Using graph prior to learn network Granger causality

Lucas Zoroddu
Centre Borelli
ENS Paris Saclay
Gif-sur-Yvette, France
lucas.zoroddu@ens-paris-saclay.fr

Pierre Humbert
Laboratoire de mathématiques d'Orsay
Université Paris-Saclay
Orsay, France
pierre.humbert@universite-paris-saclay.fr

Laurent Oudre
Centre Borelli
ENS Paris Saclay
Gif-sur-Yvette, France
laurent.oudre@ens-paris-saclay.fr

***Abstract*—Learning graphs of Granger causalities from multivariate time series is essential for understanding relationships between several entities. This approach has found wide applications in various domains such as economics, finance, neurosciences, genetic etc. However, accurately estimating Granger causality graphs in high-dimensional settings with limited samples remains a challenge. In this study, we introduce a model that leverages prior knowledge in the form of a noisy graph to learn a graph of Granger causalities assuming sparsity. We demonstrate the convergence of our fitting algorithm and present experimental results on synthetic and real-world datasets to show the advantages of our method over existing alternatives, in particular in settings with limited available samples.**

***Index Terms*—Graph Learning, Granger causality, Vector Autoregressive Models**

I. Introduction

Multivariate time series analysis plays a crucial role in understanding and forecasting real-world phenomena involving multiple interdependent variables. Within such complex systems, Granger causality [1] graph has emerged as a powerful tool to uncover directional relationships and causal influences. It has therefore been the basis for a wide range of applications in fields such as economics [2], neuroscience [3], gene regulation [4], protein-protein interactions [5] etc.

In order to learn a Granger causality graph, a standard approach is to estimate the parameters of a Vector AutoRegressive model (VAR) from an observed multivariate time series [6]. However, this is a non-trivial problem, especially in a high-dimensional context with few samples. To address this challenge, several regularization methods have been proposed, from both a frequentist and a Bayesian point of view. From the frequentist perspective, Basu and Michailidis [7] investigated a Group Lasso penalty and different variants of sparsity-inducing penalties were also presented in [8], [9], and [10]. From the Bayesian perspective, several prior distributions on the parameters of the VAR were investigated. These include Gaussian prior [11], Gaussian-inverted Wishart prior [12], or hierarchical normal priors [13]. However, as underlined in [14], the resulting Granger causality graph can be fairly dense or, on the contrary, very disconnected, which can be in contradiction with scientific background knowledge.

To address this issue, it is possible to take into account additional information summarized in the form of a knowledge graph. For example, in the context of gene network analysis, genes may be organized into distinct pathways, and it is often observed that connections within a pathway are more frequent than connections between pathways [15]. Thus, the authors of [4] ensure that the learned Granger graph follows this prior knowledge by adding a particular penalization term to the optimization problem. Another approach studied in [14] is to assume a tree-rank prior distribution, forcing the learned Granger graph to be a subgraph of the union of spanning trees. Another important example is when dealing with signals driven by a physical process and recorded by sensors at several locations. In this case, the Euclidean k-NN graph can make sense and is often used in the Graph Signal Processing community [16] as a prior in order to perform filtering, denoising or prediction. Nevertheless, note that, most methods in the literature presuppose perfect knowledge of prior data, which is unrealistic.

Contributions: In this paper, we introduce a VAR model that incorporates prior knowledge about the relationship between time series in the form of a graph. Unlike existing methods, since the prior graph is rarely accurate, we assume in the model that this prior is noisy. To estimate the associated graph of Granger causalities, we compute the MAP using a 2 block coordinate descent algorithm that is proven to converge to a set of stationary points. Finally, we perform experiments on both synthetic and real-world data showing better performances than state-of-the-art models in settings with few samples across several levels of noise.

