Online Graph Topology Learning from Matrix-valued Time Series

Yiye Jiang\textsuperscript{1,2}, Jérémie Bigot\textsuperscript{1} & Sofian Maabout \textsuperscript{2}

\textsuperscript{1}Institut de Mathématiques de Bordeaux, Université de Bordeaux
\textsuperscript{2}Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux

**Abstract**

This paper is concerned with the statistical analysis of matrix-valued time series. These are data collected over a network of sensors (typically a set of spatial locations) along time, where a vector of features is observed per time instant per sensor. Thus each sensor is characterized by a vectorial time series. We would like to identify the dependency structure among these sensors and represent it by a graph. When there is only one feature per sensor, the vector auto-regressive models have been widely adapted to infer the structure of Granger causality. The resulting graph is referred to as causal graph. Our first contribution is then extending VAR models to matrix-variate models to serve the purpose of graph learning. Secondly, we propose two online procedures respectively in low and high dimensions, which can update quickly the estimates of coefficients when new samples arrive. In particular in high dimensional regime, a novel Lasso-type is introduced and we develop its homotopy algorithms for the online learning. We also provide an adaptive tuning procedure for the regularization parameter. Lastly, we consider that, the application of AR models onto data usually requires detrending the raw data, however, this step is forbidden in online context. Therefore, we augment the proposed AR models by incorporating trend as extra parameter, and then adapt the online algorithms to the augmented data models, which allow us to simultaneously learn the graph and trend from streaming samples. In this work, we consider primarily the periodic trend. Numerical experiments using both synthetic and real data are performed, whose results support the effectiveness of the proposed methods.

**Keywords:** Graph learning, matrix-variate data, auto-regressive models, homotopy algorithms.

1 Introduction

The identification of graph topology responds to increasing needs of data representation and visualization in many disciplines, such as meteorology, finance, neuroscience and social science. It
is crucial to reveal the underlying relationships between data entries, even in the settings where the natural graphs are available. For example in Mei & Moura [19], a temperature graph is inferred from the multivariate time series recording temperatures of cities around the continental United States over one-year period. It differs from the distance graph, however exhibits a better performance in predicting weather trends. Many methods have been proposed to infer graphs for various data processes and application settings [7, 24]. The problem of graph learning is that, given the observations of multiple features represented by random variables or processes, we would like to build or infer the cross-feature relationship that takes the form of a graph, with the features termed as nodes. According to data nature and the type of relationship, there are two main lines of work in the graph learning domain using statistic tools.

The first line considers the features represented by $N$ random variables $x \in \mathbb{R}^N$ with iid observations. Moreover, it assumes that $x \sim N(0, P^{-1})$. The works are interested in inferring the conditional dependency structure among $x_i, i = 1, \ldots, N$, which is encoded in the sparsity structure of precision matrix $P$. The resulting models are known as Gaussian graphical models [9, 20], and the sparse estimators are called graphical lasso. There are also variants for stationary vector processes, see Bach & Jordan [1] and Songsiri & Vandenberghe [25], whereas the relationship considered is still the conditional dependence.

The second line considers $N$ scalar processes, denoted by $x_t \in \mathbb{R}^N$, and the inference of Granger causality relationship among them from the observed time series. By contrast to the Gaussian assumption, this line supposes that $x_t$ is vector autoregressive (VAR) process

$$x_t = b + \sum_{l=1}^{p} A_l x_{t-l} + z_t, \quad t \in \mathbb{Z},$$

where $z_t \sim WN(0, \Sigma)$ is a white noise process with variance $\Sigma$, and $b \in \mathbb{R}^N, A_l \in \mathbb{R}^{N \times N}$ are the coefficients. The Granger causality is defined pairwise: $x_{it}$ is said to Granger cause $x_{jt}$, $j \neq i$, if $x_{jt}$ can be predicted more efficiently when the knowledge of $x_{it}$ in the past and present is taken into account. More technical definition see Lütkepohl [17, Section 2.3.1]. The causal graph then refers to such a graph where each node represents a scalar process, and the edges represent Granger causality.

The advantage of VAR assumption is that the topology of causal graph is encoded in the sparsity structure of the coefficient matrices. More specifically, if the processes are generated by a stationary VAR($p$) model, then $x_{it}$ does not cause $x_{jt}$ if and only if all the $ji$-th entries of the true coefficient matrices $A^l_{ji} = 0, l = 1, \ldots, p$, [17, Corollary 2.2.1]. Thus, we can retrieve the graph topology from the common sparsity structure in $A^l$. In low-dimensional regime, this structure can be identified through Wald test, which tests linear constraints for the coefficients.

The works in literature therefore focus on the inference of causal graphs in high-dimensional regime. The inference of the exact Granger causal graph is mainly considered in Bolstad et al. [3], Zaman et al. [27]. Bolstad et al. [3] propose to consider the group lasso penalty, $\lambda \sum_{i \neq j} \| (A^1_{ij}, \ldots, A^p_{ij}) \|_{\ell_2}$, to the usual least squares problem of VAR($p$) models, in order to
infer the common sparsity structure of coefficient matrices $A^l, l = 1, \ldots, p$. Zaman et al. [27] develop the online procedure for this estimation problem. Mei & Moura [19] define a variant of VAR model, where the sparse structure of coefficients $A^l$ does not directly equal the graph topology, but the topology of $l$-hop neighbourhoods \(^1\). More specifically, they suppose that $A^l = c_{0l}I + c_{1l}W + \ldots + c_{ll}W^l$, where $W$ is the adjacency matrix to infer, and $I$ is the identity matrix. Such models can thus capture the influence from more nodes. The estimation of the underlying adjacency matrix relies on the Lasso penalty to promote the sparsity.

