1-1-2015

Sparse approximate inference for spatio-temporal point process models

Botond Cseke
University of Edinburgh

Andrew Zammit-Mangion
University of Wollongong, azm@uow.edu.au

Tom Heskes
Radboud University Nijmegen

Guido Sanguinetti
University of Edinburgh

Sparse approximate inference for spatio-temporal point process models

Abstract

Spatio-temporal log-Gaussian Cox process models play a central role in the analysis of spatially distributed systems in several disciplines. Yet, scalable inference remains computationally challenging both due to the high resolution modelling generally required and the analytically intractable likelihood function. Here, we exploit the sparsity structure typical of (spatially) discretised log-Gaussian Cox process models by using approximate message-passing algorithms. The proposed algorithms scale well with the state dimension and the length of the temporal horizon with moderate loss in distributional accuracy. They hence provide a flexible and faster alternative to both non-linear filtering-smoothing type algorithms and to approaches that implement the Laplace method or expectation propagation on (block) sparse latent Gaussian models. We infer the parameters of the latent Gaussian model using a structured variational Bayes approach. We demonstrate the proposed framework on simulation studies with both Gaussian and point-process observations and use it to reconstruct the conflict intensity and dynamics in Afghanistan from the WikiLeaks Afghan War Diary.

Publication Details

Cseke, B., Zammit-Mangion, A., Heskes, T. & Sanguinetti, G. (2016). Sparse approximate inference for spatio-temporal point process models. Journal of the American Statistical Association, 111 (516), 1746-1763.

Sparse Approximate Inference for Spatio-Temporal Point Process Models

Botond Cseke[*†1], Andrew Zammit-Mangion[‡2], Tom Heskes[§3], and Guido Sanguinetti[¶1]

[1]School of Informatics, University of Edinburgh

[2]National Institute for Applied Statistics Research Australia, School of Mathematics and Applied Statistics, University of Wollongong

[3]Faculty of Science, Radboud University Nijmegen

October 10, 2015

[*]Email: t-botcse@microsoft.com. B. Cs. was funded by BBSRC under grant BB/I004777/1.

[†]Current affiliation: Microsoft Research Cambridge, Machine Learning and Perception.

[‡]Email: azm@uow.edu.au. A. Z.-M. was partially funded by NERC under grant NE/I027401/1 while at the University of Bristol.

[§]Email: t.heskes@science.ru.nl. T.H. is a professor in Artificial Intelligence at R.U.N.

[¶]Email: gsanguin@inf.ed.ac.uk. G.S. is funded by ERC under grant MLCS-306999.

Abstract

Spatio-temporal log-Gaussian Cox process models play a central role in the analysis of spatially distributed systems in several disciplines. Yet, scalable inference remains computationally challenging both due to the high resolution modelling generally required and the analytically intractable likelihood function. Here, we exploit the sparsity structure typical of (spatially) discretised log-Gaussian Cox process models by using approximate message-passing algorithms. The proposed algorithms scale well with the state dimension and the length of the temporal horizon with moderate loss in distributional accuracy. They hence provide a flexible and faster alternative to both non-linear filtering-smoothing type algorithms and to approaches that implement the Laplace method or expectation propagation on (block) sparse latent Gaussian models. We infer the parameters of the latent Gaussian model using a structured variational Bayes approach. We demonstrate the proposed framework on simulation studies with both Gaussian and point-process observations and use it to reconstruct the conflict intensity and dynamics in Afghanistan from the WikiLeaks Afghan War Diary.

Keywords: latent Gaussian models, log-Gaussian Cox process, variational approximate inference, expectation propagation, sparse approximate inference, structure learning, conflict analysis.

1 Introduction

Dynamic models of spatially distributed point processes are widespread in scientific applications of computational statistics, ranging from environmental sciences (Wikle et al., 2001) to epidemiology (Diggle et al., 2005; Ahn et al., 2014) and ecology (Hooten and Wikle, 2008) to name but a few. The prevalence of such data is dramatically increasing due to advances in remote sensing technologies, and novel application domains are fast emerging in the social sciences due to the large scale data sets collected, for example through social networks. Log-Gaussian Cox processes (LGCPs) introduced in (Møller et al., 1998) are an important modelling paradigm for such systems, due to their ability to elegantly explain event-based data through the introduction of an auxiliary latent Gaussian field.

Despite their importance, inference in LGCPs remains computationally challenging. Markov chain Monte Carlo (MCMC) is frequently employed and has desirable asymptotic properties; however, despite considerable advances (Andrieu et al., 2010; Girolami and Calderhead, 2011; Yuan et al., 2012), the computational costs of sampling approaches remain high for high-dimensional latent fields, and in the presence of heterogeneous data sets. Deterministic approximations can provide a computationally effective alternative for computing the posterior distribution over the latent field, which is often the computational bottleneck in high dimensions. Current approaches can be broadly classified as blocked or dynamic. Blocked approaches cast the (time-discretised) model as a latent (sparse) Gaussian model with all state variables concatenated into a single large vector, and apply the Laplace method (Rue et al., 2009; Lindgren et al., 2011) or a corresponding expectation propagation (EP) algorithm (e.g., Cseke and Heskes, 2011). The computational cost of inference in the blocked approach is dominated by a sparse partial matrix inversion (Takahashi et al., 1973); these costs may become untenable for very high dimensions, and it is not always clear what further (robust) approximations could be used to alleviate these problems (e.g., Wist and Rue, 2006; Simpson et al., 2013). Dynamic approaches address the state inference problem using a filtering-smoothing (forward-backward) dynamic programming approach within a variational or EP approximation framework (e.g., Zammit-Mangion et al., 2012a; Ypma and Heskes, 2005; Hartikainen et al., 2011). Due to the non-conjugate likelihood, dynamic

approaches also have to resort to approximations, typically exploiting the message passing formulation of inference in graphical models (e.g., Lauritzen, 1996; Koller and Friedman, 2009). The cost of the forward-backward algorithm is typically cubic in the dimension of the state space due to the predictive update step in the Kalman filter.

In this paper we build on the dynamic approach to inference in spatio-temporal LGCPs, extending it in several ways in order to achieve efficiency in high-dimensional settings. First, we cast the model as a dynamic latent Gaussian model using time discretisation and basis-function projection, the weights of which define the state variables in a latent state-space model representation. Following this we derive a variational, joint state-parameter inference method for approximating the full posterior distribution over the states and unknown parameters. This approximate distribution is factored over states, parameters governing the state-interaction structure, and noise parameters, respectively. In the case of LGCPs, the message-passing algorithm for computing the approximate posterior distribution over the states is not analytically tractable since the likelihood is non-Gaussian. The key contribution of the paper is the derivation of an approximate message-passing algorithm for dealing with this intractability that does not suffer from the computational limitations arising from high-dimensional state spaces. We achieve this by enforcing a sparse structure of the messages, and adopting efficient sparse linear algebraic methods (Davis, 2006) in the local computations of the message-passing algorithm. This circumvents the limitation of typical forward-backward algorithms that invariably involve operations that destroy sparsity, for example due to matrix multiplication and marginalisation (as in the Kalman filter's update step). We show that the approximate message passing scheme we propose is an instance of expectation propagation (Minka, 2001) that can also be derived from the view of an expectation constrained approximate inference framework (Heskes et al., 2005; Opper and Winther, 2005).

The method naturally allows for a compromise between computation speed and accuracy. To show this we introduce a class of constraints that result in Gaussian messages having precision structures that are increasingly representative: (i) diagonal (factored messages), (ii) spanning tree (iii) chordal and finally (iv) fully connected (full messages). The latter case (iv) corresponds to the filtering-smoothing type algorithm that uses expectation propagation

to cater for the non-Gaussian parts of the model.

Comparisons in simulated case studies show that the proposed algorithms scale well with state dimension and, depending on the complexity of the messages, we can carry out approximate inference on thousands of state variables and hundreds of time-steps with reasonable time and memory requirements.

The text is structured as follows. In Section 2 we introduce log-Gaussian Cox processes and present the discretisation and numerical approximation steps that simplify this model to a dynamic latent Gaussian model with non-Gaussian likelihood terms. In Section 3 we describe the variational inference framework applied to this problem, and derive a class of dynamic message-passing algorithms that exploit the sparsity resulting from the discretisation. In Section 4 we carry out extensive simulation studies, discuss the performance of these algorithms and use them to extract the micro-dynamics of conflict events in the Afghan war (Zammit-Mangion et al., 2012a). Section 5 concludes the work.

2 Model

In this paper we are interested in the dynamic modelling of two-dimensional point patterns. The data consists of location- and time-stamped events $\mathcal{Y} = \{(\boldsymbol{s}_i, t_i)\}_i$ where the locations $\boldsymbol{s}_i$ are points in a two-dimensional compact domain $\mathcal{S} \subset \mathbb{R}\times\mathbb{R}$ and the time-stamps t_i are in a time interval $\mathcal{T} = [0, \max(\{t_i\}_i)]$. In order to model this type of data, we use log-Gaussian Cox process models (Møller et al., 1998) discretised in both space and time. We discretise the domain $\mathcal{S}$ by using a triangular lattice and using the corresponding piecewise linear finite element functions as basis functions. We discretise time by first dividing the time interval $\mathcal{T}$ into T time windows $\{\mathcal{T}_t\}_t$ of equal size Δ_t, that is $\mathcal{T}_t = [t\Delta_t, (t+1)\Delta_t)$ and $T\Delta_t = \max(\{t_i\}_i)$. We then treat the data $\mathcal{Y}$ as a set of spatial point processes indexed by t. Specifically, we let $\mathcal{Y} = \cup_t \mathcal{Y}_t$ where each $\mathcal{Y}_t$ contains the (spatial-only) events occurring in the window $\mathcal{T}_t$. The choice of Δ_t is often determined by the application and is in practice a difficult choice with potentially important computational repercussions. While a detailed discussion of this issue is beyond the scope of this article, we note that this choice is often expert-driven; failing that, non-parametric methods to quantify data autocorrelations (Zammit-Mangion et al.,

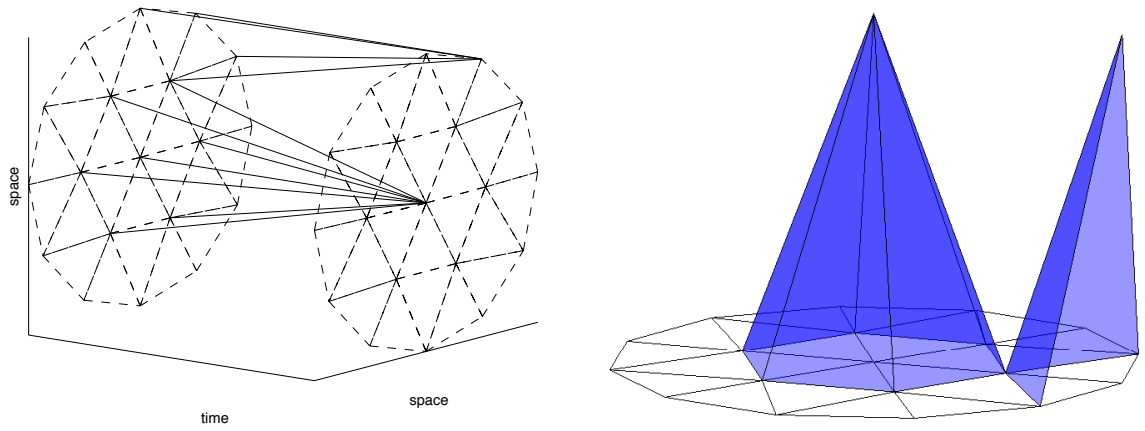

Figure 1: An illustration of the spatio-temporal discretisation employed. The right panel illustrates two basis functions defined according to the triangular finite element spatial discretisation. The bases are shown for two nodes/vertices, one in the interior and one on the boundary of the domain. The left panel illustrates the temporal connectivity for some of the nodes resulting from the spatio-temporal discretisation described in Section 2. Similarly, the temporal connectivities are shown only for an interior and a boundary node/vertex.

2012a) or other Bayesian model selection criteria (e.g., Kang et al., 2015) may be employed to determine a suitable discretisation.

We define the log intensity function of the point process as a linear combination of the n piecewise linear basis functions $\phi_j : \mathcal{S} \to \mathbb{R}; j = 1, \ldots, n$. That is, $\lambda(\boldsymbol{s}, t) \approx \exp\{\boldsymbol{x}_t^T \boldsymbol{\phi}(\boldsymbol{s})\}$, where $\boldsymbol{\phi}(\boldsymbol{s}) = (\phi_1(\boldsymbol{s}), \ldots, \phi_n(\boldsymbol{s}))^T$ and the weights (states) $\boldsymbol{x}_t \in \mathbb{R}^n$. We further assume that the weights $\boldsymbol{x}_t$ follow a linear dynamical system

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{\epsilon}_t, \tag{1}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ with both $\boldsymbol{A}$ and $\boldsymbol{Q}$ sparse. This linear dynamical model for the log intensity $u(\boldsymbol{s}, t) = \log \lambda(\boldsymbol{s}, t)$ can be derived from spatio-temporal models commonly employed in practice, such as the integro-difference equation (IDE) (Wikle, 2002), and the stochastic partial differential equation (SPDE) (Zammit-Mangion et al., 2012b). Sparsity in $\boldsymbol{A}$ and $\boldsymbol{Q}$ follows either from gridding the domain or from employing a Galerkin reduction on an infinite-dimensional system in $u(\boldsymbol{s}, t), \boldsymbol{s} \in \mathcal{S}, t \in \mathcal{T}$ using the basis functions $\{\phi_j(\boldsymbol{s})\}_j$.

