Journal of Machine Learning Research 25 (2024) 1-53 Submitted 7/23; Revised 5/24; Published 7/24

Sparse Graphical Linear Dynamical Systems

Emilie Chouzenoux EMILIE.CHOUZENOUX@INRIA.FR
Center for Visual Computing
Inria, University Paris Saclay
91190 Gif-sur-Yvette, France

Víctor Elvira VICTOR.ELVIRA@ED.AC.UK
School of Mathematics
University of Edinburgh
EH9 3FD Edinburgh, UK

Editor: Alexandre Bouchard

Abstract

Time-series datasets are central in machine learning with applications in numerous fields of science and engineering, such as biomedicine, Earth observation, and network analysis. Extensive research exists on state-space models (SSMs), which are powerful mathematical tools that allow for probabilistic and interpretable learning on time series. Learning the model parameters in SSMs is arguably one of the most complicated tasks, and the inclusion of prior knowledge is known to both ease the interpretation but also to complicate the inferential tasks. Very recent works have attempted to incorporate a graphical perspective on some of those model parameters, but they present notable limitations that this work addresses. More generally, existing graphical modeling tools are designed to incorporate either static information, focusing on statistical dependencies among independent random variables (e.g., graphical Lasso approach), or dynamic information, emphasizing causal relationships among time series samples (e.g., graphical Granger approaches). However, there are no joint approaches combining static and dynamic graphical modeling within the context of SSMs. This work proposes a novel approach to fill this gap by introducing a joint graphical modeling framework that bridges the graphical Lasso model and a causal-based graphical approach for the linear-Gaussian SSM. We present *DGLASSO (Dynamic Graphical Lasso)*, a new inference method within this framework that implements an efficient block alternating majorization-minimization algorithm. The algorithm's convergence is established by departing from modern tools from nonlinear analysis. Experimental validation on various synthetic data showcases the effectiveness of the proposed model and inference algorithm. This work will significantly contribute to the understanding and utilization of time-series data in diverse scientific and engineering applications where incorporating a graphical approach is essential to perform the inference.

Keywords: State-space models, graph inference, sparsity, graphical lasso, majorization-minimization, proximal algorithm

1. Introduction

Time series appear in applications of most fields of science and engineering, ranging from biomedicine, Earth observation, or network analysis, to name a few. The literature of time-series analysis is vast across different fields. For instance, in statistics and signal processing, it is mostly dominated by auto-regressive moving-averaging (ARMA) models and their extensions (Hamilton, 2020). The machine learning literature has recently seen a plethora of neural-network-based models, including long short-term memory (LSTM) models (Hochreiter and Schmidhuber, 1997), recurrent neural networks (RNNs) (Hüsken and Stagge, 2003), gated recurrent unit networks (GRNs) (Cho et al., 2014), and attention mechanisms (Shih et al., 2019).

Graphical inference. Graphical modeling is an important family of approaches for time series analysis in statistical machine learning. The literature in multivariate times series is abundant in models for representing correlation without exploiting the temporal structure (Maathuis et al., 2019). We refer to these approaches as static graphical models. Let us mention the famous Graphical Lasso algorithm (Friedman et al., 2008), and its multiple variants (Chandrasekaran et al., 2012; Benfenati et al., 2020; Belilovsky et al., 2017; Bach and Jordan, 2004; Ying et al., 2020; Mazumder and Hastie, 2012; Fattahi and Sojoudi, 2019; Pircalabelu and Claeskens, 2020), for inference of sparse Gaussian graphical models. Modeling (non-independant) dynamical behaviors through (directed) graph representations, and in particular, causality, has also been explored (Eichler, 2012; Ioannidis et al., 2019; Giannakis et al., 2018; Zhang, 2008; Witte et al., 2020). Graph inference can also be tackled through a supervised machine learning machinery, leading to graph representation learning (Chami et al., 2022), and graphical neural networks (Seo et al., 2018; Cini et al., 2023), with successful applications in the context of time series (Jin et al., 2023) and beyond (Krzywda et al., 2022). Note also that ARMA processes have also been studied from the graphical perspective within the machine learning literature (e.g., in Songsiri and Vandenberghe (2010)).

State-space models (SSMs). State-space models (SSMs) have became very popular in the last decades (Hamilton, 1994; Kim and Nelson, 1999; Sarkka, 2013; Newman et al., 2023) for time series modeling. SSMs characterize complex systems through discrete-time models composed of a hidden (or latent) state that evolves in a Markovian manner. SSMs are composed of a state model, which can mimic realistically the complicated dynamics of the system, and the observation model, which links the temporal observations to the hidden state at the same time step. In SSMs, the Bayesian filtering task consists on computing the posterior probability density function (pdf) of the hidden state at a given time step, given all observations from the time series available up to that time. However, in most SSMs the posterior pdf is intractable and must be approximated, generally through particle filters (Djuric et al., 2003; Doucet et al., 2009; Naesseth et al., 2019). One relevant exception is the linear-Gaussian state-space model (LG-SSM), which allows for exact inference, when the model parameters are known, through the Kalman filter and the Rauch-Tung-Striebel (RTS) smoother (Sarkka, 2013, Chapter 8). The LG-SSM is arguably the most popular model and is still subject of intense research also in the machine learning community (see for instance a Kalman filter for high-dimensional spaces with a low-rank covariance approximation in Schmidt et al. (2023)).

Parameter learning. One of the main challenges in SSMs lies in learning the model parameters, which considers both probabilistic and point-wise approaches. The probabilistic (or fully Bayesian) methods can be applied for a wide class of SSMs (i.e., beyond LG-SSMs) and include for instance particle Markov chain Monte Carlo (Andrieu et al., 2010), particle

Gibbs (Lindsten et al., 2014), sequential Monte Carlo squared (Chopin et al., 2013), and nested particle filters (Crisan and Miguez, 2018; Pérez-Vieites and Míguez, 2021; Pérez-Vieites and Elvira, 2023) (see (Kantas et al., 2015) for a review). Point-wise approaches are generally based on maximum likelihood estimation of the model parameters in LG-SSMs (Sarkka, 2013, Chap. 12). A first family of methods implements optimizers such as quasi-Newton (Olsson et al., 2007) or Newton-Raphson (Gupta and Mehra, 1974), requiring recursive likelihood derivatives expressions (Segal and Weinstein, 1988, 1989; Nagakura, 2021; Gupta and Mehra, 1974) (see the discussion in (Cappe et al., 2005, Sec. 10.2.4)). The second family of point-wise methods is based on expectation-minimization (EM) algorithm (Shumway and Stoffer, 1982)(Cappe et al., 2005, Sec. 10.4)(Sarkka, 2013, Sec. 12.2.3). EM exhibits a simplicity of implementation in the LG-SSM context, which might explain its wide use on practical applied fields, such as finance, electrical engineering, and radar (Sharma et al., 2020, 2021; Frenkel and Feder, 1999). The benefits and drawbacks of these algorithms are discussed in (Shumway and Stoffer, 1982, Sec. 1).

Dynamical graphical modeling, and other machine learning approaches. There is a clear link between state-space modeling and dynamical graphical modeling, as emphasized in (Barber and Cemgil, 2010). For instance, in Sagi et al. (2023), the extended Kalman filter (EKF) is enhanced by taking a graphical perspective. Ioannidis et al. (2018) take an approach of jointly estimating latent processes and the topology of a related graph. Alippi and Zambon (2023) also adopt a graphical perspective in linear-Gaussian models (or linearized versions of them through the EKF) in order to learn model parameters through deep neural networks. This approach bears links with differentiable particle filters (DPFs) (Corenflos et al., 2021; Chen et al., 2021; Chen and Li, 2023), where the proposal of the particle filters but also the dynamics of the latent state are generally learned through gradient-based methods, including modern neural-network architectures. It is also worth mentioning other tight links between SSMs and neural network architectures, such as LSTM and RNNs, for time series processing, as emphasized for instance in Doerr et al. (2018); Rangapuram et al. (2018); Lim (2021); Coskun et al. (2017).

The approaches for graphical model inference and model parameter estimation are often very similar algorithmically speaking, with the difference being mostly related to the interpretation of the parameters and their representation. The graphical modeling brings new insights such as the choice of specific priors (typically, sparsity) and algorithms (e.g., binary edge selection tools). Sparsity usually plays a key role (Brendan and Tenenbaum, 2010), since a graph with few edges can be enforced by imposing a sparsity prior at inference stage. A typical choice is the ℓ_1 (i.e., Lasso) penalty (Friedman et al., 2008; Meinshausen and Bühlmann, 2006; Chouzenoux and Elvira, 2020), which has the advantage of being convex.[1] Other types of priors have also been explored in Ying et al. (2020); Benfenati et al. (2020); Chandrasekaran et al. (2012); Kumar et al. (2020); Hippert-Ferrer et al. (2022), with the aim of imposing specific graph structures (e.g., low rank, bounded spectrum, block sparsity). Proximal algorithms (Combettes and Pesquet, 2011), including augmented Lagrangian methods (Komodakis and Pesquet, 2015), are typically the answer of choice for accounting for the various class of priors of interest in graphical inference. Discrete optimization techniques for edge selection have also been explored (see Benfenati et al. (2018)

[1] See also Gao et al. (2015) for state-space model inference under ℓ_1 prior, although not related to graphical modeling

and references therein). Dynamic graph inference has also been performed using proximal tools (Ioannidis et al., 2019). In the particular case of state-space models, as we explained earlier, several methods, for instance based on Newton or EM, are available for parameter inference. But most available methods did not introduce any graphical aware prior. In our recent works (Chouzenoux and Elvira, 2023; Elvira and Chouzenoux, 2022; Chouzenoux and Elvira, 2024), we introduced more sophisticated EM-based schemes, with proximal updates, to cope with generic convex regularizations. The goal was to estimate the transition matrix within an LG-SSM, adopting a Granger-based graphical interpretation of such matrix. In other words, it became possible to identify which components of the latent space contain information for a one-step-ahead prediction of a given component of the same latent vector. A fully Bayesian approach to estimate the transition matrix was taken in Cox and Elvira (2022, 2023) where the space of sparse matrices was explored via reversible jump Markov chain Monte Carlo (Green and Hastie, 2009).

The need for better graphical modeling in SSMs. Despite this prolific literature, there exists a gap that we propose to fill in this work. Graphical modeling tools are either dedicated to represent static information related to statistical dependence between independent realizations of random variables (e.g., graphical Lasso approach), or to represent dynamic information across different time series. For the latter case, time series are either processed directly (Ioannidis et al., 2019; Songsiri and Vandenberghe, 2010; Mei and Moura, 2017), or through a state-space modeling including hidden state (Elvira and Chouzenoux, 2022). Furthermore, deep learning approaches for SSM parameter estimation (Coskun et al., 2017; Revach et al., 2022; Buchnik et al., 2023) implements supervised training mechanisms which require consequent number of data batches.

However, we are not aware of any joint approach which includes both static and dynamic graphical modeling (hence, two graphs with two distinct purposes), applicable in the context of state-space models, and without the need for supervision. Our contribution lies within this goal, as we describe in the next section.

1.1 Summary of our main contributions

Our contributions are as follows:

- We introduce a joint graphical modeling for representing static and dynamical behaviors in the hidden state of a linear Gaussian state-space model, bridging the gap between the graphical Lasso model from Friedman et al. (2008) and the dynamic model from Elvira and Chouzenoux (2022).

- We present a novel Bayesian inference method, called *DGLASSO (Dynamic Graphical Lasso)*, that performs the graph representation learning task, given a single observed time series. DGLASSO estimates both graphs, under a sparsity prior.

- We propose an original optimization algorithm for the estimation task, implementing a block alternating proximal algorithm with efficient majorization-minimization inner steps. We establish the convergence of our algorithm using recent tools from nonlinear analysis.

- We then perform an extensive experimental validation of the proposed model and inference algorithm by means of experiments on various synthetic datasets. For reproducibility purpose, the code for DGLASSO algorithm, is made publicly available.[2]

2. Preliminaries

2.1 Notation

Bold symbols are used for matrix and vectors. We denote by $\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}}$ the Euclidean norm of $\mathbf{x} \in \mathbb{R}^N$, where $\top$ states from the transpose operation and $\mathbb{R}^N$ is the N-dimensional Euclidean space. We also introduce $\|\mathbf{X}\|_F$, $\|\mathbf{X}\|_2$ and $\mathrm{tr}(\mathbf{X})$, the Frobenius norm, the spectral norm (i.e., largest singular value), and the trace, respectively, of elements $\mathbf{X} = (X(n,\ell))_{1\leq n \leq N, 1 \leq \ell \leq M} \in \mathbb{R}^{N\times M}$. $\mathbf{Id}_N$ is the identity matrix of $\mathbb{R}^N$. $\mathcal{S}_N$ denotes symmetric matrices of $\mathbb{R}^{N\times N}$. Both $\mathbb{R}^{N\times N}$ and $\mathcal{S}_N$ are Hilbert spaces, endowed with the Frobenius norm $\|\cdot\|_F$ and the trace scalar product $\langle \mathbf{X}, \mathbf{Y}\rangle = \mathrm{tr}(\mathbf{XY})$. $\mathcal{S}_N^+$ (resp. $\mathcal{S}_N^{++}$) is the set of $N \times N$ symmetric positive semidefinite (resp. definite) matrices of $\mathbb{R}^N$. Given a sequence of elements $\{\mathbf{x}_k\}_{k=1}^K$ of length $K \geq 1$ and size N, we denote each element as $\mathbf{x}_k = (x_k(n))_{1\leq n\leq N}$ and we use the notation $\mathbf{x}_{k_1:k_2}$ to refer to the subsequence $\{\mathbf{x}_k\}_{k=k_1}^{k_2}$, for $1 \leq k_1 < k_2 \leq K$. For convex analysis concepts, we rely on the notation in the reference textbook by Bauschke and Combettes (2017). We denote $\Gamma_0(\mathcal{H})$ the set of proper, lower semi continuous convex functions from a Hilbert space $\mathcal{H}$ to $(-\infty, +\infty]$ (Bauschke and Combettes, 2017, Chap. 9), and ∂f the subdifferential of $f \in \Gamma_0(\mathcal{H})$ (Bauschke and Combettes, 2017, Chap. 16). With a slight abuse of notation, we make use of the extended form of the minus logarithm determinant function, defined as

$$(\forall \mathbf{P} \in \mathcal{S}_N) \quad -\log\det(\mathbf{P}) = \begin{cases} -\log|\mathbf{P}|, & \text{if } \mathbf{P} \in \mathcal{S}_N^{++}, \\ +\infty, & \text{otherwise,} \end{cases} \tag{1}$$

with $|\mathbf{P}|$ the product of the eigenvalues of $\mathbf{P}$. According to Bauschke and Combettes (2017), the function in Eq. (1) belongs to $\Gamma_0(\mathcal{S}_N)$.

2.1.1 Graphs

We introduce here our notation for describing graphs. Most of our notation is inherited from Bühlmann and Van De Geer (2011).

Let us define a graph $\mathcal{G}$ made of set of N vertices $\mathcal{V} = \{v^{(n)} \text{ s.t. } n \in \{1,\dots,N\}\}$ and of a set of edges $\mathcal{E} = \{e^{(n,\ell)} \text{ s.t. } (n,\ell) \in \mathbb{E}\}$. The latter gathers ordered pairs of distinct vertices, and as such, $\mathbb{E} \subset \{1,\dots,N\}^2$. Undirected graphs are made of undirected edges, that is such that $(n,m) \in \mathbb{E}$ and $(m,n) \in \mathbb{E}$, for every $(m,n) \in \{1,\dots,N\}^2$. In contrast, directed graphs consist of directed edges, where we say that some $e^{(n,\ell)} \in \mathcal{E}$ is directed (from n to ℓ) if $(\ell,n) \notin \mathbb{E}$. We can also distinguish reflexive graphs if self-loops are allowed (i.e., one can have $(n,n) \in \mathcal{E}$), and nonreflexive graphs otherwise. Given these definitions, one can simply bound the cardinality of $\mathbb{E}$ for each category. For instance, for an undirected nonreflexive graph, $\mathrm{Card}(\mathbb{E}) \leq N(N-1)/2$, while a directed nonreflexive graph has $\mathrm{Card}(\mathbb{E}) \leq N(N-1)$,

[2] https://pages.saclay.inria.fr/emilie.chouzenoux/Logiciel.html

and a directed reflexive graph has $\text{Card}(\mathbb{E}) \leq N^2$. Such graph definitions are binary, as it only described presence/absence of edges between vertex pairs. In this work, we require the notion of a weighted graph, where the edges $(e^{(n,\ell)})_{(n,\ell)\in\mathbb{E}}$ are associated to real valued weights $(\omega^{(n,\ell)})_{(n,\ell)\in\mathbb{E}}$. The edge positions and weight values of a graph are summarized in a matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$, where, for every $(n,\ell) \in \mathbb{E}$, $M(n,\ell) = \omega^{(n,\ell)}$ and, for every $(n,\ell) \in \{1,\ldots,N\}^2 \notin \mathbb{E}$, $M(n,\ell) = 0$. An undirected (resp. directed) graph is thus associated to a symmetric (resp. non symmetric) matrix $\mathbf{M}$. A reflexive (resp. non reflexive) graph has non zero (resp. zero) entries in the diagonal of $\mathbf{M}$. The number of edges is simply obtained as the number of non-zero entries in $\mathbf{M}$. We finally define the so-called *binary support* of the (possibly sparse) matrix $\mathbf{M}$. For every $\mathbf{M} \in \mathbb{R}^{N\times N}$, $\text{supp}(\mathbf{M}) = \mathbf{S} \in \{0,1\}^{N\times N}$, with, for every $(n,\ell) \in \{1,\ldots,N\}^2$, $S(n,\ell) = 0$ if and only if $M(n,\ell) = 0$ (i.e., $(n,\ell) \in \mathbb{E}$).

2.2 Dynamical modeling

Let us consider an observed multivariate time series $\{\mathbf{y}_k\}_{k=1}^K \in \mathbb{R}^{N_y}$, with $K \geq 1$, and dimension $N_y \geq 1$. A model describes the evolution of time-series across time. This modeling allows for various machine learning tasks, such as forecasting (i.e., prediction of the future time series, for $k > K$), parameter estimation (e.g., Andrieu et al. (2010)), classification (e.g.., Hüsken and Stagge (2003)), change point detection (e.g., Aminikhanghahi and Cook (2017)), among many other tasks. A vast amount of models are available in the literature of time series analysis. Here we focus on the powerful state-space modeling (SSM) representation, a central block in some recent high-performance generative models such as Mamba (Gu and Dao, 2023).

2.3 Considered model

We consider the linear-Gaussian state-space model (LG-SSM), widely studied in the statistics, control theory, and signal processing literature, and recently popular again in the machine learning literature (see for instance its integration with RNNs and CNNs in Gu et al. (2021)). The LG-SSM is described, for $k = 1,\ldots,K$, as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_k, \tag{2}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k, \tag{3}$$

where,

- $\{\mathbf{x}_k\}_{k=1}^K \in \mathbb{R}^{N_x}$ and $\{\mathbf{y}_k\}_{k=1}^K \in \mathbb{R}^{N_y}$, are the hidden state and the observations at each time k respectively,
- $\mathbf{A} \in \mathbb{R}^{N_x\times N_x}$ is the transition matrix that we aim at estimating,
- $\{\mathbf{H}_k\}_{k=1}^K \in \mathbb{R}^{N_y\times N_x}$ maps the observation model matrices, possibly varying with k, that are assumed to be known,
- $\{\mathbf{q}_k\}_{k=1}^K \sim \mathcal{N}(0,\mathbf{Q})$ is the i.i.d. state noise process, assumed to follow a zero-mean Gaussian model with covariance matrix $\mathbf{Q} \in \mathcal{S}_{N_x}^{++}$ that we also aim at estimating,

- $\{\mathbf{r}_k\}_{k=1}^K \sim \mathcal{N}(0, \mathbf{R}_k)$ is the i.i.d. observation noise process, again zero-mean Gaussian with known covariance matrices $\mathbf{R}_k \in \mathcal{S}_{N_y}^{++}$.

Throughout the paper, we denote $\mathbf{P} = \mathbf{Q}^{-1}$ the precision matrix of the state noise. We furthermore assume an initial state distributed such that $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ with known $\boldsymbol{\mu}_0 \in \mathbb{R}^{N_x}$ and $\boldsymbol{\Sigma}_0 \in \mathcal{S}_{N_x}^{++}$. The state noises and the observation noises are mutually independent and also independent of the initial state $\mathbf{x}_0$.

In the next subsection, we provide some reminders about filtering/smoothing procedures for time series described by an LG-SSM.

2.4 Filtering and smoothing algorithms in linear dynamical systems

Both filtering and smoothing algorithms consist on the computation of a posterior probability density function (pdf) of the hidden state $\{\mathbf{x}_k\}_{k=1}^K$. For every $k \in \{1, \ldots, K\}$, the filtering distribution is $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, where we denote as $\mathbf{y}_{1:k} = \{\mathbf{y}_j\}_{j=1}^k$ the set of observations available up to the time step k, i.e., no future observations can be used to estimate $\mathbf{x}_k$. The filtering problem is suitable for online processing of the observations. The smoothing distribution is $p(\mathbf{x}_k|\mathbf{y}_{1:K})$, where K is the final time-step for which there is an available observation, i.e., for $k \in \{1, \ldots, K-1\}$ (note that it is also possible to condition on a subset of future observations, e.g., $p(\mathbf{x}_k|\mathbf{y}_{1:k+\tau})$ with $\tau \in \mathcal{N}$).

The filtering and smoothing distributions are in general intractable for most SSMs of interest. The LG-SSM is one of the exceptions that admit closed-form solutions.