II. Preliminaries

A VAR model of order $d \geq 1$, denoted VAR(d), explains the values of a multivariate time series at time t using a linear combination of its d previously observed values. Formally, given d matrices $(\mathbf{C}^\tau)_{\tau=1}^d$ in $\mathbb{R}^{p\times p}$, a VAR(d) is defined at each time $t = 1, 2, \dots$ by:

$$X[t] = \sum_{\tau=1}^{d} \mathbf{C}^\tau X[t-\tau] + \varepsilon[t] \,, \tag{1}$$

where $X[t] = (X_1[t], ..., X_p[t])$ is a random p-dimensional time series and $\varepsilon[t] \sim \mathcal{N}(0, \sigma_X^2 I_p)$, $\sigma_X > 0$, is some innovation noise. In practice, VAR models are often used to analyze

certain aspects of the relationships between several variables of interest. Indeed, it can be shown that the matrices $\{\mathbf{C}^\tau\}_{\tau=1}^d$ in Equation (1) capture specific temporal dependencies between the p time series and are associated with the notion of Network Granger causality (NGC).

Definition 1 (Network Granger Causality)**.** *For i, j in $[\![1, p]\!]$, the i-th time series $X_i := (X_i[t])_{t=1,2,...}$ is called a Granger cause of $X_j := (X_j[t])_{t=1,2,...}$ if at least one element of $\left\{\mathbf{C}^\tau_{i,j} \mid \tau = 1, ..., d\right\}$ is non zero.*

NGC can be seen as an extension of the Granger causality between 2 variables to p variables. Note that, like for the case $p = 2$, NGC does not necessarily capture true causal relationships, but rather indicates the power of prediction of some variables to others. Nevertheless, NGC remains a powerful tool for understanding interactions between random time series, and its estimation is of practical interest.

III. Modeling framework

For simplicity, in the following, we only consider VAR(1) models i.e. $d = 1$ and we only need to learn one matrix $\mathbf{C}$. Note that this assumption is not limiting since a *VAR(d)* model can always be written as a *VAR(1)* model [6] so our method is still applicable in the general case with slight modifications. In general, estimating the parameters of the VAR model (1), i.e. the matrix $\mathbf{C}$, requires the observation of a long stationary realisation of the p-dimensional time series. However, in many applications, we only observe short replicas of the time series and additional information must be incorporated into the model to obtain accurate estimates.

A. Model

In this section, we propose to leverage prior knowledge on the structure of the matrix $\mathbf{C}$ taking the form of a graph. More specifically, we assume that the coefficients $\{\mathbf{C}_{i,j}\}_{i,j}$ are drawn from independent centered Laplace distribution with variances equal to the adjacency matrix coefficients of a given graph $\left\{\mathbf{A}^*_{i,j}\right\}$. The choice of a Laplacian distribution is motivated by the will to learn a sparse graph. Thus, if $\mathbf{A}^*_{i,j}$ is small, the value of the associated coefficient $\mathbf{C}_{i,j}$ is close to zero (meaning that there is no Granger causality between the two time series). However, since the prior information is rarely accurate for most applications, we will assume that we do not know the true matrix $\mathbf{A}^*$ but rather a noisy version denoted $\mathbf{A}^{\text{prior}}$. This assumption will lead to a more robust model that can refine the prior over iterations.

Formally, given an adjacency matrix $\mathbf{A}^*$, we consider the statistical model defined by:

$$\begin{aligned} &\mathbf{A}^{\text{prior}}_{i,j} \sim \mathcal{N}(\mathbf{A}^*_{i,j}, \sigma^2_{\mathbf{A}}) \ , \ \sigma_{\mathbf{A}} > 0 \ , \ i, j = 1, ..., p \\ &\mathbf{C}^*_{i,j} \sim \mathit{Laplace}(0, \mathbf{A}^*_{i,j}) \ , \ i, j = 1, ..., p \\ &X[t] \mid \mathbf{C}^* \sim \mathcal{N}(\mathbf{C}^* X[t-1], \sigma^2_X I_p) \ , \ \sigma_X > 0 \ . \end{aligned} \tag{2}$$