The likelihood of the intensity $\lambda(\boldsymbol{s}, t)$ can be written as

$$p(\mathcal{Y} \mid \lambda) \propto \exp\Big\{ - \int_{\mathcal{S}\times\mathcal{T}} d\boldsymbol{s}dt \; \lambda(\boldsymbol{s}, t)\Big\} \times \prod_i \lambda(\boldsymbol{s}_i, t_i).$$

Using the spatial and temporal discretisation schemes outlined above, we re-write this likelihood as $p(\mathcal{Y} \mid \lambda) \propto \prod_t p(\mathcal{Y}_t \mid \boldsymbol{x}_t)$ where

$$\begin{aligned} p(\mathcal{Y}_t|\boldsymbol{x}_t) &\propto\!\!\!\propto \exp\Big\{ - \Delta_t \int_{\mathcal{S}} d\boldsymbol{s} \; e^{\boldsymbol{x}_t^T \boldsymbol{\phi}(\boldsymbol{s})}\Big\} \times \prod_{\boldsymbol{s}\in\mathcal{Y}_t} e^{\boldsymbol{x}_t^T \boldsymbol{\phi}(\boldsymbol{s})} \\ &= L_1(\boldsymbol{x}_t) \times L_2(\boldsymbol{x}_t; \mathcal{Y}_t). \end{aligned} \tag{2}$$

This likelihood can be split into two components: the first component $L_1(\boldsymbol{x}_t)$ is directly related to the *void probability* of the process. We adopt the approach in Simpson et al. (2011) and numerically approximate the integral as:

$$\begin{aligned} \log L_1(\boldsymbol{x}_t) &\approx -\Delta_t \sum_{j=1}^{p} \widetilde{\eta}_j \exp(\boldsymbol{\phi}^T(\bar{\boldsymbol{s}}_j)\boldsymbol{x}_t) \\ &= -\boldsymbol{\eta}^T \exp(\boldsymbol{W}\boldsymbol{x}_t), \end{aligned} \tag{3}$$

where the vector $\boldsymbol{\eta}$ denotes the integration weights $\Delta_t\widetilde{\boldsymbol{\eta}}$ and the matrix $\boldsymbol{W} = [\boldsymbol{\phi}(\bar{\boldsymbol{s}}_1), \ldots, \boldsymbol{\phi}(\bar{\boldsymbol{s}}_p)]^T$ contains the values of the basis at the chosen p integration points $\{\bar{\boldsymbol{s}}_j\}_j$. The second component of the likelihood, $L_2(\boldsymbol{x}_t; \mathcal{Y}_t)$, denotes the contributions from the observed events and can be represented as

$$\log L_2(\boldsymbol{x}_t; \mathcal{Y}_t) = \sum_{\boldsymbol{s}\in\mathcal{Y}_t} \boldsymbol{\phi}^T(\boldsymbol{s})\boldsymbol{x}_t = \boldsymbol{h}_t^T \boldsymbol{x}_t, \tag{4}$$

where $h_t^j = \sum_{\boldsymbol{s}\in\mathcal{Y}_t} \phi_j(\boldsymbol{s})$ is the sum of basis functions evaluated at the events' spatial coordinates. The approximate log-likelihood can hence be written, up to a constant, as

$$\log p(\mathcal{Y}_t|\boldsymbol{x}_t) \approx -\boldsymbol{\eta}^T \exp(\boldsymbol{W}\boldsymbol{x}_t) + \boldsymbol{h}_t^T \boldsymbol{x}_t. \tag{5}$$

Both compact basis functions and gridded domains induce sparsity in $\boldsymbol{W}$. In particular, if one chooses the integration points to be the vertices of a triangulation or the centres of gridded cells, then $\boldsymbol{W}$ simplifies to the identity matrix of size $n \times n$, $\boldsymbol{I}_{n\times n}$, where $n = p$. The integration weights $\boldsymbol{\eta}$ then correspond to the volumes (scaled by Δ_t) of the basis with unit weight (Simpson et al., 2011). Note from (5) that, since $\boldsymbol{W}$ is diagonal, the non-Gaussian terms depend only on x_t^j. In the following we use $\psi_{t,j}(x_t^j) = \exp\{-\eta_j \exp(x_t^j)\}$ to denote the non-Gaussian component of the likelihood terms $\tilde{\psi}_{t,j}(x_t^j) = \psi_{t,j}(x_t^j)\exp(h_t^j x_t^j)$ appearing in (5).

In our setting $\boldsymbol{Q}$ is a diagonal matrix, while we assume that the structure of the transition matrix $\boldsymbol{A}$ follows that of the neighbourhood graph that results from the discretisation. The matrix $\boldsymbol{A}$ hence describes small-scale, possibly directional, spatio-temporal dynamics, and it is reasonable to assume that $\boldsymbol{A}$ only has a select amount of non-zero elements on the neighbourhood structure. For this reason we impose a spike and slab prior on these structural elements by introducing a set of binary auxiliary variables $\boldsymbol{Z} = (z_{ij})_{i,j}$ with $z_{ij} \in \{0, 1\}$ and where $\boldsymbol{Z}$, like $\boldsymbol{A}$, has the same sparsity structure resulting from the discretisation. We then define the conditional prior $p(a_{ij} \mid z_{ij}, v_{\text{slab}}) = \mathcal{N}(a_{ij}; 0, v_{\text{slab}})^{z_{ij}}\delta(a_{ij})^{1-z_{ij}}$ where $\delta(\cdot)$ is the Dirac delta function, and assume a Bernoulli prior $p(z_{ij} \mid p_{\text{slab}}) = Ber(z_{ij}; p_{\text{slab}})$ (spike and slab prior). Consequently, we can use the posterior distribution of the variable z_{ij} to quantify the relevance of the transition coefficient a_{ij}. We use a Gamma prior $Gam(q_{ii}; k, \tau)$ for the diagonal elements of the precision matrix $\boldsymbol{Q}$. We conclude our model specification by letting $\boldsymbol{x}_1$ be Gaussian with mean $\boldsymbol{m}_1$ and covariance matrix $\boldsymbol{V}_1$. The hyper-parameters $\boldsymbol{\theta} = \{\boldsymbol{m}_1, \boldsymbol{V}_1, v_{\text{slab}}, p_{\text{slab}}, k, \tau\}$ are fixed.

With the above assumptions we can write down the joint probabilistic model as

$$\begin{aligned} p(\mathcal{Y}, \boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{Q} \mid \boldsymbol{\theta}) = p(\boldsymbol{x}_1) \prod_t p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{A}, \boldsymbol{Q}) \prod_j \tilde{\psi}_{t+1,j}(x_{t+1}^j) \\ \times \prod_{i\sim j} p(a_{ij} \mid z_{ij}, \boldsymbol{\theta}) p(z_{ij} \mid \boldsymbol{\theta}) \prod_j p(q_{jj} \mid \boldsymbol{\theta}), \end{aligned} \tag{6}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_t\}_t$ and $\{a_{ij}\}_{i\sim j}$ denotes all the structurally non-zero elements (following from

the neighbourhood graph) in $\boldsymbol{A}$. The prior distributions are given by

$$\begin{aligned} p(\boldsymbol{x}_1) &= \mathcal{N}(\boldsymbol{x}_1; \boldsymbol{m}_1, \boldsymbol{V}_1), \\ p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{A}, \boldsymbol{Q}) &= \mathcal{N}(\boldsymbol{x}_{t+1}; \boldsymbol{A}\boldsymbol{x}_t, \boldsymbol{Q}^{-1}), \\ p(a_{ij} \mid z_{ij}, v_{\text{slab}}) &= \mathcal{N}(a_{ij}; 0, v_{\text{slab}})^{z_{ij}} \delta(a_{ij})^{1-z_{ij}}, \\ p(z_{ij} \mid \boldsymbol{\theta}) &= Ber(z_{ij}; p_{\text{slab}}), \\ p(q_{jj} \mid \boldsymbol{\theta}) &= Gam(q_{jj}; k, \tau). \end{aligned}$$

3 Inference

In many application areas such as the ones mentioned in Section 1, the dimension of the state space and the length of the time horizon is in the range of hundreds or thousands which makes MCMC sampling from the posterior distribution computationally demanding. For this reason, we resort to variational approximate inference methods; we seek structured factorised approximations to the posterior distribution. Variational approximate inference methods formulate inference as an optimisation problem by using the Kullback-Leibler divergence $\mathrm{D}[\cdot||p]$ as optimisation objective (e.g., Jordan et al., 1999). These methods have been successfully applied in many areas of engineering and machine learning where large scale analytically intractable probabilistic models are common. In latent Gaussian models they provide tractable Gaussian approximations to analytically intractable Bayesan posteriors (e.g., Opper and Archambeau, 2009; Saul et al., 1996).

To apply variational inference to our problem, we approximate the posterior distribution $p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{Q} \mid \mathcal{Y}, \boldsymbol{\theta})$ in (6) with a factored distribution

$$q(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{Q} \mid \boldsymbol{\theta}) = q_X(\boldsymbol{X} \mid \boldsymbol{\theta}) q_{AZ}(\boldsymbol{A}, \boldsymbol{Z} \mid \boldsymbol{\theta}) q_Q(\boldsymbol{Q} \mid \boldsymbol{\theta}),$$

where the factors are the solution to the optimisation

$$\underset{q_X, q_{AZ}, q_Q}{\text{minimise}} \, \mathrm{D}[q_X(\boldsymbol{X} \mid \boldsymbol{\theta}) q_{AZ}(\boldsymbol{A}, \boldsymbol{Z} \mid \boldsymbol{\theta}) q_Q(\boldsymbol{Q} \mid \boldsymbol{\theta}) \,||\, p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{Q} \mid \mathcal{Y}, \boldsymbol{\theta})] \,. \tag{7}$$

To simplify notation, hereafter we omit the dependence of the distribution of interest on the hyper-parameters $\boldsymbol{\theta}$.

The optimality conditions of (7) can be used to define a component-wise fixed point iteration that is known to correspond to a coordinate descent towards a local optimum of the objective. These updates are

$$q_X(\boldsymbol{X})^{new} \propto p(\boldsymbol{x}_1) \prod_{t,j} \psi_{t+1,j}(x_{t+1}^j) \exp\Big\{ \sum_t \langle \log p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{A}, \boldsymbol{Q}) \rangle_{q_{AZ}, q_Q} \Big\}, \tag{8}$$

$$q_{AZ}(\boldsymbol{A}, \boldsymbol{Z})^{new} \propto \prod_{i \sim j} p(a_{ij} \mid z_{ij}) p(z_{ij}) \exp\Big\{ \sum_t \langle \log p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{A}, \boldsymbol{Q}) \rangle_{q_X, q_Q} \Big\}, \tag{9}$$

$$q_Q(\boldsymbol{Q})^{new} \propto \prod_i p(q_{ii} \mid \boldsymbol{\theta}) \exp\Big\{ \sum_t \langle \log p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{A}, \boldsymbol{Q}) \rangle_{q_X, q_{AZ}} \Big\}, \tag{10}$$

where we use $\langle \cdot \rangle_q$ to denote the expectation with respect to a distribution q. These updates are performed in a circular fashion to achieve a coordinate descent in each step. If the first and second moments of q_X are known, then (9) and (10), and hence also the exponential term in the update (8), can be easily found. However, non-Gaussian components of (8) prevent us from directly computing the moments of q_X.

To deal with the analytical intractability of q_X, we propose to compute the required expectations by applying further approximate inference techniques. We view q_X from a graphical model (Lauritzen, 1996) or factor graph (Kschischang et al., 2001) perspective and propose a novel large-scale extension of an approximate inference technique called expectation propagation (Opper and Winther, 2000; Minka, 2001). We show that the approximations we arrive at can be embedded into the wider, principled framework of variational approximate inference by using expectation constraints (Heskes et al., 2005; Opper and Winther, 2005). We extend this framework to accommodate our structured variational inference approach for the joint model $p(\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{Q} \mid \mathcal{Y}, \boldsymbol{\theta})$. In the following sections we present the algorithmic approach. A detailed, generic derivation of the proposed method is given in the Supplementary Material.

3.1 The model q_X

The model for q_X is a dynamic latent Gaussian model where the non-Gaussian terms $\psi_{t,j}(x_t^j)$ depend on only one state-space component x_t^j. We collect the Gaussian terms into $\Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \propto \exp\{\langle \log p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{A}, \boldsymbol{Q}) \rangle_{q_A q_Z, q_Q}\} \times \exp(\boldsymbol{h}_{t+1}^T \boldsymbol{x}_{t+1})$ and define q_X as

$$q_X(\boldsymbol{X}) \propto \prod_t \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \times \prod_{t,j} \psi_{t+1,j}(x_{t+1}^j). \tag{11}$$

The model is tree structured, and thus inference can be done by using the message-passing algorithm (e.g., Lauritzen, 1996) or the sum-product algorithm (e.g., Kschischang et al., 2001).

Message passing algorithms operate on so-called factor graphs. These are bipartite graphs for which the node sets consist of factors of a probability distribution and the corresponding variables (or groups thereof). A factor is connected to all variables (or groups thereof) that are subsets of its arguments. Message passing is a dynamic programming algorithm that computes marginal probability distributions in factor graphs with tree structure, that is, that do not contain any loops. To each edge of this graph we assign a pair of messages, one in each direction. The message from a variable node to a factor node is computed as the product of all incoming messages from the other factors. The message from a factor node to a variable node is the marginal of the product of the factor and all other incoming messages to that factor. The dynamic nature of the algorithm guarantees that once all messages are computed, the marginals over the variables (sometimes termed beliefs) can be formed as the product of the factors and the incoming messages (e.g., Kschischang et al., 2001).

In our case, the factors are $\{\Psi_{t,t+1}\}_t$ and $\{\prod_j \psi_{t,j}\}_t$, while the groups of variables are the states $\{\boldsymbol{x}_t\}_t$. The message passing updates corresponding to the factor graph representation

of this model (illustrated in Figure 2), read

$$
\begin{aligned}
\lambda^0_{t+1,j}(x^j_{t+1}) &\propto \psi_{t+1,j}(x^j_{t+1}),\\
\xi_{t+1}(\boldsymbol{x}_{t+1}) &\propto \int d\boldsymbol{x}_t\, \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})\hat{\xi}_t(\boldsymbol{x}_t),\\
\eta_t(\boldsymbol{x}_t) &\propto \int d\boldsymbol{x}_{t+1}\, \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})\hat{\eta}_{t+1}(\boldsymbol{x}_{t+1}),\\
\lambda^l_{t+1,j}(x^j_{t+1}) &\propto \int d\boldsymbol{x}^{\setminus j}_{t+1}\, \xi_{t+1}(\boldsymbol{x}_{t+1})\eta_{t+1}(\boldsymbol{x}_{t+1}) \prod_{k\neq j} \lambda^0_{t+1,k}(x^k_{t+1}),\\
\hat{\xi}_t(\boldsymbol{x}_t) &\propto \xi_t(\boldsymbol{x}_t) \prod_j \lambda^0_{t,j}(x^j_t),\\
\hat{\eta}_{t+1}(\boldsymbol{x}_{t+1}) &\propto \eta_{t+1}(\boldsymbol{x}_{t+1}) \prod_j \lambda^0_{t+1,j}(x^j_{t+1}).
\end{aligned}
$$

The messages ξ_{t+1}, η_t and $\lambda^0_{t+1,j}$ are factor-to-variable messages sent from $\Psi_{t,t+1}$ and $\psi_{t+1,j}$ to $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}_t$, while $\hat{\xi}_t, \hat{\eta}_{t+1}$ and $\lambda^l_{t+1,j}$ are the corresponding variable-to-factor messages.

By denoting $\alpha_t = \hat{\xi}_t$ and $\beta_t = \eta_t$ and writing the marginal densities corresponding to the factors as

$$q_X(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \propto \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})\alpha_t(\boldsymbol{x}_t)\beta_{t+1}(\boldsymbol{x}_{t+1}) \prod_j \lambda^0_{t+1,j}(x^j_{t+1}), \tag{12}$$

$$q_X(x^j_{t+1}) \propto \psi_{t+1,j}(x^j_{t+1})\lambda^l_{t+1,j}(x^j_{t+1}), \tag{13}$$

we can rewrite the algorithm in the following form:[1]

$$\lambda^0_{t+1,j}(x^j_{t+1})^{new}\, \lambda^l_{t+1,j}(x^j_{t+1}) \propto q_X(x^j_{t+1}), \tag{14}$$

$$\lambda^0_{t+1,j}(x^j_{t+1})\, \lambda^l_{t+1,j}(x^j_{t+1})^{new} \propto \int d\boldsymbol{x}_t d\boldsymbol{x}^{\setminus j}_{t+1}\, q_X(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}), \tag{15}$$

$$\alpha_{t+1}(\boldsymbol{x}_{t+1})^{new}\, \beta_{t+1}(\boldsymbol{x}_{t+1}) \propto \int d\boldsymbol{x}_t\, q_X(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}), \tag{16}$$

$$\alpha_t(\boldsymbol{x}_t)\, \beta_t(\boldsymbol{x}_t)^{new} \propto \int d\boldsymbol{x}_{t+1}\, q_X(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}). \tag{17}$$

In the typical approach, one performs forward-backward updates w.r.t α_t and β_t whilst also doing a $\lambda^l_{t,j}$ and $\lambda^0_{t,j}$ update at each time step. This algorithm is analogous to the well known

[1] See Supplementary Material for details.

