GRAPHEM: EM ALGORITHM FOR BLIND KALMAN FILTERING UNDER GRAPHICAL SPARSITY CONSTRAINTS

Émilie Chouzenoux[1] and Víctor Elvira[2]

[1] Université Paris-Saclay, CentraleSupélec, Inria, CVN, Gif-sur-Yvette, France
[2] School of Mathematics, University of Edinburgh, United Kingdom

ABSTRACT

Modeling and inference with multivariate sequences is central in a number of signal processing applications such as acoustics, social network analysis, biomedical, and finance, to name a few. The linear-Gaussian state-space model is a common way to describe a time series through the evolution of a hidden state, with the advantage of presenting a simple inference procedure due to the celebrated Kalman filter. A fundamental question when analyzing multivariate sequences is the search for relationships between their entries (or the modeled hidden states), especially when the inherent structure is a non-fully connected graph. In such context, graphical modeling combined with parsimony constraints allows to limit the proliferation of parameters and enables a compact data representation which is easier to interpret by the experts. In this work, we propose a novel expectation-maximization algorithm for estimating the linear matrix operator in the state equation of a linear-Gaussian state-space model. Lasso regularization is included in the M-step, that we solved using a proximal splitting Douglas-Rachford algorithm. Numerical experiments illustrate the benefits of the proposed model and inference technique, named GraphEM, over competitors relying on Granger causality.

Index Terms— Kalman filtering, state-space model, graphical inference, sparsity, proximal methods, EM algorithm.

1. INTRODUCTION

State-space models allow for the statistical description of systems in countless applications in science and engineering. They model a hidden state that evolves over (discrete) time, which is only partially observed through transformed and noisy partial observations. In its simple form, the state-space models have a Markovian dependency in the hidden-state, and each observation depends on the current state, being conditionally independent on the rest of states.

In this framework, a common statistical task is the estimation of the hidden state conditioned to the available observations. When the estimation is done in a probabilistic manner, this is commonly known as Bayesian filtering. Similarly, when the estimation of each state is done using the whole sequence of observations, the task is called Bayesian smoothing. The linear-Gaussian state-space model has been widely used in a plethora of problems and admits an exact computation of both filtering and smoothing distribution through the Kalman filter and the Rauch-Tung-Striebel (RTS) smoother [1, Chapter 8]. For non-linear state-space models, there exist many other methods such as the extended KF (EKF), the unscented KF (UKF) [2], or the particle filters (PF) [3]. However, all these algorithms need the model parameters to be known. In the Kalman filter literature, there exist methods for the estimation of the static parameters, based on the expectation-maximization (EM) algorithm or optimization-based methods [1, Chapter 12].

In this paper, we propose a method called GraphEM for estimating the linear matrix of the state equation of a linear-Gaussian state-space model. While this parameter is arguably the hardest to be estimated, it provides very valuable information about the hidden process, not only for inference purposes but also for understanding the uncovered relations among the state dimensions, in the line of graphical modeling methods for time series [4–6]. Such representation of multivariate sequences interactions has applications in several domains such as biology [7, 8], social network analysis [9] and neurosciences [10].

Without any loss of generality, the GraphEM method is developed with the following setup in mind. The multidimensional state contains several uni-dimensional time series, each of them corresponding to a particular node. The matrix in the state equation encodes the linear dependency between the state of a given node and the previous states in the neighbor nodes. GraphEM aims at estimating the directed graph, i.e., the connections between nodes and the associated weights, that represents the (causal) dependencies between the states. GraphEM imposes a sparsity constraint in the linear matrix which promotes that each dimension of the state depends of only small subset of states in the previous time step. The method relies on the EM framework, adapted for

maximum a posteriori (MAP) estimation thanks to the introduction of an ℓ_1 penalization function. The E-step amounts to apply the Kalman filtering and RTS smoothing recursion. The M-step aims at solving a Lasso-like problem, that we address through a Douglas-Rachford proximal splitting algorithm, benefiting from great practical performance and sounded convergence guarantees [11]. Experimental results on several synthetic datasets allow to assess the fast practical convergence rate of our method, and its ability to estimate efficiently the causality graph, in particular when compared to state-of-the-art techniques relying on Granger causality.