B. Estimation of the parameters

Given N independent trajectories $X_1[1 : t], ..., X_N[1 : t]$ drawn from the statistical model (2) and a matrix $\mathbf{A}^{\text{prior}}$, we want to estimate both $\mathbf{C}^*$ (VAR parameters) and $\mathbf{A}^*$ (denoised adjacency matrix) by computing the Maximum a Posteriori (MAP):

$$\begin{aligned} \widehat{\mathbf{A}}, \widehat{\mathbf{C}} = \underset{\mathbf{A},\mathbf{C}}{\arg\max} \ L(\mathbf{A}, \mathbf{C} \mid \{X_i[1:t]\}_{i=1}^N, \mathbf{A}^{\text{prior}}) \\ \text{subject to } \ \mathbf{A} \geq 0, \ \mathbf{A} \in S_p(\mathbb{R}) \ , \end{aligned} \tag{3}$$

where $L(\cdot)$ is the likelihood function and $S_p(\mathbb{R})$ is the set of symmetric matrices. Recall that we consider a VAR(1) model, hence $t = 2$.

Remark 1. *We impose $\mathbf{A} \in S_p(\mathbb{R})$ because this variable represents the adjacency matrix of the denoised undirected prior graph knowledge. Keeping $\mathbf{A}$ symmetric also allows us to impose spectral or adjacency constraints and use the framework introduced in [17]*

Proposition 1. *Computing the MAP leads to the following optimization problem:*

$$\begin{aligned} \min_{\mathbf{A}\,,\,\mathbf{C}} \ & \frac{1}{N}\sum_{n=1}^{N} \|X_n[t] - \mathbf{C}X_n[t-1]\|_2^2 \\ & + \lambda \sum_{1\leq i<j\leq p} \frac{|\mathbf{C}_{i,j}| + |\mathbf{C}_{j,i}|}{\mathbf{A}_{i,j}} + 2\lambda \sum_{1\leq i<j\leq p} \log(2\mathbf{A}_{i,j}) \\ & + \gamma \left\|\mathbf{A} - \mathbf{A}^{\mathit{prior}}\right\|_F^2 \end{aligned}$$

$$\textit{subject to } \ \mathbf{A}_{i,j} \geq 0 \ , \ \mathbf{A}_{i,j} = \mathbf{A}_{j,i} \ , \ 1 \leq i < j \leq p \tag{4}$$

where λ and γ are hyper parameters. Note that we only optimize over the upper diagonal values and we set $\mathbf{A}_{j,i} = \mathbf{A}_{i,j}$ for $i < j$ to address the symmetry constraint of Problem 3.

Proof. (Sketch) The likelihood of the posterior distribution is calculated by a direct application of the Bayes' formula. □

Equation in Proposition (1) contains 3 terms:

1) $\frac{1}{N}\|X[t] - \mathbf{C}X[t-1]\|_2^2$ corresponds to the Least Square problem objective: this term al- lows to measure the difference between the original signals and their reconstructions. Recall that we only consider in this section VAR(1) model, hence $t = 2$.
2) $\sum_{1\leq i<j\leq p} \frac{|\mathbf{C}_{i,j}| + |\mathbf{C}_{j,i}|}{\mathbf{A}_{i,j}} + 2\sum_{1\leq i<j\leq p} \log(2\mathbf{A}_{i,j})$ is the penalization term that takes into account the graph prior knowledge. This penalization is inspired by the one used in Adaptive Lasso models, where the terms $1/\mathbf{A}_{i,j}$ act as weights. The higher the $\mathbf{A}_{i,j}$, the closer i and j are in the graph, and the lower the penalty for $\mathbf{C}_{i,j}$ and $\mathbf{C}_{j,i}$. It should be noted that the additional term composed of the sum of log is a normalization term linked to the associated statistical model (2).
3) $\left\|\mathbf{A} - \mathbf{A}^{\text{prior}}\right\|_F^2$ is a regularization term to take into account that $\mathbf{A}^{\text{prior}}$ is not necessarily the optimal prior knowledge and could therefore be further refined. Actually, adding this term in the optimization problem is equivalent to imposing

a normal prior distribution to the coefficient of $\mathbf{A}_{i,j}$ (see (2)). In practice, this term allows to increases the robustness to the prior knowledge noise.