In this paper, we consider the setting that each feature is represented by a vector process $x_{it} \in \mathbb{R}^F$. By extending the classical VAR(1) model for a matrix-valued time series $X_t = (x_{1t}, \ldots, x_{Nt}) \in \mathbb{R}^{N \times F}$, we wish to learn a graph of $N$ nodes that represents the causality structure among $x_{it}, i = 1, \ldots, N$. An example data set of this setting is given in Figure 1. Secondly, for the estimation, we also would like to develop online procedures in both low and high dimensions.

The extensions for matrix-variate, and more generally tensor-variate observations, $X_t \in \mathbb{R}^{m_1 \times \ldots \times m_K}$ have been considered for the models from the first line, for example see Greenewald et al. [12], Kalaitzis et al. [16], Wang et al. [26]. They pointed out that we may apply straightforwardly the vector models to the vectorized data vec($X_t$), however, the resulting models will

---

\(^1\)For a node, its neighbours are in the 1-hop neighbourhood of the node. All the neighbours of its neighbours are in the 2-hop neighbourhood of the node, so far and so forth.
suffer from the quadratic growth of the number of edges with respect to the graph size $\prod_{k=1}^{K} m_k$. The samples will be soon insufficient, and the computational issues will appear. Therefore, they propose to apply vector models to the vectorized data with certain structures imposed on parameter matrices that encode information on data dimensions. The most considered structures are Kronecker sum (KS) or/and Kronecker product (KP) \(^2\). For example, to extend Gaussian graphical models for matrix data, Kalaitzis et al. [16] propose the matrix Gaussian model: $\text{vec}(X) \sim \mathcal{N}(\mu, \Sigma)$, where $\Sigma = \Psi \otimes \Theta$, and $\Psi \in \mathbb{R}^{N \times N}$, $\Theta \in \mathbb{R}^{F \times F}$. That is they impose a KS structure on the precision matrix. Since the KS structure in an adjacency matrix corresponds to a Cartesian product of subgraphs, the embedding of KS structure leads to an interpretable graphical model in the way that the total conditional dependence structure is a product graph of two sub-graphs respectively for the raw and column dimensions.

Therefore, we follow the same idea to extend VAR(1). We first apply the classical VAR(1) onto the vectorized process $\text{vec}(X_t)$. Secondly, we propose to impose KS structure on the coefficient matrix $A$. At this point, we refer to the other work in literature, Chen et al. [5], also on the extension of VAR(1) to a matrix-variate process. In contrast to our KS construction, they propose to impose KP structure in the coefficient matrix. The comparison of these two constructions will be given in the next section. Additional to this difference, our work is especially devised for the purpose of graph learning, thus we promote sparsity in our coefficient estimators and focus on the development of online inference, which are not considered by the work of Chen et al. [5].

**Outlines.** In the rest of this paper, we first present the proposed matrix-variate AR(1) model in Section 2. We also explain how the total causal graph factorizes into two subgraphs, and how to interpret this relationship. In Section 3, we develop two online algorithms respectively for low and high dimensional settings. One is based on the projected ordinary least squares (OLS) estimator and Wald test, the other is based on a novel Lasso-type problem and the induced homotopy algorithms. For the regularization parameter in the Lasso approach, we also provide an automatic tuning procedure. In Section 4, we consider the real applications where the observed time series are not stationary and cannot be detrended in a preliminary step. We thus augment the previous stationary data model by integrating the trends as extra parameters, and then adapt the approaches to this setting. The augmented methods can update the trends and the graphs simultaneously. This work primarily consider the periodic trends which consist in finite values. Lastly, we present the results from numerical experiments using both synthetic and real data in Section 5. All proofs and large algorithms are gathered in the technical appendices. All the notations are collected in Table 1.

\(^2\)Let $C$ be an $m \times n$ matrix and $D$ a $p \times q$ matrix, the Kronecker product between $C$ and $D$, denoted by $C \otimes D$

\[
\begin{pmatrix}
C_{11}D & \cdots & C_{1n}D \\
\vdots & \ddots & \vdots \\
C_{m1}D & \cdots & C_{mn}D
\end{pmatrix}
\]

is the $pm \times qn$ block matrix. When $C$ and $D$ are square matrices with $m = n, p = q$, we can also define the Kronecker sum between them as: $C \oplus D = C \otimes I_p + I_m \otimes D$, where $I_k$ denote the $k \times k$ identity matrix.
Vectorized representation of a matrix.

Inverse vectorized representation of a vector, such that $\text{ivec} \circ \text{vec} = \text{id}$.

$[\cdot]$ Extraction by index. The argument in $[\cdot]$ can be a vector or a matrix. For a vector, the index argument can be a scalar or an ordered list of integers. For example, $[v]_k$ extracts the $k$-th entry of $v$, while $[v]_K = ([v]_k)_i$ extracts a sub-vector indexed by $K = (k_i)_i$ in order.