Estimating the filtering and smoothing distributions is in general a challenging problem, since obtaining these distributions of interest is possible only in few models of interest. For instance, for the LG-SSM described in (2)-(3), it is possible to obtain the filtering and smoothing distributions, for $k = 1, \ldots, K$, in the case where the model parameters $\mathbf{A}$, $\mathbf{Q}$, $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$ are known. Interestingly, these distributions can be obtained in an efficient sequential manner. In particular, the Kalman filter (KF) (Kalman, 1960) allows to obtain recursively (in a forward manner) the sequence of filtering distributions. Its smoothing counterpart, the Rauch-Tung-Striebel (RTS) smoother (Briers et al., 2010), runs backwards to obtain the sequence of smoothing distributions. We note that both algorithms require the model parameters to be known, which in the case of the LG-SSM presented in the previous section are $\mathbf{A}$, $\mathbf{Q} = \mathbf{P}^{-1}$, $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$. Algorithm 1 describes KF, which at each time step $k \in \{1, \ldots, K\}$ performs the (a) prediction/propagation step, where the mean $\boldsymbol{\mu}_{k|k-1}$ and covariance $\boldsymbol{\Sigma}_{k|k-1}$ of the (state) predictive distribution are obtained; and the (b) update step, where the mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ of the filtering distribution are obtained. Algorithm 2 describes the RTS smoother. In this case, the iteration starts at $k = K$ and runs backwards. It can be interpreted as a refinement from the mean and covariance matrices of the filtering distribution, given by Kalman, updating them with information present in future observations. However, note that the observations are not re-used in the RTS algorithm, i.e., all the required information in the observations is absorbed by the filtering distributions, which are used to produce the smoothing distributions.

```
Algorithm 1 Kalman Filter
    Input. Prior parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$; model parameters $\mathbf{A}$, $\mathbf{P}$, $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$; set of observations $\{\mathbf{y}_k\}_{k=1}^K$.
    Recursive step. For $k = 1, \dots, K$
        (a) Prediction/propagation step.
            $\boldsymbol{\mu}_{k|k-1} = \mathbf{A}\boldsymbol{\mu}_{k-1}$    (4)
            $\boldsymbol{\Sigma}_{k|k-1} = \mathbf{A}\boldsymbol{\Sigma}_{k-1}\mathbf{A}^\top + \mathbf{P}^{-1}$    (5)
        (b) Update step.
            $\boldsymbol{\nu}_k = \mathbf{H}_k\boldsymbol{\mu}_{k|k-1}$    (6)
            $\mathbf{v}_k = \mathbf{y}_k - \boldsymbol{\nu}_k$    (7)
            $\mathbf{S}_k = \mathbf{H}_k\boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k$    (8)
            $\mathbf{K}_k = \boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1}$    (9)
            $\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k$    (10)
            $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top$    (11)
    Output. $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. Then, for each $k = 1, ..., K$:
        • state filtering pdf: $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
        • observation predictive pdf: $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{y}_k; \boldsymbol{\nu}_k, \mathbf{S}_k)$
```

2.5 Problem statement

Algorithms 1 and 2 are simple and efficient. However, they require the knowledge of the model parameters. In this paper, we assume $\{\mathbf{H}_k\}_{k=1}^K$, and $\{\mathbf{R}_k\}_{k=1}^K$ to be known, and we address the problem of obtaining the filtering and smoothing distributions when matrices $\mathbf{A}$ and $\mathbf{P}$ are unknown, and must be estimated jointly with the filtering/smoothing step. To do so, we introduce a double graphical modeling of the state equations, where matrices $\mathbf{A}$ and $\mathbf{P}$ now represent the weights of graphs with a specific, and complementary, statistical interpretation. We then propose an efficient and convergent inference approach to estimate both graphs under sparse priors given an observed sequence $\{\mathbf{y}_k\}_{k=1}^K$, while also obtaining its filtering and smoothing distributions at every time steps. The graph representation learning and the time series inference is performed per each time-series observation (i.e., it does not require a training phase in identically distributed time series). This contrasts with the deep learning based techniques, deployed for instance in Revach et al. (2022); Seo et al. (2018), that include a supervision stage averaging over a dataset of multiple time series.

Algorithm 2 *RTS Smoother*

Input. *Filtering parameters* $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=0}^K$ *from the Kalman filter; model parameters* $\mathbf{A}$ *and* $\mathbf{P}$.

Initialization. *Set* $\boldsymbol{\mu}_K^s = \boldsymbol{\mu}_K$ *and* $\boldsymbol{\Sigma}_K^s = \boldsymbol{\Sigma}_K$.

Recursive step. *For* $k = K, K-1, ..., 0$

$$\boldsymbol{\mu}_{k+1}^- = \mathbf{A}\boldsymbol{\mu}_k \tag{12}$$

$$\boldsymbol{\Sigma}_{k+1}^- = \mathbf{A}\boldsymbol{\Sigma}_k\mathbf{A}^\top + \mathbf{P}^{-1} \tag{13}$$

$$\mathbf{G}_k = \boldsymbol{\Sigma}_k\mathbf{A}^\top\left(\boldsymbol{\Sigma}_{k+1}^-\right)^{-1} \tag{14}$$

$$\boldsymbol{\mu}_k^s = \boldsymbol{\mu}_{k|k-1} + \mathbf{G}_k\left(\boldsymbol{\mu}_{k+1}^s - \boldsymbol{\mu}_{k+1}^-\right) \tag{15}$$

$$\boldsymbol{\Sigma}_k^s = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{G}_k\left(\boldsymbol{\Sigma}_{k+1}^s - \boldsymbol{\Sigma}_{k+1}^-\right)\mathbf{G}_k^\top \tag{16}$$

Output. $\{\boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s\}_{k=1}^K$. *Then, for each* $k = 1, ..., K$:

- *state smoothing pdf:* $p(\mathbf{x}_k|\mathbf{y}_{1:K}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s)$

3. Proposed model and inference method

We now introduce our graphical modeling perspective of the LG-SSM. This novel view gives a graph-based interpretation of the transition matrix $\mathbf{A}$, and the precision matrix $\mathbf{P}$. Then, we present the optimization methodology in order to estimate such matrices, and, as such, learning the joint graph representation of a given observed time series.

3.1 Sparse dynamical graphical model

Our novel approach interprets the transition matrix $\mathbf{A}$ and the precision matrix $\mathbf{P}$ as (weighted) directed and undirected graphs, respectively. We now particularize this perspective for each matrix, deepening into the interpretability of such novel view, the complementarity of both graphs, and its benefits during the inference process.

3.1.1 State transition matrix

Matrix $\mathbf{A}$ governs the hidden process in (2) and can be seen as the matrix parameter of an order-one vector auto-regressive (VAR) unobserved process. For every $n \in \{1, \dots, N_x\}$ and $\ell \in \{1, \dots, N_x\}$, the entry $A(n, \ell)$ contains the information of how the n-th time series $\{x_k(n)\}_{k=1}^K$ is affected by the ℓ-th time series $\{x_k(\ell)\}_{k=1}^K$ in consecutive time steps.

More precisely, we can express the update of the n-th dimension of the latent state in the generative model as

$$x_k(n) = \sum_{\ell=1}^{N_x} A(n,\ell) x_{k-1}(\ell) + q_k(n). \tag{17}$$

Thus, if $A(n,\ell) = 0$, it implies that the ℓ-th time series of the latent state does not provide any information to predict the n-th time series one time step ahead conditioned to observing the information in all time series for which $A(n,m) \neq 0$, $m \in \{1,\dots,N_x\} \setminus \ell$.

We express this one-step-ahead conditional independence by denoting

$$x_k(n) \perp\!\!\!\perp x_{k-1}(\ell) | \{x_{k-1}(i)\}_{i \in \mathcal{I}_n}, \tag{18}$$

with $\mathcal{I}_n = \{m | A(n,m) \neq 0, m \in \{1,\dots,N_x\}\}$, i.e., due to the Markovian structure, conditionally on the k-th components of the time series indexed by $\mathcal{I}_n$, all other past components of all time series are independent to $x_k(n)$. Even more, it is possible to establish that $p(x_k(n)|\mathbf{x}_{1:k-1}) = p(x_k(n)|\{x_{k-1}(i)\}_{i\in\mathcal{I}_n})$, again due to the Markovian structure.

Our interpretation is clearly connected to Granger causality (Granger, 1969), and more in particular to conditional Granger causality (Luengo et al., 2019).[3] In particular, if $A(n,\ell) \neq 0$ for some $(n,\ell) \in \{1,\dots,N_x\}^2$, it means that $x_{k-1}(\ell)$ causes (in conditional Granger sense) $x_k(n)$, for every $k \in \{1,\dots,K\}$. The conditional Granger causality directional relationships within the entries of the multivariate latent state time series $\mathbf{x}$ can hence be represented as a graphical model made of a directed graph with N_x vertices, whose weights are those of the transpose matrix $\mathbf{A}^\top$. Moreover, self-loops occur at each vertex associated to a non-zero diagonal entry in $\mathbf{A}$. The perspective of matrix $\mathbf{A}$ interpreted as a (weighted) directed graph bears some links with the work of graphical Granger causality by Shojaie and Michailidis (2010), although we here model the interactions in the latent space instead of directly on the observations. In our case the graphical modeling is simpler, since the work by Shojaie and Michailidis (2010) considers all possible interactions in the observation space across time (i.e., the interpreted graph size is $K \cdot N_y$). The price to pay of our modeling is the difficulty in inferring $\mathbf{A}$, since it governs the latent process, hence it is never observed. We propose an advanced methodology to address the inferential task in Section 3.

Illustrative example. In Figure 1, we display an illustrative example of the graphical model associated to the following state equations for $N_x = 5$, for every $k \in 1,\dots,K$, with $\mathbf{q}_k \in \mathbb{R}^{N_x}$ the latent state noise

$$\begin{cases} x_k(1) = \;0.9\, x_{k-1}(1) + 0.7 x_{k-1}(2) + q_k(1), \\ x_k(2) = -0.3\, x_{k-1}(3) + q_k(2), \\ x_k(3) = \;0.8\, x_{k-1}(5) + q_k(3), \\ x_k(4) = -0.1\, x_{k-1}(2) + q_k(4), \\ x_k(5) = \;0.5\, x_{k-1}(3) + q_k(5). \end{cases} \tag{19}$$

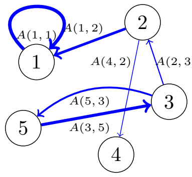
[3] We recall here a vanilla version of the Granger-causality hypothesis test. Let us consider two observed univariate time-series $\mathbf{z}_i = [z_{1,i}, z_{2,i}, ..., z_{K,i}]$ and $\mathbf{z}_j = [z_{1,j}, z_{2,j}, ..., z_{K,j}]$, and the two following statistical models: (A) $z_{k,i} = a_1 z_{k-1,i} + \varepsilon_k$; and (B) $z_{k,i} = a_1 z_{k-1,i} + b_1 z_{k-1,j} + \gamma_k$. We say that $\mathbf{z}_j$ Granger-causes $\mathbf{z}_i$ if, when fitting the two auto-regressive (AR) models (with order $p = 1$ in our example), model B is statistically significantly better than model A, i.e., the variance of γ_k is significantly smaller than the variance of ε_k.

$$\mathbf{A} = \begin{pmatrix} 0.9 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & -0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 \\ 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \end{pmatrix}$$

(a) Matrix **A**.

$$\mathrm{supp}(\mathbf{A}) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

(b) Binary support.



(c) Associated directed graph.

Figure 1: Graphical model associated to (19). Matrix **A** (a), its binary support (b) and associated directed graph (c). The edges are defined as non-zero entries of $\mathbf{A}^\top$. Non-zero diagonal entries result in self-loops (here, in vertex 1). The thickness of arrows is proportional to the absolute entries of $\mathbf{A}^\top$.

We display the matrix **A**, the associated binary matrix supp(**A**) and the resulting directed graph under this interpretation.

3.1.2 State noise precision matrix

Matrix **Q** denotes the noise covariance in the state Eq. (2). Since the noise is assumed to be Gaussian, this matrix, and more precisely, the associated precision matrix $\mathbf{P} = \mathbf{Q}^{-1}$, also has a direct interpretation in terms of graphical modeling, using the notion of Gaussian graphical model (GGM) (Bühlmann and Van De Geer, 2011, Section 13.4)(Uhler, 2017). Since we consider **Q** constant during the whole time series, let us denote the multivariate state noise r.v. at any time step as $\mathbf{q} \sim \mathcal{N}(0, \mathbf{Q})$. The GGM consists in a graphical modeling of the independence (or not) between the scalar random variables $q(1), \dots, q(N_x)$. It is easy to prove that,

$$q(n) \perp\!\!\!\perp q(\ell) | \{q(j), j \in 1, \dots, N_x \backslash \{n, \ell\}\} \iff P(n, \ell) = P(\ell, n) = 0, \tag{20}$$

i.e., the entries n and ℓ of **q** are independent given all other entries if and only if the entry $P(n, \ell)$ is zero (and obviously also $P(\ell, n)$ since the precision matrix is symmetric). Note that it is possible to condition in the l.h.s. of (20) only to the entries $q(j)$ for which $P(n, j) \neq 0$ and the equivalence would still hold. The GGM interprets the precision matrix (i.e., the inverse of the covariance matrix, $\mathbf{P} = \mathbf{\Sigma}^{-1}$) as a weighted undirected graph. In particular, $(n, \ell) \notin \mathbb{E}$ if and only if $P(n, \ell) = P(\ell, n) = 0$.

This GGM construction is at the core of the famous GLASSO (Graphical Lasso) formulation (Friedman et al., 2008)(Maathuis et al., 2019, Section 9.7), whose goal is to build the maximum a posteriori estimator of **P** given realizations of the random vector **q** under a sparsity assumption on matrix **P**. The sparsity is here interpreted as a way to eliminate spurious edges in the graph associated to **P**.

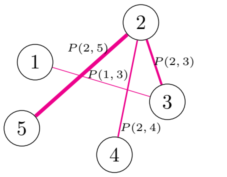
Illustrative example. In Figure 2, we display an illustrative example on the GGM associated to a given precision matrix **P** for $N_x = 5$. We show the associated binary support matrix supp(**P**) and the resulting undirected graph under this interpretation. Although self-loops (i.e., non-zero diagonal elements in **P**) occur, we removed them from the graphical representation for ease of readability.

$$\mathbf{P} = \begin{pmatrix} 2 & 0 & -0.1 & 0 & 0 \\ 0 & 0.9 & 0.3 & -0.2 & 0.5 \\ -0.1 & 0.3 & 0.8 & 0 & 0 \\ 0 & -0.2 & 0 & 2 & 0 \\ 0 & 0.5 & 0 & 0 & 1.5 \end{pmatrix}$$

(a) Matrix $\mathbf{P}$.

$$\mathrm{supp}(\mathbf{P}) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

(b) Support matrix.



(c) Associated undirected graph.

Figure 2: Matrix $\mathbf{P}$ (a), its binary support (b), and the associated undirected graph (c) with edge thickness proportional to the absolute entries of $\mathbf{P}$. Self-loops are removed, for readability purpose.

3.1.3 Proposed unifying view

We now summarize the graphical perspective on both $\mathbf{A}$ and $\mathbf{Q}$ and describe a unifying approach, where sparsity plays a key role. Matrix $\mathbf{A}$ is interpreted as the weight matrix of a directed graph with N_x vertices. Sparsity (i.e., absence of edge in the graph) in $\mathbf{A}$ is interpreted as pair-wise partial/conditional independence, given a subset of the remaining time series, for a one-step ahead prediction of the hidden state. Matrix $\mathbf{P} = \mathbf{Q}^{-1}$ is interpreted as the weight matrix of an undirected graph, related to a GGM describing the noise in the latent space. Sparsity in $\mathbf{P}$ is interpreted as pair-wise partial/conditional independence of two dimensions of the additive state noise, given a subset of the remaining dimensions. Both graphs have N_x nodes (i.e., N_x vertices), and a maximum of N_x^2 edges for $\mathbf{A}$ (resp. $N_x(N_x-1)$ for $\mathbf{P}$), possibly including self-loops, associated to weights defined as the entries of $\mathbf{A}$ or $\mathbf{P}$.

Our perspective in the state process of the LG-SSM in (2) is that $\mathbf{A}$ encodes the way the information flows in consecutive time-steps between the nodes (state dimensions) of the network (vector state). Thus, its properties shape how the energy/information is transferred and dissipated (under the noise). In contrast, $\mathbf{P} = \mathbf{Q}^{-1}$ encodes how information that is not in the system at time $k-1$ enters in the system at time k. In that respect, the interpreted graph with weight matrix $\mathbf{P}$ encodes the dependency of the new information across the nodes of the network.

We adopt the above perspective to estimate both $\mathbf{A}$ and $\mathbf{Q}$ by promoting properties in both graphs. Specifically, we introduce sparsity priors on the matrices, as the sparsity property is key to reach interpretability and compactness of the whole model. In particular, it allows to understand the inner structure of the latent space. Moreover, it can be helpful to speed up computations as the sparsity level is increased, e.g., when running the Kalman filter and RTS smoother. Our proposed method DGLASSO (Dynamic Graphical Lasso) hence aims at providing the maximum a posteriori (MAP) estimator of $\mathbf{A}$ and $\mathbf{P}$ (i.e., the weight matrices related to the graphical modeling of the latent state correlation and causality) under Lasso sparsity regularization on both matrices, given the observed sequence $\mathbf{y}_{1:K}$. A visual representation of DGLASSO graphical model is given in Figure 3. The figure summarizes the relationships among the state entries of an LG-SSM using matrices $(\mathbf{A}, \mathbf{P})$ from Figures 1 and 2.

Related works: Our approach DGLASSO generalizes important existing sparse graphical inference ones. For instance, our model with $\mathbf{A} = 0$ (degenerate case) has no memory, and all the energy/information of the system is lost at each time step, thus the state dimensions only incorporate exogenous energy/information through the additive noises. This degenerate case is the same model than GLASSO (Friedman et al., 2008) in the case when $\mathbf{R}_k \equiv 0$, and same than the robust GLASSO model (Benfenati et al., 2020, Sec.5.2) when $\mathbf{R}_k \equiv \sigma_{\mathbf{R}}^2 \mathbf{Id}_{\mathbf{N_y}}$. In contrast, if the state noise covariance matrix $\mathbf{Q}$ is known, DGLASSO coincides with our recent GraphEM framework (Elvira and Chouzenoux, 2022). Probably the closer related work is (Ioannidis et al., 2019), which also introduces a joint graph modeling within an LG-SSM, capturing order-one causal relationships and instantaneous influence (i.e., order zero), through two sparse graphs. Their proposed inference method is an alternating optimization technique, that infers the two graphs under Lasso prior, jointly with the estimation of hidden state. In contrast with DGLASSO, in (Ioannidis et al., 2019), (i) the state model follows a structural vector autoregressive model (SVAR) where instantaneous causality and noise are distinguished, while DGLASSO assumes an order-one VAR in the hidden state; and (ii) the cost function does not result from a Bayesian modeling, and as such it is not related to a maximum a posteriori loss for the graph variables, (iii) the state estimation is point wise defined as the solution of an handcrafted optimization problem, while DGLASSO preserves a full Bayesian interpretation and hence allows the complete characterization of the filtering/smoothing state distributions. In particular, (Ioannidis et al., 2019) model does not recover GLASSO as a particular case.

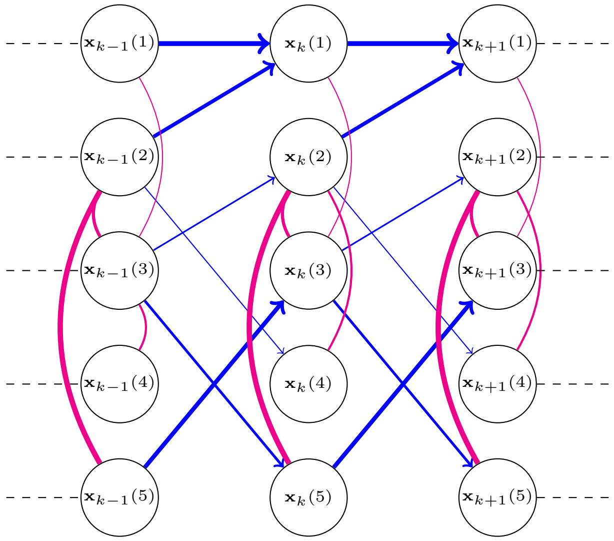


Figure 3: Summary representation of the DGLASSO graphical model, for the example graphs presented in Figs. 1 and 2. Blue (oriented) edges represent Granger causality between state entries among consecutive time steps, encoded in matrix $\mathbf{A}$ (Fig. 1). Magenta edges represent static (i.e., instantaneous) relationships between the state entries, at every time step, due to correlated state noise described by matrix $\mathbf{P}$ (Fig. 2).

3.2 Optimization problem

The considered MAP inference problem in DGLASSO reads as an optimization problem that we formulate hereafter. More specifically, let us denote the posterior of the unknown parameter, $p(\mathbf{A}, \mathbf{P}|\mathbf{y}_{1:K})$, where the hidden states have been marginalized. It is direct to show, using Bayes rule and composition with the (strictly increasing) logarithmic function, that the maximum of $p(\mathbf{A}, \mathbf{P}|\mathbf{y}_{1:K}) \propto p(\mathbf{A}, \mathbf{P})p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P})$, with $p(\mathbf{A}, \mathbf{P})$ some prior on the parameters $\mathbf{A}$ and $\mathbf{P}$, coincides with the minimum of the following loss function:

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) \triangleq \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) + \mathcal{L}_0(\mathbf{A}, \mathbf{P}). \tag{21}$$

with $\mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) \triangleq -\log p(\mathbf{y}_{1:K}|\mathbf{A}, \mathbf{P})$ and $\mathcal{L}_0(\mathbf{A}, \mathbf{P}) = -\log p(\mathbf{A}, \mathbf{P})$. According to (Sarkka, 2013, Chap. 12),

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = \sum_{k=1}^{K} \tfrac{1}{2} \log\det(2\pi \mathbf{S}_k) + \frac{1}{2}\mathbf{z}_k^\top \mathbf{S}_k^{-1} \mathbf{z}_k, \tag{22}$$

where $\mathbf{z}_k = \mathbf{y}_k - \mathbf{H}_k \mathbf{A} \boldsymbol{\mu}_{k-1}$ and $\mathbf{S}_k$ is the covariance matrix of the predictive distribution $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{A}, \mathbf{P}) = \mathcal{N}(\mathbf{y}_k; \mathbf{H}\mathbf{A}\boldsymbol{\mu}_{k-1}, \mathbf{S}_k)$, both being obtained by the KF in Algorithm 1 run for given $(\mathbf{A}, \mathbf{P})$ (see (Sarkka, 2013, Section 4.3)).