C. A-AdaptiveLasso (AALasso)

The function to minimize in Proposition (1) is not convex in $(\mathbf{A}, \mathbf{C})$ since it is not convex in $\mathbf{A}$. However, as the function is convex in $\mathbf{C}$ (adaptive lasso problem) and we have a closed form for the roots of the derivative in $\mathbf{A}$, alternating minimization is a good way to solve this problem.

a) $\mathbf{C}$ update: For fixed $\mathbf{A}$, the optimization problem (4) with respect to $\mathbf{C}$ is:

$$\min_{\mathbf{C}} \ \frac{1}{N}\sum_{n=1}^{N}\left\|X^{(n)}[t]-\mathbf{C}X^{(n)}[t-1]\right\|_2^2 + \lambda \sum_{1\le i<j\le p}\frac{|\mathbf{C}_{i,j}|+|\mathbf{C}_{j,i}|}{\mathbf{A}_{i,j}} \ . \quad (5)$$

. From Eq. (5), we see that the optimization step in $\mathbf{C}$ is an Adaptive Lasso problem with weights equal to $1/\mathbf{A}_{i,j}$ [18]. In this study we used the `cvxpy` [19] library to solve Problem 5.

b) $\mathbf{A}$ update:

For fixed $\mathbf{C}$, the optimization problem (4) with respect to $\mathbf{A}$ is:

$$\min_{\mathbf{A}} \ \lambda \sum_{1\le i<j\le p}\frac{|\mathbf{C}_{i,j}|+|\mathbf{C}_{j,i}|}{\mathbf{A}_{i,j}} + 2\lambda \sum_{1\le i<j\le p}\log(2\mathbf{A}_{i,j}) + \gamma\left\|\mathbf{A}-\mathbf{A}^{\text{prior}}\right\|_F^2 ,$$
$$\text{subject to } \ \mathbf{A}_{i,j}\ge 0\ ,\ \mathbf{A}_{i,j}=\mathbf{A}_{j,i}\ \ ,\ \ 1\le i<j\le p\ .$$

To address the symmetry constraint, a straightforward way is to optimize over the upper diagonal values and to set $\mathbf{A}_{j,i} = \mathbf{A}_{i,j}$ for $i < j$. The minimisation can then be carried out by directly calculating the exact minimum, which is given in the next proposition.

Proposition 2. *The roots of the derivative with respect to $\mathbf{A}_{l,m}$ of the objective function* (6) *are:*

$$z_k = \frac{\mathbf{A}_{l,m}^{prior}}{3} + e^{2ik\pi/3}\sqrt[3]{\frac{1}{2}\left(-q+\sqrt{\frac{\Delta}{27}}\right)} + e^{-2ik\pi/3}\sqrt[3]{\frac{1}{2}\left(-q-\sqrt{\frac{\Delta}{27}}\right)} \ \ , \ \ k=0,1,2\ , \quad (6)$$

where

$$\begin{cases} p = -\frac{(\mathbf{A}_{l,m}^{prior})^2}{3}+\frac{\lambda}{2\gamma} \\ q = -\frac{\mathbf{A}_{l,m}^{prior}}{3}\left(\frac{8\gamma(\mathbf{A}_{l,m}^{prior})^2}{9}-2\lambda\right) - \lambda\left(|\mathbf{C}_{l,m}|+|\mathbf{C}_{m,l}|\right) \\ \Delta = 4p^3+27q^2\ . \end{cases}$$

Furthermore, there exists at least one positive root for the derivative with respect to $\mathbf{A}_{l,m}$, and the global minimum on the interval $]0,+\infty[$ is attained at one of these roots.