The paper is structured as follows. In Section 2 we briefly describe the model and the filtering/smoothing algorithms, and introduce the EM framework for parameters estimation in state-space models. The novel GraphEM algorithm is presented in Section 3. The paper concludes with a numerical example in Section 4 and some remarks in Section 5.

2. BACKGROUND

2.1. Linear state-space model

Let us consider the following Markovian state-space model,

$$
\begin{aligned}
\mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_k, \\
\mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \mathbf{r}_k,
\end{aligned}
\tag{1}
$$

for $k = 1, \ldots, K$, where $\mathbf{x}_k \in \mathbb{R}^{N_x}$ is the hidden state at time k, $\mathbf{y}_k \in \mathbb{R}^{N_y}$ is the associated observation, $\mathbf{A} = \mathbb{R}^{N_x \times N_x}$, $\mathbf{H} = \mathbb{R}^{N_y \times N_x}$, $\{\mathbf{q}_k\}_{k=1}^K \sim \mathcal{N}(0, \mathbf{Q})$ is the i.i.d. state noise process and $\{\mathbf{r}_k\}_{k=1}^K \sim \mathcal{N}(0, \mathbf{R})$ is the i.i.d. observation noise process. The state process is initialized as $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0, \mathbf{P}_0)$ with known $\bar{\mathbf{x}}_0$ and $\mathbf{P}_0$.

2.2. Kalman filtering and smoothing

The interest in the linear-Gaussian model of Eq. (1) is usually in the computation of the sequence of filtering distributions $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, where we use the shorthand $\mathbf{y}_{1:k} = \{\mathbf{y}_j\}_{j=1}^k$. The Kalman filter provides an exact form of those as $p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k|\mathbf{m}_k, \mathbf{P}_k)$, for every $k = 1, \ldots, K$ [12]. The sequence of smoothing distributions (conditioned to the whole set of observations), can be also computed exactly by the RTS smoother yielding $p(\mathbf{x}_k|\mathbf{y}_{1:K}) = \mathcal{N}(\mathbf{x}_k|\mathbf{m}_k^s, \mathbf{P}_k^s)$. In both Kalman filter and RTS smoother, it is necessary to know the exact model, i.e., the parameters $\mathbf{A}$, $\mathbf{H}$, $\mathbf{Q}$, and $\mathbf{R}$. The explicit filter and the smoother algorithms can be found in [1].

2.3. EM framework for parameters estimation

In this work, we are interested in the joint estimation of the unknown $\mathbf{A}$ and the hidden states. Parametric estimation in state-space models has been considered in the literature through three main types of methods: EM algorithms [13, 14], optimization based methods [15], and Monte Carlo methods [16]. We will retain here the EM-based framework since it is well-suited for linear Gaussian models and it is flexible while benefiting from sounded convergence guarantees. EM is a method to iteratively find a maximum likelihood (ML) estimate of the parameters when the direct optimization of the posterior distribution is not feasible [17, 18]. It is also highly connected to majoration-minimization (MM) approaches, as it can be viewed as a special class for those [19]. The EM algorithm indeed alternates between a majoration step consisting in building an upper bound on the neg-log-likelihood function (E-step), and the minimization of this upper bound (M-step). In the context of state-space models with parameters $\boldsymbol{\theta}$, at each iteration $i \in \mathbb{N}$ of EM method, one majorizes $-\log p(\mathbf{y}_{1:K}|\boldsymbol{\theta}) \leq \mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i-1)})$ [13] with $\boldsymbol{\theta}^{(i-1)}$ the parameters value of the previous EM iterate and

$$
\begin{aligned}
&\mathcal{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i)}) \\
&= -\int p(\mathbf{x}_{0:K}|\mathbf{y}_{1:K}, \boldsymbol{\theta}^{(i)}) \log p(\mathbf{x}_{0:K}, \mathbf{y}_{1:K}|\boldsymbol{\theta}) \mathrm{d}\mathbf{x}_{0:K}.
\end{aligned}
\tag{2}
$$

Then, the M-step consists in defining $\boldsymbol{\theta}^{(i)}$ as a minimizer for $\mathcal{Q}(\cdot; \boldsymbol{\theta}^{(i)})$, so that, by construction, the sequence $(\boldsymbol{\theta}^{(i)})_{i \in \mathbb{N}}$ decreases monotonically the ML cost function, and convergence guarantees to the ML estimate can be obtained under suitable assumptions. Note that, although the EM algorithm was originally an algorithm for computing ML estimates, it can also be modified for computation of MAP estimates, as we show in the next section.