For a matrix, the index argument can be a pair of scalars or a pair of ordered lists of integers. For example, $[M]_{k,k'}$ extracts the $(k,k')$-th entry of $M$, while $[M]_{K,K'} = ([M]_{k_i,k_j})_{i,j}$ extracts a sub-matrix indexed by $K = (k_i)_i$ in row order, and $K' = (k'_j)_j$ in column order. When $K = K'$, we denote $[M]_{K,K'}$ by $[M]_K$.

$[M]_{:,i}$ Extraction of the $i$-th column vector of matrix $M$.

$[M]_{i,:}$ Extraction of the $i$-th row vector of matrix $M$.

$svec(M)$ Vectorized representation of the upper diagonal part of matrix $M$, that is, $(M)_{1,2}, [M]_{1,3}, \ldots, [M]_{2,3}, \ldots$.

$\text{diag}(M)$ Diagonal vector of matrix $M$.

$\text{offd}(M)$ $M$ with the diagonal elements replaced by zeros.

Table 1: Notations.

2 Causal Product Graphs and Matrix-variate AR(1) Models

We firstly compare the data assumptions led by KS and KP structures. It is known that when KP and KS structures are present in adjacency matrices, it implies in both cases that, the corresponding graphs can factorize into smaller graphs. To see the difference, let $A_F, A_N$ be the adjacency matrices of two graphs $G_F, G_N$, then the KP $A_F \otimes A_N$ and the KS $A_F \oplus A_N$ are respectively the adjacency matrices of their tensor product graph $G_N \times G_F$ and Cartesian product graph $G_N \square G_F$ [24]. We illustrate these two product graphs in Figure 2.

Figure 2 shows that, the two product graphs differ greatly. For example, the lattice-like structure of the Cartesian product preserves the subgraphs in all sections of both dimensions. By contrast, the tensor product focuses on the cross-dimensional connection, yet abandoning the intra-dimensional dependency. This later property actually refers to, in the Gaussian process literature, the cancellation of inter-task transfer, see for example, Bonilla et al. [4, Section 2.3]. Therefore when the nodes represent $x_{it}^f$, $i = 1, \ldots, N$, $f = 1, \ldots, F$, imposing KP structure [5]

\footnote{For the formal definitions of Cartesian and tensor products of graphs, we refer to Chen & Chen [6], Hammack et al. [13], Imrich & Peterin [15].}
Figure 2: Comparison of the Cartesian and the tensor products of graphs. The node set of both product graphs is the Cartesian product of the components’ node sets, yet follows the different adjacencies. The example is based on Sandryhaila & Moura [24, Figure 2].

implies assuming no causality dependencies among \( x^f_{it}, i = 1, \ldots, N \) for each \( f \) fixed, which represent the observations of the feature \( f \) at different nodes across the network. By contrast, the coefficient matrix \( A \) of KS structure is able to take such dependencies into account during inference, which are in effect present in many applications. This justifies our choice.

We now present the complete model setting. The matrix-variate stochastic process \( X_t \in \mathbb{R}^{N \times F} \) is said matrix-variate AR(1) process if the multivariate process \( x_t := \text{vec}(X_t) \) is a VAR(1) process

\[
\begin{align*}
    x_t &= Ax_{t-1} + z_t, \quad t \in \mathbb{Z},
\end{align*}
\]

with \( A \) having the particular KS structure \( \mathcal{K}_G \)

\[
\begin{align*}
    \mathcal{K}_G &= \{ M \in \mathbb{R}^{NF \times NF} : \exists M_F \in \mathbb{R}^{F \times F}, M_N \in \mathbb{R}^{N \times N}, \text{ such that,} \quad \text{offd}(M) = M_F \oplus M_N, \text{ with, } \text{diag}(M_F) = 0, \text{ diag}(M_N) = 0, \quad (2.2) \\
    &\quad M_F = M_F^T, \quad M_N = M_N^T \}. \quad (2.3)
\end{align*}
\]

By constraining \( A \in \mathcal{K}_G \), we impose the KS structure into the off-diagonal part of coefficient matrix \( A \), modelling the total causality structure by a Cartesian product graph \( \mathcal{G} \) parameterized by the spatial graph \( \mathcal{G}_N \) and the feature graph \( \mathcal{G}_F \). In this formal model construction, the diagonal of \( A \) is free from the structure constraint. In return, we require no self-loops in the component graphs by raising \( \text{diag}(M_F) = 0, \text{ diag}(M_N) = 0 \). This is to primarily address the non-identifiability problem of Kronecker sum, since \( A_F \oplus A_N = (A_F + cI_F) \oplus (A_N - cI_N) \) holds for any scalar \( c \). On the other hand, full parameterized diagonal adds the self-loops on all nodes of the total graph \( \mathcal{G} \), which also brings more flexibility to the model.
Note that, the last constraint in $K_G$ requires the component graphs hence the product graph to be symmetric. This is because we notice that, the existing causal graphs are usually directed, which disables their further use in the methods, which require undirected graphs as prior knowledge, like kernel methods, and graph Fourier transform related methods. Therefore, we focus on learning undirected graphs. Nevertheless, we stress that the derived approaches do not depend on the specific structure of coefficient, thus can be adapted to for example the relaxed constraint set without the symmetry assumption.