As already mentioned, the introduction of priors that induce sparsity is advantageous due to several reasons. First, it generally reduces over-fitting, particularly when K is low compared to the number of parameters to be estimated. Also, it enhances the interpretability. The zero elements in $\mathbf{A}$ and $\mathbf{P}$ have a clear interpretation of conditional independence between the time series. Ideally, we would use the ℓ_0 (pseudo) norm of $\mathbf{A}$ and $\mathbf{P}$, i.e., penalizing the number of non-zero entries of the matrix. However, this penalty is known to have undesirable properties such as being non-convex, non continuous, and associated to a improper law $p(\mathbf{A})$. We thus propose instead the regularization term

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_0(\mathbf{A}, \mathbf{P}) = \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1. \tag{23}$$

The ℓ_1 norm in (23), defined as the sum of absolute values of the matrix entries, is a proper convex function, that leads to the so-called Lasso regularization (Bach et al., 2012). Note that this penalty, used in numerous works of signal processing and machine learning (Tibshirani, 1996; Chaux et al., 2007), including graph signal processing (Friedman et al., 2008; Benfenati et al., 2020), is associated with a joint Laplace prior distribution on $\mathbf{A}$ and $\mathbf{P}$. Such joint distribution factorizes (i.e., the prior assumes independence on both parameters), the means are zero, and the scale parameters are proportional to, respectively, λ_A and λ_P. The larger the regularization parameter λ_A (or λ_P), the higher sparsity of $\mathbf{A}$ (or $\mathbf{P}$), with the degenerate case of a null $\mathbf{A}$ (and $\mathbf{P}$) when the regularization parameter grows.

3.3 General minimization procedure

The expressions (22)-(23) provide an effective way to evaluate (21). However, due to the recursive form in (22), it is challenging to derive direct quantities (e.g., gradient) for $\mathcal{L}_{1:K}$. Moreover, despite its simple expression, the regularization term (23) is non differentiable. For both reasons, the minimization of (21) is a challenging question.

We propose a block alternating majorization-minimization (MM) technique to infer the MAP estimates of $(\mathbf{A}, \mathbf{P})$. Our method presents the advantage of sound convergence guarantees and the ability to incorporate sparsity priors on both $\mathbf{A}$ and $\mathbf{P}$. The general idea of MM is to replace a complicated optimization problem by a sequence of more tractable ones (Sun et al., 2016; Hunter and Lange, 2004). Surrogate approximations of the cost function are built iteratively, following a majorization principle. For any estimates $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$ (i.e., in the interior domain of definition of $\mathcal{L}$) of $(\mathbf{A}, \mathbf{P})$, a majorizing approximation is constructed for the likelihood term. It is required to satisfy both

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A}, \mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) \geq \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}), \tag{24}$$

and also the so-called tangency condition

$$\mathcal{Q}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = \mathcal{L}_{1:K}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}). \tag{25}$$

The MM algorithm then alternates between **M**ajorization step to build function $\mathcal{Q} + \mathcal{L}_0$ satisfying conditions (24) and (25), and **M**inimization step to minimize this majorizing approximation. In our proposed approach, we adopt a block alternating implementation of the MM method, where only one variable (i.e., $\mathbf{A}$ or $\mathbf{P}$) is updated at each iteration, following a cyclic rule (Jacobson and Fessler, 2007; Hong et al., 2016). This alternating strategy considerably simplifies the majorizing function construction as well as its minimization. We furthermore add so-called proximal terms in both updates. Let us recall that, for a function $\phi : \mathcal{H} \mapsto (-\infty, +\infty] \in \Gamma_0(\mathcal{H})$, with $\mathcal{H}$ a Hilbert space with endowed norm $\|\cdot\|$, the proximity operator[4] of function f at $\widetilde{\mathbf{V}} \in \mathcal{H}$ is defined as (Combettes and Pesquet, 2011)

$$\operatorname{prox}_{\phi}(\widetilde{\mathbf{V}}) = \underset{\mathbf{V} \in \mathcal{H}}{\operatorname{argmin}} \left(\phi(\mathbf{V}) + \frac{1}{2} \|\mathbf{V} - \widetilde{\mathbf{V}}\|^2 \right). \tag{26}$$

In our context, the considered Hilbert space is either $\mathbb{R}^{N_x \times N_x}$ for the update of the transition matrix or $\mathcal{S}_{N_x}$ for the update of the precision matrix, and the endowed norm is in both cases the Frobenius norm $\|\cdot\|_F$. Introducing proximity terms thus amounts to adding to each majorizing function a quadratic distance to the previous iterate, weighted by a positive factor. This modification preserves the MM interpretation of the method, while ensuring improved stability and convergence guarantees. As we show below, the iterates belong to the interior of domain of the loss function by construction. Namely for every $i \in \mathbb{N}$, $(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \in \mathbb{R}^{N_x \times N_x} \times \mathcal{S}_{N_x}^{++}$, so the precision matrix remains invertible along the iterations and the algorithm is well defined. The resulting DGLASSO approach is summarized in Algorithm 3. DGLASSO aims at providing ultimately the MAP estimates for the matrix parameters $(\mathbf{A}, \mathbf{P})$ of the considered LG-SSM, through the minimization of (21). The covariance state noise MAP estimate is straightforwardly obtained by inversion of the precision state noise matrix provided as an output of DGLASSO. The state filtering/smoothing pdf associated to each estimates are finally computed by running KF/RTS loops when setting $(\mathbf{A}, \mathbf{P})$ equal to DGLASSO outputs. Next sections are dedicated to the (i) construction of the majorizing function, (ii) discussion about the resolution of each inner step, and (iii) convergence analysis.

[4] See also `http://proximity-operator.net/`

```
Algorithm 3 DGLASSO algorithm

    Inputs. Prior parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$; model parameters $\{\mathbf{H}_k\}_{k=1}^K$ and $\{\mathbf{R}_k\}_{k=1}^K$; set of observations $\{\mathbf{y}_k\}_{k=1}^K$; hyper-parameters $(\lambda_A, \lambda_P) > 0$; stepsizes $(\theta_A, \theta_P) > 0$; precisions $(\varepsilon, \xi) > 0$.

    Initialization. Set $(\mathbf{A}^{(0)}, \mathbf{P}^{(0)}) \in \mathbb{R}^{N_x \times N_x} \times \mathcal{S}_{N_x}^{++}$.

    Recursive step. For $i = 0, 1, \ldots$:

     (a) //Update transition matrix
      (i) Build surrogate function satisfying (24) and (25) at $\widetilde{\mathbf{A}} = \mathbf{A}^{(i)}$ and $\widetilde{\mathbf{P}} = \mathbf{P}^{(i)}$.
     (ii) Run Algorithm 4 with precision $\xi$ to solve

          $\mathbf{A}^{(i+1)} = \operatorname{prox}_{\mathbf{A} \to \theta_A \mathcal{Q}(\mathbf{A}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \theta_A \lambda_A \|\mathbf{A}\|_1} \left(\mathbf{A}^{(i)}\right),$
          $\qquad = \underset{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}}{\operatorname{argmin}} \; \mathcal{Q}(\mathbf{A}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A} - \mathbf{A}^{(i)}\|_F^2.$      (27)

     (b) //Update noise precision matrix
      (i) Build surrogate function satisfying (24) and (25) at $\widetilde{\mathbf{A}} = \mathbf{A}^{(i+1)}$ and $\widetilde{\mathbf{P}} = \mathbf{P}^{(i)}$.
     (ii) Run Algorithm 5 with precision $\xi$ to solve

          $\mathbf{P}^{(i+1)} = \operatorname{prox}_{\mathbf{P} \to \theta_P \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \theta_P \lambda_P \|\mathbf{P}\|_1} \left(\mathbf{P}^{(i)}\right),$
          $\qquad = \underset{\mathbf{P} \in \mathcal{S}_{N_x}}{\operatorname{argmin}} \; \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2\theta_P} \|\mathbf{P} - \mathbf{P}^{(i)}\|_F^2.$      (28)

    If $\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F \le \varepsilon \|\mathbf{A}^{(i)}\|_F$ and $\|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F \le \varepsilon \|\mathbf{P}^{(i)}\|_F$, stop the recursion by returning $(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)})$.

    Output. MAP estimates of the transition and state noise precision matrices.
```

3.4 Building the majorizing function

In this section, we derive a theorem regarding the expression of the loss function $\mathcal{L}_{1:K}$ and a valid majorant function for it.

Theorem 1 *The loss function can be expressed as*[5]

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A}, \mathbf{P}) = \frac{1}{2} \sum_{k=1}^{K} \Big((\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P} \, (\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) \Big)$$

$$- \frac{K}{2} \log \det(2\pi \mathbf{P}) + \log p(\mathbf{x}_{0:K} | \mathbf{y}_{1:K}, \mathbf{A}, \mathbf{P}) + \log p(\mathbf{x}_0) - \sum_{k=1}^{K} p(\mathbf{y}_k | \mathbf{x}_k). \quad (29)$$

Moreover, consider the outputs of Algorithms 1 and 2 for a given $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ *and* $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x^{++}}$. *Denote*

$$\begin{cases} \widetilde{\mathbf{\Psi}} & = \frac{1}{K} \sum_{k=1}^{K} \left(\mathbf{\Sigma}_k^s + \boldsymbol{\mu}_k^s (\boldsymbol{\mu}_k^s)^\top \right), \\ \widetilde{\mathbf{\Delta}} & = \frac{1}{K} \sum_{k=1}^{K} \left(\mathbf{\Sigma}_k^s \mathbf{G}_{k-1}^\top + \boldsymbol{\mu}_k^s (\boldsymbol{\mu}_{k-1}^s)^\top \right), \\ \widetilde{\mathbf{\Phi}} & = \frac{1}{K} \sum_{k=1}^{K} \left(\mathbf{\Sigma}_{k-1}^s + \boldsymbol{\mu}_{k-1}^s (\boldsymbol{\mu}_{k-1}^s)^\top \right), \end{cases} \quad (30)$$

where, for every $k \in \{1, \ldots, K\}$, $\mathbf{G}_k = \mathbf{\Sigma}_k \widetilde{\mathbf{A}}^\top (\widetilde{\mathbf{A}} \mathbf{\Sigma}_k \widetilde{\mathbf{A}}^\top + \widetilde{\mathbf{P}}^{-1})^{-1}$ *(see Algorithm 2). Then, conditions* (24) *and* (25) *hold with*

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A}, \mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = \frac{K}{2} \operatorname{tr} \Big(\mathbf{P} (\widetilde{\mathbf{\Psi}} - \widetilde{\mathbf{\Delta}} \mathbf{A}^\top - \mathbf{A} \widetilde{\mathbf{\Delta}}^\top + \mathbf{A} \widetilde{\mathbf{\Phi}} \mathbf{A}^\top) \Big)$$

$$- \frac{K}{2} \log \det(2\pi \mathbf{P}). \quad (31)$$

As a consequence, for every $(\theta_A, \theta_P) > 0$,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) \leq \mathcal{Q}(\mathbf{A}, \mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) - \mathcal{Q}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}})$$

$$+ \lambda_A \|\mathbf{A}\|_1 + \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2\theta_A} \|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 + \frac{1}{2\theta_P} \|\mathbf{P} - \widetilde{\mathbf{P}}\|_F^2, \quad (32)$$

with equality holding for $\mathbf{A} = \widetilde{\mathbf{A}}$ *and* $\mathbf{P} = \widetilde{\mathbf{P}}$.

Proof See Appendix A. ■

Theorem 1 allows to build, for any tangent point $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}})$, a majorizing approximation (32) for $\mathcal{L}$. Function (32) depends on three matrices $(\widetilde{\mathbf{\Psi}}, \widetilde{\mathbf{\Delta}}, \widetilde{\mathbf{\Phi}})$ through (30), themselves depending on the tangent point $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}})$, as highlighted by the tilde symbol. DGLASSO method leverages this property designing a block alternating MM scheme. At each iteration $i \in \mathbb{N}$, two steps are conducted, namely the update of (a) the transition matrix, (b) the noise precision matrix. Each steps follows an MM structure, that is it first builds a majorizing approximation for $\mathcal{L}$ at the current estimates, using Theorem 1, and then minimizes it with respect to the active variable ($\mathbf{A}$ in step (a), or $\mathbf{P}$ in step (b)). Processing the variables in two distinct steps allows to build upon the desirable convex structure of (32) with respect to one of the variable, the other being fixed. The good performance of MM approaches combined with block alternating steps have been illustrated in (Hong et al., 2016; Chouzenoux et al., 2016; Hien et al., 2020). In particular, convergence guarantees are at reach, as we will show in Section 4.

[5] Note that the l.h.s. in (29) does not depend on $\mathbf{x}_{0:K}$, so the r.h.s. is valid for any arbitrary value of $\mathbf{x}_{0:K}$ with non-zero probability under $p(\mathbf{x}_{0:K})$, i.e., for all $\mathbf{x}_{0:K} \in \mathbb{R}^{(K+1)N_x}$.

3.5 Resolution of the inner problems

We now discuss the structure and resolution of the inner problems (27) and (28) arising in Algorithm 3.

Minimization with respect to A: Let $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$. By definition of the proximity operator (26) and the majorant expression in (31), Eq. (27) requires to

$$\text{minimize}_{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}} \mathcal{C}_1(\mathbf{A}), \tag{33}$$

where, for every $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$,

$$\mathcal{C}_1(\mathbf{A}) \triangleq \frac{\theta_A K}{2} \mathrm{tr}\left(\widetilde{\mathbf{P}}(\widetilde{\mathbf{\Psi}} - \widetilde{\mathbf{\Delta}}\mathbf{A}^\top - \mathbf{A}\widetilde{\mathbf{\Delta}}^\top + \mathbf{A}\widetilde{\mathbf{\Phi}}\mathbf{A}^\top)\right) + \theta_A \lambda_A \|\mathbf{A}\|_1 + \frac{1}{2}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2.$$

Remarkably, the problem above is a special instance of a multivariate Lasso regression problem (Tibshirani, 1996), for which many efficient iterative solvers are available. The specificity here is that the problem is strongly convex thanks to the proximal term. We thus suggest the use of the Dykstra-like algorithm by Bauschke and Combettes (2008), whose iterations are recalled in the Appendix B. This method presents the advantage of fast convergence rate, ease of implementation, and no parameter tuning.

Minimization with respect to P: Let $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$. The update of Eq. (28) solves a minimization problem with generic form

$$\text{minimize}_{\mathbf{P} \in \mathcal{S}_{N_x}} \mathcal{C}_2(\mathbf{P}), \tag{34}$$

where, for every $\mathbf{P} \in \mathcal{S}_{N_x}$, we denote

$$\mathcal{C}_2(\mathbf{P}) \triangleq \frac{\theta_P K}{2} \mathrm{tr}\left(\mathbf{P}\widetilde{\mathbf{\Pi}}\right) - \frac{\theta_P K}{2} \log\det \mathbf{P} + \theta_P \lambda_P \|\mathbf{P}\|_1 + \frac{1}{2}\|\mathbf{P} - \widetilde{\mathbf{P}}\|_F^2,$$

$$\widetilde{\mathbf{\Pi}} \triangleq \widetilde{\mathbf{\Psi}} - \widetilde{\mathbf{\Delta}}\widetilde{\mathbf{A}}^\top - \widetilde{\mathbf{A}}\widetilde{\mathbf{\Delta}}^\top + \widetilde{\mathbf{A}}\widetilde{\mathbf{\Phi}}\widetilde{\mathbf{A}}^\top. \tag{35}$$

Here we have used the definition of the proximity operator (26) and the majorant expression in (31) (ignoring the constant multiplicative term in the logarithm). Remarkably, (34) reads as a regularized form of the famous GLASSO problem (Friedman et al., 2008), and gets actually equivalent to it when $\theta_P \to \infty$. Matrix $\widetilde{\mathbf{\Pi}}$ in (35) plays the same role as the empirical covariance matrix in GLASSO, and $\frac{2\lambda_P}{K}$ acts as the weight on the ℓ_1 term. The proximal term works as a Tikhonov-like regularizer, ensuring the strong convexity of the problem, and thus the uniqueness of its minimizer. Moreover, by the definition of the log-determinant in Eq. (1), the solution of (34) belongs to $\mathbf{P} \in \mathcal{S}_{N_x}^{++}$, i.e., the precision matrix is symmetric and invertible, and thus a valid covariance matrix can be deduced from it by inversion. Standard GLASSO solvers can be easily modified to solve (34). We present in the Appendix B the complete derivations, when applying the Dykstra-like algorithm from Bauschke and Combettes (2008), which presents the advantage of fast convergence, and ease of implementation.

3.6 Discussion

Strengths and weaknesses of the MM method: As highlighted in our introductory section, state-of-the-art approaches for SSM parameter estimation are grounded either on the EM (or MM) framework or on gradient (or Newton) updates (through sensitivity equations). In the present work, we adopt the former strategy. We here discuss the pros and cons for such choice for the minimization of (21), some aspects being also illustrated by the numerical experiments presented in Section 5.1.6.

A potential limitation of the proposed minimization algorithm is its computational complexity. From Theorem 1, the construction of the majorizing function at each tangent point $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}})$ requires to run the KF Algorithm 1 and the RTS Algorithm 2. This is indeed necessary for the computation, using (30), of the three matrices, $(\widetilde{\mathbf{\Psi}}, \widetilde{\mathbf{\Delta}}, \widetilde{\mathbf{\Phi}})$. This feature was already present in previous EM-based approaches for LG-SSM parameter estimations (see for instance (Sarkka, 2013, Chap. 12) and Elvira and Chouzenoux (2022)). Due to its block alternating form, the proposed DGLASSO algorithm requires to build twice per iteration a majorizing approximation, which means that KF/RTS are ran twice per iteration. Moreover, inner solvers are required, for the minimization of the majorant functions, increasing again the method complexity. In contrast, gradient-based methods relying on sensitivity equations (Gupta and Mehra, 1974) only require the KF recursions (and not the RTS ones) for building their updates, and do not make use of inner iterations, which might seem appealing from the computational viewpoint.

Gradient-based approaches however face some difficulties. The ℓ_1 norm is non differentiable, preventing from the use of standard first (or second) order minimization techniques as those proposed in Olsson et al. (2007). This could be fixed either by smoothing the ℓ_1 terms at the price of adding extra hyperparameters or by implementing suitable updates (e.g., subgradient or proximal operations). An additional issue comes from the logarithmic determinant term on $\mathbf{P}$, involved in the likelihood term (29). This makes the definition domain of the loss function restricted to the set of symmetric positive definite matrices $\mathbf{P}$. Moreover, onto this set, the gradient loss is not Lipschitz, and its norm explodes for matrices $\mathbf{P}$ close to non invertible. DGLASSO accounts for the singularity of the log-det term thanks to proximal-based updates in the inner solver of Algorithm 5 detailed in Appendix B. In contrast, gradient-based solvers require a tedious manual stepsize tuning (typically, backtracking linesearch), with no proved guarantee of the existence of a valid stepsize. As a consequence, we observed in our experiments in Sec. 5.1.6 that a very large number of iterations was required to reach stability for these methods, which largely undermines their apparent advantage in terms of iteration complexity.

Finally, an important feature of the proposed method is that it inherits the sound convergence properties from the MM paradigm, as detailed in our Section 4. DGLASSO iterates are proved to reach a cluster point of the MAP loss function. No such results can be derived for the gradient-based competitors, due to the non-convexity of the loss, and the singularity of its gradient, as we already mentioned.

Regularization parameters tuning: DGLASSO formulation requires the setting of the weights (λ_A, λ_P) balancing the sparsity prior and the data fidelity. The automatic tuning of DGLASSO regularization hyperparameters in the context of real data is a challenging problem. Among promising avenues, let us mention several recent approaches based on supervised learning, either implementing deep learning architectures (Shrivastava et al., 2020; Revach et al., 2022; Buchnik et al., 2023; Shrivastava et al., 2022), or bi-level approaches (Pouliquen et al., 2023; Franceschi et al., 2017; Bertrand et al., 2022) combined with SURE statistical estimators (Deledalle et al., 2014; Luo et al., 2014). Note however that the aforementioned works consider either static graphical models such as GLASSO, or Kalman filtering without graph formulation. Their extension to our DGLASSO model remains an open question that we leave as a future research line. In our experimental section, synthetic data will be considered, allowing an empirical finetuning through qualitative metrics computed on the ground truth model.

4. Convergence analysis

We now present our convergence proof for the proposed DGLASSO algorithm presented in Algorithm 3. Our analysis is made assuming that the inner steps (27) and (28) are solved in an exact manner. The extension of the result to the case of an inexact resolution of the subproblems is discussed at the end of the section.