3. PROPOSED GRAPHEM APPROACH

In this section we introduce the GraphEM algorithm for the estimation of the $\mathbf{A}$ matrix under sparsity constraints encoded in the prior distribution $p(\mathbf{A})$ (see a discussion about the prior choice in Section 3.1). GraphEM aims at finding the maximum of $p(\mathbf{A}|\mathbf{y}_{1:K}) \propto p(\mathbf{A})p(\mathbf{y}_{1:K}|\mathbf{A})$, i.e., the MAP estimate of $\mathbf{A}$. It is direct to show that the maximum of $p(\mathbf{A}|\mathbf{y}_{1:K})$ coincides with the minimum of $\varphi_K(\mathbf{A}) = -\log p(\mathbf{A}) - \log p(\mathbf{y}_{1:K}|\mathbf{A})$. In the model (1), φ_K can be expressed recursively for $k = 1, \ldots, K$:

$$
\varphi_k(\mathbf{A}) = \varphi_{k-1}(\mathbf{A}) - \log p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{A}) \tag{3}
$$

$$
= \varphi_{k-1}(\mathbf{A}) + \frac{1}{2}\log|2\pi\mathbf{S}_k(\mathbf{A})|
$$

$$
+ \frac{1}{2}\mathbf{z}_k(\mathbf{A})^\top \mathbf{S}_k(\mathbf{A})^{-1}\mathbf{z}_k(\mathbf{A}), \tag{4}
$$

where $\mathbf{z}_k(\mathbf{A}) = \mathbf{y}_k - \mathbf{H}\mathbf{A}\mathbf{m}_{k-1}(\mathbf{A})$ and $\mathbf{S}_k(\mathbf{A})$ is the covariance matrix of the predictive distribution $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{y}_k; \mathbf{H}\mathbf{A}\mathbf{m}_{k-1}(\mathbf{A}), \mathbf{S}_k(\mathbf{A}))$, both being side products of the Kalman filter ran for a given $\mathbf{A}$ (see [1, Section 4.3]). Moreover, $\varphi_0(\mathbf{A}) = -\log p(\mathbf{A})$ is the regularization function. The direct minimization of φ_K is made difficult due

to its recursive form. The EM approach allows to construct an upper bound for it, that is more tractable and thus easier to minimize. Let us denote $\mathbf{G}_k = \mathbf{P}_k(\mathbf{A}')^\top(\mathbf{A}'\mathbf{P}_k(\mathbf{A}')^\top + \mathbf{Q})^{-1}$ the output of the RTS smoother for a given $\mathbf{A}' \in \mathbb{R}^{N_x \times N_x}$ and set $\boldsymbol{\Sigma} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}_k^s + \mathbf{m}_k^s(\mathbf{m}_k^s)^\top$, $\boldsymbol{\Phi} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}_{k-1}^s + \mathbf{m}_{k-1}^s(\mathbf{m}_{k-1}^s)^\top$, $\mathbf{C} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{P}_k^s\mathbf{G}_{k-1}^\top + \mathbf{m}_k^s(\mathbf{m}_{k-1}^s)^\top$. Then, following [1, Section 12.2.3], we can show that the following function, parametrized by $\mathbf{A}'$, majorizes the MAP objective function φ_K for every $\mathbf{A} \in \mathbb{R}^{N_x \times N_x}$:

$$\mathcal{Q}(\mathbf{A};\mathbf{A}') = \frac{K}{2}\text{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\top)\right)$$
$$+ \varphi_0(\mathbf{A}) + \mathcal{C} \quad (5)$$

where tr is the trace operator, and $\mathcal{C}$ is a constant term independent from $\mathbf{A}$. The GraphEM algorithm is then constructed as described in Alg. 1. The algorithm iterates alternating between an E-step and an M-step, ensuring the monotonic decrease of $\left\{\varphi_K(\mathbf{A}^{(i)})\right\}_{i\in\mathbb{N}}$. In the E-step, the function $\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i-1)})$ is formed by running the Kalman filter and RTS smoother where the state matrix in the model is set to $\mathbf{A}^{(i-1)}$. In the M-step, the matrix $\mathbf{A}^{(i)}$ is updated by minimizing the latest generated majorizing function $\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i-1)})$. In the following, we provide more details about the choice of the sparsity promoting prior and the M-step implementation.

