.22) yields

$$P_i(x, \mathrm{d}x') \ge Q_i(x, x'_i)\alpha_i(x, x')\delta_{x_{-i}}(\mathrm{d}x'_{-i})\mathrm{d}x'_i \ge \underline{Q}_i\underline{\alpha}_i\delta_{x_{-i}}(\mathrm{d}x'_{-i})\mathrm{d}x'_i\,, \tag{11.23}$$

where $\underline{\alpha}_i := \underline{\pi}\underline{Q}_i/\overline{\pi}\overline{Q}_i$. Now, the (systematic-scan) Metropolis-Hastings transition kernel writes

$$\begin{aligned}
P(x, \mathrm{d}x') \quad &:= P_1P_2P_3(x, \mathrm{d}x') \\
&= \int P_1(x, \mathrm{d}y)P_2P_3(y, \mathrm{d}x')\,, \\
&\ge \int \underline{Q}_1\underline{\alpha}_1\mathrm{d}y_1P_2P_3(y_1, x_2, x_3, \mathrm{d}x')\,, \\
&\ge \int \underline{Q}_1\underline{\alpha}_1\mathrm{d}y_1 \int \underline{Q}_2\underline{\alpha}_2\mathrm{d}z_2P_3(y_1, z_2, x_3, \mathrm{d}x')\,, \\
&\ge \int \underline{Q}_1\underline{\alpha}_1\mathrm{d}y_1 \int \underline{Q}_2\underline{\alpha}_2\mathrm{d}z_2\underline{Q}_3\left\{1 \wedge \frac{\pi(x')\overline{Q}_3}{\underline{\pi}\underline{Q}_3}\right\}\mathrm{d}x'_3\delta_{y_1}(\mathrm{d}x'_1)\delta_{z_2}(\mathrm{d}x'_2)\,, \\
&\ge \left\{\prod_{i=1}^{2}\underline{Q}_i\underline{\alpha}_i\right\}\underline{Q}_3\left\{1 \wedge \frac{\pi(x')\underline{Q}_3}{\overline{\pi}\overline{Q}_3}\right\}\mathrm{d}x'\,,
\end{aligned}$$

since $\iint \mathrm{d}y_1\delta_{y_1}(\mathrm{d}x'_1)\mathrm{d}z_2\delta_{z_2}(\mathrm{d}x'_2) = \mathrm{d}x'_1\mathrm{d}x'_2$. Hence, defining ν as the absolutely continuous probability measure with pdf $\nu(x) \propto 1 \wedge \pi(x')\underline{Q}_3/\overline{\pi}\overline{Q}_3$, we have

$$P(x, \mathrm{d}x') \ge \beta\nu(\mathrm{d}x')\,, \tag{11.24}$$

with $\beta := \left\{\prod_{i=1}^{2}\underline{Q}_i\underline{\alpha}_i\right\}\underline{Q}_3\int 1\wedge\pi(x')\underline{Q}_3/\overline{\pi}\overline{Q}_3\mathrm{d}x'$. We conclude from Eq. (11.24) that the whole state space $\mathcal{S}$ is small for P and that therefore P is uniformly ergodic (with geometric rate $1 - \beta$), see e.g. Theorem 8 in [28]. □

Acknowledgments

Tha authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Funding

Part of this research has been carried out while R. R. was affiliated with the Institute of Statistics and Mathematics, Vienna University of Economics and Business, Vienna, Austria; and funded through the Vienna Science and Technology Fund (WWTF) Project MA14-031. F. M. is supported in part by NSERC of Canada through the Discovery Grants program. This research was also supported by the Insight Centre for Data Analytics through Science Foundation Ireland grant SFI/12/RC/2289.

References

[1] Alquier, P., Friel, N., Everitt, R. and Boland, A. (2016). Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing* **26** 29–47. MR3890764

[2] Boland, A., Friel, N. and Maire, F. (2017). Efficient MCMC for Gibbs Random Fields using pre-computation. *arXiv preprint arXiv:1710.04093*. MR3592054

[3] Durante, D. and Dunson, D. B. (2016). Locally adaptive dynamic networks. *The Annals of Applied Statistics* **10** 2203–2232. MR3750873

[4] Durante, D., Dunson, D. B. and Vogelstein, J. T. (2017). Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association* 1–15.

[5] Friel, N., Rastelli, R., Wyse, J. and 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** 6629-6634. MR0132566

[6] Gilbert, E. N. (1961). Random plane networks. *Journal of the Society for Industrial and Applied Mathematics* **9** 533–543.