4.1 Descent property

Lemma 1 *Assuming exact resolution of* (27) *and* (28)*, the sequence* $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ *produced by DGLASSO algorithm satisfies*

$$(\forall i \in \mathbb{N}) \quad \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) \leq \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}). \tag{36}$$

Proof Let $i \in \mathbb{N}$. First, let us use the Theorem 1 at $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = (\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)})$ (Inequality (a)), and the definition of $\mathbf{P}^{(i+1)}$ in (28) (Inequality (b)) as

$$\begin{aligned}\mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) &\overset{(a)}{\leq} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i+1)}\|_1 + \frac{1}{2\theta_A}\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2 + \frac{1}{2\theta_P}\|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i)}\|_F^2 \qquad (37) \\ &\overset{(b)}{\leq} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 + \frac{1}{2\theta_A}\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2 + \frac{1}{2\theta_P}\|\mathbf{P}^{(i)} - \mathbf{P}^{(i)}\|_F^2. \qquad (38)\end{aligned}$$

Inequality (38) simplifies into

$$\mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) \leq \mathcal{L}_{1:K}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 = \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}). \tag{39}$$

Applying Theorem 1 now at $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = (\mathbf{A}^{(i)}, \mathbf{P}^{(i)})$ (Inequality (a)), and the definition of $\mathbf{A}^{(i+1)}$ in (27) (Inequality (b)), leads to

$$\begin{aligned}\mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) &\overset{(a)}{\leq} \mathcal{Q}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i+1)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 + \frac{1}{2\theta_A}\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i)}\|_F^2 + \frac{1}{2\theta_P}\|\mathbf{P}^{(i)} - \mathbf{P}^{(i)}\|_F^2 \qquad (40) \\ &\overset{(b)}{\leq} \mathcal{Q}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) - \mathcal{Q}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \\ &\quad + \lambda_A \|\mathbf{A}^{(i)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 + \frac{1}{2\theta_A}\|\mathbf{A}^{(i)} - \mathbf{A}^{(i)}\|_F^2 + \frac{1}{2\theta_P}\|\mathbf{P}^{(i)} - \mathbf{P}^{(i)}\|_F^2, \qquad (41)\end{aligned}$$

which simplifies into

$$\mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \leq \mathcal{L}_{1:K}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \lambda_A \|\mathbf{A}^{(i)}\|_1 + \lambda_P \|\mathbf{P}^{(i)}\|_1 = \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}), \qquad (42)$$

and concludes the proof. ■

If the cost function $\mathcal{L}$ is lower bounded (e.g., if it is coercive), Lemma 1 implies the convergence of sequence $\left\{\mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)})\right\}_{i\in\mathbb{N}}$ to a finite value and, as such, the existence of cluster points in $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$. This is however a rather weak convergence result and we propose hereafter a thorough analysis relying on recent tools of nonlinear analysis (Attouch et al., 2010; Bolte et al., 2014) combined with the works (Hien et al., 2023, 2020) on the convergence of block alternating MM schemes.

4.2 Convergence guarantees

Theorem 2 *Consider the sequence $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ generated by DGLASSO, assuming exact resolution of both inner steps (27) and (28). If the sequence $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ is bounded, then $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ converges to a critical point of $\mathcal{L}$.*

Proof The convergence analysis relies in proving that the exact form of DGLASSO algorithm is a special instance of TITAN algorithm from (Hien et al., 2023), and as such, inherits (Hien et al., 2023, Theorem 6) and (Hien et al., 2023, Theorem 8), under our assumptions.

- Let us introduce the following notations.

$$\begin{aligned}(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad f(\mathbf{A}, \mathbf{P}) = \frac{1}{2}\sum_{k=1}^{K}\Big((\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P}\,(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})\Big) \\ + \log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \mathbf{A}, \mathbf{P}) - \log p(\mathbf{x}_0) - \sum_{k=1}^{K}\log p(\mathbf{y}_k|\mathbf{x}_k), \qquad (43)\end{aligned}$$

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad g_1(\mathbf{A}) = \lambda_A \|\mathbf{A}\|_1, \qquad (44)$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad g_2(\mathbf{P}) = -\frac{K}{2}\log\det(2\pi\mathbf{P}) + \lambda_P \|\mathbf{P}\|_1, \qquad (45)$$

so that

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A}, \mathbf{P}) = f(\mathbf{A}, \mathbf{P}) + g_1(\mathbf{A}) + g_2(\mathbf{P}), \tag{46}$$

with f lower semi-continuous function, $g_1 \in \Gamma_0(\mathbb{R}^{N_x \times N_x})$ and $g_2 \in \Gamma_0(\mathcal{S}_{N_x})$. Moreover, let us denote

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{J}(\mathbf{A}, \mathbf{P}) = \frac{K}{2} \operatorname{tr}\left(\mathbf{P}(\widetilde{\mathbf{\Psi}} - \widetilde{\mathbf{\Delta}}\mathbf{A}^\top - \mathbf{A}\widetilde{\mathbf{\Delta}}^\top + \mathbf{A}\widetilde{\mathbf{\Phi}}\mathbf{A}^\top) \right), \tag{47}$$

and, for every $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad u_1(\mathbf{A}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = \mathcal{J}(\mathbf{A}, \widetilde{\mathbf{P}}) + \frac{1}{2\theta_A} \|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 + f(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) - \mathcal{J}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}), \tag{48}$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad u_2(\mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = \mathcal{J}(\widetilde{\mathbf{A}}, \mathbf{P}) + \frac{1}{2\theta_P} \|\mathbf{P} - \widetilde{\mathbf{P}}\|_F^2 + f(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) - \mathcal{J}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}). \tag{49}$$

By Theorem 1, the following majorization properties hold for every $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad u_1(\mathbf{A}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) \geq f(\mathbf{A}, \widetilde{\mathbf{P}}), \tag{50}$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad u_2(\mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) \geq f(\widetilde{\mathbf{A}}, \mathbf{P}), \tag{51}$$

and we have the tangency condition, for every $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$,

$$u_1(\widetilde{\mathbf{A}}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = f(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}), \tag{52}$$

$$u_2(\widetilde{\mathbf{P}}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = f(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}). \tag{53}$$

Then, straightforward computations allow to rewrite the iterates of Algorithm 3 as follows:

$$(\forall i \in \mathbb{N}) \quad \begin{cases} \mathbf{A}^{(i+1)} = \underset{\mathbf{A} \in \mathbb{R}^{N_x \times N_x}}{\operatorname{argmin}} \; u_1(\mathbf{A}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + g_1(\mathbf{A}), \\ \mathbf{P}^{(i+1)} = \underset{\mathbf{P} \in \mathcal{S}_{N_x}}{\operatorname{argmin}} \; u_2(\mathbf{P}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + g_2(\mathbf{P}), \end{cases} \tag{54}$$

which identifies with the iterative scheme TITAN from Hien et al. (2023), in the case of two blocks and setting the extrapolation step to zero. The rest of the proof amounts to check the fulfillment of the assumptions required for (Hien et al., 2023, Theorem 6) and (Hien et al., 2023, Theorem 8).

- Let us denote, for every $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x}) \quad \tilde{u}_1(\mathbf{A}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = u_1(\mathbf{A}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) + g_1(\mathbf{A}), \tag{55}$$

$$(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \tilde{u}_2(\mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = u_2(\mathbf{P}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) + g_2(\mathbf{P}). \tag{56}$$

Functions u_1 and u_2 are quadratic and strongly convex with respective strong convexity constants θ_A^{-1} and θ_P^{-1}. Since both g_1 and g_2 are convex, functions $\tilde{u}_1$ and $\tilde{u}_2$ are also strongly convex, with respective strong convexity constants θ_A^{-1} and θ_P^{-1}. Let $i \in \mathbb{N}$.

According to the optimality conditions of both equations in (54), there exists $\mathbf{T}_1^{(i+1)} \in \partial \tilde{u}_1(\mathbf{A}^{(i+1)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{T}_2^{(i+1)} \in \partial \tilde{u}_2(\mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \in \mathcal{S}_{N_x}$ such that

$$\begin{cases} \operatorname{tr}\left(\mathbf{T}_1^{(i+1)}(\mathbf{A}^{(i)} - \mathbf{A}^{(i+1)})\right) \geq 0, \\ \operatorname{tr}\left(\mathbf{T}_2^{(i+1)}(\mathbf{P}^{(i)} - \mathbf{P}^{(i+1)})\right) \geq 0. \end{cases} \tag{57}$$

Moreover, by strong convexity of both $\tilde{u}_1$ and $\tilde{u}_2$,

$$\begin{cases} \tilde{u}_1(\mathbf{A}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \tilde{u}_1(\mathbf{A}^{(i+1)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \operatorname{tr}\left(\mathbf{T}_1^{(i+1)}(\mathbf{A}^{(i)} - \mathbf{A}^{(i+1)})\right) \\ \qquad\qquad + \frac{1}{2\theta_A}\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2, \\ \tilde{u}_2(\mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \geq \tilde{u}_2(\mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \operatorname{tr}\left(\mathbf{T}_2^{(i+1)}(\mathbf{P}^{(i)} - \mathbf{P}^{(i+1)})\right) \\ \qquad\qquad + \frac{1}{2\theta_P}\|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i+1)}\|_F^2. \end{cases} \tag{58}$$

Hence, using (57),

$$\begin{cases} \tilde{u}_1(\mathbf{A}^{(i)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \tilde{u}_1(\mathbf{A}^{(i+1)}; \mathbf{A}^{(i)}, \mathbf{P}^{(i)}) + \frac{1}{2\theta_A}\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2, \\ \tilde{u}_2(\mathbf{P}^{(i)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \geq \tilde{u}_2(\mathbf{P}^{(i+1)}; \mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \frac{1}{2\theta_P}\|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i+1)}\|_F^2, \end{cases} \tag{59}$$

and, using (50)-(51)-(52)-(53),

$$\begin{cases} \mathcal{L}(\mathbf{A}^{(i)}, \mathbf{P}^{(i)}) \geq \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) + \frac{1}{2\theta_A}\|\mathbf{A}^{(i+1)} - \mathbf{A}^{(i+1)}\|_F^2, \\ \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i)}) \geq \mathcal{L}(\mathbf{A}^{(i+1)}, \mathbf{P}^{(i+1)}) + \frac{1}{2\theta_P}\|\mathbf{P}^{(i+1)} - \mathbf{P}^{(i+1)}\|_F^2. \end{cases} \tag{60}$$

It means that the so-called NDSP (nearly sufficiently decreasing property) condition from (Hien et al., 2023) holds with, for every $i \in \mathbb{N}$, $(\gamma_1^{(i)}, \gamma_2^{(i)}, \eta_1^{(i)}, \eta_2^{(i)}) \equiv (0, 0, \theta_A^{-1}, \theta_P^{-1})$. Remark that our proof for (60) constitutes an alternative proof for Lemma 1.

- According to (Gupta and Mehra, 1974), function f is continuously differentiable. As $g_1 \in \Gamma_0(\mathbb{R}^{N_x \times N_x})$ and $g_2 \in \Gamma_0(\mathcal{S}_{N_x})$, we have for every $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$ and $\mathbf{P} \in \mathcal{S}_{N_x}$,

$$\begin{cases} \partial_A\left(f(\mathbf{A}, \mathbf{P}) + g_1(\mathbf{A})\right) = \nabla_A f(\mathbf{A}, \mathbf{P}) + \partial g_1(\mathbf{A}), \\ \partial_P\left(f(\mathbf{A}, \mathbf{P}) + g_2(\mathbf{P})\right) = \nabla_P f(\mathbf{A}, \mathbf{P}) + \partial g_2(\mathbf{P}), \end{cases} \tag{61}$$

so that (Hien et al., 2023, Assumption 3(i)) holds. Moreover, by construction of the majorizing function $\mathcal{J}$, it is also continuously differentiable and we have the coincidence of the gradient at the majorization point, namely for every $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$,

$$\nabla_A f(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = \nabla \tilde{u}_1(\widetilde{\mathbf{A}}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}), \tag{62}$$

$$\nabla_P f(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}) = \nabla \tilde{u}_2(\widetilde{\mathbf{P}}; \widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}). \tag{63}$$

Thus, according to (Hien et al., 2023, Lem.2), (Hien et al., 2023, Assumption 2) is verified. Moreover, as the restriction of $\mathcal{J}$ to both of its variables is quadratic, the partial gradients ∇u_1 and ∇u_2 are linear and thus Lipschitz continuous on any bounded subset of $\mathbb{R}^{N_x \times N_x}$ and $\mathcal{S}_{N_x}$, respectively, which yields the fulfillment of (Hien et al., 2023, Assumption 3).

• Since NSDP and (Hien et al., 2023, Assumption 2) hold, the result in (Hien et al., 2023, Theorem 6) allows to show that, if the sequence $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$ is bounded, then every limit point $(\mathbf{A}^*, \mathbf{P}^*)$ of it is a critical point of $\mathcal{L}$. Moreover, (Hien et al., 2023, Assumption 3) is satisfied and (Hien et al., 2023, Cond. 1) and (Hien et al., 2023, Cond. 4) are trivially met in our case. We can also show that the loss function $\mathcal{L}$ satisfies the Kurdyka-Lojasiewicz property (Bolte et al., 2014) (using a similar proof than (Chouzenoux et al., 2019, Lemma 2)). Thus, (Hien et al., 2023, Theorem 8) holds which concludes our proof.

■

4.3 Discussion

Kurdyka-Lojasiewicz inequality in non-convex optimization: The proof of Theorem 2 is grounded on the recent works by Hien et al. (2023, 2020), generalizing the works by Bolte et al. (2014); Chouzenoux et al. (2016), for establishing the convergence of block alternating MM schemes under the Kurdyka-Lojasiewicz (KL) inequality assumption (Lojasiewicz, 1963). The latter assumption is a powerful tool of nonlinear functional analysis that has been popularized in the seminal paper by Attouch et al. (2010). In their paper, the authors show how to prove the convergence of the iterates generated by a large family of minimization schemes, under the sole requirement that the function to minimize satisfies the KL inequality. The latter requirement is very mild, as it holds for a large class of functions, non necessarily convex, as soon as they can be embedded within an o-minimal structure. Semi-algebraic and analytical functions are examples of such functions. In our analysis, for the sake of conciseness, we skipped the proof showing that $\mathcal{L}$ satisfies KL as it is identical to the one of (Chouzenoux et al., 2019, Lemma 2).

Inner updates: Theorem 2 assumes that both inner steps (27) and (28) are solved in an exact manner. Due to strong convexity of $\mathcal{C}_1$ and $\mathcal{C}_2$, each problem has a unique solution, which ensures the well posedness of the formulation. However, the resolution of (27) and (28) does not take a closed form (except when λ_A or λ_P equals zero, in such case we retrieve the MLEM updates from (Sarkka, 2013, Chap. 12)). Iterative inner solvers are thus necessary, and we proposed some efficient ones in Section 3.5. The extension of our convergence study to the case of inexact resolution of (27) and (28) is not straightforward, up to our knowledge. One could exit Algorithm (4) (resp. Algorithm (5)) as soon as $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) decreases, which might be satisfied after only a few iterations. In such case, Lemma 1 holds and a weak convergence result can be deduced, namely the convergence of the loss function sequence, and the existence of cluster points for $\{\mathbf{A}^{(i)}, \mathbf{P}^{(i)}\}_{i\in\mathbb{N}}$. Convergence of iterates arising from an inexact form of DGLASSO is difficult to guarantee, due to the intricated form of function $\mathcal{L}$ (i.e., non-convex, non-Lipschitz smooth, function). Inexact proximal schemes for KL losses, with advanced stopping conditions, have been studied in various works, such as Attouch et al. (2013); Cherni et al. (2020); Chouzenoux et al. (2014); Zheng et al. (2023); Bonettini et al. (2018, 2021), to name a few. We are not aware of any study covering the block alternating MM scheme considered here, and thus left the convergence study of the inexact implementation of DGLASSO as future work. In practice, we impose a rather demanding condition on the stopping rule for the inner solvers of (27) and (28) (typically, $\xi = 10^{-3}$ with a maximum of 20000 iterations), and did not observe any numerical instabilities of the proposed algorithm.

5. Experiments

We perform a thorough evaluation study on various controlled scenarios where the ground truth matrices denoted $(\mathbf{A}^*, \mathbf{P}^*)$ (as well as $\mathbf{Q}^* = (\mathbf{P}^*)^{-1}$) are predefined, and the time series $\{\mathbf{y}_k, \mathbf{x}_k\}_{k=1}^K$ are built directly from the LG-SSM model (2)-(3) using such matrices. The goal is then to provide estimates $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}}, \widehat{\mathbf{Q}})$ of $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$, given the observation of $\{\mathbf{y}_k\}_{k=1}^K$. Except otherwise stated, all the compared methods have moreover access to a perfect knowledge of $(\mathbf{R}_k, \mathbf{H}_k, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, on top of accessing the time series $\{\mathbf{y}_k\}_{k=1}^K$. The hidden state $\{\mathbf{x}_k\}_{k=1}^K$ is, by definition, not assumed to be known, and is only used to compute quality metrics on test sets. We first work on synthetic data in Section 5.1, where the structure, the sparsity level, the conditioning, of the sought matrices $(\mathbf{A}^*, \mathbf{P}^*)$; are controlled. This allows us to evaluate our method on multiple cases, to discuss its parameter tuning, and to compare it to benchmarks in terms of inference quality and complexity. We then address in Section 5.2, a set of problems of graph inference arising in weather variability analysis, using four synthetic datasets built upon the Neurips CauseMe data challenge (Runge et al., 2020). This second set of experiments aims at evaluating the ability of DGLASSO to model and estimate a large class of graph structures (here, 200 different graphs per dataset), in comparison with other state-of-the-art graph detection methods.

All codes are run on a Desktop Dell Latitude computer, with 11th Gen Intel(R) Core(TM) i7-1185G7 at 3.00GHz, equipped with 32Go Ram, using Matlab R2021a software. In all the experiments, we set the precision parameters in DGLASSO algorithm to $(\xi, \varepsilon) = (10^{-3}, 10^{-3})$, with a maximum number of 50 iterations for the outer loop, and 20000 iterations for the inner solvers. Similar stopping conditions are used for the benchmarks of competitors for ensuring fair comparisons. Moreover, the DGLASSO stepsize parameters are set to $(\theta_A, \theta_P) = (1, 1)$, as this simple choice led to good performance in our experiments. The code is publicly available, for reproducibility purpose.[6]

5.1 Controlled data

5.1.1 Datasets

We set $K = 10^3$, $\mathbf{R}_k = \sigma_{\mathbf{R}}^2 \mathbf{Id}_{N_y}$ for every $k \in \{1, \ldots, K\}$, $\boldsymbol{\mu}_0 \in \mathbb{R}^{N_x}$ is a vector of ones, $\boldsymbol{\Sigma}_0 = \sigma_0^2 \mathbf{Id}_{N_x}$ with $(\sigma_{\mathbf{R}}, \sigma_0) = (10^{-1}, 10^{-4})$. Matrix $\mathbf{H}_k$ is set to identity matrix for every $k \in \{1, \ldots, K\}$, so that $N_x = N_y$. This setting models a one-to-one correspondence between states and observations. Identifiability issues are hence avoided.

We set $N_x = N_y = 9$, and we rely on block-diagonal matrices $(\mathbf{A}^*, \mathbf{P}^*)$, made of 3 blocks with dimensions 3×3. Matrices $\mathbf{A}^*$ are randomly set as

$$\mathbf{A}^* = \mathbf{U}\mathrm{Diag}(\max(\boldsymbol{\lambda}, 0.99))\mathbf{U}^\top, \tag{64}$$

where $(\mathbf{U}, \boldsymbol{\lambda})$ are obtained by the singular value decomposition

$$\mathbf{U}\mathrm{Diag}(\boldsymbol{\lambda})\mathbf{U}^\top = \mathrm{BDiag}\{(B_j)_{1 \leq j \leq 3}\}, \tag{65}$$

[6] https://pages.saclay.inria.fr/emilie.chouzenoux/Logiciel.html

with

$$(\forall (j,n,\ell) \in \{1,2,3\}^3) \quad B_j(n,\ell) = \rho_j^{|\sigma_j(n)-\ell|}. \tag{66}$$

Equation (66) builds matrices of auto-regressive processes of order one. $\text{BDiag}\{(B_j)_{1\leq j\leq 3}\}$ denotes the block diagonal matrix formed with the $(B_j)_{1\leq j\leq 3}$ matrices, ρ_j is a scalar uniformly sampled into $[0,1]$, and σ_j is a random bijective permutation from $\{1,2,3\}$ to $\{1,2,3\}$. The latter permutation allows to break the symmetry property of block elements of $\mathbf{A}^*$. The capping operation in Equation (64) implements a spectral projection (Benfenati et al., 2020), so that the spectral norm of $\mathbf{A}^*$ is lower or equal than 0.99, and, as such, ensures the stability of the resulting SSM.

The diagonal blocks of $\mathbf{P}^* = \text{BDiag}\{(\Omega_j)_{1\leq j\leq 3}\}$ are randomly set following the procedure from Moré and Toraldo (1989), described as

$$(\forall j \in \{1,2,3\}) \quad \Omega_j = \left(\mathbf{Id}_3 - 2\frac{\mathbf{p}_j\mathbf{p}_j^\top}{\|\mathbf{p}_j\|}\right) \text{Diag}\left\{(c^{(n-1)/2})_{1\leq n\leq 3}\right\} \left(\mathbf{Id}_3 - 2\frac{\mathbf{p}_j\mathbf{p}_j^\top}{\|\mathbf{p}_j\|}\right), \tag{67}$$

where each $\mathbf{p}_j \in \mathbb{R}^3$ has uniformly sampled entries in $[-1,1]$. Parameter $c > 0$ controls the conditioning number of matrices $(\Omega_j)_{1\leq j\leq 3}$, and thus the one of $\mathbf{P}^*$. We set $\log_{10}(c) \in \{0.1, 0.2, 0.5, 1\}$, leading to datasets A, B, C, and D, respectively.

DGLASSO provides estimates $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$ as a direct output. The estimate $\widehat{\mathbf{Q}}$ is simply deduced as $\widehat{\mathbf{Q}} = (\widehat{\mathbf{P}})^{-1}$. We initialize DGLASSO with $\mathbf{P}^{(0)} = 10^{-1}\mathbf{Id}_{N_x}$, and $\mathbf{A}^{(0)}$ equal to a stable auto-regressive order one matrix with entries $A^{(0)}(n,m) = (10^{-1})^{|n-m|}$ projected onto the set of matrices with spectral norm equal to 0.99. The setting of regularization parameters (λ_A, λ_P) is discussed in the dedicated Section 5.1.4. Performance of the method for varying initializations is discussed in Appendix C.1.

5.1.2 Comparisons to other methods

In our experiments, we also compare DGLASSO with other model inference techniques for LG-SSM.

First, we consider the EM method from Shumway and Stoffer (1982) (denoted after by MLEM) to compute $(\widehat{\mathbf{A}}, \widehat{\mathbf{Q}})$ as maximum likelihood estimates (i.e., no regularization is employed) of matrices $(\mathbf{A}^*, \mathbf{Q}^*)$, the estimation $\widehat{\mathbf{P}}$ being defined here as the inverse of $\widehat{\mathbf{Q}}$. Note that no inner loop is required in the MLEM procedure, as solutions to the M-step have a closed form expression (Shumway and Stoffer, 1982).