We then focus on the analysis of stationary process. We recall the stationarity condition for VAR(1) model in Lütkepohl [17]. We also need the conditions of center limit theorem (CLT), which permit the consistent estimator and its Wald test in the low-dimensional domain. We conclude all these assumptions by data generating model (2.5). We assume the samples $X_t \in \mathbb{R}^{N \times F}$ are generated by the following model given the initial sample $X_0$

$$x_t = Ax_{t-1} + z_t, \quad \text{with} \quad A \in K_G, \quad \|A\|_2 < 1, \quad t \in \mathbb{N}^+,$$

(2.5)

where $x_t = \text{vec}(X_t)$, $\|A\|_2$ equals the largest singular value of $A$, $z_t \in \mathbb{R}^{NF}$ is white noise with non-singular covariance matrix $\Sigma$ and bounded fourth moments, and $x_0 = \sum_{j=0}^{\infty} A^j z_{t-j}$.

Note that we firstly assume the process mean is zero and derive the main frameworks in Section 3. In Section 4, we will study the model with non-zero but time-variant process mean, namely, process trend, and we will adapt the derived frameworks to the augmented model.

Applying $\text{ivec}(\cdot)$ on both sides of Model (2.5), we can obtain its matrix representation

$$X_t = D \circ X_{t-1} + A_N X_{t-1} + X_{t-1} A_F^T + Z_t,$$

(2.6)

where $\circ$ is Hadamard product, $D \in \mathbb{R}^{N \times F} = \text{ivec}($diag$(A))$, $A_N$ and $A_F$ are the adjacency matrices such that offd$(A) = A_F \oplus A_N$, and $Z_t = \text{ivec}(z_t)$. In Model (2.6), $A_N X_{t-1}$ describes the spatial dependency, where each column of $X_{t-1}$ can be viewed as a graph signal on the same sensor graph $G_N$. Similarly, each row of $X_{t-1}$ can be seen as a graph signal on the feature graph $G_F$.

## 3 Online Graph Learning

In this section, we develop two learning frameworks to estimate $A$ recursively. The first method is valid in low dimensional regime, where the number of samples along time is assumed to be sufficiently large with respect to the number of parameters. By contrast, the second method based on a Lasso-type problem requires fewer samples, and it is thus adapted to high-dimensional regime. We especially consider a general learning framework where the partial sparsity is pursued in the estimation of only $A_N$. This is motivated by the fact that, merely a very small number of features $F$ are usually present in applications. Thus, the feature graph can be reasonably assumed fully-connected. On the other hand, since the partial sparsity constraint is also a technically more complicated case for the proposed high dimensional learning method, given its corresponding
resolution, the adaption to the case of fully sparsity does not require novel techniques. In the following section, we firstly introduce the tools on constraint set $\mathcal{K}_G$, which are crucial to derive the proposed frameworks.

### 3.1 Orthonormal Basis and Projection Operator of $\mathcal{K}_G$

$\mathcal{K}_G$ defined as Equation (2.2) is a linear space of dimension $NF + \frac{1}{2}F(F-1) + \frac{1}{2}N(N-1)$. We now endow $\mathcal{K}_G$ with the Frobenius inner product of matrix, that is $\langle B, C \rangle_F = \text{tr}(B^TC)$. The orthogonal basis of $\mathcal{K}_G$ is then given in the following Lemma.

**Lemma 3.1.** The set of matrices $U_k$, $k \in K := \{1, \ldots, NF + \frac{1}{2}F(F-1) + \frac{1}{2}N(N-1)\}$, defined below form an orthogonal basis of $\mathcal{K}_G$

$$U_k = \begin{cases} E_k, & k \in K_D := \{1, \ldots, NF\}, \\ \frac{1}{2N}E_k \otimes I_N, & k \in K_F := NF + \{1, \ldots, \frac{1}{2}F(F-1)\}, \\ \frac{1}{N}I_F \otimes E_k, & k \in K_N := NF + \frac{1}{2}F(F-1) + \{1, \ldots, \frac{1}{2}N(N-1)\}, \end{cases} \quad (3.1) $$

where when $k \in K_D$, $E_k \in \mathbb{R}^{NF \times NF}$, with $[E_k]_{i,j} = 1$, if $i = j = k$, otherwise 0, when $k \in K_F$, $E_k \in \mathbb{R}^{F \times F}$ is almost a zero matrix except

$$[E_k]_{1,2} = [E_k]_{2,1} = 1, \text{ if } k = NF + 1, \\
[E_k]_{1,3} = [E_k]_{3,1} = 1, \text{ if } k = NF + 2, \\
[E_k]_{2,3} = [E_k]_{3,2} = 1, \text{ if } k = NF + F, \\
\vdots \\
[E_k]_{F-1,F} = [E_k]_{F,F-1} = 1, \text{ if } k = NF + \frac{1}{2}F(F-1), \quad (3.2)$$

when $k \in K_N$, $E_k \in \mathbb{R}^{N \times N}$ is almost a zero matrix except

$$[E_k]_{1,2} = [E_k]_{2,1} = 1, \text{ if } k = NF + \frac{1}{2}F(F-1) + 1, \\
[E_k]_{1,3} = [E_k]_{3,1} = 1, \text{ if } k = NF + \frac{1}{2}F(F-1) + 2, \\
[E_k]_{N-1,N} = [E_k]_{N,N-1} = 1, \text{ if } k = NF + \frac{1}{2}F(F-1) + \frac{1}{2}N(N-1). \quad (3.3)$$

In Figure 3, we give an example of this orthogonal basis of $\mathcal{K}_G$ for $N = 3, F = 2$, where $U_k$ are visualized with respect to their non-zero entries. We can find that each $U_k$ relates to one variable of $\text{diag}(M), M_F$ and $M_N$, and characterises how it contributes to the structure of $M$ by repeating at multiple entries. Thus, taking the inner product with $U_k$ actually calculates the average value of an arbitrary matrix over these entries. This is important to understand how to project an arbitrary matrix onto $\mathcal{K}_G$.