Algorithm 1 GraphEM algorithm
***
Initialization of $\mathbf{A}^{(0)}$
For $i = 1, 2, \ldots$
    (E-step) Run the Kalman filter and RTS smoother by setting $\mathbf{A}' := \mathbf{A}^{(i-1)}$ and construct $\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i-1)})$ using Eq. (5).
    (M-step) Update $\mathbf{A}^{(i)} = \text{argmin}_A\left(\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i-1)})\right)$ (see Section 3.2)

3.1. Choice of the prior

Sparsity is a key feature in statistical data processing to limit the degrees of freedom in parametric models. It has been widely used for graphical model inference [20, 21], since a sparse matrix better allows to reveal interpretable and compact network of interdependencies between the entities. Since the ℓ_0 count measure is barely tractable, many statistical or learning approaches have invested in computable proxies, such as the ℓ_1 norm, also called Lasso penalty. In our problem, this amounts to set $\varphi_0(\mathbf{A}) = \gamma\|\mathbf{A}\|_1$ with γ a positive penalty parameter. The ℓ_1 penalty is non-smooth, though convex and can be efficiently handled thanks to proximal optimization methods, as we will show hereafter.

3.2. Computation in the M-step

At an iteration $i \in \mathbb{N}$, the M-step minimizes the function $\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i-1)})$, obtained by plugging $\mathbf{A}' := \mathbf{A}^{(i-1)}$ in Eq.(5).

This minimization problem, sometimes called Lasso regression [22], has been much studied in the literature of optimization [23, 24], and most of the methods proposed to solve it rely on the proximity operator [25, 26].[1] We recall that, for a proper, lower semi-continuous and convex function $f : \mathbb{R}^{N_x \times N_x} \mapsto]-\infty, +\infty]$, the proximity operator of f at $\widetilde{\mathbf{A}} \in \mathbb{R}^{N_x \times N_x}$ is defined as

$$\text{prox}_f(\widetilde{\mathbf{A}}) = \text{argmin}_{\mathbf{A}}\left(f(\mathbf{A}) + \frac{1}{2}\|\mathbf{A} - \widetilde{\mathbf{A}}\|_F^2\right). \quad (6)$$

Let us decompose $\mathcal{Q}(\mathbf{A};\mathbf{A}^{(i-1)}) = f_1(\mathbf{A}) + f_2(\mathbf{A})$, where $f_1(\mathbf{A}) = \frac{K}{2}\text{tr}\left(\mathbf{Q}^{-1}(\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^\top - \mathbf{A}\mathbf{C}^\top + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\top)\right)$, and $f_2 = \varphi_0$ is the prior that we have discussed in Section 3.1. Function f_1 is quadratic, so that, for $\theta > 0$, we have

$$\text{prox}_{\theta f_1}(\widetilde{\mathbf{A}}) =$$
$$\text{mat}\left(\left[\mathbf{Id}\otimes(K\mathbf{Q}^{-1}) + (\theta\boldsymbol{\Phi}^{-1})\otimes\mathbf{Id}\right]^{-1}\text{vec}(K\mathbf{Q}^{-1}\mathbf{C}\boldsymbol{\Phi}^{-1})\right),$$

due to the fact that the Lyapunov equation $\mathbf{X}\mathbf{A} + \mathbf{A}\mathbf{Y} = \mathbf{Z}$ has for solution, $\mathbf{A} = \text{mat}((\mathbf{Id}\otimes\mathbf{X} + \mathbf{Y}^\top\otimes\mathbf{Id})^{-1}\text{vec}(\mathbf{Z}))$, with vec and mat the re-ordering operations, using lexicographic order and $\otimes$ the Kronecker product. Note that if $\mathbf{Q} = \sigma_\mathbf{Q}^2\mathbf{Id}$, i.e., the state noise is isotropic, the expression simplifies into

$$\text{prox}_{\theta f_1}(\widetilde{\mathbf{A}}) = \left(\frac{\theta K}{\sigma_\mathbf{Q}^2}\mathbf{C} + \widetilde{\mathbf{A}}\right)\left(\frac{\theta K}{\sigma_\mathbf{Q}^2}\boldsymbol{\Phi} + \mathbf{Id}\right)^{-1}.$$