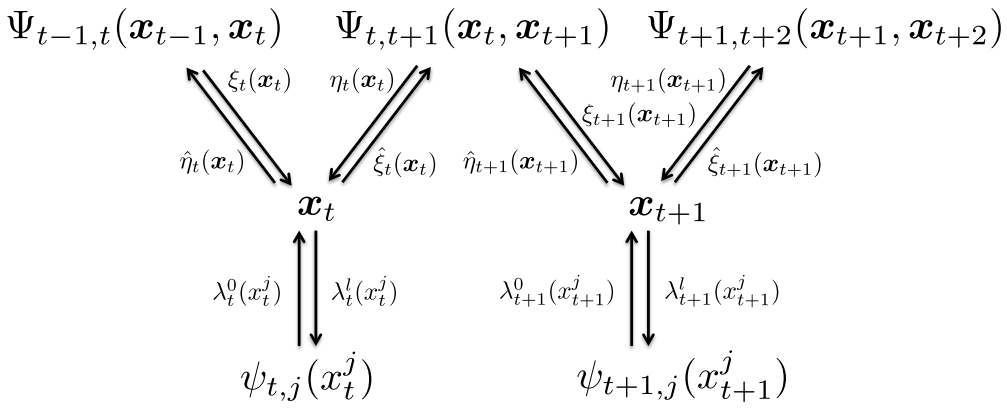

Figure 2: Illustration of the factor graph and the message-passing algorithm for the model q_X.

Rauch–Tung–Striebel smoothing algorithm for linear dynamical systems with a Gaussian likelihood.

In order to make the message passing tractable with a non-Guassian likelihood, we use Gaussian messages between the Gaussian and non-Gaussian terms. Moreover, we propose to use messages with restricted precision structures between the Gaussian factors to exploit sparsity and significantly reduce computing time; details are presented in the following subsections and in the Supplementary Material.

3.1.1 Approximations for non-Gaussian likelihoods

Due to the non-Gaussianity of $\psi_{t+1,j}$, the updates (14)–(17) cannot be computed analytically. To deal with this, we recast $\lambda^0_{t+1,j}(x^j_{t+1})$ as a Gaussian message by defining $\tilde{q}_{t+1}(x^j_{t+1}) \propto \psi_{t+1,j}(x^j_{t+1})\lambda^l_{t+1,j}(x^j_{t+1})$, and, based on (14), introducing the approximation

$$\lambda^0_{t+1,j}(x^j_{t+1})^{new}\lambda^l_{t+1,j}(x^j_{t+1}) = \text{Project}\big[\tilde{q}_{t+1}(x^j_{t+1}); \mathcal{N}\big]. \tag{18}$$

The operation Project$[q(\boldsymbol{x}); \mathcal{N}]$ is the projection of a distribution $q(\boldsymbol{x})$ into the Gaussian family $\mathcal{N}$ in the moment matching Kullback-Leibler sense, that is,

$$\text{Project}\big[q(\boldsymbol{x}); \mathcal{N}\big] = \underset{\tilde{q}\in\mathcal{N}}{\text{argmin}}\, \text{D}\big[q(\boldsymbol{x})||\tilde{q}(\boldsymbol{x})\big].$$

This projection finds a Gaussian distribution $\tilde{q}(\boldsymbol{x})$ that has the first and second moments identical to those of $q(\boldsymbol{x})$. Note that the method operates by projecting the marginal distributions resulting from the message passing and not the messages themselves (Minka, 2001; Heskes and Zoeter, 2002; Minka, 2005).

A natural consequence of the projection (18) is that the messages no longer compute the marginals $q_X(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ and $q_X(x_{t+1}^j)$ given in (12) and (13), but instead local approximate marginals, which we denote as $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ and $\tilde{q}_{t+1}(x_{t+1}^j)$, respectively. A further consequence of the approximation is that the resulting marginals only satisfy the weak consistency conditions $\text{Project}[\tilde{q}_{t+1}(x_{t+1}^j); \mathcal{N}] = \tilde{q}_{t,t+1}(x_{t+1}^j)$, that is, $\tilde{q}_{t+1}(x_{t+1}^j)$ and $\tilde{q}_{t,t+1}(x_{t+1}^j)$ only match in their first two moments. This becomes apparent when the approach is derived from an expectation constrained inference perspective, see Supplementary Material for more details.

The resulting algorithm that caters for the non-Gaussian components of the likelihood can be written as follows:

$$\lambda_{t+1,j}^0(x_{t+1}^j)^{new}\, \lambda_{t+1,j}^l(x_{t+1}^j) \propto \text{Project}\big[\tilde{q}_{t+1}(x_{t+1}^j); \mathcal{N}\big], \tag{19}$$

$$\lambda_{t+1,j}^0(x_{t+1}^j)\, \lambda_{t+1,j}^l(x_{t+1}^j)^{new} \propto \int d\boldsymbol{x}_t d\boldsymbol{x}_{t+1}^{\setminus j}\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}), \tag{20}$$

$$\alpha_{t+1}(\boldsymbol{x}_{t+1})^{new}\, \beta_{t+1}(\boldsymbol{x}_{t+1}) \propto \int d\boldsymbol{x}_t\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}), \tag{21}$$

$$\alpha_t(\boldsymbol{x}_t)\, \beta_t(\boldsymbol{x}_t)^{new} \propto \int d\boldsymbol{x}_{t+1}\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}), \tag{22}$$

where, similar to (12) and (13),

$$\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \propto \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \alpha_t(\boldsymbol{x}_t) \beta_{t+1}(\boldsymbol{x}_{t+1}) \prod_j \lambda_{t+1,j}^0(x_{t+1}^j), \tag{23}$$

$$\tilde{q}_t(x_{t+1}^j) \propto \psi_{t+1,j}(x_{t+1}^j) \lambda_{t+1,j}^l(x_{t+1}^j). \tag{24}$$

Note how, since the newly defined $\lambda_{t+1,j}^0(x_{t+1}^j)$ is Gaussian, the computations for the forward and backward approximate messages in (21) and (22) are now tractable. As such, from an implementation point of view, the only modification to the algorithm of (14)–(17) is the replacement of (14) with (18). The $\text{Project}[\tilde{q}_{t+1}(x_{t+1}^j); \mathcal{N}]$ step in (18) can be done

by univariate numerical quadrature. Due to the accuracy of these univariate methods, the numerical error in computing the moments is negligible. The (19) and (20) updates are performed for all j at once; this corresponds to the so-called parallel EP scheduling in Gerven et al. (2009) and Cseke and Heskes (2011).

Although the resulting message-passing algorithm is not exact, similar approximate message-passing algorithms have been successfully used in various models (e.g., Murphy et al., 1999; Minka, 2001; Heskes and Zoeter, 2002). These have been derived from different perspectives for various tasks such as the cavity method in statistical physics (Opper and Winther, 2000), assumed density filtering-based factor-graph inference (Minka, 2001, 2005) or expectation constrained approximate inference (Heskes and Zoeter, 2002; Heskes et al., 2005; Opper and Winther, 2005). In latent Gaussian models with log-concave likelihoods (such as the model considered here), the fixed point iteration over the messages typically exhibits fast convergence and provides good quality approximations (e.g., Minka, 2001; Kuss and Rasmussen, 2005; Seeger, 2008).

3.1.2 Exploiting sparsity

The approximate messages introduced above make inference tractable since all messages (α_t, β_t, $\lambda^0_{t+1,j}$, $\lambda^l_{t+1,j}$) and the relevant approximate marginals $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ become Gaussian. However, in the case of large state spaces, the $O(n^3)$ computational and $O(n^2)$ storage costs resulting from the computation of the temporal messages α_t and β_t can become prohibitive. To lessen these costs, and thus render inference scalable, we propose further approximations made possible by exploiting the structural sparsity of $\boldsymbol{A}$ and $\boldsymbol{Q}$.

From (23) we can see that, by restricting the precision matrix of the temporal messages α_t and β_{t+1} to be sparse, one can keep $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$'s precision matrix sparse. This allows us to use fast sparse linear algebraic methods, such as the sparse Cholesky factorisation and partial matrix inversions, to compute the required moments. To introduce our proposed approximations which result in temporal messages with sparse precision structures, we proceed as follows.

Let $\boldsymbol{f}(\boldsymbol{x}) = (x_1, \ldots, x_n, \{-x_i x_j/2\}_{i\sim j})$ denote the sufficient statistic of a Gaussian Markov random field where $i \sim j$ follows the connectivity of a graph with n vertices, $G(\boldsymbol{f})$, to

be specified later. Let Project$[q(\boldsymbol{x}); \mathcal{N}_{\boldsymbol{f}}]$ denote the Kullback-Leibler moment matching projection to the Gaussian family with precision structure defined by $G(\boldsymbol{f})$. We make use of the form of (21) and (22) to define the approximate message updates

$$\alpha_{t+1}(\boldsymbol{x}_{t+1})^{new}\beta_{t+1}(\boldsymbol{x}_{t+1}) \propto \text{Project}\big[\int d\boldsymbol{x}_t\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}); \mathcal{N}_{\boldsymbol{f}}\big], \tag{25}$$

$$\beta_t(\boldsymbol{x}_t)^{new}\alpha_t(\boldsymbol{x}_t) \propto \text{Project}\big[\int d\boldsymbol{x}_{t+1}\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}); \mathcal{N}_{\boldsymbol{f}}\big]. \tag{26}$$

These projections ensure that the forward and backward messages α_t and β_t have a sparse precision structure, defined by $G(\boldsymbol{f})$, at all times. Similarly to the approach presented in the previous section, this new approximate message-passing algorithm (i.e., equations (19), (20),(25) and (26)) is computed iteratively until a fixed point is reached. As in Section 3.1.1, once convergence is achieved, the above definition of the messages results in the weak consistency conditions

$$\text{Project}\big[\int d\boldsymbol{x}_{t-1}\, \tilde{q}_{t-1,t}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t); \mathcal{N}_{\boldsymbol{f}}\big] = \text{Project}\big[\int d\boldsymbol{x}_{t+1}\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}); \mathcal{N}_{\boldsymbol{f}}\big].$$

See Supplementary Material for details.

In the following we detail the computational issues related to the newly introduced approximate message-passing algorithm. Specifically, we show that (i) fast linear algebraic methods can be applied to exploit sparsity and (ii) when and under which conditions on $G(\boldsymbol{f})$ we can do the projection Project$\big[\cdot; \mathcal{N}_{\boldsymbol{f}}\big]$ efficiently.

A. Efficient methods for moment computations.

In Section 3.1.1 we have already shown that we carry out (19) by computing the first and second moments of $\tilde{q}_{t,j}$ using univariate numerical quadrature. The update (20) is performed by computing the univariate canonical parameters corresponding to the marginal $\tilde{q}_{t,t+1}(x^j_{t+1})$, while updates in (25) and (26) are performed by computing the canonical parameters resulting from the projections to Gaussians with restricted precision structure. In the following we show how the computation of these canonical parameters can be carried out using efficient moment computations through sparse matrix inversion and log-determinant optimisation.

The computation of the marginal $\int d\boldsymbol{x}_t d\boldsymbol{x}^{\setminus j}_{t+1}\, \tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ in (20), for all j, reduces

to the computation of the marginal means and variances of $\boldsymbol{x}_{t+1}$ in $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$, which can be computationally expensive. The crucial idea that leads to significant computational savings is that we perform the computations on the joint $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{+1})$ that now has a sparse precision structure. Let $(\boldsymbol{h}_{\alpha_t}, \boldsymbol{Q}_{\alpha_t})$ and $(\boldsymbol{h}_{\beta_{t+1}}, \boldsymbol{Q}_{\beta_{t+1}})$ denote the canonical parameters of the messages α_t and β_{t+1}, respectively. Further, concatenate the parameters of $\lambda^0_{t+1,j}$ into the representation $(\boldsymbol{h}_{\lambda^0_{t+1,\cdot}}, \boldsymbol{Q}_{\lambda^0_{t+1,\cdot}})$ where, $\boldsymbol{Q}_{\lambda^0_{t+1,\cdot}}$ is diagonal. Recall the definition of $\Psi_{t,t+1}$ from (11) and that of $\tilde{q}_{t,t+1}$ in (23). The linear parameter $\boldsymbol{h}_{t,t+1}$ and the precision matrix $\boldsymbol{Q}_{t,t+1}$ of $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ can be written as $\boldsymbol{h}^T_{t,t+1} = [\boldsymbol{h}^T_{\alpha_t}, \boldsymbol{h}^T_{t+1} + \boldsymbol{h}^T_{\beta_{t+1}} + \boldsymbol{h}^T_{\lambda^0_{t+1,\cdot}}]$ and

$$\boldsymbol{Q}_{t,t+1} = \begin{bmatrix} \left\langle \boldsymbol{A}^T \boldsymbol{Q} \boldsymbol{A} \right\rangle_{q_{AZ}, q_Q} + \boldsymbol{Q}_{\alpha_t} & -\left\langle \boldsymbol{A} \right\rangle^T_{q_{AZ}} \left\langle \boldsymbol{Q} \right\rangle_{q_Q} \\ -\left\langle \boldsymbol{Q} \right\rangle_{q_Q} \left\langle \boldsymbol{A} \right\rangle_{q_{AZ}} & \left\langle \boldsymbol{Q} \right\rangle_{q_Q} + \boldsymbol{Q}_{\beta_{t+1}} + \boldsymbol{Q}_{\lambda^0_{t+1,\cdot}} \end{bmatrix}. \tag{27}$$

To compute the required moments of $\boldsymbol{x}_{t+1}$, we (i) solve the system $[\boldsymbol{Q}_{t,t+1}]^{-1}[\boldsymbol{h}_{t,t+1}]$ and (ii) compute the diagonal of $[\boldsymbol{Q}_{t,t+1}]^{-1}$. We do this by a sparse Cholesky factorisation of a fill-in reduction reordering of $\boldsymbol{Q}_{t,t+1}$ (Davis, 2006) followed by (i) solving triangular sparse linear systems and (ii) doing a partial inversion by solving the Takahashi equations (Takahashi et al., 1973). The Takahashi equations compute all the covariance elements that correspond to non-zeros in the Cholesky factor, and thus to all non-zeros in $\boldsymbol{Q}_{t,t+1}$. This property is pivotal to rendering the Project$[\,\cdot\,; \mathcal{N}_{\boldsymbol{f}}]$ step computationally efficient.