Second, we consider three model inference techniques that explicitly incorporate a sparse graphical prior knowledge on the sought matrices. Namely, we compare to GLASSO (Friedman et al., 2008), that considers a simple static and noiseless version of the LG-SSM (i.e., $\widehat{\mathbf{A}}$ is empirically set to zero and observation noise is neglected). Matrix $\widehat{\mathbf{P}}$ is then obtained through a maximum a posteriori formulation under an ℓ_1 prior. The convex GLASSO loss is minimized using the proximal splitting method described in (Benfenati et al., 2020, Algorithm 1). Matrix $\widehat{\mathbf{Q}}$ is deduced by inversion of the resulting $\widehat{\mathbf{P}}$. We also compare with the robust GLASSO (rGLASSO) approach introduced in Benfenati et al. (2020), that explicitly accounts for the expression of $\mathbf{R}_k$ in the maximum a posteriori loss expression. For the sake of fair comparison, we use hereagain an ℓ_1 penalty to obtain $\widehat{\mathbf{P}}$ although more sophisticated priors could be encompassed by rGLASSO. For both aforementioned methods, we rely on the Matlab toolbox provided by the authors.[7] Finally, we provide the results obtained with the GRAPHEM method we recently introduced in Elvira and Chouzenoux (2022). GRAPHEM provides a maximum a posteriori estimate $\widehat{\mathbf{A}}$ under an ℓ_1 prior, while $\widehat{\mathbf{Q}}$ is empirically set to $\sigma_{\mathbf{Q}}^2 \mathbf{Id}_{N_y}$, with a finetuned $\sigma_{\mathbf{Q}} > 0$. The Matlab toolbox provided by the authors[8] is used to produce the results for this method.

Third, we compare DGLASSO with a quasi-Newton approach also aiming at minimizing (21). The log-likelihood gradient is computed using the sensitivity equations from Nagakura (2021), while subgradient updates are used to handle the Lasso penalties. The chain rule from Petersen and Pedersen (2012) is used to build gradients with respect to $\mathbf{P} = \mathbf{Q}^{-1}$. The minimization is performed using a Quasi-Newton routine from the Optimization Toolbox from Matlab, with default options and a maximum of 100 iterations.

All the comparative methods are programmed in the same language, they are initialized using a similar strategy as our DGLASSO method, and similar stopping criteria and hyper-parameter tuning approach are employed, for fair comparisons.

5.1.3 Evaluation metrics

We first evaluate the results of our method, as well as the comparative benchmarks, through quality assessment metrics. Namely, we use the relative mean square error (RMSE) between the ground truth matrices $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$ and the estimated $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}}, \widehat{\mathbf{Q}})$ (when available). For instance,

$$\operatorname{RMSE}(\mathbf{A}^*, \widehat{\mathbf{A}}) = \frac{\|\mathbf{A}^* - \widehat{\mathbf{A}}\|_{\mathrm{F}}^2}{\|\mathbf{A}^*\|_{\mathrm{F}}^2}. \tag{68}$$

$\operatorname{RMSE}(\mathbf{P}^*, \widehat{\mathbf{P}})$ and $\operatorname{RMSE}(\mathbf{Q}^*, \widehat{\mathbf{Q}})$ are defined in a similar fashion. We also compute area-under-curve (AUC) and F1 score comparing the non-zero entries (that is, the graph edges positions) of the sparse matrices $(\mathbf{A}^*, \mathbf{P}^*)$ and their estimates $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$. The absolute value of the entry of each matrix is used as a detection threshold for the AUC score. A threshold value of 10^{-10} is used for the detection hypothesis for the F1 score (i.e., a weight greater than 10^{-10}, in absolute value, is considered as an edge). We furthermore evaluate the ability of the estimated model parameters to actually describe and infer the time series (both observed and hidden states). To that end, we build test time series $(\mathbf{x}^{\text{test}}, \mathbf{y}^{\text{test}})$, not seen by the algorithms, constructed by running the ground truth LG-SSM (i.e., with ground truth matrix parameters $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$). We then run KF and RTS algorithms 1 and 2, respectively, using either the ground truth matrices $(\mathbf{A}^*, \mathbf{P}^*, \mathbf{Q}^*)$ or their estimations $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}}, \widehat{\mathbf{Q}})$, to build, for every $k \in \{1, \dots, K\}$, the predictive distribution means $(\boldsymbol{\mu}_k^*, \boldsymbol{\nu}_k^*, \boldsymbol{\mu}_k^{\mathrm{s}*})$ and $(\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\nu}}_k, \widehat{\boldsymbol{\mu}}_k^{\mathrm{s}})$, respectively.

This allows in particular to compute the cumulative normalized mean squared error (cNMSE) between the predictive distribution means using either the ground truth model matrices or the estimated ones. Namely, we calculate

$$\operatorname{cNMSE}(\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}}) = \frac{\sum_{k=1}^K \|\boldsymbol{\nu}_k^* - \widehat{\boldsymbol{\nu}}\|^2}{\sum_{k=1}^K \|\boldsymbol{\nu}_k^*\|^2}, \tag{69}$$

[7] http://www-syscom.univ-mlv.fr/ benfenat/Software.html

[8] https://pages.saclay.inria.fr/emilie.chouzenoux/LogicielEN.html

as well as cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$, and cNMSE$(\boldsymbol{\mu}^{\mathrm{s}*}, \widehat{\boldsymbol{\mu}}^{\mathrm{s}})$.

Finally, we evaluate the negative logarithm of the marginal likelihood $\mathcal{L}_{1:K}(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$ as defined in (22), on the test time series.

5.1.4 Influence of regularization parameters setting

We here assess the influence on the results of the setting of the DGLASSO hyper-parameters (λ_A, λ_P), accounting for the sparsity priors on $\mathbf{A}$ and $\mathbf{P}$, respectively. To that aim, we ran DGLASSO for 100 values of hyperparameters (λ_A, λ_P), regularly spaced on a log-scale grid between 1 and 10^2, and repeated the experience on 50 randomly generated time series, for dataset A. We display in Figure 4 the averaged values, over the random runs, of several quantitative metrics, as a function of hyperparameters (λ_A, λ_P). We also report in the caption the averaged RMSE scores obtained when running DGLASSO with $(\lambda_A, \lambda_P) = (0, 0)$ (i.e., MLEM result). As it can be observed, both F1/RMSE for the estimation of the transition matrix $\mathbf{A}$ are mostly governed by the value of λ_A, while the quality scores for the state noise precision matrix $\mathbf{P}$ are mostly influenced by λ_P. Note that the RMSE scores on $\mathbf{Q}$, not shown here, follow similar evolution than RMSE on $\mathbf{P}$. Just inspecting F1/RMSE appears not informative enough to set parameters (λ_A, λ_P), as each metric and each parameter seems to push towards a different goal. The maps of cNMSE$(\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}})$ and of the marginal likelihood log-loss $\mathcal{L}_{1:K}$ show very similar behavior. Note that the later is a practically interesting metric, because it does not require the knowledge of the ground truth matrices. On this example, it however appears not discriminative enough. Typically, it stays almost constant for a (too) wide value range of (λ_A, λ_P), which confirms again the ill-posedness of the minimization problem. The maps for cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$ and cNMSE$(\boldsymbol{\mu}^{\mathrm{s}*}, \widehat{\boldsymbol{\mu}}^{\mathrm{s}})$ are very similar. Interestingly, the minimization of both these quantities, related to the state mean distributions, seems to provide a meaningful value range for the regularization parameters, narrowing down around values that achieve an optimal compromise between (i) good estimation of the sought matrices, (ii) good estimation of the sparse matrices support, and (iii) good predictive behavior for time series inference by KF/RTS techniques. Similar conclusions were reached for the other three datasets. Note additionally that the DGLASSO results outperform for a wide range of (λ_A, λ_P) those obtained with MLEM, confirming the advantage of introducing the proposed sparsity prior. This will be more deeply examined in an upcoming section.

In our forthcoming experiments, we opt for simple procedure to set the hyperparameters, depending on the task of interest. Namely, in this Section 5.1, we aim at an optimal compromise between both graph estimation, that reaches the best predictive power of the learned models, when evaluated on an unseen time series. This leads us to fix (λ_A, λ_P) through a rough grid search among $\{1, 5, 8, 10\}^2$ to minimize cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$ averaged on few (typically 5) runs.

5.1.5 Influence of conditioning number

We report in Table 1 the results, averaged over 50 realizations of the time series generation. We do not provide the scores for the estimation of $\mathbf{A}$ (resp. $(\mathbf{P}, \mathbf{Q})$) for GLASSO/rGLASSO (resp. GRAPHEM), as those methods are not designed for such task. The general trend in the results is a slight decrease of the quality of estimation for all methods, when c_Q increases

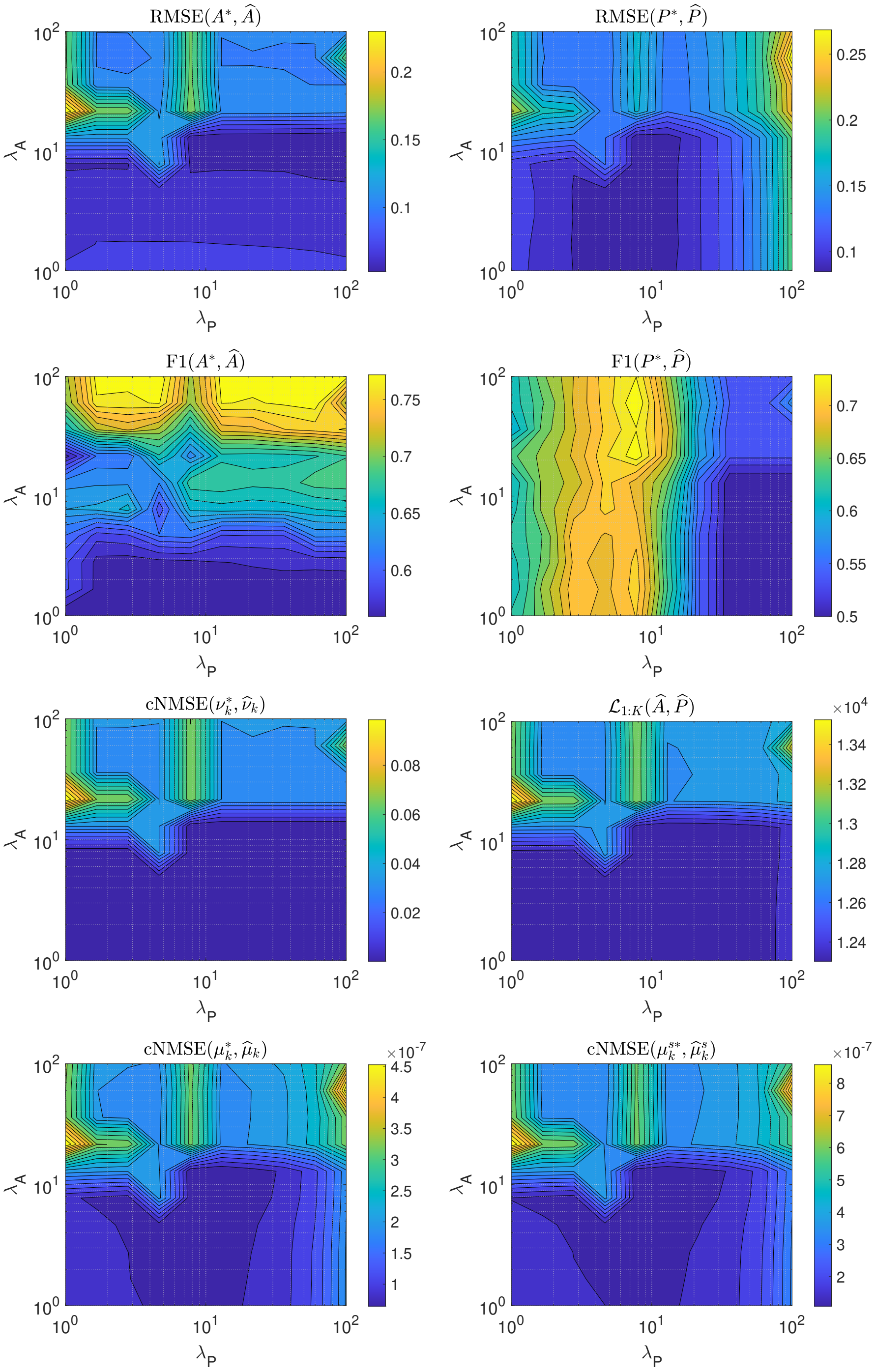


Figure 4: Evolution of RMSE, F1, cNMSE and loss scores on the estimation of $\mathbf{A}$ (left) and $\mathbf{P}$ (right) by DGLASSO, as a function of hyperparameters (λ_A, λ_P), for dataset A (averaged on 10 runs). As a comparison, the averaged RMSE scores for $(\lambda_A, \lambda_P) = (0, 0)$ (i.e., MLEM) on this example were $(0.077, 0.106)$ for $(\mathbf{A}, \mathbf{P})$, respectively.

(i.e., from dataset A to D). This is expected, as an ill-conditioned matrix $\mathbf{Q}$ complicates the mathematical structure of the likelihood term.

Regarding the estimation of $\mathbf{A}$, GRAPHEM method presents the best results for the three considered metrics. DGLASSO is second best, while MLEM follows. As already stated, GLASSO/rGLASSO do not estimate $\mathbf{A}$ (i.e., they assume this matrix to be zero, that is the process is i.i.d. without any Markov structure). Regarding the estimation of $(\mathbf{P}, \mathbf{Q})$, DGLASSO is outperforming the benchmarks in terms of RMSE score. The second best, in terms of RMSE is MLEM. GLASSO and rGLASSO got very bad RMSE scores, probably because of the model mismatch induced by assuming $\mathbf{A}$ to be zero. The edge detection performance of DGLASSO are excellent, except in few cases where rGLASSO gets slightly better results. MLEM gets poorer results than DGLASSO in all metrics. In particular it exhibits a bad F1 score, as it does not include sparse prior, and thus does not succeed to eliminate any spurious edges.

Regarding the distribution mean calculation, it is remarkable that DGLASSO outperforms in all examples the benchmarks, which shows that the proposed formulation allows to provide model parameters that are best suited for inference using KF/RTS at test phase. The marginal likelihood log-loss computed on test set is also minimized with the quantities provided by our method. This could appear as counter-intuitive, as the method is not specifically designed to minimize this loss (while MLEM explicitly aims at minimizing this loss on the train set). The advantage of DGLASSO is that it accounts for prior knowledge, making the inferred model more robust, and less prone to overfitting, which is translated into better behavior on an independent test time series.

We display in Figure 5 box plots assessing the variability to the RMSE and F1 scores, for both MLEM and DGLASSO methods, on 50 runs on dataset A and dataset D. The RMSE values are in most runs lower (i.e., better) for the proposed method. Both methods are quite stable with respect to the time series generation, as the plots show few outliers. For dataset D, corresponding to a more challenging case with an ill-conditioned matrix $\mathbf{P}^*$, the results are slightly more spread, especially for the metrics related to the recovery quality of this matrix. Remark that the F1 scores of MLEM are constant, and are equal to 0.5. As already pointed out, this is expected as MLEM is not designed to perform an edge detection task.

We refer the reader to Appendix C.2 for extra experiments on the synthetic datasets, assessing the performance of DGLASSO when the sparsity level of $\mathbf{A}^*$ increases.

5.1.6 Complexity analysis

We now examine the time complexity of the method, as well as of MLEM, when processing dataset A, for various values of the time series length K, namely $K \in \{100, 200, 500, 1000, 2000, 5000\}$ (we recall that our previous experiments were all done with $K = 1000$). We display in Figure 6(left) the computing time in seconds for computing $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$, for DGLASSO and MLEM, averaged over 50 realizations. We also display, in Figure 6(middle) for the same experiment, the RMSE between $\mathbf{A}^*$ and $\widehat{\mathbf{A}}$, and in Figure 6(right) the metric cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$. One can notice that our method is slightly more demanding in terms of computational time, requiring about twice the time compared to MLEM in order to reach convergence. But both DGLASSO and MLEM scale similarly. Moreover, as already observed in our previous ex-

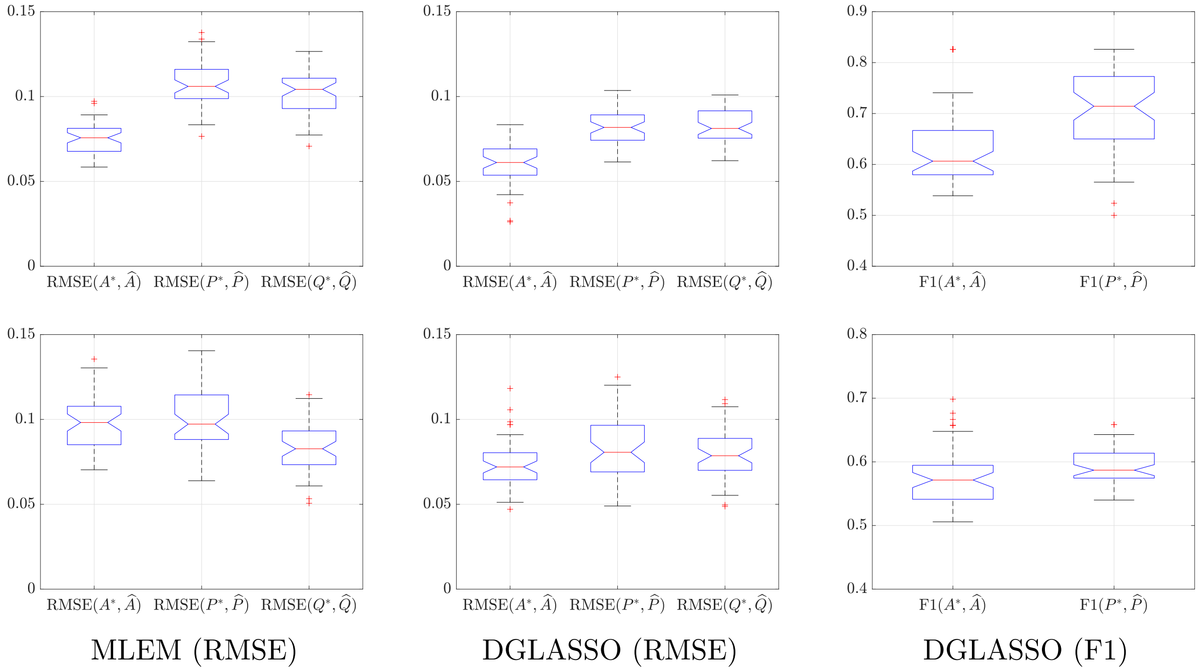


Figure 5: Box plots for quantitative metrics when running MLEM, and DGLASSO, on 50 randomly generated LG-SSM time series, for dataset A (top) and dataset D (bottom). F1 score is not reported for MLEM, as this method does not perform edge detection, resulting in a constant F1 score around 0.5. DGLASSO outperforms MLEM with better (i.e., lower) RMSE scores for most runs and good F1 scores. Dataset D is more challenging in terms of inference, thus yielding to more spread results for both methods.

Table 1: Results for the four considered datasets A to D, with an increasing conditioning number of $\mathbf{P}^*$ equal to $\log_{10}(c) \in \{0.1, 0.2, 0.1, 1\}$, respectively. We evaluate the methods in terms of estimation quality for $(\mathbf{A}, \mathbf{P}, \mathbf{Q})$, using either RMSE as defined in (68), and edge detection scores (AUC, F1), as well as in terms of inference quality on test set using cNMSE and marginal likelihood metrics defined in (69).