Theorem 1. *(Convergence of AALasso)*
The sequence $\left\{(\mathbf{C}^{(r)},\mathbf{A}^{(r)})\right\}_{r=1,2,\dots}$ generated by the 2 blocks alternating minimization is defined and bounded. Moreover, every cluster point is a stationary point of the MAP (1).

Proof. (Sketch) The proof consists of verifying that our objective function satisfies the assumptions in point (3) of Theorem 4.1 of [20]. □

Algorithm 1: Fitting algorithm.

input : N_{iter}, λ, γ, $\mathbf{A}^{\text{prior}}$
output: $\widehat{\mathbf{C}}$, $\widehat{\mathbf{A}}$
$\mathbf{A}^{(0)} \leftarrow \mathbf{A}^{\text{prior}}$
for $i \leftarrow 1$ **to** N_{iter} **do**
  $C^{(i)} \leftarrow f_{\mathbf{C}}(\mathbf{C}, \mathbf{A}^{(i-1)})$ where $f_{\mathbf{C}}$ denotes the update in (III-C).
  $\mathbf{A}^{(i)} \leftarrow f_{\mathbf{A}}(\mathbf{C}^{(i)}, \mathbf{A})$ where $f_{\mathbf{A}}$ denotes the update in (III-C).
return $\mathbf{C}^{(N_{\text{iter}})}, \mathbf{A}^{(N_{\text{iter}})}$.

IV. Experiments

Several experiments were conducted using both synthetic and real datasets to assess the performance of AALasso. Quantitative results are exclusively provided for synthetic data due to the unavailability of ground truth graphs for the majority of real-world datasets. Consequently, synthetic data allow to evaluate the performance of our method, while real data are used to present visual results.

A. Task and evaluation metrics

In these experiments, the objective is to learn a Network Granger causality (NGC) from given multivariate time series and a prior network. We suppose that these series follow a VAR(1) model, hence, learning the NGC is equivalent to fit the VAR parameters, i.e., given $X \in \mathbb{R}^{p\times N}$ and $\mathbf{A}^{\text{prior}}$ in $\mathbb{R}^{p\times p}$ we want to estimate the matrix $\mathbf{C}$.
Since VAR models are usually employed for forecasting tasks, a standard metric to evaluate estimators is the normalized Root Mean Square Error (nRMSE) of the one step predictions. Let $X[t]$ be a multivariate time series and $\hat{X}[t]$ be the reconstruction using the fitted VAR model at time t, the nRMSE is defined by:

$$\text{nRMSE}(\hat{X}) := \sqrt{\frac{\sum_t \left\|\hat{X}[t]-X[t]\right\|_2^2}{\sum_t \|X[t]\|_2^2}} \ .$$

Note that, the main objective of this paper is to learn the underlying NGC, so we are more interested in learning a relevant graph than allowing a good reconstruction (even though the two tasks are correlated). To evaluate the quality of

the learned graph, we compute the F1-score between $\widehat{\mathbf{C}}$ and $\mathbf{C}^*$ (see e.g. [21] for more information). This metric is defined as follows:

$$\text{F1-score} := \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where

$$\text{precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

and

$$\text{recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}.$$

Note that the F1-score is only available for synthetic data as we need to have access to the true graph (the true VAR model). Although these two measures are related (a good graph should lead to a good reconstruction), it should be noted that a good reconstruction can be achieved by a relatively dense graph. Given that sparsity is a desired property, the F1 score is used to understand whether the learned graph can efficiently reconstruct time series while avoiding irrelevant edges.