In the following, we show how to derive the canonical parameters for α_{t+1}; a similar procedure is used to derive those for β_t. From (25), it follows that we have to project $\tilde{q}_{t,t+1}(\boldsymbol{x}_{t+1})$ into the Gaussian family $\mathcal{N}_{\boldsymbol{f}}$. Let $\boldsymbol{m}_{t,t+1} = \boldsymbol{Q}^{-1}_{t,t+1}\boldsymbol{h}_{t,t+1}$ and $\boldsymbol{V}_{t,t+1} = \boldsymbol{Q}^{-1}_{t,t+1}$. Further, let $\boldsymbol{m}^{[t+1]}_{t,t+1}$ and $\boldsymbol{V}^{[t+1]}_{t,t+1}$ denote the marginal mean and variance of $\boldsymbol{x}_{t+1}$. Then, Project$[\tilde{q}_{t,t+1}(\boldsymbol{x}_{t+1}); \mathcal{N}_{\boldsymbol{f}}]$ reduces to finding the matrix $\boldsymbol{Q}_{\alpha_{t+1}}$ which solves

$$\begin{aligned} \underset{\boldsymbol{Q}_{\alpha_{t+1}}}{\text{minimise}} \quad & \operatorname{tr}\left(\boldsymbol{V}^{[t+1]}_{t,t+1}\boldsymbol{Q}_{\alpha_{t+1}}\right) - \log\det \boldsymbol{Q}_{\alpha_{t+1}} \\ \text{s.t.} \quad & [\boldsymbol{Q}_{\alpha_{t+1}}]_{ij} = 0, \text{ for all } (i,j) \notin G(\boldsymbol{f}), \end{aligned} \tag{28}$$

and $\boldsymbol{h}_{\alpha_{t+1}} = \boldsymbol{Q}^{-1}_{\alpha_{t+1}}\boldsymbol{m}^{[t+1]}_{t,t+1}$.

The optimisation (28) can be solved by gradient-based methods or the Newton method

and the calculations are computationally expensive. However, when the graph $G(\boldsymbol{f})$ is *chordal* (e.g., Lauritzen, 1996), optimality conditions lead to equations that can be solved exactly (without expensive optimisation) by using only the values $[\boldsymbol{V}_{t,t+1}^{[t+1]}]_{ij}$ with $(i,j) \in G(\boldsymbol{f})$ (Dahl et al., 2008). Recall that the partial matrix inversion of $\boldsymbol{Q}_{t,t+1}$ always computes these values, and hence no further covariance computations are needed. For this reason, in this paper we especially consider restricting temporal messages to have chordal precision structure (that is, we set $G(\boldsymbol{f})$ to be chordal). The algorithm for computing Project$[\cdot; \mathcal{N}_{\boldsymbol{f}}]$ for chordal graphs follows from Dahl et al. (2008) and is presented in the Supplementary Material. It has a complexity that scales approximately cubically with the largest clique size in $G(\boldsymbol{f})$, which is generally much less than n.

B. Choosing chordal structures for spatial applications.

In principle, any chordal graph structure can be used, however, since in this work we are concerned with spatial applications, it is natural to motivate the choice of $G(\boldsymbol{f})$ based on the neighbourhood graph of the spatial lattice. In general, the neighbourhood graph corresponding to a spatial lattice is *not* chordal. Therefore, in order to take advantage of the efficient optimisation resulting from the use of chordal structures, we need to include extra edges in the neighbourhood graph such that the resulting $G(\boldsymbol{f})$ is chordal. Notice that by the use of chordal completions we are using a *larger* family $\boldsymbol{f}$ than the neighbourhood structure, so that no additional approximation error is introduced in this way.

It is well known that sparse Cholesky factorisations create chordal matrices that contain the original sparse matrix structure (Davis, 2006). For this reason, we propose to construct chordal graph structures by carrying out (symbolic) sparse Cholesky factorisations of the adjacency matrix given by the spatial lattice. Our choice of chordal graphs is motivated by computational arguments: we aim to construct chordal structures that include the neighbourhood graph and are maximally sparse, so that the precision matrix in (27) is as sparse as possible. There is a substantial literature on maximising the sparsity of the sparse Cholesky factors by row-column permutations, see Davis (2006) and references therein. In this paper we use (i) the approximate minimum degree permutation (Amestoy et al., 1996), denoted by `amd`, (ii) the symmetric reverse Cuthill-McKee permutation (Cuthill and McKee, 1969), denoted by `rcm`, and (iii) the nested dissection permutation (Brainman and Toledo, 2002),

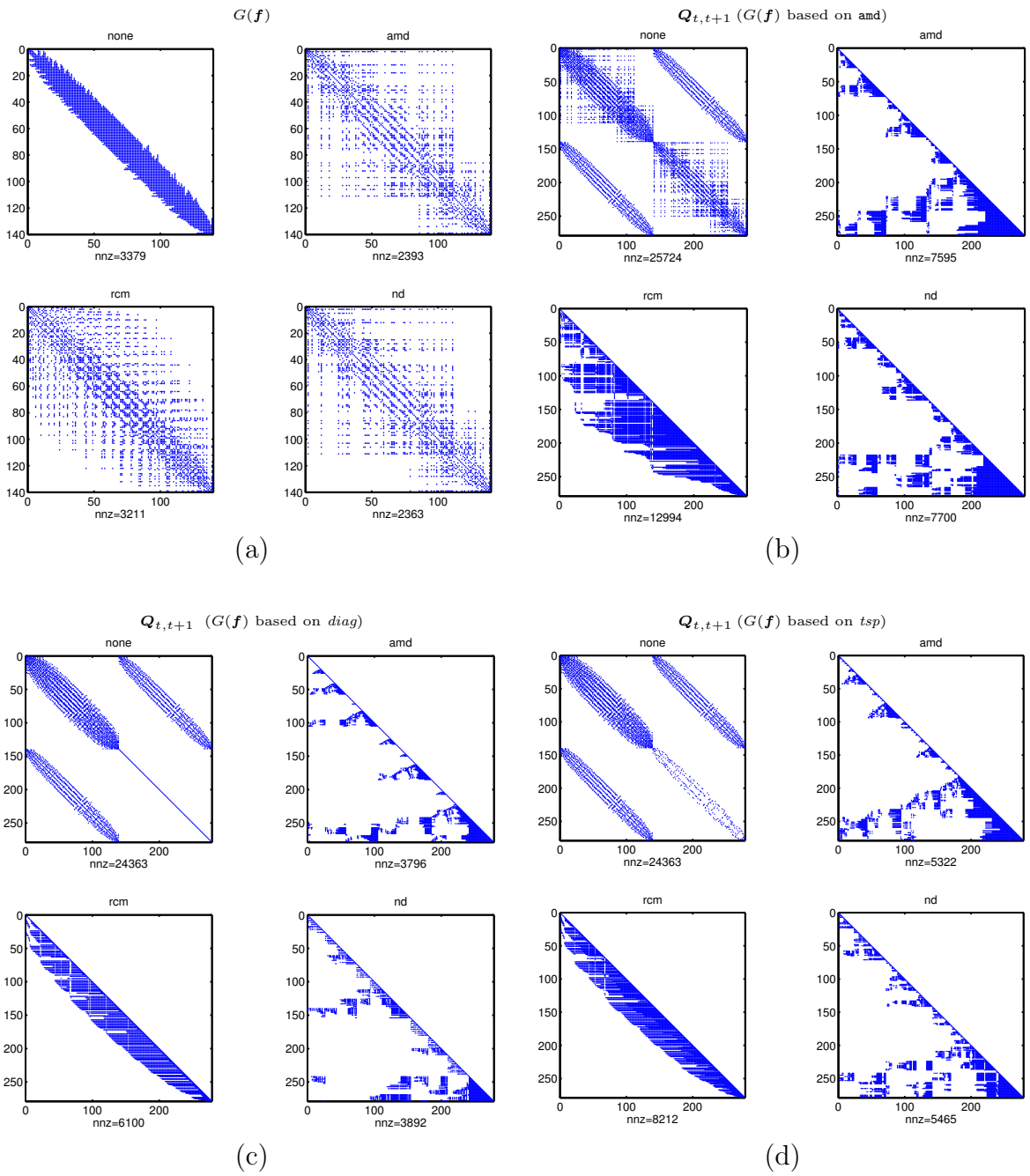

Figure 3: Illustration of the sparsity structures of the graph $G(\boldsymbol{f})$ and the matrix $\boldsymbol{Q}_{t,t+1}$ on the lattice model illustrated in Figure 1. The panels in (a) show the chordal completions of the sparsity structure of the lattice obtained by symbolic Cholesky factorisations using "fill-in" reducing permutations. Panels (b), (c) and (d) show the structure of $\boldsymbol{Q}_{t,t+1}$ for a choice of $G(\boldsymbol{f})$ as well as the structure of its Cholesky factors for various reordering permutations. For more details see Sections 3.1.2 and 3.1.3.

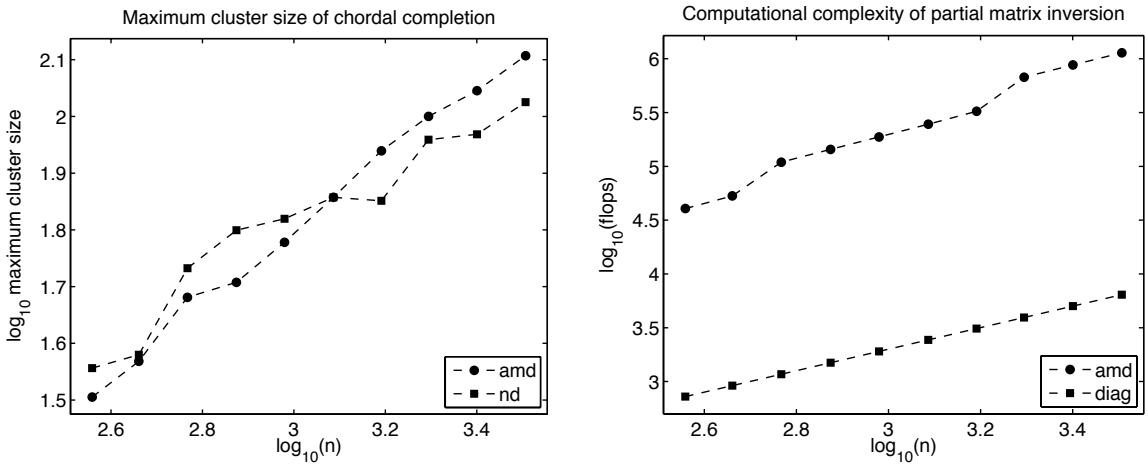

Figure 4: Computational complexities of sparse matrix operations as a function of state dimension. The left panel shows the maximum clique size of the chordal graphs generated from a triangular mesh by using `amd` and `nd` permutations. The right panel shows the empirical computational complexity estimates for the partial matrix inversion of $\boldsymbol{Q}_{t,t+1}$.

denoted by `nd`.

The panels of Figure 3 show the sparsity structures of $G(\boldsymbol{f})$, $\boldsymbol{Q}_{t,t+1}$ and the corresponding Cholesky factor for various choices of row-column permutations for a triangular lattice with $n = 140$. The matrix $\boldsymbol{A}$ corresponds to a lattice structure as illustrated in Figure 1 and $\boldsymbol{Q}$ is diagonal. The group of panels (a) show the chordal graphs $G(\boldsymbol{f})$ generated from the lattice using the various permutations. The group of panels (b) show the structure of the $\boldsymbol{Q}_{t,t+1}$s for which $G(\boldsymbol{f})$ is the chordal completion of the neighbourhood graph following an `amd` permutation. The top-left panel of this group shows $\boldsymbol{Q}_{t,t+1}$, while the other panels in this group show its corresponding Cholesky factors following the `amd`, `rcm`, and `nd` permutations. The panels (c) and (d) consider different graphs $G(\boldsymbol{f})$, *diag* and *tsp*, where *diag* corresponds to a diagonal structure and *tsp* corresponds to a spanning tree of the neighbourhood graph. These graph structures are discussed further in Section 3.1.3.

The panels of Figure 4 show the maximum clique size of the chordal graph generated from a triangular mesh and the empirical computational complexity estimates for the partial matrix inversion of $\boldsymbol{Q}_{t,t+1}$. We generated triangular meshes of various size on a circular domain and used the `amd` and `nd` permutations to complete them to chordal graphs. The left panel shows that the maximum clique size scales approximately as $O(n^{\gamma})$ with $\gamma \sim 1/2$

for both permutations, implying that the time required to solve (28) scales approximately as $O(n^{3/2})$ or $O(n^2)$ at most (recall the cubic complexity of the projection step w.r.t. maximum clique size). The right panel shows an estimate of the number of flops required for solving the Takahashi equations given the sparse Cholesky factor of $\boldsymbol{Q}_{t,t+1}$. The *chordal* component in this plot was generated by using `amd` and `nd` permutation-based chordal completions for $G(\boldsymbol{f})$ and an `amd` permutation for the sparse Cholesky factorisation on $\boldsymbol{Q}_{t,t+1}$. The flop-count estimates show that the computational complexity of the *chordal* methods scales approximately as $O(n^{3/2})$ or $O(n^2)$ at most, and that the computational complexity of the *diag* is significantly lower. These results are in line with the complexity estimates in George (1973) and Rue and Held (2005).

3.1.3 Approximations and message passing schedules

The choice of $G(\boldsymbol{f})$, and the scheduling of the updates in the fixed point iteration of equations (19), (20),(25) and (26), govern the accuracy and the computational complexity of the inference algorithm. In the following we detail our choices and relate some of these to the current literature.

The computational complexity of the approximations is dominated by the partial matrix inversion of $\boldsymbol{Q}_{t,t+1}$, that is in turn directly determined by the structure of $G(\boldsymbol{f})$. We thus consider three classes of $G(\boldsymbol{f})$: (i) *full*, where $G(\boldsymbol{f})$ is fully connected, which corresponds to an approximate inference approach of propagating full Gaussian messages (Ypma and Heskes, 2005), (ii) *chordal*, where $G(\boldsymbol{f})$ is a chordal graph, corresponding to messages having precision matrices with restricted sparsity structure (the spanning tree structure *tsp* is a special case), and (iii) *diag*, where $G(\boldsymbol{f})$ is a disconnected graph, corresponding to factorised temporal messages. With *diag*, only marginal means and variances are propagated, see Murphy and Weiss (2001) for an algorithmically similar approach for models with discrete variables.

When $G(\boldsymbol{f})$ is fully connected, (*full* temporal messages), $\boldsymbol{Q}_{t,t+1}$ has dense diagonal blocks, and hence the computational complexity scales as $O(n^3T)$. When $G(\boldsymbol{f})$ is chordal, this complexity is reduced; empirical studies showed that the resulting complexity is around $O(n^2T)$. In case of *tsp* and *diag*, we expect a complexity of around $O(n^{3/2}T)$ (Rue et al., 2009, Section 2.1).