It is easy to verify that $\langle U_k, U_{k'} \rangle_F = 0$ for any $k \neq k'$ in $K$, and $(U_k)_k$ spans $\mathcal{K}_G$. Thus the normalized matrices $U_k/\|U_k\|_F, k \in K$ form an orthonormal basis of $\mathcal{K}_G$. We introduce the
orthogonal projection onto $K_G$ and provide an explicit formula to calculate it using $(U_k/\|U_k\|_F)_k$ in Proposition 3.2.

**Proposition 3.2.** For a matrix $A \in \mathbb{R}^{NF \times NF}$, its orthogonal projection onto $K_G$ is defined by

$$
\text{Proj}_G(B) = \arg \min_{M \in K_G} \|B - M\|_F^2.
$$

(3.4)

Then given the orthonormal basis $U_k/\|U_k\|_F, k \in K$, the projections can be calculated explicitly as

$$
\text{Proj}_G(B) = \sum_{k \in K} \langle U_k, B \rangle \frac{1}{\|U_k\|_F^2} U_k.
$$

(3.5)

The projection is very straightforward to understand. To obtain a variable in $\text{diag}(M), M_F$ and $M_N$ related to $U_k$, we use $\langle U_k, B \rangle$ to calculate the average value of $B$ as explained previously. Then this average value is repeated at the corresponding entries to construct the structure, by multiplying locator $U_k/\|U_k\|_F^2$.

Furthermore the orthogonality of the basis implies the direct sum

$$
K_G = K_D \oplus K_F \oplus K_N,
$$

(3.6)

where $K_D, K_F,$ and $K_N$ are respectively spanned by $(U_k)_{k \in K_D}$, $(U_k)_{k \in K_F}$, and $(U_k)_{k \in K_N}$. Given the construction of $(U_k)_k$, Equation (3.6) actually reveals the product graph decomposition, note that equally we have

$$
K_D = \{M \in \mathbb{R}^{NF \times NF} : \text{offd}(M) = 0\},
$$
\[ K_F = \{ M \in \mathbb{R}^{NF \times NF} : \exists M_F \in \mathbb{R}^{F \times F}, \text{ such that,} \ M = M_F \otimes I_N, \ \text{with, diag}(M_F) = 0, M_F = M_F^T \} , \]

\[ K_N = \{ M \in \mathbb{R}^{NF \times NF} : \exists M_N \in \mathbb{R}^{N \times N}, \text{ such that,} \ M = I_F \otimes M_N, \ \text{with, diag}(M_N) = 0, M_N = M_N^T \} . \]

The projection onto these subspaces can also be computed analogously

\[ \text{Proj}_D(B) = \sum_{k \in K_D} \langle U_k, B \rangle \frac{1}{\|U_k\|_F^2} U_k, \text{ that is the diagonal part of } B. \quad (3.7) \]

\[ \text{Proj}_F(B) = \sum_{k \in K_F} \langle U_k, B \rangle \frac{1}{\|U_k\|_F^2} U_k = \left[ \sum_{k \in K_F} \langle U_k, B \rangle E_k \right] \otimes I_N, \quad (3.8) \]

\[ \text{Proj}_N(B) = \sum_{k \in K_N} \langle U_k, B \rangle \frac{1}{\|U_k\|_F^2} U_k = I_F \otimes \left[ \sum_{k \in K_N} \langle U_k, B \rangle E_k \right]. \quad (3.9) \]

We use \( \text{Proj}_{G_F}(B) \) and \( \text{Proj}_{G_N}(B) \) to denote the small matrices \( \sum_{k \in K_F} \langle U_k, B \rangle E_k \) and \( \sum_{k \in K_N} \langle U_k, B \rangle E_k \), with an extra subscript \( G \), with which we will represent the proposed estimators of \( A_F, A_N \) in the following sections. Finally, we have

\[ \text{Proj}_G(B) = \text{Proj}_D(B) + \text{Proj}_{G_F}(B) \oplus \text{Proj}_{G_N}(B). \quad (3.10) \]