		Estimation of **A**			Estimation of **P**			Estim. **Q**	State distrib.		Predictive distrib.	
	Method	RMSE	AUC	F1	RMSE	AUC	F1	RMSE	cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$	cNMSE$(\boldsymbol{\mu}^{s*}, \widehat{\boldsymbol{\mu}}^{s})$	cNMSE$(\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}})$	$\mathcal{L}_{1:K}(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$
Dataset A	DGLASSO	0.061	0.843	**0.641**	**0.082**	0.778	0.698	**0.083**	$\mathbf{6.394 \times 10^{-8}}$	$\mathbf{1.050 \times 10^{-7}}$	$\mathbf{2.984 \times 10^{-4}}$	**12 307.169**
	MLEM	0.076	0.817	0.500	0.105	0.857	0.500	0.102	1.095×10^{-7}	1.803×10^{-7}	4.843×10^{-4}	12 341.205
	GLASSO	NA	NA	NA	0.818	0.804	0.496	1 073.510	4.485×10^{-6}	7.180×10^{-6}	1.000	28 459.294
	rGLASSO	NA	NA	NA	0.764	**0.924**	0.598	31.689	2.826×10^{-6}	5.492×10^{-6}	1.000	22 957.693
	GRAPHEM	**0.045**	**0.895**	**0.847**	NA	NA	NA	NA	4.364×10^{-6}	6.944×10^{-6}	2.980×10^{-4}	29 035.030
Dataset B	DGLASSO	0.068	0.833	0.603	**0.070**	0.893	**0.835**	**0.071**	$\mathbf{7.490 \times 10^{-8}}$	$\mathbf{1.236 \times 10^{-7}}$	$\mathbf{3.281 \times 10^{-4}}$	**11 806.744**
	MLEM	0.080	0.815	0.500	0.106	0.898	0.500	0.100	1.299×10^{-7}	2.133×10^{-7}	4.619×10^{-4}	11 833.448
	GLASSO	NA	NA	NA	0.827	0.826	0.505	341.873	5.069×10^{-6}	8.072×10^{-6}	1.000	27 744.964
	rGLASSO	NA	NA	NA	0.734	**0.930**	0.608	33.896	3.215×10^{-6}	6.187×10^{-6}	1.000	22 530.036
	GRAPHEM	**0.047**	**0.893**	**0.848**	NA	NA	NA	NA	5.158×10^{-6}	8.036×10^{-6}	2.912×10^{-4}	29 031.412
Dataset C	DGLASSO	0.070	0.829	0.581	**0.090**	0.954	**0.830**	**0.078**	$\mathbf{1.896 \times 10^{-7}}$	$\mathbf{2.994 \times 10^{-7}}$	$\mathbf{3.956 \times 10^{-4}}$	**10 311.104**
	MLEM	0.081	0.810	0.500	0.097	**0.974**	0.500	0.094	2.583×10^{-7}	4.180×10^{-7}	5.053×10^{-4}	10 326.410
	GLASSO	NA	NA	NA	0.901	0.805	0.489	3.926×10^{17}	0.012	0.012	1.000	26 634.892
	rGLASSO	NA	NA	NA	0.805	0.928	0.614	29.530	7.195×10^{-6}	1.320×10^{-5}	1.000	21 322.247
	GRAPHEM	**0.049**	**0.892**	**0.857**	NA	NA	NA	NA	1.055×10^{-5}	1.641×10^{-5}	3.912×10^{-4}	29 023.369
Dataset D	DGLASSO	0.073	0.835	0.575	**0.083**	**1.000**	0.598	**0.080**	$\mathbf{5.127 \times 10^{-7}}$	$\mathbf{8.243 \times 10^{-7}}$	$\mathbf{3.373 \times 10^{-4}}$	**7 911.943**
	MLEM	0.098	0.808	0.500	0.095	**1.000**	0.500	0.084	6.296×10^{-7}	1.027×10^{-6}	4.219×10^{-4}	7 923.850
	GLASSO	NA	NA	NA	0.964	0.941	0.550	187.823	2.348×10^{-5}	3.701×10^{-5}	1.000	23 684.178
	rGLASSO	NA	NA	NA	0.882	0.956	**0.645**	28.703	1.886×10^{-5}	3.239×10^{-5}	1.000	20 100.491
	GRAPHEM	**0.061**	**0.892**	**0.864**	NA	NA	NA	NA	2.503×10^{-5}	3.839×10^{-5}	3.743×10^{-4}	29 016.321

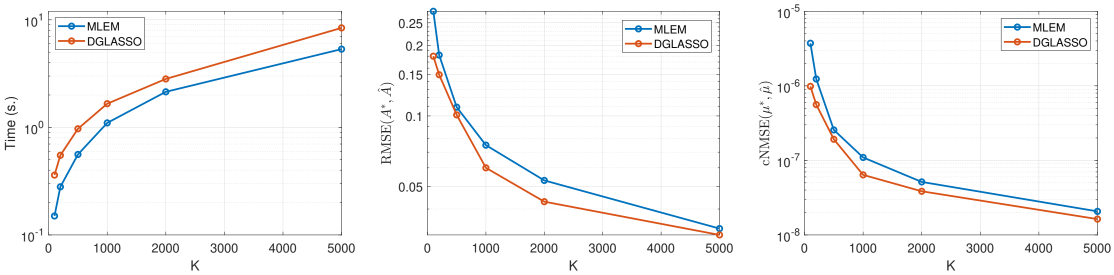
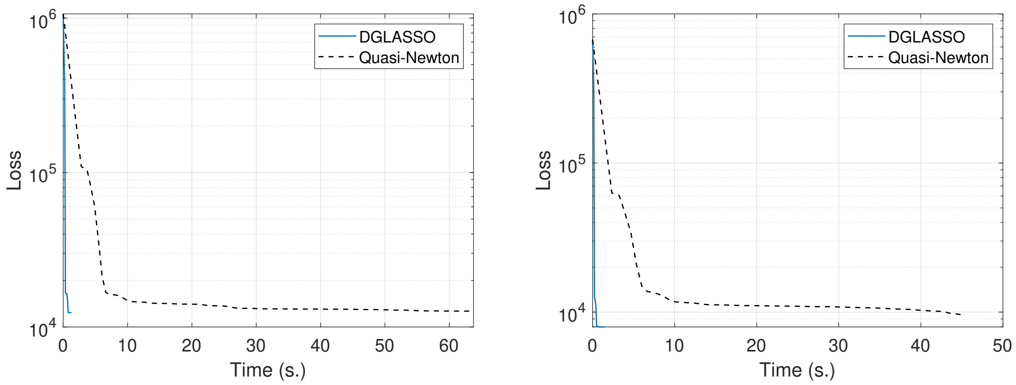


Figure 6: Evolution of the complexity time (left), RMSE$(\mathbf{A}^*, \widehat{\mathbf{A}})$ (middle) and cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$ (right) metrics, as a function of the time series length K, for experiments on dataset A averaged over 50 runs. Both methods scale similarly, with DGLASSO requiring about twice the time to run than MLEM.

periments, DGLASSO outperforms MLEM on both metrics shown here, in all tested values of K. As expected, the results are better for higher values of K, at the price of an increased computational time. Interestingly, the regularization still yields improved results for very large K.

We end the complexity study by showing a comparison between DGLASSO and a Quasi-Newton (Q-N) implementation to solve the same minimization problem. Figure 7 displays, for both methods, the evolution of the loss function along time, for examples from Datasets A and D. The regularization hyperparameters are set the same for both algorithms to ensure a fair comparison. One can notice that both algorithms reach a similar value of the loss at convergence. DGLASSO requires very few iterations (typically, less than 10) and short time



DGLASSO:	$\mathrm{RMSE}(\mathbf{A}^*, \widehat{\mathbf{A}}) = 0.070$, $\mathrm{RMSE}(\mathbf{P}^*, \widehat{\mathbf{P}}) = 0.081766$	$\mathrm{RMSE}(\mathbf{A}^*, \widehat{\mathbf{A}}) = 0.101$, $\mathrm{RMSE}(\mathbf{P}^*, \widehat{\mathbf{P}}) = 0.082$
Q-N:	$\mathrm{RMSE}(\mathbf{A}^*, \widehat{\mathbf{A}}) = 0.077$, $\mathrm{RMSE}(\mathbf{P}^*, \widehat{\mathbf{P}}) = 0.3015$	$\mathrm{RMSE}(\mathbf{A}^*, \widehat{\mathbf{A}}) = 0.359$, $\mathrm{RMSE}(\mathbf{P}^*, \widehat{\mathbf{P}}) = 0.809$

Figure 7: Loss evolution versus computational time in seconds, and RMSE scores, for DGLASSO and Quasi-Newton algorithms, running on an example from Dataset A (left) and Dataset D (right).

to stabilize, while Q-N displays a slower convergence profile despite its quasi-Newton (i.e., second-order) acceleration strategy. In terms of qualitative results, both algorithms reach a similar RMSE for the estimation of matrix $\mathbf{A}$ on Dataset A, while DGLASSO outperforms Q-N on Dataset D. In both scenarios, the estimation of $\mathbf{P}$ is considerably more accurate with the proposed DGLASSO. We explain these differences in terms of convergence speed and inference quality, by the gradient singularity of the loss with respect to variable $\mathbf{P}$, leading to numerical instabilities of the Q-N approach. This is especially the case for Dataset D, characterized by a poorly conditioned $\mathbf{P}^*$.

5.2 Synthetic weather data

5.2.1 Experimental settings

We now evaluate our method on synthetic datasets arising from causal graph discovery studies in the field of weather variability tracking. Specifically, we consider two sets of 200 sparse matrices $\mathbf{A}^* \in \mathbb{R}^{N_x}$, with $N_x = 5$ or 10 respectively, representing the ground truth causal graphs used to produce `WEATH` datasets in the Neurips 2019 data challenge (Runge et al., 2020).[9] For each $\mathbf{A}^*$, we create times series $(\mathbf{x}_k, \mathbf{y}_k)_{k=1}^K$ using (2)-(3), with $K = 10^3$, $\mathbf{H}^* = \mathbf{Id}_{N_x}$ (i.e., $N_y = N_x$), and $(\sigma_\mathbf{R}, \sigma_0) = (10^{-1}, 10^{-4})$. We set $\mathbf{Q}^*$ as a block diagonal matrix of J blocks with dimensions $(B_j)_{1 \leq j \leq J}$, with $\sum_{j=1}^J B_j = N_x$. Here, we consider two settings for the conditioning of $\mathbf{Q}^*$, namely one with a condition number close to one (i.e., $\mathbf{Q}^*$ is close to identity matrix) and another with high condition number. The main characteristics of the datasets and their names are summarized in Table 2.

We evaluate our results in terms on quality assessment metrics of the estimated $\widehat{\mathbf{A}}$ when compared to its ground truth $\mathbf{A}^*$, as this is the quantity of interest for these datasets. We compute $\mathrm{RMSE}(\widehat{\mathbf{A}}, \mathbf{A}^*)$, as well as the precision, recall, specificity, accuracy, and F1

[9] https://causeme.uv.es/static/datasets/TestWEATH/

score for detecting the non-zero entries of the transition matrix (that is, the graph edges positions). A threshold value of 10^{-10} on the absolute entries is used for the detection hypothesis.

As for comparison, we also provide the results obtained with MLEM and GRAPHEM approaches, as in our previous experiments. The same stopping criterion than in our previous set of experiments has been used. Since the datasets are synthetic, the ground truth is available. The hyperparameters (λ_A, λ_P) for DGLASSO, and λ_A for GRAPHEM, are finetuned on the rough grid $\{1, 5, 10\}$, to minimize $\text{RMSE}(\widehat{\mathbf{A}}, \mathbf{A}^*)$, on one example randomly chosen, per each dataset. We then keep the parameters fixed for all the dataset, to limit overfitting behavior. In addition to these EM-based methods, we provide comparisons with two Granger-causality approaches for graphical modeling, namely pairwise Granger Causality (PGC) and conditional Granger Causality (CGC). Those approaches are based on sparse vector autoregressive (VAR) models. We allow the order of the VAR process to be up to $p = 11$, CGC is run with a maximum distance of 5 causal links, and in each experiment we display the best results (in F1 score) for the statistical tests performed with the significance level $\alpha \in \{0.01, 0.05, 0.1\}$ (see more details in Luengo et al. (2019)). As PGC and CGC do not provide a weighted graph estimation, no RMSE score is computed in those cases.

5.2.2 Results

We summarize our results in Table 3. We also report the averaged inference time for all methods. Remarkably, the proposed method outperforms all benchmarks in all the considered metrics, related to the inference of the graph weights (RMSE metric) and the edge detection task (binary classification metrics). As expected, the result quality tends to slightly degrade when a more complicated structure is assumed for $\mathbf{Q}^*$ (see, for instance, WeathN5a versus WeathN5b), and when the dataset size increases (see, for instance, WeathN5a versus WeathN10a). MLEM has very poor edge detection metrics, since it does not exploit any sparsity prior on the sought matrix. GRAPHEM has better behavior, but, in these datasets, it stays way behind the quality of DGLASSO, by all metrics (which was not the case for the controlled synthetic data). This shows that our method really makes the change when dealing with complex graphs and correlated noise in the observations. In terms of computational time, the proposed method has similar complexity than the two other EM-based approaches. The results of CGC and PGC are satisfactory in the first two examples although the binary metrics are far from the performance of DGLASSO. PGC still gives meaningful results in the last two examples, but CGC has a low performance due to a high number of false negatives (i.e., many existing links are not discovered). We also note that while PGC and CGC have significantly lower running times for the case with $N_y = 5$, the computational cost for $N_y = 10$ is closer to DGLASSO while the performance gap is higher. Thus, in this examples we show that as the dimension grows, the computational cost of all methods is comparable while the proposed method has clearly a better performance. We also display some examples of inferred graphs in Figures 8 and 9. We can observe that DGLASSO is able to capture in a very accurate manner the structure of the graphs, despite the wide variability of the dataset. MLEM and GRAPHEM capture the main edges, but

Dataset	$N_x = N_y$	$(B_j)_{1\leq j\leq J}$	$\log_{10}(\text{cond}(\mathbf{Q}^*))$
WeathN5a	5	$(2, 3)$	0.1
WeathN5b	5	$(2, 3)$	1
WeathN10a	10	$(5, 5)$	0.1
WeathN10b	10	$(5, 5)$	1

Table 2: Details about the datasets. Each dataset is associated with 200 examples for matrix $\mathbf{A}^*$.

Table 3: Results for climate datasets along with computing times. All the metrics are averaged over the 200 examples of the dataset.

	method	RMSE	accur.	prec.	recall	spec.	F1	Time (s.)
WeathN5a	DGLASSO	**0.108**	**0.937**	0.894	0.998	0.894	**0.937**	0.608
	MLEM	0.140	0.413	0.413	**1.000**	0.000	0.584	0.596
	GRAPHEM	0.127	0.703	0.595	**1.000**	0.496	0.742	0.606
	PGC	-	0.772	**0.902**	0.515	**0.953**	0.652	0.019
	CGC	-	0.672	0.828	0.285	0.945	0.415	0.026
WeathN5b	DGLASSO	**0.166**	**0.773**	0.668	0.992	0.619	**0.788**	0.630
	MLEM	0.197	0.413	0.413	**1.000**	0.000	0.584	0.376
	GRAPHEM	0.186	0.629	0.536	**1.000**	0.368	0.694	0.470
	PGC	-	0.675	**0.677**	0.469	0.819	0.544	0.017
	CGC	-	0.634	0.659	0.263	**0.895**	0.369	0.023
WeathN10a	DGLASSO	**0.202**	**0.948**	**0.898**	0.925	0.954	**0.890**	1.363
	MLEM	0.264	0.219	0.219	**1.000**	0.000	0.359	0.834
	GRAPHEM	0.224	0.511	0.311	1.000	0.374	0.473	1.445
	PGC	-	0.879	0.904	0.504	**0.983**	0.644	0.232
	CGC	-	0.773	0.539	0.211	0.932	0.278	0.358
WeathN10b	DGLASSO	**0.192**	**0.866**	**0.633**	0.994	0.829	**0.769**	0.557
	MLEM	0.342	0.219	0.219	**1.000**	0.000	0.359	0.989
	GRAPHEM	0.219	0.855	0.620	0.994	0.816	0.757	0.655
	PGC	-	0.799	0.558	0.473	0.890	0.506	0.154
	CGC	-	0.750	0.407	0.218	**0.900**	0.265	0.178

their graphs are perturbed by several spurious edges, which shows how important it is to adopt a joint graph modeling, with sparsity priors on each.

6. Conclusion

This paper proposes a joint graphical modeling approach that incorporates a graphical perspective on the dynamics of the hidden state of a linear Gaussian state-space model. In particular, we propose a joint approach that considers a sparse undirected graph as a prior on the precision matrix of the hidden state noise and a sparse directed graph for the transition matrix that models the state dynamics. By bridging the gap between the static graphical Lasso model and the dynamic state-space model, we provide a novel comprehensive framework for interpretable inference with time-series data. The presented inference method, based on an efficient block alternating majorization-minimization algorithm, enables simultaneous estimation of both graphs and the construction of the filtering/smoothing distri-

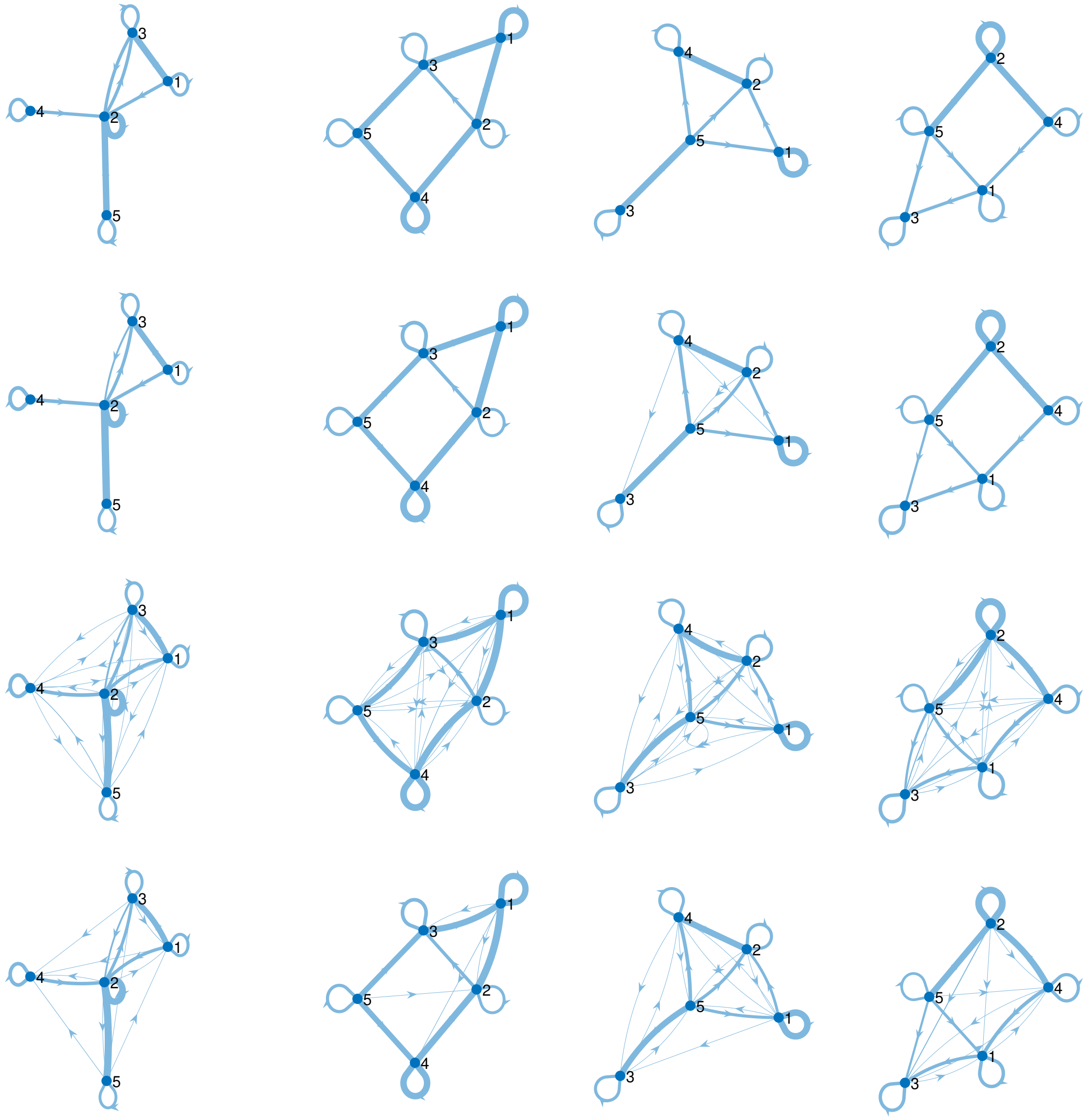


Figure 8: Graph inference results on 4 examples extracted from the dataset *WeathN5a*. From top to bottom: Original graph representation of $\mathbf{A}^*$, and of its estimation $\widehat{\mathbf{A}}$, using DGLASSO, MLEM, GRAPHEM, respectively.

bution for the time series. The established convergence of our algorithm, departing from recent nonlinear analysis tools, enhances the reliability and practicality of our approach. Through extensive experimental validation on synthetic data, we have demonstrated the effectiveness and potential of our proposed model and inference algorithm. The results showcase its ability to uncover meaningful insights from time-series data, contributing not only to better forecasting performance but also to a better understanding of complex phenomena in various scientific and engineering domains. Future research can further explore

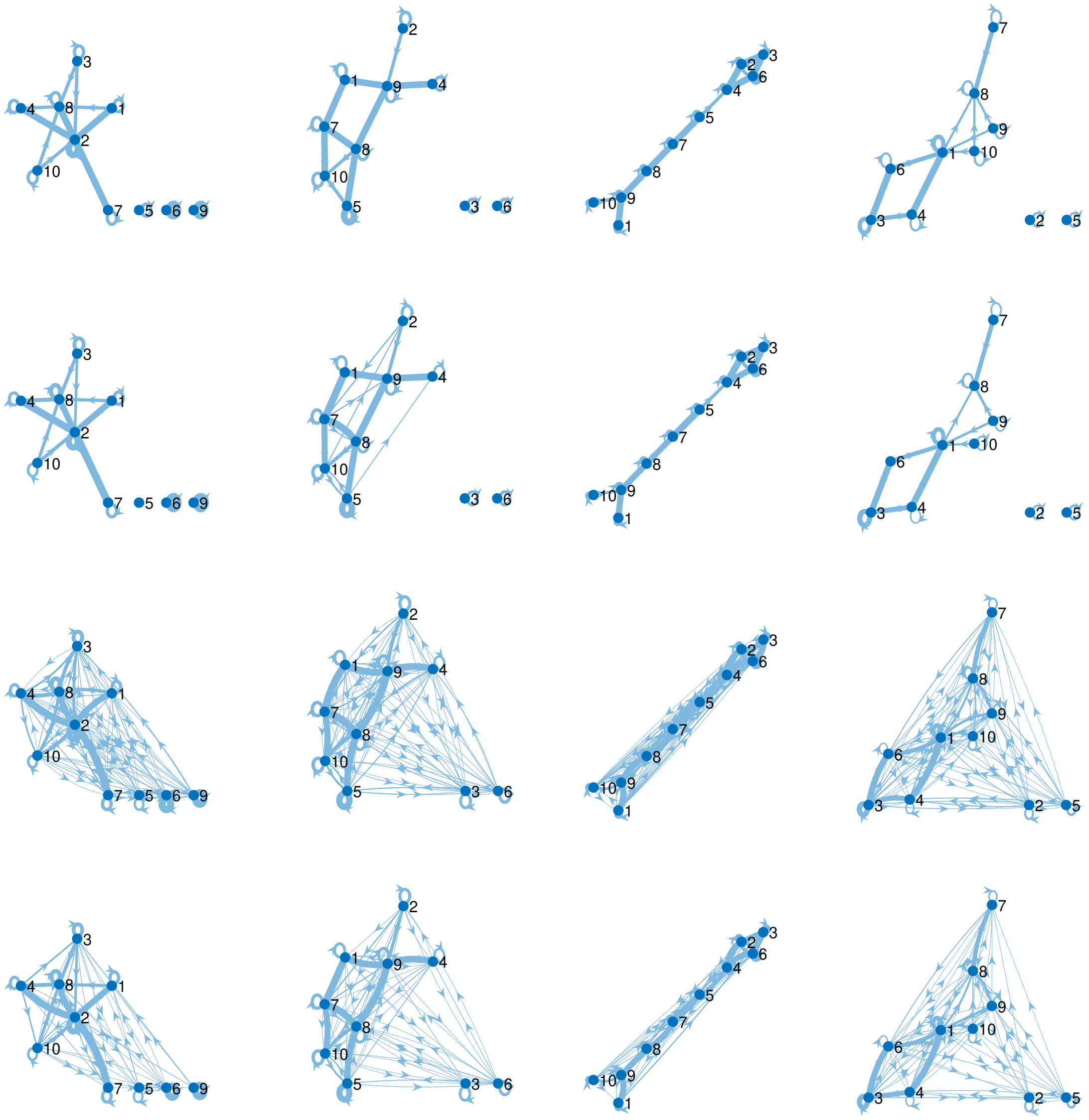


Figure 9: Graph inference results on 4 examples extracted from the dataset *WeathN10a*. From top to bottom: Original graph representation of $\mathbf{A}^*$, and of its estimation $\widehat{\mathbf{A}}$, using DGLASSO, MLEM, GRAPHEM, respectively.

automatic hyperparameter tuning, and extend the presented framework to tackle even more challenging state-space models, and more complex graphical structures.