In terms of message scheduling we differentiate between the following choices: (i) *Static*. Here the forward backward updates (25) and (26) for all time steps are iterated until convergence and then the (19) and (20) updates are performed. In this scheduling, the forward-backward (25)–(26) iteration corresponds to an approximate partial matrix inversion while the updates (19) and (20) correspond to the EP steps in a blocked model (Minka, 2001; Cseke and Heskes, 2011). (ii) *Sequential*. Here the (19) and (20) updates are iterated until convergence at each time step, followed by a (25)-(26) update; these steps are performed in a forward backward fashion. In this scheduling, the (19) and (20) steps correspond to an EP algorithm approximating the Kalman update steps at each time point. (iii) *Dynamic*. Here, in order to minimise the number of expensive partial matrix inversions, we use a greedy scheduling strategy. With this strategy, at every step we select the message that has the largest (last) update (in terms of canonical parameters), and update both the receiver and the source of this message, be it either $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ or $\tilde{q}_{t+1}(x^j_{t+1})$. For example, suppose that α_t has the largest recent update. Then, we update $\tilde{q}_{t,t+1}$ and its outgoing messages β_t and α_{t+1} followed by an update of $\tilde{q}_{t-1,t}$ and its outgoing messages, thus providing a new update also for α_t. Simulation studies showing the computation savings of the greedy algorithm are given in Section 4, while the scheduling options are discussed further un the Supplementary Material.

By constructing longer scheduling queues (ranking the change in messages, instead of choosing the maximal change), one can distribute the computation in the dynamic scheduling scheme to several processing units and achieve a further reduction of computing time. We distribute the computation by selecting and scheduling locally independent receiver-source pair updates to different computational cores. We adapt the greedy approach by keeping a ranking of the message updates and then in each cycle proceeding from the top of the ranking in selecting pairs of approximate marginals $\tilde{q}_{t,t+1}$ to update. We select as many disjunct pairs as available computational units, proceed with the computation and repeat these cycles until convergence. Currently, we can achieve about a five fold reduction in computation time using eight cores in our Matlab[2] implementation. We believe that this can be further improved by fine-tuning our ranking and scheduling scheme and by making

[2]http://www.mathworks.com

better use of Matlab's distributed computing toolbox.

3.2 The models q_{AZ} and q_Q

In this paper we are interested in learning dynamics for models with diagonal $\boldsymbol{Q}$, therefore, we only present the inference for such models. For diagonal $\boldsymbol{Q}$, the distribution q_{AZ} factorises over the rows of $\boldsymbol{A}$ (denoted $\boldsymbol{a}_i$) and the rows of $\boldsymbol{Z}$ (denoted $\boldsymbol{z}_i$), that is, $q_{AZ}(\boldsymbol{A}, \boldsymbol{Z}) = \prod_i q^i_{AZ}(\boldsymbol{a}_i, \boldsymbol{z}_i)$ (see (9) in Section 2) with each q^i_{AZ} being a multivariate conditional Gaussian. Due to the sparse lattice structure, q^i_{AZ} simplifies to a distribution with low, say $6 - 10$, dimensions, and therefore the marginal moments can be computed exactly in reasonable time. The first and second moments of $\boldsymbol{a}_i$ needed to update q_X and q_Q are computed from $q^i_{AZ}(\boldsymbol{a}_i)$, whereas $q^i_{AZ}(z_{ij})$ can be used as a measure of the relevance of a_{ij}. In cases when $\boldsymbol{Q}$ is not diagonal, the model for q_{AZ} has a high dimensionality and exact inference is intractable. In such cases we can resort once again to approximate message passing. The form of this non-factorising model and the corresponding message-passing algorithm are outlined in the Supplementary Material.

In general we choose conjugate priors for $\boldsymbol{Q}$ or we keep $\boldsymbol{Q}$ fixed. When $\boldsymbol{Q}$ is diagonal, from (10) we can see that q_Q factorises as $q_Q(\boldsymbol{Q}) = \prod_i q_Q(q_{ii})$ and that due to conjugacy, the marginals $q_Q(q_{ii})$ are Gamma distributed.

4 Experiments

In this section we assess the speed and accuracy of the inference methods we introduced and show the potential use of this approach in the WikiLeaks Afghan War Diary data studied in Zammit-Mangion et al. (2012a). The algorithms have been coded in Matlab and for partial matrix inversion we use the implementation of Gerven et al. (2011–2015), which is implemented in the C programming language.

4.1 Accuracy of state inference in 1D models

In this section we assess the accuracy of the state inference methods in 1D Gaussian and Poisson models by using fixed parameters. Although, these models do not fall in the class of models we discussed so far, they are well suited for an empirical assessment of the quality of approximations we introduce. We use the Gaussian model to assess the accuracy of restricted (temporal) message passing inference schemes *diag* and *chordal* by comparing them to the exact *full*. By replacing the Gaussian likelihood with a Poisson likelihood we assess the loss of accuracy due to non-Gaussian likelihoods. Note that the Poisson likelihood ay time t and location j is formally identical to $\tilde{\psi}_{t,j}$.

4.1.1 Models and accuracy measures

In both the Gaussian and the Poisson case, we consider a diffusion model on a 1D grid with $n-1$ grid intervals (n state space variables, $\boldsymbol{x}_t \in \mathbb{R}^n$) and T time points. We define $\boldsymbol{A}$ as a symmetric banded matrix with various bandwidths n_{neighb} and $a_{ij} = (1-\epsilon_{\boldsymbol{A}})/(1+2n_{\text{neighb}})$ with values for nodes close to the boundaries rescaled accordingly to obtain a constant row-sum $1-\epsilon_{\boldsymbol{A}}$. We define the system noise inverse covariance $\boldsymbol{Q}$ as a linear combination of a first order intrinsic field's precision matrix and a unit diagonal matrix. We introduce a parameter s to control the correlation decay in $\boldsymbol{Q}^{-1}$ and normalise $\boldsymbol{Q}^{-1}$ to obtain a chosen variance value v_x. Formally, $\boldsymbol{Q}$ is defined as

$$\boldsymbol{Q}(v_x, s) = v_x^{-1}\sqrt{\text{diag}(\boldsymbol{R}(s)^{-1})}\boldsymbol{R}(s)\sqrt{\text{diag}(\boldsymbol{R}(s)^{-1})}, \quad \text{where} \quad \boldsymbol{R}(s) = \boldsymbol{I} + 10^s\boldsymbol{R}_1,$$

and $\boldsymbol{R}_1$ denotes the tri-diagonal precision matrix corresponding to the quadratic form $\sum_i (x_{i+1}-x_i)^2$. The left panel of Figure 5 shows how s influences the correlation decay.

The observation models are defined as follows. In the Gaussian case we assume that we observe the field with added Gaussian observation noise with variance v_{obs}, and that the field is only partially observed: we sample the locations of the observations from the $n \times T$ space-time grid uniformly with probability p_{obs}. In the Poisson case the observations are Poisson random numbers with mean $\exp\{x_t^i\}$ and we sample at all locations of the space-time grid.

As mentioned above, in the Gaussian case we focus on the accuracy of *diag* and *chordal* and compare them to the exact *full* to assess the loss in accuracy due to the restricted

temporal messages. Since the objective of the state inference method is to approximate the two-time-slice marginals $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$, the accuracy measure we choose is the symmetric KL divergence w.r.t. the exact two time slice marginals $p_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ computed by *full*. Therefore, we define the accuracy measure

$$S(\{p_{t,t+1}\}_t, \{\tilde{q}_{t,t+1}\}_t) = \frac{1}{2(T-1)} \sum_{t=1}^{T-1} \Big\{ \mathrm{D}[\tilde{q}_{t,t+1} || \, p_{t,t+1}] + \mathrm{D}[p_{t,t+1} || \, \tilde{q}_{t,t+1}] \Big\}. \tag{29}$$

In the Poisson case the quality of the approximation is affected both by the restricted temporal messages and the non-Gaussian nature of the problem. Our aim here is to assess the joint effect of both sources of inaccuracy.

In the Gaussian case the KL measure seems to be a reasonable choice to assess the distributional accuracy. However, in the Poisson case the exact marginals are not available, therefore, we opt for the quantile-quantile (Q-Q) summaries as a measure of accuracy. Since the accuracy measure should reflect the local nature (we only approximate marginals) of the algorithm, we use the local Gaussian approximation $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ to compute the normalised residuals $\hat{\boldsymbol{\epsilon}}_{t,t+1}$ w.r.t. the state values $\tilde{\boldsymbol{x}}_{t,t+1}$ used in the data generation. Formally, we define the residuals as

$$\hat{\boldsymbol{\epsilon}}_{t,t+1} = \boldsymbol{L}_{t,t+1}^T (\tilde{\boldsymbol{x}}_{t,t+1} - \boldsymbol{m}_{t,t+1}),$$

where $\boldsymbol{m}_{t,t+1}$ and $\boldsymbol{Q}_{t,t+1} = \boldsymbol{L}_{t,t+1}\boldsymbol{L}_{t,t+1}^T$ are the mean and precision corresponding to $\tilde{q}_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$. We then use the quantile values to assess how well the standard normal distribution fits $\hat{\epsilon}_{t,t+1}^j$ for all j and t. Specifically, we use the mean absolute deviation from the standard normal quantiles as a measure of accuracy.

4.1.2 Simulation results

In the Gaussian case we considered models with $n = 64$, $T = 100$ and $n_{\text{neighb}} \in \{1, 2, 4, 8\}$. We chose a system noise variance $v_x = (0.5)^2$, an observation noise variance $v_{\text{obs}} = (0.25)^2$ and we set $p_{\text{obs}} = 0.75$. We chose $s \in \{-1, 0, 1\}$ leading to the correlation functions shown on the left panel of Figure 5. We simulated the models starting from a sample from the stationary

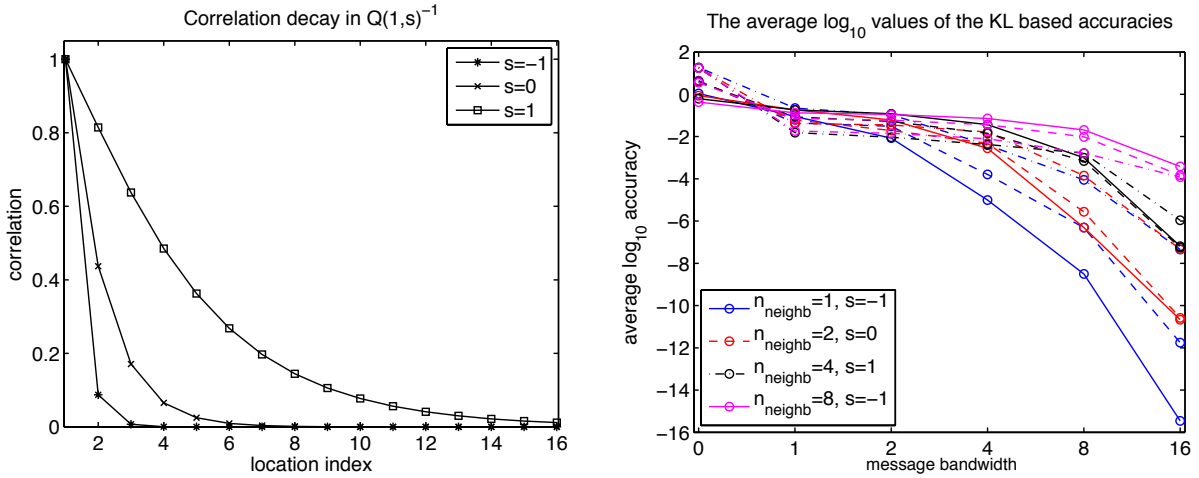

Figure 5: Quantifying accuracy for the 1D Gaussian model. The left panel shows the correlation decay in $\boldsymbol{Q}(1,s)^{-1}$ w.r.t. the location index i for $s \in \{-1,0,1\}$ while the right panel shows the average over $n_{\text{exp}} = 25$ runs of the $\log_{10}$ of KL accuracy score (29) for various choices of n_{neighb} (colour) and s (line style) as a function of the message bandwidth n_{msg}. Lower values are indicative of better performance.

distribution. All these parameter choices together with $\epsilon_{\boldsymbol{A}} = 0.025$ lead to simulated samples with rich variations in the latent field $\{\boldsymbol{x}_t\}_t$ in the given time window. We simulated $n_{\text{exp}} = 25$ runs for each model and we computed the KL based score in (29) for *diag* ($n_{\text{msg}} = 0$) and for the *chordal* models corresponding to messages with bandwidths $n_{\text{msg}} \in \{1,2,4,8,16\}$ in their precision matrix, thus varying the accuracy of the approximation. Note that $n_{\text{msg}} = 63$ corresponds to the exact *full* method and due to the univariate nature of the problem, all chordal methods (`amd`, `nd` and `rcm`) lead to the same banded structure in the temporal messages' precision matrix. For each inference method, the inference scheme was run until the change in the maximum absolute value in the message parameters became smaller than 10^{-8}. The right panel of Figure 5 shows the average log KL accuracies w.r.t. the message bandwidth n_{msg} for various choices of n_{neighb} and s. The accuracy plots show that for the *diag* method ($n_{\text{msg}} = 0$) the accuracy is dominated by s and that the chordal methods lead to a significant improvement in accuracy. The general pattern in the variation of the accuracy w.r.t. n_{neighb} and s is that, as expected, smaller correlation in $\boldsymbol{Q}^{-1}$ and fewer neighbours in $\boldsymbol{A}$ lead to better accuracy. For all cases the accuracy increases as the message bandwidth n_{msg} increases thus validating the usefulness of the *chordal* inference methods.

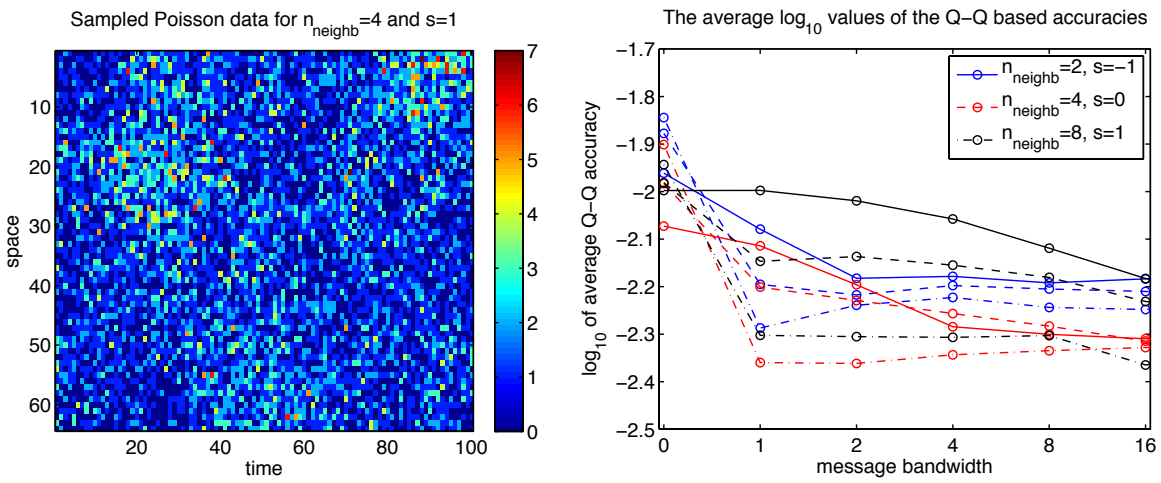

Figure 6: Quantifying accuracy for the 1D Poisson model. The left panel shows sampled Poisson data from the model while the right panel shows the logarithm of the average (over $n_{\text{exp}} = 25$ runs) mean absolute quantile deviations (50 bins) for various choices of n_{neighb} (colour) and s (line style) as a function of message bandwidth n_{msg}. Lower values are indicative of better performance.