### 3.2 Approach 1: Projected OLS Estimators and Wald Test

In low dimensional regime, VAR model (2.5) can be fitted by the ordinary least squares method. Assume that we start receiving samples \( x_0, x_1, ..., x_t \) from time \( \tau = 1 \), the OLS estimator for an intercept-free VAR(1) model is given by

\[ \tilde{A}_t = \hat{\Gamma}_t(1) \left[ \hat{\Gamma}_t(0) \right]^{-1}, \quad (3.11) \]

where

\[ \hat{\Gamma}_t(0) = \frac{1}{t} \sum_{\tau=1}^{t} x_{\tau-1} x_{\tau-1}^\top \]

and

\[ \hat{\Gamma}_t(1) = \frac{1}{t} \sum_{\tau=1}^{t} x_{\tau} x_{\tau-1}^\top \]

are respectively the consistent estimators of auto-covariance matrices \( \Gamma(0) \) and \( \Gamma(1) \), with \( \Gamma(h) = \mathbb{E}(x_t x_{t-h}^\top), h \geq 0 \). Moreover, the additional conditions in Model (2.5) permit the asymptotic properties [17, Section 3.2.2]
1. $\tilde{A}_t \overset{P}{\to} A$,

2. $\sqrt{t} \text{vec}(\tilde{A}_t - A) \overset{d}{\to} \mathcal{N}(0, \Sigma_{ols})$, where $\Sigma_{ols} = [\Gamma(0)]^{-1} \otimes \Sigma$.

However, due to model misspecification and limited samples, $\tilde{A}_t$ will not have the same structure as $A \in \mathcal{K}_G$. Therefore, the projection of $\tilde{A}_t$ onto $\mathcal{K}_G$ needs to be performed, which leads to the projected OLS estimator:

$$\tilde{A}_t := \text{Proj}_G(\tilde{A}_t).$$

Given the representation (3.10), it is natural to define the estimators of $\text{diag}(A), A_F$, and $A_N$ by $\text{Proj}_D(\tilde{A}_t), \text{Proj}_G(\tilde{A}_t)$, and $\text{Proj}_{\text{OLS}}(\tilde{A}_t)$, respectively, denoted by $\tilde{A}_{D,t}, \tilde{A}_{F,t}$, and $\tilde{A}_{N,t}$. We now establish the Wald test with $\tilde{A}_{N,t}$ to identify the sparsity structure of the true $A_N$. To this end, we provide the CLT in Theorem 3.3.

**Theorem 3.3.** Assume samples $x_0, x_1, \ldots, x_t$ satisfy the assumptions of Model (2.5), then the CLT holds for $\tilde{A}_{N,t}$, as $t \to +\infty$,

$$\sqrt{t} \text{vec}(\tilde{A}_{N,t} - A_N) \overset{d}{\to} \mathcal{N}(0, \Sigma_N),$$

where

$$\Sigma_N = \sum_{k,k' \in \mathcal{K}_N} \text{vec}(U_k)^\top \Sigma_{ols} \text{vec}(U_{k'}) \left( \text{vec}(E_k)^\top \text{vec}(E_{k'}) \right).$$

The proof is done through applying Cramér-Wold theorem on $\sqrt{t} \text{vec}(\tilde{A}_{N,t} - A_N)$, given the linearity of $\text{Proj}_G(\tilde{A}_t)$ and the CLT of classical OLS estimator $\tilde{A}_t$. For details, see Appendix A, where we also derive a CLT for $\tilde{A}_{N,t}$.

It is straightforward to understand the asymptotic distribution of $\tilde{A}_{N,t}$. The asymptotic covariance between its two entries is assigned the mean of covariance values $\text{vec}(U_k)^\top \Sigma_{ols} \text{vec}(U_{k'})$, following the construction of the corresponding estimators $\langle U_k, \tilde{A}_t \rangle$ and $\langle U_{k'}, \tilde{A}_t \rangle$ as averages as well.

Based on this large sample result, we now test the nullity of $P$ given variables $[A_N]_{i_k,j_k}, k = 1, \ldots, P$, with $i_k < j_k$ as

$$H_0 : \alpha = 0 \text{ versus } H_1 : \alpha \neq 0,$$

where $\alpha \in \mathbb{R}^P := \left( \cdots, [A_N]_{i_k,j_k}, \cdots \right)^\top$. The test statistic is given by

$$\lambda_{W,t} = t \hat{\alpha}_t^\top \left[ \hat{\Sigma}_{W,t} \right]^{-1} \hat{\alpha}_t,$$

where $\hat{\alpha}_t \in \mathbb{R}^P := \left( \cdots, [\tilde{A}_{N,t}]_{i_k,j_k}, \cdots \right)^\top$, and $\hat{\Sigma}_{W,t} \in \mathbb{R}^{P \times P}$ is defined as

$$\left[ \hat{\Sigma}_{W,t} \right]_{k,k'} = \text{vec}(U_{h_k})^\top \hat{\Sigma}_{ols,t} \text{vec}(U_{h_{k'}}),$$
such that $U_{hk}$ is the matrix corresponding to variable $[A_N]_{ik,jk}$,

$$\hat{\Sigma}_{ols,t} = \left[\hat{\Gamma}_t(0)\right]^{-1} \otimes \hat{\Sigma}_t,$$

and

$$\hat{\Sigma}_t = \hat{\Gamma}_t(0) - \hat{\Gamma}_t(1) \left[\hat{\Gamma}_t(0)\right]^{-1} \hat{\Gamma}_t(1)^	op,$$

are the consistent estimators. CLT (3.3) implies the following result.