Acknowledgments

É.C. acknowledges support from the European Research Council Starting Grant MAJORIS ERC-2019-STG-850925. The work of V. E. is supported by the *Agence Nationale de la Recherche* of France under PISCES (ANR-17-CE40-0031-01), the Leverhulme Research Fellowship (RF-2021-593), and by ARL/ARO under grant W911NF-22-1-0235.
The authors thank Gustau Camps-Valls and his team, for providing the dataset from Runge et al. (2020).

Appendix A. Proof of Theorem 1

For any initial state $\mathbf{x}_{0:K}$, with non zero probability, the neg-log-likelihood $(\mathbf{A},\mathbf{P}) \to (\mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) = -\log p(\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P}))$ reads, according to Bayes' rule,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) = -\log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P}) + \log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A},\mathbf{P}). \tag{70}$$

Again, note that the l.h.s. does not depend on $\mathbf{x}_{0:K}$, so the r.h.s. is valid for any arbitrary value of $\mathbf{x}_{0:K}$ with non-zero probability under $p(\mathbf{x}_{0:K})$, i.e., for all $\mathbf{x}_{0:K} \in \mathbb{R}^{(K+1)N_x}$. According to Eqs. (2)-(3)

$$\log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P}) = \log p(\mathbf{x}_0) + \sum_{k=1}^{K} \log p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{A},\mathbf{P}) + \sum_{k=1}^{K} \log p(\mathbf{y}_k|\mathbf{x}_k). \tag{71}$$

Moreover, using again Eq. (2) and the statistical model of the state noise,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \tag{72}$$

$$\sum_{k=1}^{K} \log p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{A},\mathbf{P}) = -\frac{1}{2}\sum_{k=1}^{K}\Big((\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P}\,(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1}) - \log\det(2\pi\mathbf{P})\Big), \tag{73}$$

$$= -\frac{1}{2}\sum_{k=1}^{K}\Big((\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})^\top \mathbf{P}\,(\mathbf{x}_k - \mathbf{A}\mathbf{x}_{k-1})\Big) + \frac{K}{2}\log\det(2\pi\mathbf{P}), \tag{74}$$

which concludes the first part of the proof.
Let us now consider some $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ and $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$. We start by recalling some known results arising from the EM methodology (Dempster et al., 1977; Wu, 1983), that we specify here for our context for the sake of clarity. First, we notice that

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) = -\int \log p(\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P}) p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) \mathrm{d}\mathbf{x}_{0:\mathrm{K}}, \tag{75}$$

since $\log p(\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P})$ is constant with respect to the integration variable, and the distribution $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$ integrates to one. Then, according to (70) and (75), the expectation of the neg-log-likelihood multiplied by $p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$ over all possible values of the unknown state reads:

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x \times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) = \overbrace{-\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) \log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P}) \mathrm{d}\mathbf{x}_{0:\mathrm{K}}}^{q(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})}$$
$$+ \overbrace{\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) \log p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\mathbf{A},\mathbf{P}) \mathrm{d}\mathbf{x}_{0:\mathrm{K}}}^{h(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})}. \tag{76}$$

In particular, for $(\mathbf{A},\mathbf{P}) = (\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$, $\mathcal{L}_{1:K}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) = q(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) + h(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$ so that

$$\begin{aligned}(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) - \mathcal{L}_{1:K}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) &= q(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) - q(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})\\ &+ h(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) - h(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}). \quad (77)\end{aligned}$$

Using Gibbs's inequality, $h(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) \le h(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$, with equality at $(\mathbf{A},\mathbf{P}) = (\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$. Thus, using (77), that is

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) \le q(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) - q(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}), \quad (78)$$

where equality holds at $(\mathbf{A},\mathbf{P}) = (\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$. Notice that, for any function reading

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) = q(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) + \mathrm{ct}_{/\mathbf{A},\mathbf{P}}, \quad (79)$$

we obviously still have

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}_{1:K}(\mathbf{A},\mathbf{P}) \le \mathcal{Q}(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) - \mathcal{Q}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}), \quad (80)$$

where equality hereagain holds at $(\mathbf{A},\mathbf{P}) = (\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$. Using (80), (21), (23) and noticing that, for every $(\theta_A,\theta_P) > 0$,

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \frac{1}{2\theta_A}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 \ge 0, \quad \frac{1}{2\theta_P}\|\mathbf{P} - \widetilde{\mathbf{P}}\|_F^2 \ge 0, \quad (81)$$

with equality holding at $(\mathbf{A},\mathbf{P}) = (\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$, we deduce the desired majorizing property

$$\begin{aligned}(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{L}(\mathbf{A},\mathbf{P}) &\le \mathcal{Q}(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) + \mathcal{L}_{1:K}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) - \mathcal{Q}(\widetilde{\mathbf{A}},\widetilde{\mathbf{P}};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})\\ &+ \lambda_A\|\mathbf{A}\|_1 + \lambda_P\|\mathbf{P}\|_1 + \frac{1}{2\theta_A}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2 + \frac{1}{2\theta_P}\|\mathbf{P} - \widetilde{\mathbf{P}}\|_F^2, \quad (82)\end{aligned}$$

with equality holding at $(\mathbf{A},\mathbf{P}) = (\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$. The remaining of the proof amounts to expliciting the expression for $(\mathbf{A},\mathbf{P}) \to \mathcal{Q}(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}})$ satisfying (79) with function q defined as in (76):

$$\begin{aligned}&(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x})\\ &\qquad q(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) = -\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K},\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) \log p(\mathbf{x}_{0:K},\mathbf{y}_{1:K}|\mathbf{A},\mathbf{P}) \mathrm{d}\mathbf{x}_{0:\mathrm{K}}. \quad (83)\end{aligned}$$

Following (Sarkka, 2013, Theorem 12.4) (see also an alternative proof in (Elvira and Chouzenoux, 2022, Sec. III-B)), (76)-(79) hold for

$$(\forall \mathbf{A} \in \mathbb{R}^{N_x\times N_x})(\forall \mathbf{P} \in \mathcal{S}_{N_x}) \quad \mathcal{Q}(\mathbf{A},\mathbf{P};\widetilde{\mathbf{A}},\widetilde{\mathbf{P}}) = \frac{K}{2}\mathrm{tr}\left(\mathbf{P}(\widetilde{\mathbf{\Psi}} - \widetilde{\mathbf{\Delta}}\mathbf{A}^\top - \mathbf{A}\widetilde{\mathbf{\Delta}}^\top + \mathbf{A}\widetilde{\mathbf{\Phi}}\mathbf{A}^\top)\right) - \frac{K}{2}\log\det(2\pi\mathbf{P}), \quad (84)$$

and

$$\widetilde{\mathbf{\Psi}} = \frac{1}{K}\sum_{k=1}^{K}\left(\mathbf{\Sigma}_k^{\mathrm{s}} + \boldsymbol{\mu}_k^{\mathrm{s}}(\boldsymbol{\mu}_k^{\mathrm{s}})^\top\right), \tag{85}$$

$$\widetilde{\mathbf{\Delta}} = \frac{1}{K}\sum_{k=1}^{K}\left(\mathbf{\Sigma}_k^{\mathrm{s}}\mathbf{G}_{k-1}^\top + \boldsymbol{\mu}_k^{\mathrm{s}}(\boldsymbol{\mu}_{k-1}^{\mathrm{s}})^\top\right), \tag{86}$$

$$\widetilde{\mathbf{\Phi}} = \frac{1}{K}\sum_{k=1}^{K}\left(\mathbf{\Sigma}_{k-1}^{\mathrm{s}} + \boldsymbol{\mu}_{k-1}^{\mathrm{s}}(\boldsymbol{\mu}_{k-1}^{\mathrm{s}})^\top\right). \tag{87}$$

Hereabove, $(\boldsymbol{\mu}_k^{\mathrm{s}}, \mathbf{\Sigma}_k^{\mathrm{s}})_{0 \le k \le K-1}$ denotes the mean and covariance of the smoothing distribution obtained when running Algs. 1-2 using $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}})$. Moreover, the matrix

$$\mathbf{G}_k = \mathbf{\Sigma}_k\widetilde{\mathbf{A}}^\top\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}_k\widetilde{\mathbf{A}}^\top + \widetilde{\mathbf{P}}^{-1}\right) \tag{88}$$

is defined as in Algorithm 2. This concludes the proof.

Appendix B. Proximal algorithms to solve the inner steps

We present Algorithms 4 and 5, that are proximal splitting algorithms to solve, respectively, the inner problems (33) and (34). Specifically, both algorithms are special instances of the Dykstra-like splitting algorithm from Bauschke and Combettes (2008) (see also (Combettes and Pesquet, 2011, Sec.5)), for the minimization of the sum of two convex but non-differentiable functions. Sequence $(\mathbf{A}_n)_{n\in\mathbb{N}}$ (resp. $(\mathbf{P}_n)_{n\in\mathbb{N}}$) is guaranteed to converge to the solution of problem (33) (resp. (34)). The proximity steps involved in Algorithms 4 and 5 have closed form expressions that can be found for instance in Bauschke and Combettes (2017). We explicit them hereafter for the sake of completeness.

Proximity of ℓ_1. Let $\gamma > 0$ and $\widetilde{\mathbf{V}} \in \mathbb{R}^{N_x \times N_x}$. Then,

$$\begin{aligned} &\mathrm{prox}_{\gamma\ell_1}(\widetilde{\mathbf{V}}) && (89)\\ &= \left(\mathrm{sign}(\widetilde{V}(n,\ell)\max(0, \widetilde{V}(n,\ell) - \gamma)\right)_{1\le n,\ell\le N_x}, && (90)\end{aligned}$$

which amounts to applying the soft thresholding operator with weight γ on every entry of the matrix input $\widetilde{\mathbf{V}}$.

Proximity of quadratic term. Let $\gamma > 0$ and $\widetilde{\mathbf{W}} \in \mathbb{R}^{N_x \times N_x}$. Then, by definition,

$$\widehat{\mathbf{Z}} = \mathrm{prox}_{\mathbf{W}\to\gamma\mathrm{tr}\left(-\widetilde{\mathbf{P}}\widetilde{\Delta}\mathbf{W} - \widetilde{\mathbf{P}}\mathbf{W}\widetilde{\Delta}^\top + \widetilde{\mathbf{P}}\mathbf{W}\widetilde{\Psi}\mathbf{W}^\top\right)}\left(\widetilde{\mathbf{W}}\right) \tag{91}$$

$$= \underset{\mathbf{W}\in\mathbb{R}^{N_x\times N_x}}{\mathrm{argmin}}\ \gamma\mathrm{tr}\left(-\widetilde{\mathbf{P}}\widetilde{\Delta}\mathbf{W}^\top - \widetilde{\mathbf{P}}\mathbf{W}\widetilde{\Delta}^\top + \widetilde{\mathbf{P}}\mathbf{W}\widetilde{\Psi}\mathbf{W}^\top\right) + \frac{1}{2}\|\mathbf{W} - \widetilde{\mathbf{W}}\|_{\mathrm{F}}^2. \tag{92}$$

The optimality condition for (92) gives

$$-\gamma\widetilde{\mathbf{P}}\widetilde{\mathbf{\Delta}} - \gamma\widetilde{\mathbf{\Delta}}^\top\widetilde{\mathbf{P}} + \gamma\widetilde{\mathbf{P}}\widehat{\mathbf{Z}}\widetilde{\mathbf{\Psi}} + \gamma\widetilde{\mathbf{P}}^\top\widehat{\mathbf{Z}}\widetilde{\mathbf{\Psi}}^\top + \widehat{\mathbf{Z}} - \widetilde{\mathbf{W}} = \mathbf{0}. \tag{93}$$

Since $\widetilde{\mathbf{P}} \in \mathcal{S}_{N_x}^{++}$, and $\widetilde{\mathbf{\Psi}} \in \mathcal{S}_{N_x}$ (by construction), we have equivalently,

$$-\gamma\widetilde{\mathbf{\Delta}} - \gamma\widetilde{\mathbf{P}}^{-1}\widetilde{\mathbf{\Delta}}^\top\widetilde{\mathbf{P}} + 2\gamma\widehat{\mathbf{Z}}\widetilde{\mathbf{\Psi}} + \widetilde{\mathbf{P}}^{-1}\widehat{\mathbf{Z}} - \widetilde{\mathbf{P}}^{-1}\widetilde{\mathbf{W}} = \mathbf{0}. \tag{94}$$

Thus,

$$\widehat{\mathbf{Z}} = \text{lyapunov}\left(\widetilde{\mathbf{P}}^{-1}, 2\gamma\widetilde{\mathbf{\Psi}}, \gamma(\widetilde{\mathbf{\Delta}} + \widetilde{\mathbf{P}}^{-1}\widetilde{\mathbf{\Delta}}^\top\widetilde{\mathbf{P}}) + \widetilde{\mathbf{P}}^{-1}\widetilde{\mathbf{W}}\right), \tag{95}$$

where $\mathsf{A} = \text{lyapunov}(\mathsf{X}, \mathsf{Y}, \mathsf{Z})$ provides the solution to the Lyapunov equation $\mathsf{XA} + \mathsf{AY} = \mathsf{Z}$.

Proximity of log-determinant term. Let $\gamma > 0$ and $\widetilde{\mathbf{W}} \in \mathcal{S}_{N_x}^{++}$. Then, by definition,

$$\widehat{\mathbf{Z}} = \text{prox}_{\mathbf{W}\to\gamma(-\log\det(\mathbf{W})+\text{tr}(\mathbf{W}\widetilde{\mathbf{\Pi}}))}\left(\widetilde{\mathbf{W}}\right) \tag{96}$$

$$= \underset{\mathbf{W}\in\mathcal{S}_{N_x}}{\text{argmin}} \; -\gamma\log\det(\mathbf{W}) + \gamma\text{tr}(\mathbf{W}\widetilde{\mathbf{\Pi}}) + \frac{1}{2}\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F^2. \tag{97}$$

Using (Bauschke and Combettes, 2017, Chap. 24) (see also Benfenati et al. (2020)), for every $\alpha > 0$,

$$\widehat{\mathbf{Z}} = \mathbf{U}\text{Diag}\left(\left(\frac{1}{2}(\omega(n) + \sqrt{\omega(n)^2 + 4\gamma})\right)_{1\leq n\leq N_x}\right)\mathbf{U}^\top \tag{98}$$

where $\boldsymbol{\omega} = (\omega(n))_{1\leq n\leq N_x}$ gathers the eigenvalues of $\widetilde{\mathbf{W}} - \gamma\widetilde{\mathbf{\Pi}} \in \mathcal{S}_{N_x}$ and $\mathbf{U} \in \mathbb{R}^{N_x\times N_x}$ is an orthogonal matrix such that

$$\widetilde{\mathbf{W}} - \gamma\widetilde{\mathbf{\Pi}} = \mathbf{U}\text{Diag}(\boldsymbol{\omega})\mathbf{U}^\top. \tag{99}$$

Appendix C. Additional experiments

We present here additional experimental results completing Section 5.1.

C.1 Robustness to initialization

DGLASSO algorithm amounts to minimizing a non-convex loss function. As such, its results might be sensitive to the initialization of the algorithm. To evaluate this aspect, we consider the computation of $(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$ given the observation of a single time series generated by the ground truth LG-SSM, when using 50 different initializations of DGLASSO algorithm. To do so, we use the same initialization strategy as discussed above, now with (a, p) randomly selected as $p \sim \mathcal{U}([0, 1])$ and $a \sim \mathcal{U}([0, 1])$. Figure 10 displays the box plots for RMSE and F1 scores obtained for dataset A. One can notice that the box plots are very concentrated, showing a good robustness of the method to its initialization, with the wider spreading observed for the F1 score on $\mathbf{A}$. Similar behavior was observed for the other three datasets.

C.2 Influence of sparsity level

We evaluate here the performance of DGLASSO, as well as the benchmarks, when varying the sparsity level of the ground truth matrices. To do so, we perform slight changes in the

```
Algorithm 4 Proximal splitting method to solve (33)
    Inputs. $\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}, \widetilde{\boldsymbol{\Psi}}, \widetilde{\boldsymbol{\Delta}}, \widetilde{\boldsymbol{\Phi}}$. Precision $\xi > 0$.
 1. Setting. Set stepsize $\vartheta \in (0, 2)$.
 2. Initialization. Set $\mathbf{V}_0 = \widetilde{\mathbf{A}}$.
 3. Recursive step. For $n = 1, 2, \ldots$:
        $\mathbf{A}_n = \mathrm{prox}_{\theta_A \lambda_A \ell_1} \left( \widetilde{\mathbf{A}} - \mathbf{V}_n \right)$
        $\mathbf{W}_n = \mathbf{V}_n + \vartheta \mathbf{A}_n$
        $\mathbf{Z}_n = \mathrm{prox}_{\mathbf{W} \to \frac{\vartheta \theta_A K}{2} \mathrm{tr}\left(-\widetilde{\mathbf{P}}\widetilde{\Delta}\mathbf{W} - \widetilde{\mathbf{P}}\mathbf{W}\widetilde{\Delta}^\top + \widetilde{\mathbf{P}}\mathbf{W}\widetilde{\Psi}\mathbf{W}^\top\right)} \left(\vartheta^{-1}\mathbf{W}_n\right)$
        $\mathbf{V}_{n+1} = \mathbf{W}_n - \vartheta \mathbf{Z}_n$.
    If $|\mathcal{C}_1(\mathbf{A}_n) - \mathcal{C}_1(\mathbf{A}_{n-1})| \leq \xi$, stop the recursion.
    Output. Transition matrix $\widehat{\mathbf{A}} = \mathbf{A}_n$.
```

```
Algorithm 5 Proximal splitting method to solve (34)
    Inputs. $\widetilde{\mathbf{A}}, \widetilde{\mathbf{P}}, \widetilde{\boldsymbol{\Psi}}, \widetilde{\boldsymbol{\Delta}}, \widetilde{\boldsymbol{\Phi}}$. Precision $\xi > 0$.
 1. Setting. Set stepsize $\vartheta \in (0, 2)$ and $\widetilde{\boldsymbol{\Pi}}$ as in (35).
 2. Initialization. Set $\mathbf{V}_0 = \widetilde{\mathbf{P}}$.
 3. Recursive step. For $n = 1, 2, \ldots$:
        $\mathbf{P}_n = \mathrm{prox}_{\theta_P \lambda_P \ell_1} \left( \widetilde{\mathbf{P}} - \mathbf{V}_n \right)$
        $\mathbf{W}_n = \mathbf{V}_n + \vartheta \mathbf{P}_n$
        $\mathbf{Z}_n = \mathrm{prox}_{\mathbf{W} \to \frac{\vartheta \theta_P K}{2} \left(-\log\det(\mathbf{W}) + \mathrm{tr}(\widetilde{\Pi}\mathbf{W})\right)} \left(\vartheta^{-1}\mathbf{W}_n\right)$
        $\mathbf{V}_{n+1} = \mathbf{W}_n - \vartheta \mathbf{Z}_n$.
    If $|\mathcal{C}_2(\mathbf{A}_n) - \mathcal{C}_2(\mathbf{A}_{n-1})| \leq \xi$, stop the recursion.
    Output. Precision matrix $\widehat{\mathbf{P}} = \mathbf{P}_n$.
```

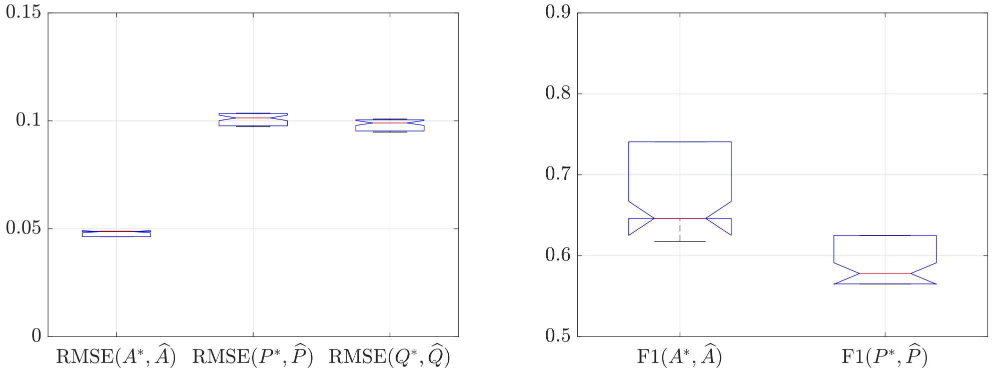


Figure 10: Box plots for the RMSE (left) and F1 (right) scores for retrieving $(\mathbf{A}, \mathbf{P}, \mathbf{Q})$ matrices, when running DGLASSO on one single LG-SSM time series using dataset A, and 50 random initializations $(\mathbf{A}^{(0)}, \mathbf{P}^{(0)})$. Noticeably, low variability is observed for all metrics.

dataset, to vary the sparsity pattern of the matrices (i.e., the edge structure of the graphs). First, we modify the ground truth matrix $\mathbf{A}^*$ by keeping $s_A \in \{27, 15, 10, 5\}$ entries of it, within the 27 block diagonal ones, to be non-zero, the others being set to zero. We then rescaled the matrix to keep a spectral norm equal to 0.99. Matrix $\mathbf{Q}^*$ is taken from dataset A. The results are reported in Table 4.