In the Poisson case we used the same latent diffusion model, generated Poisson observations (see the left panel of Figure 6) and used the same inference schemes and stopping criteria as in the Gaussian case. For each run we constructed Q-Q curves using the residuals $\hat{\epsilon}^j_{t,t+1}$ and 50 quantile bins and then measured the accuracy of the inference by computing the mean absolute deviation of the curve from the diagonal. We then averaged all the deviations over $n_{\text{exp}} = 25$ experiments for each choice of n_{neighb}, s and message bandwidth n_{msg}. The resulting accuracies are shown in the right panel of Figure 6. The plots show that, similarly to the Gaussian case, the quality of the approximation improves as we increase the message bandwidth n_{msg} and that, typically, there is a significant increase in accuracy when moving from *diag* to *chordal* methods. As in the Gaussian case, the weaker the diffusion (smaller n_{neighb}) the more accurate the method is, however, it seems that in this case correlation in $\boldsymbol{Q}^{-1}$ leads to slightly improved accuracy—compare the performance of the *chordal* methods with the *diag* one and note the bad performance of *diag* for $s \in \{0, 1\}$.

4.2 Accuracy and structure recovery in a 2D spatial model

In this section we consider a two-dimensional spatio-temporal model where we vary the transition matrix $\boldsymbol{A}$ and a diagonal inverse covariance $\boldsymbol{Q}$ and we assess the accuracy of

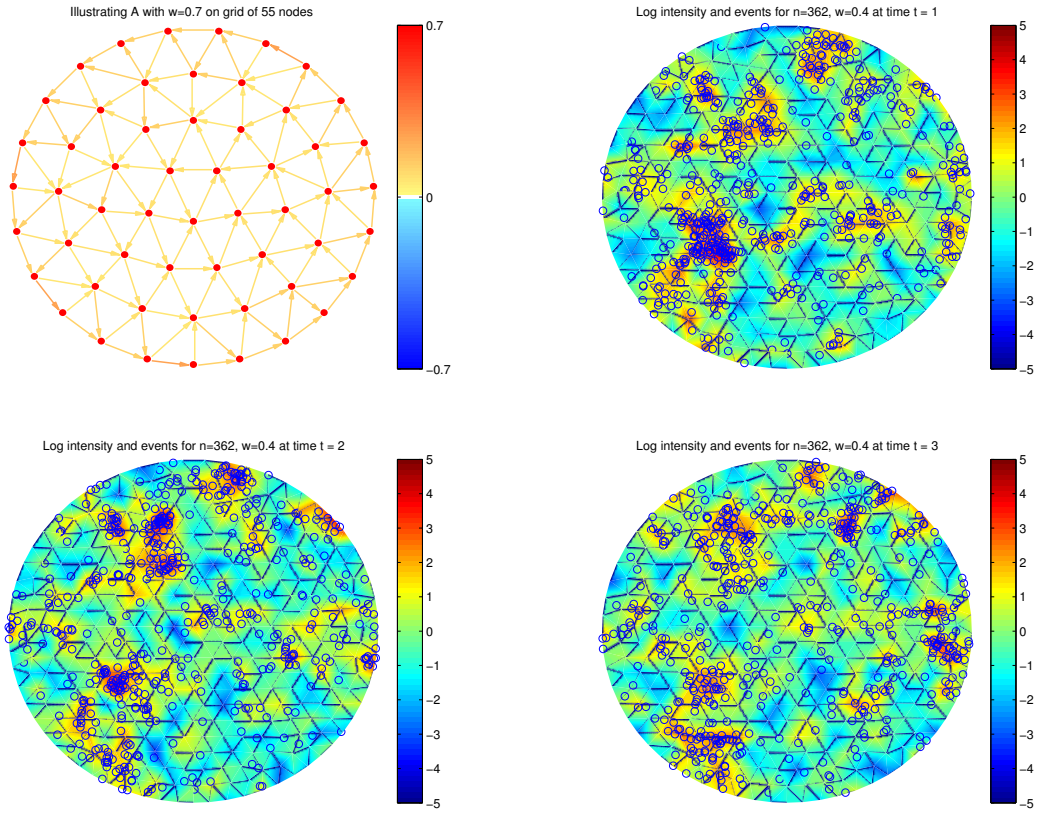

Figure 7: The top left panel illustrates the transition matrix $\boldsymbol{A}$ for a 2D model with $w = 0.7$ and a state space of size $n = 55$. The rest of the panels show the log intensity and the simulated events (open circles) corresponding to a sequence of 3 steps from a model with $n = 362, w = 0.4$ and $\sigma^2 = 1$ and a similarly structured transition matrix as in the top left panel. The resulting field, and consequently the point patterns, exhibit a rotation motion.

the state inference and the recovery rate of the network structure corresponding to $\boldsymbol{A}$. In order to obtain interesting dynamics, we define the field on a circular domain and choose the structure of the matrix $\boldsymbol{A}$ such that the model gives rise to a rotating "motion" in the Gaussian field. To achieve this, we define $\boldsymbol{A}$ as follows: we start from a symmetric structure given by a triangular lattice, that is $a_{ij} = a_{ji}$ for all $i, j = 1, \ldots, n$, and for each node i we eliminate all incoming edges (i, j) for which the corresponding vector in the lattice is not in an anti-clockwise direction w.r.t. the domain's center, see Figure 7, top-left panel. We then set the elements of the i-th row of $\boldsymbol{A}$ according to

$$a_{ii} = w, \quad \text{and} \quad a_{ij} = (1 - \epsilon_w - w)/|\mathcal{N}(i)|,$$

where $\mathcal{N}(i)$ denotes the set of neighbours of node i in this newly defined directed structure. We set ϵ_w to a small value such that the row $\boldsymbol{a}_i$ sums are lower than 1, thus resulting in a zero stationary mean value for the states. We set $\epsilon_w = 0.05$. The remaining panels of Figure 7 show samples from the field $u(s,t) = \sum_j \phi_j(s) x_t^j$ and the event data generated from it. We vary the diagonal values of $\boldsymbol{A}$ by choosing $w \in [0, 1]$ and modify the system noise $\boldsymbol{Q}^{-1} = \sigma^2 \boldsymbol{I}$ by choosing $\sigma^2 \in \{0.5, 1, 2\}$. The stationary mean and covariance of $\{\boldsymbol{x}_t\}_t$ are given by

$$\boldsymbol{m}_\infty = \boldsymbol{A}\boldsymbol{m}_\infty \quad \text{and} \quad \boldsymbol{V}_\infty = \boldsymbol{A}\boldsymbol{V}_\infty \boldsymbol{A}^T + \sigma^2 \boldsymbol{I},$$

and the mean value of the stationary intensity is

$$\langle \lambda_\infty(\boldsymbol{s}) \rangle = \exp\left\{ \boldsymbol{\phi}(\boldsymbol{s})^T \boldsymbol{m}_\infty + \frac{1}{2} \boldsymbol{\phi}(\boldsymbol{s})^T \boldsymbol{V}_\infty \boldsymbol{\phi}(\boldsymbol{s}) \right\}.$$

We simulated artificial event data by using initial samples from the stationarity distribution. The mean $\boldsymbol{m}_\infty$ is typically sufficiently close to zero to be negligible; thus the mean intensity is determined by $\boldsymbol{V}_\infty$, that is, by w and σ^2. In the first experiment we assess how the accuracy of the state inference varies in terms of w and σ^2, while in the second one we fix $\sigma^2 = 1$ and do joint inference for the parameters in $\boldsymbol{A}$ and the states in $\boldsymbol{X}$.

We assess the accuracy of the state inference on models with $n \in \{362, 1008\}$. We

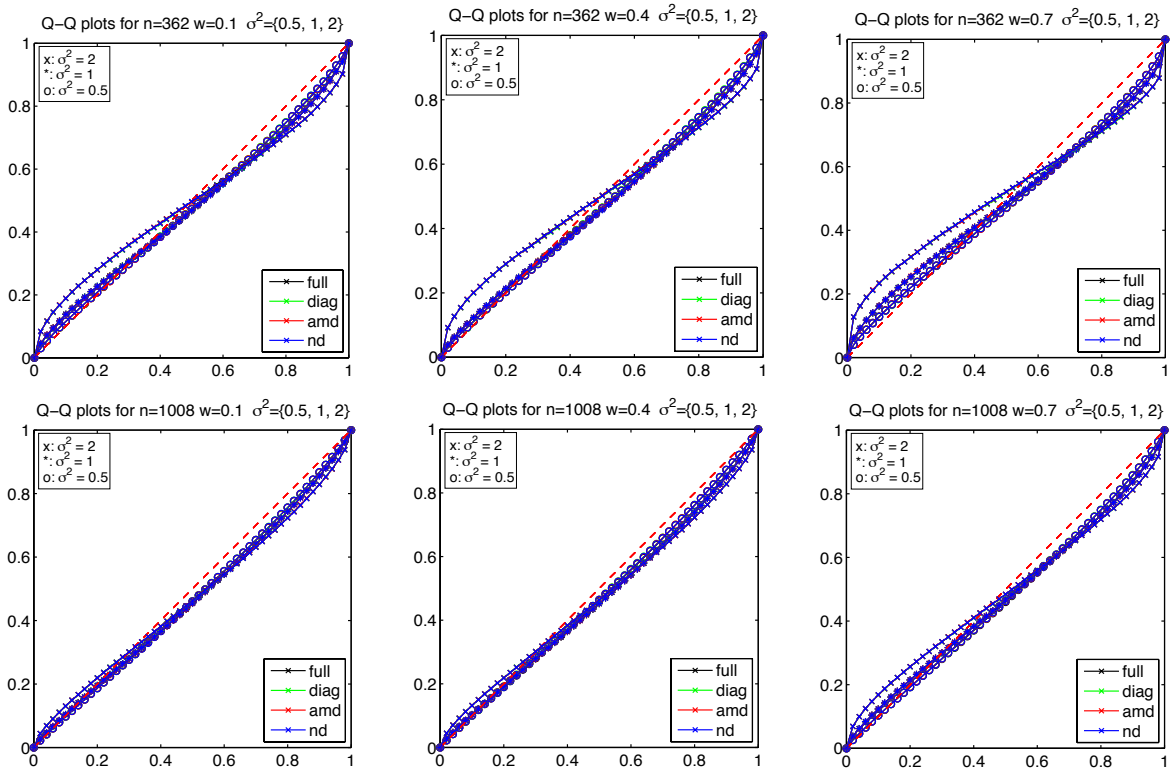

Figure 8: Accuracy of the state space approximation in a 2D spatial model. The panels show the Q-Q plots for a variety of parameter settings and methods in models with $n = 362$ (top) 1008 (bottom) and $T = 50$. The Q-Q plots were generated using the residuals $\hat{\epsilon}^j_{t,t+1}$, see Section 4.2.

generated a sequence of $T = 50$ state samples $\{\tilde{\boldsymbol{x}}_t\}_t$ starting from the stationary distributions and sampled the event data by using a standard thinning method. We then ran the state inference methods using the w and σ^2 parameters the data was generated by. The panels in Figure 8 show the Q-Q plots using the residuals $\hat{\epsilon}^j_{t,t+1}$ for various settings of w and σ^2. We can see that in this model the methods have very similar performance (the Q-Q plots overlap) and there is a decrease in performance as the values of w and σ^2 increase, as expected. The overlap of the Q-Q plots can be explained by the diagonal nature of $\boldsymbol{Q}$ (low noise correlation) and the magnitude of the Q-Q accuracies, see Figure 6 right panel. The worsening of performance due to increasing w is negligible compared to that due to increasing σ^2, which can be explained by the fact that higher system noise leads to less accurate approximations.

Recovering the structure of $\boldsymbol{A}$ is important in many spatio-temporal applications where

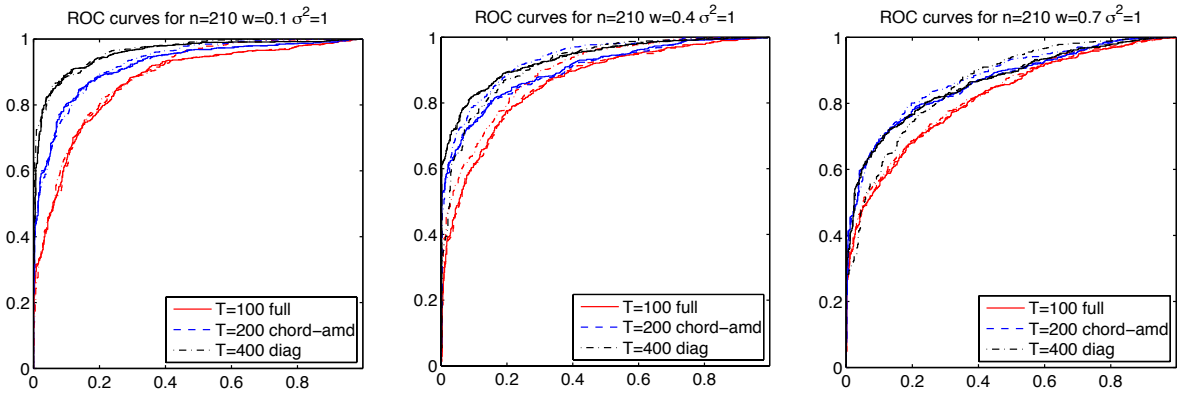

Figure 9: Structure recovery. The panels of the figure show the ROCs for various settings of w and T and for a variety of state inference methods, see Section 4.2. Colours denote the choices of T while line types denote the inference method used for q_X.

we want to infer how events spread over a certain geographic area, see for example the data in Section 4.4. To test the quality of edge recovery, we generated data by using the above described construction of $\boldsymbol{A}$ and a fixed $\sigma^2 = 1$, and inferred the (approximate) posterior distribution of $\boldsymbol{X}, \boldsymbol{A}$ and $\boldsymbol{Z}$. In Section 3 we mentioned that the (approximate) posterior distribution of the variables z_{ij} can be used to quantify the relevance of an edge (i, j). Here we use $P(z_{ij} = 1)$ as classification scores to assess whether the correct edges have been eliminated from the prior lattice structure, and we construct receiver-operator curves (ROC) to assess the quality of the structure inference. These curves show the true positive rate versus the false positive rate when varying the classification threshold between 0 and 1. The panels in Figure 9 show the ROC curves for various choices of w, T and approximation methods. We can conclude that, as expected, the quality of the recovery increases as T increases for all values of w. As expected lower values of w and thus higher diffusion speeds lead to better performance on structure recovery in a limited time window. The difference in various state inference methods is hardly noticeable in these models and is well within the expected statistical variation. This lack of difference can be explained by the accuracy results discussed above.