**Corollary 3.3.1.** The asymptotic distribution of $\lambda_{W,t}$ as $t \to +\infty$ is given by

$$\lambda_{W,t} \xrightarrow{d} \chi^2(P), \quad \text{Under } H_0.$$

**Remark 1.** We can also consider the test statistic $\lambda_{F,t} := \lambda_{W,t}/P$ as suggested in Lütkepohl [17, Section 3.6] in conjunction with the critical values from $F(P, t - NF - 1)$.

The Wald test above theoretically completes the approach. In practice, we propose to test the $p$ entries of the smallest estimate magnitudes, jointly each time, as $p$ grows from 1 to possibly largest value $|K_N|$. Specifically, for a given estimation $\hat{\mathbf{A}}_{N,t}$, we first sort its entries such that

$$|[\hat{\mathbf{A}}_{N,t}]_{i_1,j_1}| \leq |[\hat{\mathbf{A}}_{N,t}]_{i_2,j_2}| \leq \ldots \leq |[\hat{\mathbf{A}}_{N,t}]_{|K_N|,|K_N|}|.$$

Then, we set up the sequence of joint tests

$$H_0(1), H_0(2), \ldots, H_0(|K_N|),$$

where $H_0(p) : ([\hat{\mathbf{A}}_{N,t}]_{i_1,j_1}, \ldots, [\hat{\mathbf{A}}_{N,t}]_{p_0,j_{p_0}})^	op = 0$.

We perform these tests sequentially until $H(p_0 + 1)$ is rejected for some $p_0$. Lastly, we replace the entries $[\hat{\mathbf{A}}_{N,t}]_{i_1,j_1}, \ldots, [\hat{\mathbf{A}}_{N,t}]_{p_0,j_{p_0}}$ with 0 in $\hat{\mathbf{A}}_{N,t}$ as the final estimate of $\mathbf{A}_N$. Note that searching for $p_0$ resembles root-finding, since the output from each point $p$ is binary. Thus, the search can be accelerated by using the bisection, with the maximal number of steps about $\log_2(|K_N|)$.

The previous procedure is performed at the $t$-th iteration, given the OLS estimator $\hat{\mathbf{A}}_t$ and the consistent estimator $\hat{\Sigma}_{ols,t}$. When new sample $\mathbf{x}_{t+1}$ comes, $\hat{\mathbf{A}}_{t+1}$ and $\hat{\Sigma}_{ols,t+1}$ can be calculated efficiently by applying the Sherman Morrison formula on $[\hat{\Gamma}_t(0)]^{-1}$. The pseudo code is given in Algorithm 2.

### 3.3 Approach 2: Structured Matrix-variate Lasso and Homotopy Algorithms

As discussed at the introduction, a common practice in the literature to identify the sparsity structure of VAR coefficients in high dimensional regime is to adopt Lasso estimators. The one used in Bolstad et al. [3], Zaman et al. [27] is defined as the minimizer of Lasso problem (3.14) in the VAR(1) case.

$$\min_{\mathbf{A}} \frac{1}{2t} \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - \mathbf{A}\mathbf{x}_{\tau-1}\|_2^2 + \lambda_t \|\mathbf{A}\|_{\ell_1}, \quad (3.14)$$
where \( \mathbf{x}_\tau \) is a vector of sample, which can be taken as \( \text{vec}(\mathbf{X}_\tau) \) for example. Lasso (3.14) is the most standard Lasso in literature [14, Section 3.4.2]. A wide variety of frameworks from convex analysis and optimization have been adapted to compute its solutions for different scenarios, for example, coordinate descent [10], proximal gradient methods [2], and a more Lasso-specific technique least angle regression [8]. However, Lasso (3.14) is not able to estimate the structured \( \mathbf{A} \) with the sparse component \( \mathbf{A}_N \). Therefore motivated by the estimation, we propose the novel Lasso type problem (3.15)

\[
\mathbf{A}(t, \lambda_t) = \arg \min_{\mathbf{A} \in \mathcal{K}_G} \frac{1}{2t} \sum_{\tau=1}^{t} \| \mathbf{x}_\tau - \mathbf{A} \mathbf{x}_{\tau-1} \|_2^2 + \lambda_t F \| \mathbf{A}_N \|_1.
\] (3.15)

The ordinary resolution of Lasso (3.15) can be done by applying for example the proximal gradient descent [23]. In the algorithm framework, the structure constraint and the partial sparsity do not pose additional difficulties, since only the gradient with respect to \( \mathbb{R}^{NF \times NF} \) is calculated in the forward step. We show these details in Appendix E.

At this point, we focus on providing the algorithms to quickly update the previous solutions for the change in the hyperparameter value or in the data term. This different goal requires to consider specific methods. For classical Lasso, the framework of homotopy continuation methods [22] has been explored [11, 18] to calculate the fast updating. Since the homotopy algorithm is derived from the optimality condition, which is with respect to the matrices in \( \mathcal{K}_G \) for Lasso (3.15), requiring to consider the gradient with the structure, thus the existing homotopy algorithms for classical Lasso are not applicable. Therefore in the following, we first calculate the optimality condition of Lasso (3.15) in Section 3.3.1, based on the expression of projection onto \( \mathcal{K}_G \). Then we derive the two homotopy algorithms in Sections 3.3.2 and 3.3.3, respectively for the updating paths \( \mathbf{A}(t, \lambda_1) \to \mathbf{A}(t, \lambda_2) \) and \( \mathbf{A}(t, \lambda_2) \to \mathbf{A}(t+1, \lambda_2) \), together with an adaptive tuning procedure for the regularization hyperparameter.