[7] Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statistics* 455–472. MR3474046

[8] Gollini, I. and Murphy, T. B. (2014). Joint modelling of multiple network views. *Journal of Computational and Graphical Statistics*.

[9] Gormley, I. C. and Murphy, T. B. (2007). A latent space model for rank data. In *Statistical Network Analysis: Models, Issues, and New Directions* 90–102. Springer. MR2364300

[10] Handcock, M. S., Raftery, A. E. and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **170** 301–354. MR1951262

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

[12] Johndrow, J. E. and Mattingly, J. C. (2017). Error bounds for approximations of Markov chains. *arXiv preprint arXiv:1711.05382*. MR2758082

[13] Krivitsky, P. N., Handcock, M. S., Raftery, A. E. and Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social networks* **31** 204–213.

[14] Leskovec, J., Kleinberg, J. and Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **1** 2. MR3969054

[15] Maire, F., Friel, N. and Alquier, P. (2018). Informed sub-sampling MCMC: approximate Bayesian inference for large datasets. *Statistics and Computing* 1–34. `https://doi.org/10.1007/s11222-018-9817-3`. MR3419385

[16] Matias, C. and Robin, S. (2014). Modeling heterogeneity in random graphs through latent space models: a selective review. *ESAIM: Proceedings and Surveys* **47** 55–74. MR2203818

[17] Mitrophanov, A. Y. (2005). Sensitivity and convergence of uniformly ergodic Markov chains. *Journal of Applied Probability* **42** 1003–1014.

[18] Negrea, J. and Rosenthal, J. S. (2017). Error bounds for approximations of geometrically ergodic Markov chains. *arXiv preprint arXiv:1702.07441*. MR1975193

[19] Newman, M. E. J. (2001). Scientific collaboration networks. I. Network construction and fundamental results. *Physical review E* **64** 016131. MR1947255

[20] Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association* **96** 1077–1087. MR3663888

[21] Parsonage, E. and Roughan, M. (2017). Fast generation of spatially embedded random networks. *IEEE Transactions on Network Science and Engineering* **4** 112–119. MR3750874

[22] Raftery, A. E. (2017). Comment: extending the latent position model for networks. *Journal of the American Statistical Association* **112** 1531–1534.

[23] Raftery, A. E. and Lewis, S. M. (1995). The number of iterations, convergence diagnostics and generic Metropolis algorithms. *Practical Markov Chain Monte Carlo* **7** 763–773. MR3005803

[24] Raftery, A. E., Niu, X., Hoff, P. D. and 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** 901–919.

[25] Rastelli, R., Friel, N. and Raftery, A. E. (2016). Properties of latent variable network models. *Network Science* **4** 407–432. https://doi.org/10.1017/nws.2016.23

[26] Rastelli, R., Maire, F. and Friel, N. (2023). NoisyLPM GitHub repository, https://github.com/riccardorastelli/NoisyLPM. MR1624941

[27] Roberts, G. O. and Rosenthal, J. S. (1998). Two convergence properties of hybrid samplers. *The Annals of Applied Probability* **8** 397–407. MR2095565

[28] Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability surveys* **1** 20–71. MR3779696

[29] Rudolf, D. and Schweizer, N. (2017). Perturbation theory for Markov chains via Wasserstein distance. *Bernoulli* **24** 2610–2639.

[30] Ryan, C., Wyse, J. and Friel, N. (2017). Bayesian model selection for the latent position cluster model for Social Networks. *Network Science* **5** 70–91. MR3709558

[31] Salter-Townshend, M. and McCormick, T. H. (2017). Latent space models for multiview network data. *The Annals of Applied Statistics* **11** 1217–1244. MR2981116

[32] Salter-Townshend, M. and Murphy, T. B. (2013). Variational Bayesian inference for the latent position cluster model for network data. *Computational Statistics & Data Analysis* **57** 661–671. MR2958152

[33] Salter-Townshend, M., White, A., Gollini, I. and Murphy, T. B. (2012). Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **5** 243–264.

[34] Sarkar, P. and Moore, A. W. (2006). Dynamic social network analysis using latent space models. In *Advances in Neural Information Processing Systems* 1145–1152. MR3449061

[35] Sewell, D. K. and Chen, Y. (2015a). Latent space models for dynamic networks. *Journal of the American Statistical Association* **110** 1646–1657. MR3367791

[36] Sewell, D. K. and Chen, Y. (2015b). 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** 611–633. MR3449061

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

[38] Shortreed, S., Handcock, M. S. and Hoff, P. (2006). Positional estimation within a latent space model for networks. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences* **2** 24. MR0883333

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