4.3 Running times and scalability

To determine the scalability of the algorithm as we vary n, we use the setting of Section 4.2 with $\boldsymbol{A}$ fixed to encode the neighbourhood graph. The results presented in this section are insensitive to the specific parameter values. We used a set of parameters that, at stationarity, result in around 1000 events per time frame; a typical count for large datasets.

The algorithms were tested on domains with varying mesh density, $n \in \{362, 562, 1008\}$ and computing times were recorded using Matlab's profiler. The message passing was run until every parameter in the message was not changed by more than 10^{-4} in successive iterations. To ensure a fair comparison, all test results given here are with computations restricted to a single processor core.[3]

The computing times for the sequential scheduling scheme are plotted in the left panel of Figure 10. We segmented the computing times to correspond to the three main operations: (i) *temp-messages* stands for the Project$[\cdot; \mathcal{N}_{\boldsymbol{f}}]$ operation (28), (ii) *overhead* accounts for initialisations, message updating and convergence monitoring, (iii) *local lin-alg* logs the time for linear algebraic operations (dominated by the Cholesky factorisation and partial matrix inversion), and (iv) *local non-Gaussian* stands for the univariate moment computations to update $\lambda^0_{t,j}$ in (19). For clarity, we omit results for the static scheduling case which were up to an order of magnitude slower than the second worst-performing method.

The left panel in Figure 10 shows that, for small n, the *full* inference scheme is faster than the other schemes due to the fact that it is implemented more efficiently in terms of dense matrix operations (Matlab/LAPACK core routines). However, the situation changes for $n = 1008$, where we see that the *full* is slower than the best *chordal* methods and much slower than the *tsp* and *diag*. Note that the increase in total computing time is well below cubic and at most quadratic for all methods other than the *full*. It is clear from this figure that *full* will become untenable for large n (note the rate of growth w.r.t. n).

Although the scheduling itself does not affect the scalability of the algorithm, it can be seen from Figure 10, right panel, that, as expected, the greedy scheduling can greatly reduce the computing time. For instance, after the initial forward-backward, the *full* needed only a few factor updates to achieve convergence within tolerance.

[3] All algorithms were tested on an Intel Core$^{\text{TM}}$i7-2600S @ 2.80GHz personal computer with 8GB of RAM.

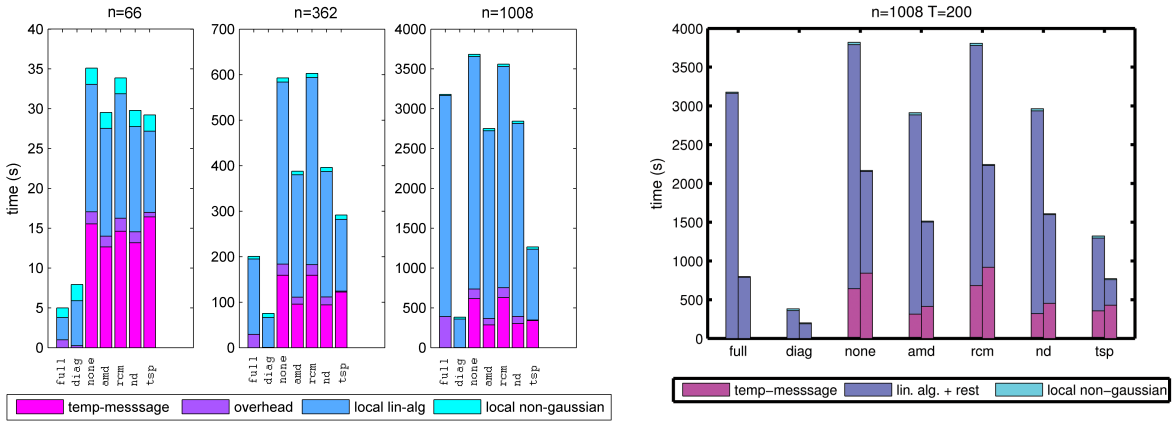

Figure 10: Left panel: Running times for various state space sizes and scheduling options. (left) Running times for the inference schemes *full*, *diag*, and chordal schemes `none`, `amd`, `rcm`, `nd` and *tsp*. Right panel: Comparison in terms of running times of the sequential (left bar) and greedy (right bar) scheduling strategies. Local operations refer to the local linear algebra whilst the temporal messages refer to the Project$[\cdot; \mathcal{N}_f]$ optimisation.

4.4 The Afghan War Diary

Spatio-temporal point-process methods have recently been shown to be a valuable tool in the study of conflict. In Zammit-Mangion et al. (2012a), a dynamic spatio-temporal model, inspired by the integro-difference equation, was used to obtain posterior estimates of conflict intensity in Afghanistan and to predict conflict levels using an iterative state-parameter update scheme on the WikiLeaks Afghan War Diary (AWD). Updates on $\boldsymbol{x}_t$ were found using an algorithm similar to the *full* described above. The spatial scales considered there were on the order of a 100 km and thus, modelling of micro-scale effects such as relocation or escalation diffusions in conflict were not possible. Conflict dynamics are known to occur at much smaller scales (Schutte and Weidmann, 2011), even at resolutions of $\approx$10 km. The goal of this section is thus primarily to show that we can perform inference at such high resolutions and, in addition, estimate the dynamics on the required spatial and temporal scales.

4.4.1 State estimation using fixed parameters

Afghanistan has an area of over 500000 km^2 and the WikiLeaks data set contains over 70000 events. The mesh we employed (using population density as a proxy for mesh density), shown in Figure 11 has the largest triangles with sides of 22km and the smallest ones with

sides of 7km. The total number of vertices amounts to $n = 9398$ in a system with $T = 313$ time points (weeks).

We constructed $\boldsymbol{A}$ using a Galerkin reduction with a mass lumping method (Bueche et al., 2000; Lindgren et al., 2011) of a diffusion equation. For illustration purposes, the diffusion constant was set to $D = 10^{-4}$ with latitude/longitude as spatial units (all of $\boldsymbol{A}$ will be estimated in the next sub-section). The matrix $\boldsymbol{Q} = 0.2 \times \boldsymbol{I}$ was taken as rough value from the full joint analysis using a low resolution model, see Zammit-Mangion et al. (2012a) for details. We carried out inference in the AWD with the *diag* algorithm, which took only a few hours on a standard PC and consumed only about 4GB of memory.

A characteristic plot showing one week of the conflict progression (first week of October 2009) is given in Figure 11. At this point, in the conflict, activity in the south in Helmand and Kandahar was reaching its peak and conflict at the Pakistani border was intensifying considerably. The insets clearly show how detailed inferences can be made.

4.4.2 Learning conflict dynamics from the AWD

In the context of conflict, the dynamic behaviour of events is usually extracted from the data by gridding the domain of interest in space and time and carrying out an empirical study on the events *per se*. For example, one could analyse the pattern of cells which contain at least one event at two consecutive time frames (Baudains et al., 2013; Schutte and Weidmann, 2011). Unfortunately, due to an explicit reliance on the use of multiple observations to obtain a reliable estimator, these methods are only able to provide global assessments of the dynamics, that is, assert whether phenomena such as escalation or relocation are present everywhere on average. Here, on the other hand, we can provide spatially-resolved maps of conflict patterns and, moreover, are able to assign a probability to the presence or absence of dynamics.

The most important phenomena considered here are containment (events repeat in the same location), escalation (events repeat and also spill over into surrounding areas), and relocation (events move from one area to the next). All of these contagion phenomena may be interpreted directly from our posterior beliefs on $\boldsymbol{A}$ and the diagonal elements of $\boldsymbol{Q}$. For example, a vertex with low values of q_{ii} and low values for the elements in its respective

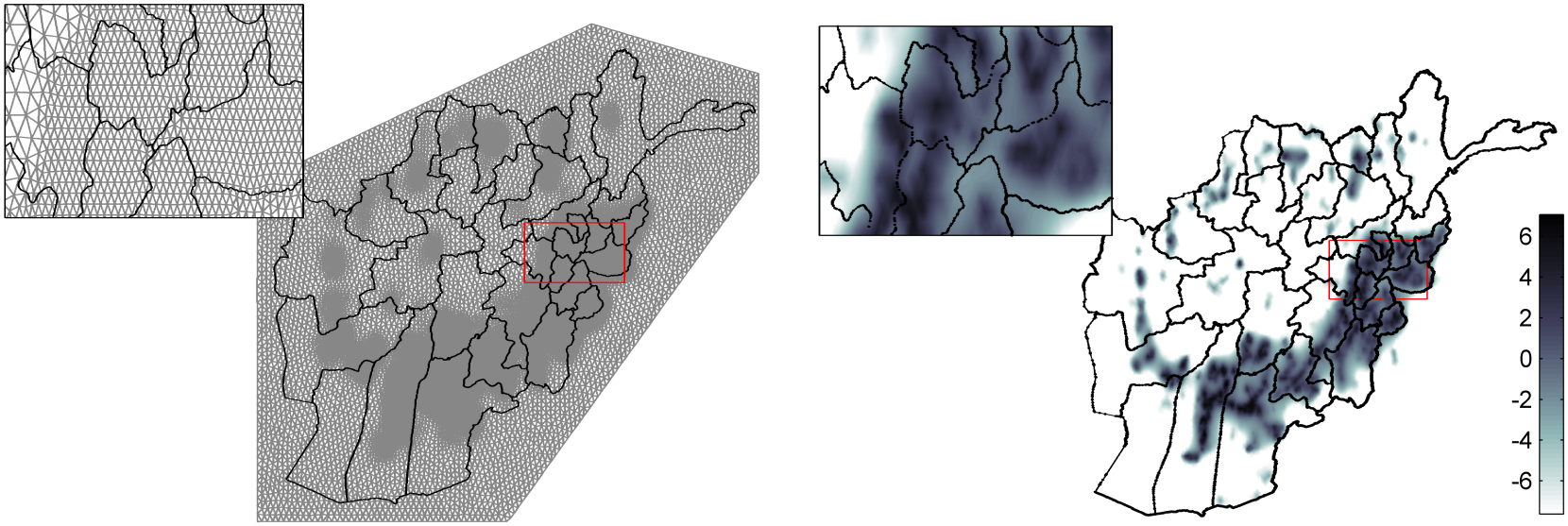

Figure 11: The mesh and one time-slice log intensity map corresponding to the AWD on the first week of October 2009.

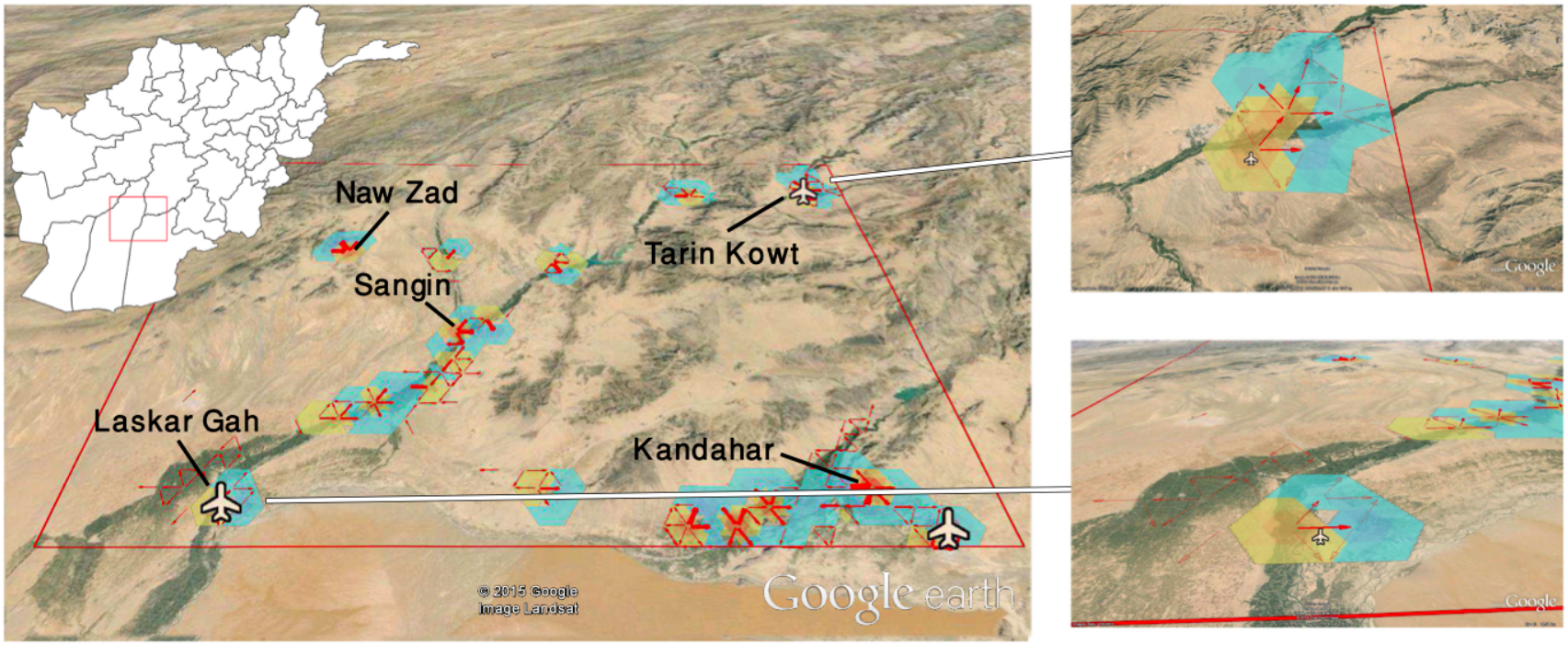

Figure 12: Left panel: The difference between the extent of sourcing (red) and sinking (blue) of conflict event intensity in part of Southern Afghanistan. The aeroplane symbols denote airport locations whilst the arrows denote the relevant edges in $\boldsymbol{A}$ with arrow thickness proportional to the element size. Right panel: Insets showing the detail at the airports near Laskar Gah and Tarin Kowt. (Google Earth is a product of Google Inc. We used the *Google EarthToolbox* (Davis, 2012) visualisation tool to construct the additional artwork.)

column in $\boldsymbol{A}$, is indicative of an area where events do occur, but that do not escalate or diffuse to surrounding areas. On the other hand, large values in $\boldsymbol{a}_i$ are indicative of an area that is susceptible to nearby conflict events whilst large values in a column of $\boldsymbol{A}$ are indicative of an area that is a large potential contributor to conflict contagion.

The last interpretation is particularly interesting, not only for retrospective analysis and prediction purposes, but also for generating insights into mechanisms which could be employed to contain contagion. One may summarise the role of a region in conflict by summing over the respective columns and rows in $\boldsymbol{A}$ (excluding the diagonal elements) in order to obtain a *source* index and a *sink* index for each vertex. The difference between these two indices can then be seen as a measure of how likely a region is to contribute to conflict in the surrounding areas and how likely conflict in a region is *due to* conflict in a neighbouring area.