Second, the proximity operator for f_2 is the simple soft thresholding operator:

$$\text{prox}_{\theta f_2}(\widetilde{\mathbf{A}}) = \left(\text{sign}(\widetilde{A}_{nm}) \times \max(0, |\widetilde{A}_{nm}| - \theta)\right)_{1\le n,m\le N_x}$$

Since function f_1 is differentiable, a natural strategy would be to adopt the iterative thresholding method [27] or an accelerated version for it [28]. Here, following the comparative analysis from [29], we will prefer an algorithm that activates both terms via their proximity operator. More precisely, we will make use of the Douglas-Rachford (DR) algorithm, a fixed-point strategy for convex optimization that benefits from sounded convergence guarantees [11] and has demonstrated its great practical performance in matrix optimization problems related to graphical inference applications [30]. This leads to Alg. 2, that generates a sequence $\{\mathbf{A}_n\}_{n\in\mathbb{N}}$ guaranteed to converge to a minimizer of $f_1 + f_2$. In practice, for the iteration i in GraphEM, we run DR method with $\theta = 1$ and initialization $\mathbf{Z}_0 = \mathbf{A}^{(i-1)}$, i.e. the majorant function tangency point. Moreover, we stop the DR loop as soon as $|(f_1 + f_2)(\mathbf{A}_{n+1}) - (f_1 + f_2)(\mathbf{A}_n)| \le \varepsilon$ (typically, $\varepsilon = 10^{-3}$).

---

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

Algorithm 2 Douglas-Rachford algorithm for M-step

Set $\mathbf{Z}_0 \in \mathbb{R}^{N_x \times N_x}$ and $\theta \in (0, 2)$
For $n = 1, 2, \ldots$
$\quad \mathbf{A}_n = \mathrm{prox}_{\theta f_2}(\mathbf{Z}_n)$
$\quad \mathbf{V}_n = \mathrm{prox}_{\theta f_1}(2\mathbf{A}_n - \mathbf{Z}_n)$
$\quad \mathbf{Z}_{n+1} = \mathbf{Z}_n + \theta(\mathbf{V}_n - \mathbf{A}_n)$
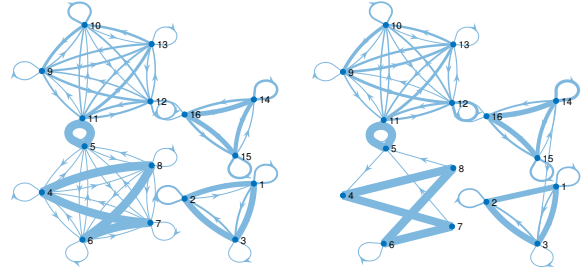
4. EXPERIMENTAL EVALUATION

The synthetic data are generated following model (1) with $N_x = N_y$. A block-diagonal matrix $\mathbf{A}$ is considered, composed with b blocks of dimensions $(b_j)_{1 \leq j \leq b}$, so that $N_x = \sum_{j=1}^{b} b_j$. Each j-th diagonal block of $\mathbf{A}$ is randomly generated. Note that a projection on the space of matrices with maximum singular value less than one is applied, so that matrix $\mathbf{A}$ leads to a stable Markov process. Furthermore, we set $\mathbf{H} = \mathrm{Id}$, $K = 10^3$ and $\mathbf{Q} = \sigma_{\mathbf{Q}}^2 \mathrm{Id}$, $\mathbf{R} = \sigma_{\mathbf{R}}^2 \mathrm{Id}$, $\mathbf{P}_0 = \sigma_{\mathbf{P}}^2 \mathrm{Id}$ with $(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$ some predefined values. Following this procedure, we created four datasets, whose parameters are provided in Table 1. Measures of quality to assess the performance of the methods are the relative mean square error (RMSE) on $\mathbf{A}$, and the precision, recall, specificity, accuracy, and F1 score for the graph edge detection, when applying a threshold of 10^{-10} on the entries of matrix $\mathbf{A}$.