B. Methods

The aim of these experiments is to show that AALasso can exploit prior knowledge to improve its performance compared to existing methods. We compare our estimator to the classical estimators: the Lasso and the Adaptive Lasso with weights equal to the least squares estimates (noted LS + ALasso, cf [18]). Moreover, since the first step of our algorithm is equivalent to solve an Adaptive Lasso problem assuming that the weights are given by $W_{i,j} = \frac{1}{\mathbf{A}^{\text{prior}}_{i,j}}$, we compare our method with this first step (denoted 1-AALasso) to demonstrate the usefulness to perform several steps. Note that we do not show the performances of the Least Squares estimator since the results are poor in settings with only few samples. Finally, note that the Lasso and LS+ALasso algorithms do not take into account the prior matrix, so the prior noise will not impact their results.

C. Synthetic data

The synthetic data were generated with respect to the statistical model (2). To define the matrix $\mathbf{A}^*$, we first generate $p = 40$ points in $[0,1]^2$ uniformly at random. Then, we construct a matrix $\mathbf{D} \in \mathbb{R}^{p \times p}$ by applying a Gaussian kernel $K_\sigma := (x,y) \mapsto \exp\left(\frac{\|x-y\|_2^2}{\sigma^2}\right)$, $(x,y) \in \mathbb{R}^2$, to the Euclidean pairwise distances, taking their median values for σ. $\mathbf{A}^*$ is obtained by randomly setting to 0 a ratio $\tau_m = 0.5$ of values of $\mathbf{D}$ (mispecified edges) and cutting to 0 values smaller than $\tau = 0.7$ to promote sparsity. Finally, VAR parameters are drawn from $Laplace(0, \mathbf{A}^*_{i,j})$ and $\mathbf{A}^{\text{prior}} = \mathbf{D} + \epsilon$, where ϵ is a symmetric matrix where subdiagonal values are sampled from iid Gaussian distribution with variance σ_A^2 (varying in $\{0.02, 0.1, 0.25, 0.35\}$ to test several level of noises). At the end, for each experiment, we generate time series following the VAR model with lengths $N = \{2 \times 40, 2 \times 100, \ldots, 2 \times 250\}$ different time series $X[t] \sim \mathcal{N}(\mathbf{C}^* X[t-1], \sigma_X^2)$, $\sigma_X^2 = 0.1$,

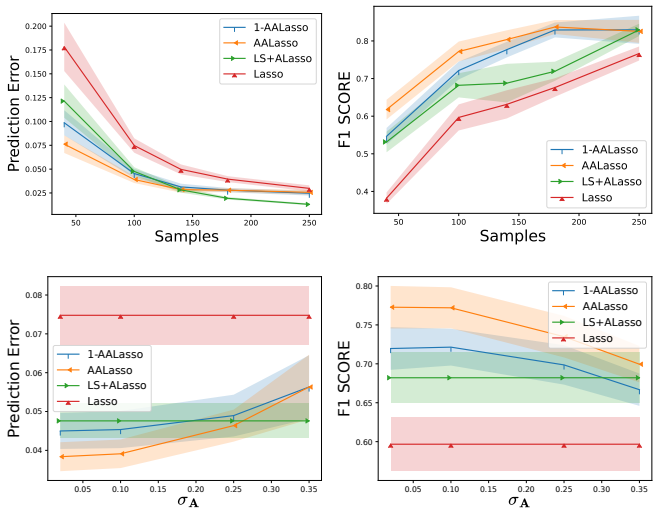


Fig. 1. Top panels: rNMSE and F1-score in function of the Signal to Noise Ratio in dB using $N = 40$ samples for training. Bottom panels: rNMSE and F1-score in function of the number of samples used for training using a SNR equals to 15. We plotted the 90% confidence intervals.