As a proof of concept, we employ the full model to obtain a map of contagion for Helmand using data between May 2006 and November 2009, shown in Fig. 12. It is beyond the scope of this work to analyse the map in detail, however, three things are of note. First, although not evident from this figure, there is no direct correlation between event intensity and contagion, suggesting that the inference is able to distinguish between containment and relocation/escalation. Second, all airports in the vicinity (three in this case: Kandahar in the South East, Laskar Gah in the South West and Tarin Kowt in the North East) are highlighted as a *source* of conflict, which is not surprising given the strategic importance of airbases. Third, several of the source hot spots are on towns and villages which have played a prominent role in the Afghan conflict, these include Naw Zad in the North West, the location of multiple offensive operations by the International Security Assistance Force (ISAF) in the latter part of the conflict and Sangin, one of the most hotly contested towns in the conflict. We note that the interpretation of the inferred dynamics is fully dependent on the spatial and temporal resolutions we employ, and may change for different mesh sizes and temporal discretisations.

To evaluate whether the inferred connectivity in $\boldsymbol{A}$ makes any improvement on the approach where the evolution of x_t^i are considered independent (diagonal $\boldsymbol{A}$), we propose to use the one step ahead predictive probabilities. We proceed as follows: we infer a connected

$\boldsymbol{A}$, $\boldsymbol{A}_{\text{conn}}$, using a *chordal* method for state inference, and we also infer a diagonal $\boldsymbol{A}$, $\boldsymbol{A}_{\text{indep}}$. We then use the mean values $\langle \boldsymbol{A}_{\text{conn}} \rangle$ and $\langle \boldsymbol{A}_{\text{indep}} \rangle$ to compute the predictive probabilities. Note that in the latter case, the inference completely decouples into independent inference tasks for all $\{x_t^i\}_t$ and a_{ii}.

The one step ahead predictive probability at time t is given by

$$p(\mathcal{Y}_{t+1} \mid \mathcal{Y}_{1:t}, \boldsymbol{A}) = \int d\boldsymbol{x}_{t,t+1}\, p(\mathcal{Y}_{t+1} \mid \boldsymbol{x}_{t+1}) \mathcal{N}(\boldsymbol{x}_{t+1}; \boldsymbol{A}\boldsymbol{x}_t, \boldsymbol{Q}^{-1}) \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{Q}_{\alpha_t}^{-1} \boldsymbol{h}_{\alpha_t}, \boldsymbol{Q}_{\alpha_t}^{-1}),$$

where $\boldsymbol{h}_{\alpha_t}$ and $\boldsymbol{Q}_{\alpha_t}$ denote the canonical parameters of the $\alpha_t(\boldsymbol{x}_t)$ forward message corresponding to the filtering algorithm. Clearly, the above quantity is not tractable, therefore, we use the corresponding marginal likelihood approximation following from our approach. The integral itself corresponds to expectation propagation based marginal likelihood approximation in latent Gaussian models and is known to be a good quality approximation for a variety of (pseudo) likelihood terms (e.g., Kuss and Rasmussen, 2005; Rasmussen and Williams, 2005). Similar latent Gaussian models where this approximation is shown to perform excellently are the stochastic volatility and spatial log-Gaussian Cox process models in Cseke and Heskes (2011). The cumulative log values of the one step ahead prediction approximations approximate the log evidence, and thus we can also use them to do model comparison. In this way we can assess which model better explains the data.

Figure 13 shows how the one step ahead predictions $p(\mathcal{Y}_{t+1} \mid \mathcal{Y}_{1:t}, \langle \boldsymbol{A}_{\text{conn}} \rangle)$ and $p(\mathcal{Y}_{t+1} \mid \mathcal{Y}_{1:t}, \langle \boldsymbol{A}_{\text{indep}} \rangle)$ compare. The plot shows that the corresponding predictive log likelihood ratio is positive for most times and that the overall log likelihood ratio (sum of one step ahead log likelihood ratios) is positive. Therefore, our qualitative conclusions about the benefit of learning micro diffusions are supported by quantitative evidence: the (approximated) predictive performance of the model increases and a connected model is more likely than an independent one. As future work, we intend to focus on areas of high conflict intensity to assess how the learned conflict dynamics varies w.r.t. the spatial resolution of the model.

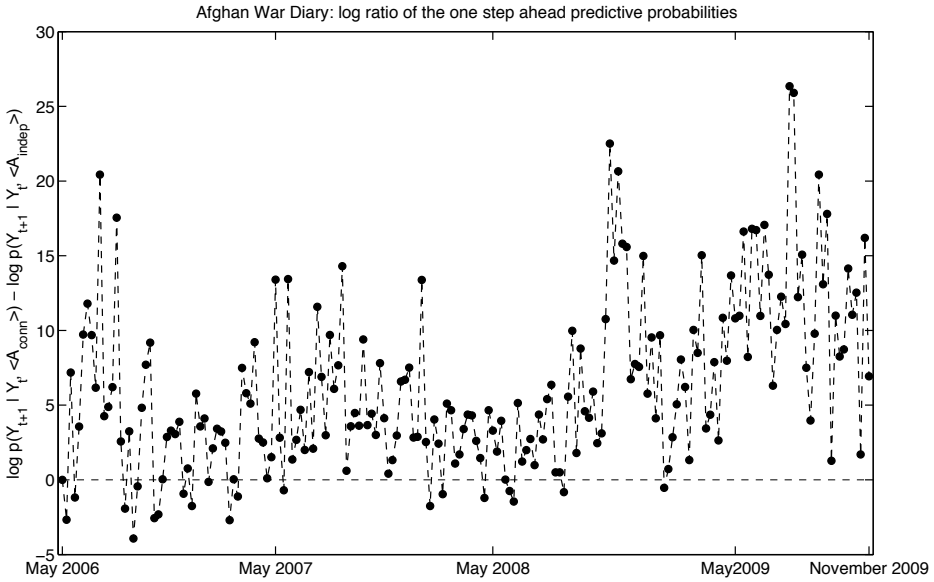

Figure 13: The log ratio of the (approximate) one step ahead predictive probabilities given by the approximations of $\log p(\mathcal{Y}_{t+1} \mid \mathcal{Y}_{1:t}, \langle \boldsymbol{A}_{\text{conn}} \rangle) - \log p(\mathcal{Y}_{t+1} \mid \mathcal{Y}_{1:t}, \langle \boldsymbol{A}_{\text{indep}} \rangle)$ for the AWD data. The plot shows that the connected model generally achieves better predictive performance and that the connected model is more likely overall—the sum of log ratios is clearly positive.

5 Conclusions

In this paper we propose a family of approximate inference methods for spatio-temporal log-Gaussian Cox process models; the algorithms are based on variational approximate inference methods and approximate message passing. Note that the method can be applied to any similar latent Gaussian model (6) with general (pseudo) likelihood $\psi_{t,i}(x_t^i)$. We show how the sparsity in the underlying dynamic model can be exploited in order to overcome the limitations in the standard forward-backward and block inference methods which can become prohibitive for large n and T.

In this paper we employ two layers of approximation. The first is the variational approximation to the posterior distribution, which is factored across the states and parameters. The second one addresses the non-tractability and computational issues associated with the resulting variational distribution over the states, q_X. Approximations due to the variational method and the EP updates for intractable likelihoods have been used extensively with considerable success in several applications. As such, one could argue that the additional approximation used in this work in order to retain sparsity in the message updates intro-

duces further errors that may be hard to quantify. However, the advantage of the proposed method is that it provides a wide range of options w.r.t. accuracy and both computational and storage complexity; in particular we propose using messages with chordal precision structures that serve as a good compromise in complexity between schemes using diagonal and full precision matrix structures. A beneficial aspect of this framework is that one can always choose to do away with these new approximations and revert to the *full* scheme when this is not computationally prohibitive. In practice, as discussed in Section 4.1.2, the quality of the sparse approximation may vary depending on the problem at hand and, while in principle our method could be adjusted by selecting a wider bandwidth, in practice accuracy/computational trade-offs may be inevitable. Finally, as shown in the Supplementary Material, these layers of approximation naturally follow from embedding the variational method into the expectation constrained framework.

We applied the proposed methods to model conflict data and we showed that by using the increased resolution resulting from our methods we can detect micro-diffusions in the Afghan War Diary data. By learning these diffusion effects we can improve the predictive performance and obtain plausible qualitative interpretations of conflict contagion. The proposed methodology can also be applied to epidemic or environmental studies where sparse latent spatial diffusion models—linear diffusions or linear approximations—can be formulated.

In the future we intend to explore the set of structures that lie between the fast and less accurate spanning tree and the somewhat larger chordal structures employed in this work. Currently, we are working on improving the distributed scheduling presented in Section 3.1.3. This is the most important area of further research as the proposed message-passing algorithm, by design, is particularly well suited to take advantage of distributed computing environments. The ease of parallelisation is a major strength of our approach w.r.t. block approaches relying on partial matrix inversions, as it is unclear how solution of the Takahashi equations could be distributed. Another notable advantage of our approach, when compared to the block model, is that it more adaptable to online settings, that is, to cases where we have streamed data.

References

J. Ahn, T. D. Johnson, D. Bhavnani, J. N. S. Eisenberg, and B. Mukherjee. A space-time point process model for analyzing and predicting case patterns of diarrheal disease in northwestern Ecuador. *Spatial and Spatio-temporal Epidemiology*, 9(0):23 – 35, 2014.

P. R. Amestoy, T. A. Davis, and I. S. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal of Matrix Analysis and Applications*, 17(4):886–905, 1996.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society (Series B)*, 72(3), 2010.

P. Baudains, S. D. Johnson, and A. M. Braithwaite. Geographic patterns of diffusion in the 2011 London riots. *Applied Geography*, 45:211–219, 2013.

I. Brainman and S. Toledo. Nested-dissection orderings for sparse LU with partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 23(4):998–1012, 2002.

D. Bueche, N. Sukumar, and B. Moran. Dispersive properties of the natural element method. *Computational Mechanics*, 25(2):207–219, 2000.

B. Cseke and T. Heskes. Approximate marginals in latent Gaussian models. *Journal of Machine Learning Research*, 12:417–454, 2011.

E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference*, pages 157–172. ACM, 1969.

J. Dahl, L. Vandenberghe, and V. Roychowdhury. Covariance selection for non-chordal graphs via chordal embedding. *Optimization Methods Software*, 23(4):501–520, 2008.

S. L. Davis. Google Earth Toolbox, https://code.google.com/p/googleearthtoolbox/, 2012. Last accessed 1 June 2014.

T. A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

P. Diggle, B. Rowlingson, and T. Su. Point process methodology for on-line spatio-temporal disease surveillance. *Environmetrics*, 16(5):423–434, 2005.

A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

M. Gerven, B. Cseke, R. Oostenveld, and T. Heskes. Bayesian source localization with the multivariate Laplace prior. In *Advances in Neural Information Processing Systems 22*, pages 1901–1909, 2009.

M. Gerven, A. Bahramisharif, J. Farquhar, and T. Heskes. Donders Machine Learning Toolbox, https://github.com/distrep/dmlt, 2011–2015. Last accessed 1 June 2012.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society (Series B)*, 73(2):123–214, 2011.

J. Hartikainen, J. Riihimäki, and S. Särkkä. Sparse spatio-temporal Gaussian processes with general likelihoods. In *Proceedings of the 21th International Conference on Artificial Neural Networks*, pages 193–200, 2011.

T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 216–223, 2002.

T. Heskes, M. Opper, W. Wiegerinck, O. Winther, and O. Zoeter. Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, 2005.

M. B. Hooten and C. K. Wikle. A hierarchical Bayesian non-linear spatio-temporal model for the spread of invasive species with application to the Eurasian Collared-Dove. *Environmental and Ecological Statistics*, 15(1):59–70, 2008.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

S. Y. Kang, J. McGree, P. Baade, and K. Mengersen. A case study for modelling cancer incidence using bayesian spatio-temporal models. *Australian & New Zealand Journal of Statistics*, 57(3):325–345, 2015.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2), 2001.

M. Kuss and C. E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, UK, 1996.

F. Lindgren, H. Rue, and J. Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society (Series B)*, 73(4):423–498, 2011.

T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT, MA, 2001.

T. P. Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research Ltd., Cambridge, UK, 2005.

J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log-Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

K. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, volume 9, pages 467–475, 1999.

K. P. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 378–385, 2001.

M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural Computation*, 21(3):786–792, 2009.

M. Opper and O. Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

M. Opper and O. Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6:2177–2204, 2005.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, Cambridge, MA, 2005.

H. Rue and L. Held. *Gaussian Markov Random Fields Theory and Applications*. Chapman and Hall/CRC Press, 2005.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of The Royal Statistical Society (Series B)*, 71(2):319–392, 2009.

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4(1), 1996.

S. Schutte and N. B. Weidmann. Diffusion patterns of violence in civil wars. *Political Geography*, 30:143–152, 2011.

M. W. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

D. Simpson, J. Illian, F. Lindgren, S. Sørbye, and H. Rue. Going off grid: Computationally efficient inference for log-Gaussian Cox processes. *arXiv:1111.0641*, 2011.

D. Simpson, I. W. Turner, C. M. Strickland, and A. N. Pettitt. Scalable iterative methods for sampling from massive Gaussian random vectors. *arXiv:1312.1476*, 2013.

D.H. Stern, R. Herbrich, and T. Graepel. Matchbox: Large scale online bayesian recommendations. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, 2009.

K. Takahashi, J. Fagan, and M.-S. Chin. Formation of a sparse impedance matrix and its application to short circuit study. In *Proceedings of the 8th PICA Conference*, 1973.

C. K. Wikle. A kernel-based spectral model for non-Gaussian spatio-temporal processes. *Statistical Modelling*, 2(1):299–314, 2002.

C. K. Wikle, R. F. Milliff, D. Nychka, and L. M. Berliner. Spatiotemporal hierarchical Bayesian modeling tropical ocean surface winds. *Journal of the American Statistical Association*, 96(454):382–397, 2001.

T. Wist and H. Rue. Specifying a Gaussian Markov random field by a sparse Cholesky triangle. *Communications in Statistics: Simulation and Computation*, 35(1):161–176, 2006.

A. Ypma and T. Heskes. Novel approximations for inference in nonlinear dynamical systems using expectation propagation. *Neurocomputing*, 69(1):85–99, 2005.

K. Yuan, M. Girolami, and M. Niranjan. Markov chain Monte Carlo methods for state-space models with point process observations. *Neural Computation*, 24(6):1462–1486, 2012.

A. Zammit-Mangion, M. Dewar, V. Kadirkamanathan, and G. Sanguinetti. Point process modelling of the Afghan war diary. *Proceeding of the National Academy of Sciences*, 109 (31):12414–12419, 2012a.

A. Zammit-Mangion, G. Sanguinetti, and V. Kadirkamanathan. Variational estimation in spatiotemporal systems from continuous and point-process observations. *IEEE Transactions on Signal Processing*, 60(7):3449–3459, 2012b.