As we can observe, the performance of MLEM, in terms of RMSE and F1 score, drop dramatically when the sparsity level on $\mathbf{A}$ increases. This is expected as this approach does not promote any sparsity prior on matrix $\mathbf{A}$. The best AUC are either obtained by MLEM, DGLASSO or rGLASSO (for $\mathbf{P}$), depending on the test cases. GLASSO/rGLASSO metrics slightly improve when $\mathbf{A}^*$ gets sparser, which is expected, as their assumption of a zero transition matrix gets more realistic. Hereagain, DGLASSO outperforms the benchmarks in most cases.

References

C. Alippi and D. Zambon. Graph Kalman filters. Technical report, 2023. https://arxiv.org/pdf/2303.12021.pdf.

S. Aminikhanghahi and D. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51:339–367, 2017.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems:an approach based on the kurdyka-lojasiewicz inequality. *Mathematics of Operation Research*, 35(2):438–457, 2010.

Table 4: Results for sparsity levels $s_A \in \{27, 15, 10, 5\}$ of $\mathbf{A}^*$, using $(\mathbf{P}^*, \mathbf{Q}^*)$ from dataset A. The first set of results, with $s_A = 27$ identifies with the first row block of Tab. 1.

		Estimation of $\mathbf{A}$			Estimation of $\mathbf{P}$			Estim. $\mathbf{Q}$	State distrib.		Predictive distrib.	
	Method	RMSE	AUC	F1	RMSE	AUC	F1	RMSE	cNMSE$(\boldsymbol{\mu}^*, \widehat{\boldsymbol{\mu}})$	cNMSE$(\boldsymbol{\mu}^{\mathrm{s}*}, \widehat{\boldsymbol{\mu}}^{\mathrm{s}})$	cNMSE$(\boldsymbol{\nu}^*, \widehat{\boldsymbol{\nu}})$	$\mathcal{L}_{1:K}(\widehat{\mathbf{A}}, \widehat{\mathbf{P}})$
$s_A = 27$	DGLASSO	0.061	0.843	0.641	**0.082**	0.778	**0.698**	**0.083**	$\mathbf{6.394 \times 10^{-8}}$	$\mathbf{1.050 \times 10^{-7}}$	$\mathbf{2.984 \times 10^{-4}}$	**12 307.169**
	MLEM	0.076	0.817	0.500	0.105	0.857	0.500	0.102	1.095×10^{-7}	1.803×10^{-7}	4.843×10^{-4}	12 341.205
	GLASSO	NA	NA	NA	0.818	0.804	0.496	1 073.510	4.485×10^{-6}	7.180×10^{-6}	1.000	28 459.294
	rGLASSO	NA	NA	NA	0.764	**0.924**	0.598	31.689	2.826×10^{-6}	5.492×10^{-6}	1.000	22 957.693
	GRAPHEM	**0.045**	**0.895**	**0.847**	NA	NA	NA	NA	4.364×10^{-6}	6.944×10^{-6}	2.980×10^{-4}	29 035.030
$s_A = 15$	DGLASSO	**0.108**	0.916	0.781	**0.077**	0.883	**0.855**	**0.075**	$\mathbf{8.268 \times 10^{-7}}$	$\mathbf{1.157 \times 10^{-6}}$	$\mathbf{5.183 \times 10^{-3}}$	**11 801.220**
	MLEM	0.159	**0.933**	0.313	0.105	**0.899**	0.500	0.101	1.298×10^{-6}	2.110×10^{-6}	6.052×10^{-3}	11 806.483
	GLASSO	NA	NA	NA	0.463	0.702	0.576	7.088	2.064×10^{-5}	3.723×10^{-5}	1.000	14 258.010
	rGLASSO	NA	NA	NA	0.425	0.723	0.598	4.562	1.379×10^{-5}	2.768×10^{-5}	1.000	14 432.765
	GRAPHEM	0.127	0.928	**0.721**	NA	NA	NA	NA	5.077×10^{-5}	6.455×10^{-5}	3.802×10^{-3}	29 034.715
$s_A = 10$	DGLASSO	0.089	0.981	**0.938**	**0.084**	0.768	**0.686**	**0.084**	$\mathbf{9.386 \times 10^{-7}}$	$\mathbf{1.415 \times 10^{-6}}$	$\mathbf{9.748 \times 10^{-3}}$	12 290.789
	MLEM	0.167	**1.000**	0.220	0.105	**0.900**	0.500	0.101	2.383×10^{-6}	3.918×10^{-6}	2.592×10^{-2}	**11 819.739**
	GLASSO	NA	NA	NA	0.365	0.667	0.500	0.469	3.026×10^{-5}	5.958×10^{-5}	1.000	13 173.462
	rGLASSO	NA	NA	NA	0.339	0.749	**0.699**	0.632	1.970×10^{-5}	4.375×10^{-5}	1.000	13 497.061
	GRAPHEM	**0.117**	**1.000**	0.745	NA	NA	NA	NA	8.956×10^{-5}	1.124×10^{-4}	1.226×10^{-2}	29 034.727
$s_A = 5$	DGLASSO	**0.100**	0.823	**0.774**	**0.084**	0.764	**0.682**	**0.085**	$\mathbf{9.217 \times 10^{-7}}$	$\mathbf{1.206 \times 10^{-6}}$	$\mathbf{9.981 \times 10^{-3}}$	**12 268.667**
	MLEM	0.234	0.893	0.116	0.106	**0.856**	0.500	0.102	2.232×10^{-6}	3.538×10^{-6}	5.610×10^{-2}	12 312.749
	GLASSO	NA	NA	NA	0.242	0.667	0.500	0.306	1.406×10^{-5}	3.095×10^{-5}	1.000	12 892.144
	rGLASSO	NA	NA	NA	0.262	0.704	0.566	0.451	1.195×10^{-5}	2.884×10^{-5}	1.000	12 834.737
	GRAPHEM	0.127	**0.931**	0.573	NA	NA	NA	NA	1.060×10^{-4}	1.225×10^{-4}	1.477×10^{-2}	29 034.456

H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms,forward–backward splitting. *Mathematical Programming*, 137(1):91–129, Feb. 2013.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, Jan. 2012.

F. R. Bach and M. I. Jordan. Learning graphical models for stationary time series. *IEEE Transactions on Signal Processing*, 52(8):2189–2199, Aug. 2004.

D. Barber and A. T. Cemgil. Graphical models for time-series. *IEEE Signal Processing Magazine*, 27(6):18–28, Nov 2010.

H. H. Bauschke and P. L. Combettes. A Dykstra-like algorithm for two monotone operators. *Pacific Journal of Optimization*, 4:383–391, 2008.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* Springer, 2017.

E. Belilovsky, K. Kastner, G. Varoquaux, and M. B. Blaschko. Learning to discover sparse graphical models. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, volume 70, pages 440–448, 2017.

A. Benfenati, E. Chouzenoux, L. Duval, J.-C. Pesquet, and A. Pirayre. A review on graph optimization and algorithmic frameworks. Technical report, 2018. https://hal.science/hal-01901499/document.

A. Benfenati, E. Chouzenoux, and J.-C. Pesquet. Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation. *Signal Processing*, 169: 107417, Apr. 2020.

Q. Bertrand, Q. Klopfenstein, M. Massias, M. Blondel, and S. Vaiter. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *Journal of Machine Learning Research*, (23):1–43, 2022.

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, Aug. 2014.

S. Bonettini, M. Prato, and S. Rebegoldi. A block coordinate variable metric linesearch based proximal gradient method. *Computational Optimization and Applications*, 71(1): 5–52, 2018.

S. Bonettini, M. Prato, and S. Rebegoldi. New convergence results for the inexact variable metric forward–backward method. *Applied Mathematics and Computation*, 392:125719, 2021.

L. Brendan and J. Tenenbaum. Discovering structure by learning sparse graphs. In *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society (CogSci 2010)*, pages 778–784, Portland, Oregon, USA, 11-14 Aug. 2010.

M. Briers, A. Doucet, and S. Maskell. Smooting algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(61), 2010.

I. Buchnik, G. Sagi, N. Leinwand, Y. Loya, N. Shlezinger, and T. Routtenberg. Gsp-kalmannet: Tracking graph signals via neural-aided Kalman filtering. Technical report, 2023. https://arxiv.org/pdf/2311.16602.

P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media, 2011.

O. Cappe, E. Moulines, and T. Ridden. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer New York, NY, 1st edition, 2005.

I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research*, 23(89): 1–64, 2022.

V. Chandrasekaran, P. Parrilo, and A. Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012.

C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs. A variational formulation for frame-based inverse problems. *Inverse Problems*, 23(4):1495–1518, June 2007.

X. Chen and Y. Li. An overview of differentiable particle filters for data-adaptive sequential Bayesian inference. Technical report, 2023. https://arxiv.org/abs/2302.09639.

X. Chen, H. Wen, and Y. Li. Differentiable particle filters through conditional normalizing flow. In *Proceedings of the IEEE 24th International Conference on Information Fusion (FUSION 2021)*, pages 1–6, 2021.

A. Cherni, E. Chouzenoux, L. Duval, and J.-C. Pesquet. SPOQ lp-over-lq regularization for sparse signal recovery applied to mass spectrometry. *IEEE Transactions on Signal Processing*, 68:6070–6084, 2020.

K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734, Doha, Qatar, 25–29 Oct. 2014.

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. Smc2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(3):397–426, 2013.

E. Chouzenoux and V. Elvira. GraphEM: EM algorithm for blind Kalman filtering under graphical sparsity constraints. In *In Proceedings of the 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020)*, pages 5840–5844, 4–8 May 2020.

E. Chouzenoux and V. Elvira. Graphit: Iterative reweighted l1 algorithm for sparse graph inference in state-space models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, Rhodes Island, Greece, 4-10 June 2023.

E. Chouzenoux and V. Elvira. Graphical inference in non-Markovian linear-Gaussian state-space models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2024)*, Seoul, South Korea, Apr. 2024.

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, 2014.

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. A block coordinate variable metric forward-backward algorithm. *Journal of Global Optimization*, 66(3):457–485, 2016.

E. Chouzenoux, T. T.-K. Lau, and J.-C. Pesquet. Optimal multivariate Gaussian fitting with applications to PSF modeling in two-photon microscopy imaging. *Journal of Mathematical Imaging and Vision*, 61(7):1037–1050, 2019.

A. Cini, D. Zambon, and C. Alippi. Sparse graph learning from spatiotemporal time series. *Journal of Machine Learning Research*, 24(242):1–36, 2023.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, page 2100–2111, Jul 2021.

H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari. Long short-term memory kalman filters:recurrent neural estimators for pose regularization. In *Proceedings of International Conference on Computer Vision (ICCV 2017)*, 22-29 Oct. 2017.

B. Cox and V. Elvira. Parameter estimation in sparse linear-Gaussian state-space models via reversible jump Markov Chain Monte Carlo. In *Proceedings of the 30th European Signal Processing Conference (EUSIPCO 2022)*, pages 797–801, Belgrade, Serbia, 29 Aug.-2 Sep. 2022.

B. Cox and V. Elvira. Sparse bayesian estimation of parameters in linear-gaussian state-space models. *IEEE Transactions on Signal Processing*, 71:1922–1937, 2023.

D. Crisan and J. Miguez. Nested particle filters for online parameter estimation in discrete-time state-space Markov models. Technical report, 2018. https://arxiv.org/abs/1308.1883.

C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré. Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. *SIAM Journal on Imaging Sciences*, 7(4): 2448–2487, 2014. doi: 10.1137/140968045.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003.

A. Doerr, C. Daniel, M. Schiegg, N.-T. Duy, S. Schaal, M. Toussaint, and T. Sebastian. Probabilistic recurrent state-space models. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, volume 80 of *Proceedings of Machine Learning Research*, pages 1280–1289, 10–15 Jul. 2018.

A. Doucet, A. M. Johansen, et al. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

M. Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153(1):233–268, Jun. 2012. ISSN 1432-2064. doi: 10.1007/s00440-011-0345-8. URL `https://doi.org/10.1007/s00440-011-0345-8`.

V. Elvira and E. Chouzenoux. Graphical inference in linear-Gaussian state-space models. *IEEE Transactions on Signal Processing*, 70:4757–4771, 2022.

S. Fattahi and S. Sojoudi. Graphical lasso and thresholding: Equivalence and closed-form solutions. *Journal of Machine Learning Research*, 20(10):1–44, 2019.

L. Franceschi, M. Donini, P. Frasconi, and M. Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, Sydney, Australia, 6th-11st Aug. 2017.

L. Frenkel and M. Feder. Recursive expectation-maximization (EM) algorithms for time-varying parameters with applications to multiple target tracking. *IEEE Transactions on Signal Processing*, 47(2):306–320, 1999.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, jul 2008.

R. Gao, F. Tronarp, and S. Sarkka. Iterated extended Kalman smoother-based variable splitting for l1-regularized state estimation. *IEEE Transactions on Signal Processing*, 67 (19):5078–5092, 2015.

G. Giannakis, Y. Shen, and G. Karanikolas. Topology identification and learning over graphs: Accounting for nonlinearities and dynamics. *Proceedings of the IEEE*, 106(5): 787–807, May 2018.

C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

P. J. Green and D. I. Hastie. Reversible jump MCMC. *Genetics*, 155(3):1391–1403, 2009.

A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. Technical report, 2023. https://arxiv.org/abs/2312.00752.

A. Gu, I. Johnson, K. Goel, K. Saab, T. Dao, A. Rudra, and C. Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

N. Gupta and R. Mehra. Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, 19(6):774–783, 1974.

J. D. Hamilton. State-space models. *Handbook of Econometrics*, 4:3039–3080, 1994.

J. D. Hamilton. *Time Series Analysis.* Princeton university press, 2020.

L. Hien, D. N. Phan, and N. Gillis. An inertial block majorization minimization framework for nonsmooth nonconvex optimization. *Journal of Machine Learning Research*, 24(18): 1–41, 2023.

T. Hien, N. Gillis, and P. Patrinos. Inertial block proximal method for non-convex non-smooth optimization. In *Proceedings of the Thirty-seventh International Conference on Machine Learning (ICML 2020)*, 2020.

A. Hippert-Ferrer, F. Bouchard, A. Mian, T. Vayer, and A. Breloy. Learning graphical factor models with Riemannian optimization. Technical report, 2022. https://arxiv.org/abs/2210.11950.

S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computations*, 9(8): 1735–1780, Nov. 1997.

M. Hong, M. Razaviyayn, Z.-Q. Luo, and J. S. Pang. A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing. *IEEE Signal Processing Magazine*, 33(1):57–77, Jan. 2016.

D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Amer. Stat.*, 58(1):30–37, Feb. 2004.

M. Hüsken and P. Stagge. Recurrent neural networks for time series classification. *Neurocomputing*, 50:223–235, 2003.

V. Ioannidis, D. Romero, and G. Giannakis. Inference of spatio-temporal functions over graphs via multikernel kriged Kalman filtering. *IEEE Transactions on Signal Processing*, 66(12):3228–3239, 2018.

V. Ioannidis, Y. Shen, and G. Giannakis. Semi-blind inference of topologies and dynamical processes over dynamic graphs. *IEEE Transactions on Signal Processing*, 67(9):2263–2274, May 2019.

M. Jacobson and J. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Transactions on Image Processing*, 16(10): 2411–2422, 2007.

M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. Technical report, 2023. https://arxiv.org/abs/2307.03759.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin. On particle methods for parameter estimation in state-space models. Technical report, 2015. https://arxiv.org/abs/1412.8695.

C.-J. Kim and C. R. Nelson. *State-space models with regime switching: classical and Gibbs-sampling approaches with applications.* MIT Press Books, 1st edition, 1999.

N. Komodakis and J.-C. Pesquet. Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, 32(6):31–54, 2015.

M. Krzywda, S. Lukasik, and A. H. Gandomi. Graph neural networks in computer vision – architectures, datasets and common approaches. Technical report, 2022. https://arxiv.org/abs/2212.10207.

S. Kumar, J. Ying, J. V. de Miranda Cardoso, and D. P. Palomar. A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research*, 21(22):1–60, 2020.

S. H. Lim. Understanding recurrent neural networks using nonequilibrium response theory. *Journal of Machine Learning Research*, 22(47):1–48, 2021.

F. Lindsten, M. I. Jordan, and T. B. Schon. Particle gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15:2145–2184, 2014.

S. Lojasiewicz. Unepropriété topologique des sous-ensembles analytiques réels. *Editions du Centre National de la Recherche Scientifique*, pages 87–89, 1963.

D. Luengo, G. Rios-Munoz, V. Elvira, C. Sanchez, and A. Artes-Rodriguez. Hierarchical algorithms for causality retrieval in atrial fibrillation intracavitary electrograms. *IEEE Journal of Biomedical and Health Informatics*, 12(1):143–155, Jan. 2019.

S. Luo, R. Song, and D. Witten. Sure screening for gaussian graphical models. Technical report, 2014. https://arxiv.org/abs/1407.7819.

M. Maathuis, M. Drton, S. Lauritzen, and M. Wainwright. *Handbook of Graphical Models*. CRC Press, Boca Raton, FL, 2019.

R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13(27):781–794, 2012.

J. Mei and M. Moura. Signal processing on graphs: Causal modeling of unstructured data. *IEEE Transactions on Signal Processing*, 65(8):2077–2092, Apr. 2017.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerical Mathematics*, 55:377–400, 1989.

C. A. Naesseth, F. Lindsten, T. B. Schön, et al. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

D. Nagakura. Computing exact score vectors for linear Gaussian state space models. *Communications in Statistics - Simulation and Computation*, 50(8):2313–2326, 2021.

K. Newman, R. King, V. Elvira, P. de Valpine, R. S. McCrea, and B. J. Morgan. State-space models for ecological time-series data: Practical model-fitting. *Methods in Ecology and Evolution*, 14(1):26–42, 2023.

R. Olsson, K. Petersen, and T. Lehn-Schioler. State-space models: from the EM algorithm to a gradient approach. *Neural Computation*, 19(4):1097–1111, 2007.

S. Pérez-Vieites and V. Elvira. Adaptive Gaussian nested filter for parameter estimation and state tracking in dynamical systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, pages 1–5, 2023.

S. Pérez-Vieites and J. Míguez. Nested gaussian filters for recursive bayesian inference and nonlinear tracking in state space models. *Signal Processing*, 189:108295, 2021.

K. Petersen and M. Pedersen. The matrix cookbook. Technical report, 2012. http://matrixcookbook.com.

E. Pircalabelu and G. Claeskens. Community-based group graphical lasso. *Journal of Machine Learning Research*, 21(64):1–32, 2020.

C. Pouliquen, P. Goncalves, M. Massias, and T. Vayer. Implicit differentiation for hyperparameter tuning the weighted Graphical Lasso. In *Proceedings of the XXIXème Colloque Francophone De Traitement Du Signal Et Des Images (GRETSI 2023)*, Grenoble, France, 28 Aug.-1st Sep. 2023.

S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NEURIPS 2018)*, volume 31. Curran Associates, Inc., 2018.

G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar. Kalmannet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

J. Runge, X.-A. Tibau, M. Bruhns, J. Muñoz-Marí, and G. Camps-Valls. The causality for climate competition. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, volume 123, pages 110–120, 2020.

G. Sagi, N. Shlezinger, and T. Routtenberg. Extended Kalman filter for graph signals in nonlinear dynamic systems. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2023)*, Rhodes Island, Greece, 4-10 June 2023.

S. Sarkka. *Bayesian Filtering and Smoothing.* 3rd edition, 2013.

J. Schmidt, P. Hennig, J. Nick, and F. Tronarp. The rank-reduced kalman filter: Approximate dynamical-low-rank filtering in high dimensions. *Advances in Neural Information Processing Systems*, 36:61364–61376, 2023.

M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient (score) of linear dynamic systems. *IEEE Transactions on Automatic Control*, 33(8):763–766, 1988. doi: 10.1109/9.1295.

M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient, the hessian, and the fisher information matrix for linear dynamic systems. *IEEE Transactions on Information Theory*, 35(3):682–687, 1989. doi: 10.1109/18.30995.

Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. In L. Cheng, A. C. S. Leung, and S. Ozawa, editors, *Neural Information Processing*, pages 362–373, Cham, 2018. Springer International Publishing.

S. Sharma, A. Majumdar, V. Elvira, and E. Chouzenoux. Blind Kalman filtering for short-term load forecasting. *IEEE Transactions on Power Systems*, 35(6):4916–4919, Nov. 2020.

S. Sharma, V. Elvira, E. Chouzenoux, and A. Majumdar. Recurrent dictionary learning for state-space models with an application in stock forecasting. *Neurocomputing*, 450:1–13, Aug. 2021.

S. Shih, F. Sun, and H. Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108:1421–1441, 2019.

A. Shojaie and G. Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010.

H. Shrivastava, X. Chen, B. Chen, G. Lan, S. Aluru, H. Liu, and L. Song. GLAD: learning sparse graph recovery. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia (virtual), 26th Apr.-1st May 2020.

H. Shrivastava, U. Chajewska, R. Abraham, and X. Chen. uglad: Sparse graph recovery by optimizing deep unrolled networks. Technical report, 2022. https://arxiv.org/abs/2205.11610.

R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.

J. Songsiri and L. Vandenberghe. Topology selection in graphical models of autoregressive processes. *The Journal of Machine Learning Research*, 11:2671–2705, 2010.

Y. Sun, P. Babu, and D. P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794 – 816, 2016.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

C. Uhler. Gaussian graphical models: An algebraic and geometric perspective. Technical report, 2017. https://arxiv.org/abs/1707.04345.

J. Witte, L. Henckel, M. H. Maathuis, and V. Didelez. On efficient adjustment in causal graphs. *Journal of Machine Learning Research*, 21(246):1–45, 2020.

C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

J. Ying, J. V. d. M. Cardoso, and D. P. Palomar. Does the ℓ_1-norm learn a sparse graph under Laplacian constrained graphical models? Technical report, 2020. https://arxiv.org/abs/2006.14925.

J. Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(47):1437–1474, 2008.

P. Zheng, E. Chouzenoux, and L. Duval. PENDANTSS: penalized norm-ratios disentangling additive noise, trend and sparse spikes. *IEEE Signal Processing Letters*, 30:215–219, 2023.