Dataset	$(b_j)_{1 \leq j \leq b}$	$(\sigma_{\mathbf{Q}}, \sigma_{\mathbf{R}}, \sigma_{\mathbf{P}})$
A	$(3, 3, 3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
B	$(3, 3, 3)$	$(1, 1, 10^{-4})$
C	$(3, 5, 5, 3)$	$(10^{-1}, 10^{-1}, 10^{-4})$
D	$(3, 5, 5, 3)$	$(1, 1, 10^{-4})$

Table 1. Description of datasets

For each dataset, we ran GraphEM algorithm with the stopping rule $|\varphi_K(\mathbf{A}^{(i)}) - \varphi_K(\mathbf{A}^{(i-1)})| \leq 10^{-3}$. In practice, a maximum number of 50 iterations is sufficient to reach this criterion. The algorithm is initialized setting a random and dense matrix $\mathbf{A}^{(0)}$, ensuring the stability of the process. Parameter γ, balancing the weight of the sparse prior, is optimized on a single realization thanks to a manual grid search, so as to maximize the accuracy score. We also provide the results obtained by the ML estimator, also computed using EM algorithm (MLEM). In this case, the M-step has a closed form solution [1, Th.12.5]. In addition, we compare with two Granger-causality approaches [31] for graphical modeling. The first algorithm, pairwise Granger Causality (PGC) explores the $N_x(N_x\text{-}1)$ possible dependencies among two nodes, at each time independently from the rest. The second approach, the conditional Granger Causality (CGC), operates similarly but for each pair of nodes, it also takes into account the information of the other $N_x - 2$ signals, in order to evaluate whether one node provides information to the other when the rest of signals are observed. We also do a manual grid search for finetuning the parameters of both PGC and CGC

	method	RMSE	accur.	prec.	recall	spec.	F1
A	GraphEM	0.081	0.9104	0.9880	0.7407	0.9952	0.8463
	MLEM	0.149	0.3333	0.3333	1	0	0.5
	PGC	-	0.8765	0.9474	0.6667	0.9815	0.7826
	CGC	-	0.8765	1	0.6293	1	0.7727
B	GraphEM	0.082	0.9113	0.9914	0.7407	0.9967	0.8477
	MLEM	0.148	0.3333	0.3333	1	0	0.5
	PGC	-	0.8889	1	0.6667	1	0.8
	CGC	-	0.8889	1	0.6667	1	0.8
C	GraphEM	0.120	0.9231	0.9401	0.77	0.9785	0.8427
	MLEM	0.238	0.2656	0.2656	1	0	0.4198
	PGC	-	0.9023	0.9778	0.6471	0.9949	0.7788
	CGC	-	0.8555	0.9697	0.4706	0.9949	0.6337
D	GraphEM	0.121	0.9247	0.9601	0.7547	0.9862	0.8421
	MLEM	0.239	0.2656	0.2656	1	0	0.4198
	PGC	-	0.8906	0.9	0.6618	0.9734	0.7627
	CGC	-	0.8477	0.9394	0.4559	0.9894	0.6139

Table 2. Results for GraphEM, MLEM, PGC and CGC.

Fig. 1. True graph (left) and GraphEM estimate (right) for dataset C.

(more information can be found in [8]). Note that PGC and CGC do not estimate a weighted graph but a binary one, so that RMSE is not calculated in those cases.

The results, averaged on 50 realizations, are presented in Table 2. MLEM does not promote sparsity in the graph which explains the poor results in terms of edge detectability. Moreover, GraphEM provides a better RMSE score on all examples. Regarding the graph structure, we can observe that GraphEM has also better detection scores, when compared with both PGC and CGC. We also display an example of graph reconstruction for dataset C in Fig. 1, illustrating the ability of GraphEM to recover the graph shape and weights.