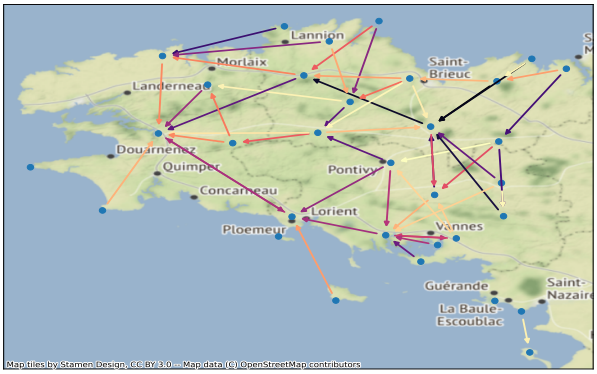
which we split into training and test sets of equal sizes. We repeated this procedure 20 times for each value of N.

For all of the 20 experiments, we performed $N_{\text{iter}} = 10$ iterations in the alternating minimization algorithm (1) using half of the training set and the parameters λ and γ were selected via cross validation minimizing the nRMSE over the second half. We compared our method both to the Lasso estimator and the Adaptive Lasso estimator using weights given by the least squares estimator (see [18] for details) and both metrics were computed on the test set.

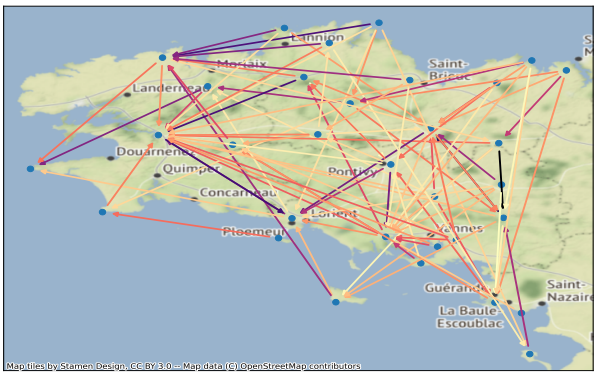
The results in Figure 1 exhibit better reconstruction and greater F1-score when utilizing AALasso rather than vanilla methods when the number of samples is lower than 140. From 40 to 140 samples, AALasso returns F1-scores from 0.6 to 0.8 while LS+ALasso F1-score ranges from 0.5 and 0.73 (an average gain of 0.1). Thus, our algorithm effectively leverages the additional information in settings with few samples, and our approach enables fine-tuning of the graph while remaining a good forecasting power. When the number of samples increases, the LS+ALasso estimator provides better reconstruction than AALasso. However, recall that we are interested in a graph learning task so the F1-score is more informative than the reconstruction error, and shows satisfying results even for relatively large number of samples (until a certain threshold, here 250 samples).
Finally, the difference of results between the first iteration and the complete optimization process of AALasso points out the interest of the alternating minimization.

D. Real data

We tested our method on the Molène dataset, which consists of temperatures recorded by sensors at 32 locations in Brittany. Also we considered the first derivative of the signals rather



(a) AALasso



(b) Lasso

Fig. 2. Results on the Molène dataset for (a) AALasso and (b) Lasso. Darker colors indicate larger weight.

than original signals in order to verify wide-sense stationary assumptions of VAR models. We trained the models with 80 points, still selecting λ (for Lasso and AALasso) and γ by cross-validation. Figure (2) compares the resulting graphs of Granger causalities using our method AALasso and a Lasso. We observe that the graph returned by AALasso is sparse while remaining almost connected and allows a good visualization of the process. Moreover, contrarily to the Lasso one, it is consistent with the Euclidean structure.

Remark 2. *Note that some points are not connected (no incoming edges), meaning that their are independent from the others.*

V. Conclusion

In this paper, we have presented a method that demonstrates its efficiency in learning Granger causalities under high-dimensional settings with limited samples. By effectively incorporating prior knowledge in the form of a noisy adjacency matrix, our method allows us to obtain better accuracy and robustness than state of the art algorithms. Moreover, the framework we present here can be extended to learn graphs with specific structure (spectral and adjacency constraints on the graph) as the ones given in [17] since the model deals with a symmetric matrix with positive values.