5. CONCLUSION

This work proposes the GraphEM algorithm, an expectation-minimization method for the estimation of the linear operator that encodes the hidden state relationships in a linear-Gaussian state-space model. Due to the use of a Lasso penalization term, we incorporate a sparsity constraint in our model. This is particularly well suited for modeling and representing the state entries interactions as a compact and interpretable graph. A proximal splitting algorithm is employed for solving the inner minimization problem of our EM approach. Numerical results illustrate the great performance of the method, when compared to several techniques for graphical modeling.

References

[1] S. Sarkka, *Bayesian Filtering and Smoothing*, 3 edition, 2013.

[2] S. J. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 401–422, March 2004.

[3] P. M. Djurić, M. F. Bugallo, and J. Míguez, "Density assisted particle filters for state and parameter estimation," in *Proceedings of the 29th IEEE ICASSP*, May 2004.

[4] M. Eichler, "Graphical modelling of multivariate time series," *Probability Theory and Related Fields*, vol. 153, no. 1, pp. 233–268, Jun. 2012.

[5] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2189–2199, Aug. 2004.

[6] D. Barber and A. T. Cemgil, "Graphical models for time-series," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 18–28, Nov 2010.

[7] A. Pirayre, C. Couprie, L. Duval, and J. Pesquet, "BRANE Clust: Cluster-assisted gene regulatory network inference refinement," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 3, pp. 850–860, May 2018.

[8] 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*, vol. 12, no. 1, pp. 143–155, Jan. 2019.

[9] C. Ravazzi, R. Tempo, and F. Dabbene, "Learning influence structure in sparse social networks," *IEEE Transactions on Control of Network Systems*, vol. PP, pp. 1–1, 12 2017.

[10] J. Richiardi, S. Achard, B. Horst, , and D. Van De Ville, "Machine learning with brain graphs," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 58–70, 2013.

[11] P.L. Combettes and J.C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 564–574, Dec. 2007.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[13] R. H Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[14] B. Thiesson, D. M. Chickering, D. Heckerman, and C. Meek, "Arma time-series modeling with graphical models," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, United States, 2004, UAI '04, pp. 552–560, AUAI Press.

[15] R. Olsson, K. Petersen, and T. Lehn-Schioler, "State-space models: from the EM algorithm to a gradient approach," *Neural Computation*, vol. 19, no. 4, pp. 1097–1111, 2007.

[16] N. Kantas, A. Doucet, and J. Singh, S.and Maciejowski, "An overview of sequential Monte Carlo methods for parameter estimation in general state-space models," in *Proceedings IFAC Symposium on System Identification (SYSID)*, 2009.

[17] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[18] 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)*, vol. 39, no. 1, pp. 1–38, 1977.

[19] D. R. Hunter and K. Lange, "A tutorial on mm algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[20] 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 - Volume 70*. 2017, ICML'17, pp. 440–448, JMLR.org.

[21] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical LASSO," *Biostatistics*, vol. 9, no. 3, pp. 432–441, jul 2008.

[22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[23] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *Machine Learning: ECML 2007*, J. N. Kok, J. Koronacki, R.L. Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds., Berlin, Heidelberg, 2007, pp. 286–297, Springer Berlin Heidelberg.

[24] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations Trends Machine Learning*, vol. 4, no. 1, pp. 1–106, Jan. 2012.

[25] P.L. Combettes and J.C. Pesquet, *Proximal Splitting Methods in Signal Processing*, Springer-Verlag, New York, 2010.

[26] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2017, 2nd edition.

[27] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[28] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[29] P. L. Combettes and L. E. Glaudin, "Proximal activation of smooth functions in splitting algorithms for convex image recovery," Tech. Rep., 2019, https://arxiv.org/pdf/1803.02919.pdf.

[30] A. Benfenati, E. Chouzenoux, and J.-C. Pesquet, "A proximal approach for a class of matrix optimization problems," Tech. Rep., 2018, https://arxiv.org/abs/1801.07452.

[31] S. L. Bressler and A. K. Seth, "Wiener–Granger causality: a well established methodology," *NeuroImage*, vol. 58, no. 2, pp. 323–329, 2011.