References

[1] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods.," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.

[2] J.H. Stock and M.W. Watson, "Dynamic Factor Models, Factor-Augmented Vector Autoregressions, and Structural Vector Autoregressions in Macroeconomics," in *Handbook of Macroeconomics*, J. B. Taylor and Harald Uhlig, Eds., vol. 2 of *Handbook of Macroeconomics*, chapter 0, pp. 415–525. Elsevier, 2016.

[3] Anil K. Seth, Adam B. Barrett, and Lionel Barnett, "Granger causality analysis in neuroscience and neuroimaging," *Journal of Neuroscience*, vol. 35, no. 8, pp. 3293–3297, 2015.

[4] Shun Yao, Shinjae Yoo, and Dantong Yu, "Prior knowledge driven causality analysis in gene regulatory network discovery," in *2013 IEEE 13th International Conference on Data Mining Workshops*, 2013, pp. 124–130.

[5] Cunlu Zou, Christophe Ladroue, Shuixia Guo, and Jianfeng Feng, "Identifying interactions in the time and frequency domains in local and global networks - a granger causality approach," *BMC Bioinformatics*, vol. 11, no. 1, 2010.

[6] Helmut Lütkepohl, *New Introduction to Multiple Time Series Analysis*, Springer, 2005.

[7] Sumanta Basu, Ali Shojaie, and George Michailidis, "Network granger causality with inherent grouping structure," *Journal of Machine Learning Research*, vol. 16, no. 13, pp. 417–453, 2015.

[8] Anders Kock and Laurent Callot, "Oracle inequalities for high dimensional vector autoregressions," *Journal of Econometrics*, vol. 186, no. 2, pp. 325–344, 2015.

[9] Jiahe Lin and George Michailidis, "Regularized estimation and testing for high-dimensional multi-block vector-autoregressive models," *Journal of Machine Learning Research*, vol. 18, no. 117, pp. 1–49, 2017.

[10] William B. Nicholson, Ines Wilms, Jacob Bien, and David S. Matteson, "High dimensional forecasting via interpretable vector autoregression," *Journal of Machine Learning Research*, vol. 21, no. 166, pp. 1–52, 2020.

[11] Christopher A. Sims, *A Nine-Variable Probabilistic Macroeconomic Forecasting Model*, pp. 179–212, University of Chicago Press, January 1993.

[12] Marta Bańbura, Domenico Giannone, and Lucrezia Reichlin, "Large bayesian vector auto regressions," *Journal of applied Econometrics*, vol. 25, no. 1, pp. 71–92, 2010.

[13] Satyajit Ghosh, Kshitij Khare, and George Michailidis, "High-dimensional posterior consistency in bayesian vector autoregressive models," *Journal of the American Statistical Association*, vol. 114, pp. 735 – 748, 2018.

[14] Leo L Duan, Zeyu Yuwen, George Michailidis, and Zhengwu Zhang, "Low tree-rank bayesian vector autoregression models," *Journal of Machine Learning Research*, vol. 24, no. 286, pp. 1–35, 2023.

[15] Benjamin Marlin, Mark Schmidt, and Kevin Murphy, "Group sparse priors for covariance estimation," *arXiv preprint arXiv:1205.2626*, 2012.

[16] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[17] Sandeep Kumar, Jiaxi Ying, José Vinícius De M. Cardoso, and Daniel P Palomar, "A unified framework for structured graph learning via spectral constraints," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 785–844, 2020.

[18] Hui Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[19] Steven Diamond and Stephen P. Boyd, "Cvxpy: A python-embedded modeling language for convex optimization," *Journal of machine learning research : JMLR*, vol. 17, 2016.

[20] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, jun 2001.

[21] Bastien Pasdeloup, Vincent Gripon, Grégoire Mercier, Dominique Pastor, and Michael G Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 481–496, 2017.