Therefore, the online algorithm consists in performing the three steps in the order:

\[
\begin{align*}
\text{Step 1 :} & \quad \lambda_t \to \lambda_{t+1}, \\
\text{Step 2 :} & \quad \mathbf{A}(t, \lambda_t) \to \mathbf{A}(t, \lambda_{t+1}), \\
\text{Step 3 :} & \quad \mathbf{A}(t, \lambda_{t+1}) \to \mathbf{A}(t+1, \lambda_{t+1}).
\end{align*}
\] (3.16)

### 3.3.1 Optimality Conditions

The key point in deriving the optimality conditions arising from the variational problem (3.15) is to transfer the structure of \( \mathbf{A} \) onto the data vector \( \mathbf{x}_{\tau-1} \), using an orthonormal basis of \( \mathcal{K}_G \). We introduce the auxiliary variable \( \mathbf{A}^0 \), such that \( \mathbf{A} = \text{Proj}_{\mathcal{G}}(\mathbf{A}^0) \), and rewrite Problem (3.15) with respect to \( \mathbf{A}^0 \)

\[
\min_{\mathbf{A}^0 \in \mathbb{R}^{NF \times NF}} \frac{1}{2t} \sum_{\tau=1}^{t} \left\| \mathbf{x}_\tau - \sum_{k \in \mathcal{K}} \langle \mathbf{U}_k, \mathbf{A}^0 \rangle \frac{1}{\| \mathbf{U}_k \|_F^2} \mathbf{U}_k \mathbf{x}_{\tau-1} \right\|_2^2 + \lambda \sum_{k \in \mathcal{K}_N} \langle \mathbf{U}_k, \mathbf{A}^0 \rangle \frac{1}{\| \mathbf{U}_k \|_F^2} \mathbf{U}_k \|_{\ell_1}.
\] (3.17)
Problem (3.17) is weakly convex, since a minimizer of (3.15) can be projected from infinitely many minimizers of (3.17). We still use $L_{\lambda,t}$ to denote the objective function above. A minimizer $A^0$ of (3.17) satisfies the optimality conditions

\[
0 \in \frac{\partial L_{\lambda,t}}{\partial A^0} = \sum_{k,k' \in K} \langle U_k, U_{k'} \hat{\Gamma}(0) \rangle \frac{1}{\|U_{k'}\|_F^2} U_k - \sum_{k \in K} \langle U_k, \hat{\Gamma}(1) \rangle \frac{1}{\|U_k\|_F^2} U_k + \lambda \sum_{k \in K_N} \partial \langle U_k, A^0 \rangle \frac{1}{\|U_k\|_F^2} U_k,
\]

(3.18)

Assume $A^0$ is a matrix which satisfies Equation (3.18), hence a minimizer of Problem (3.17). Then $A = \text{Proj}_G(A^0)$ is a minimizer of Lasso (3.15). We denote its active set $\{ k \in K_N : \langle U_k, A^0 \rangle \neq 0 \}$ by $K^A_N$, that is all the non-zero variables of $A_N$, and its non-active set by $K^0_N$, that is $K_N \setminus K^A_N$. Since $(U_k)_{k \in K}$ is an orthogonal family, Equation (3.18) is equivalent to

\[
0 = \sum_{k \in K_D \cup K_F} \left[ \sum_{k' \in K} \langle U_k, U_{k'} \hat{\Gamma}(0) \rangle \frac{1}{\|U_{k'}\|_F^2} U_k - \langle U_k, \hat{\Gamma}(1) \rangle \right] \frac{1}{\|U_k\|_F^2} U_k,
\]

(3.19)

\[
0 = \sum_{k \in K^A_N} \left[ \sum_{k' \in K} \langle U_k, U_{k'} \hat{\Gamma}(0) \rangle \frac{1}{\|U_{k'}\|_F^2} U_k - \langle U_k, \hat{\Gamma}(1) \rangle \right] \frac{1}{\|U_k\|_F^2} U_k + \lambda \sum_{k \in K^A_N} \text{sign} \langle U_k, A^0 \rangle \frac{1}{\|U_k\|_F^2} U_k,
\]

(3.20)

\[
0 = \sum_{k \in K^0_N} \left[ \sum_{k' \in K} \langle U_k, U_{k'} \hat{\Gamma}(0) \rangle \frac{1}{\|U_{k'}\|_F^2} U_k - \langle U_k, \hat{\Gamma}(1) \rangle \right] \frac{1}{\|U_k\|_F^2} U_k + \lambda \sum_{k \in K^0_N} \partial \langle U_k, A^0 \rangle \frac{1}{\|U_k\|_F^2} U_k, \text{ where } \partial \langle U_k, A^0 \rangle \in [-1, 1]
\]

(3.21)

To furthermore derive the optimality conditions of Lasso (3.15) in terms of $A$, we introduce the projections onto sub-spaces $K^1_N := \text{span}\{U_k : k \in K^1_N\}$ and $K^0_N := \text{span}\{U_k : k \in K^0_N\}$, denoted respectively by $\text{Proj}_{K^1}$ and $\text{Proj}_{K^0}$. Note that Equation (3.6) in fact admits

\[
K_G = \bigoplus_{k \in K} \text{span}\{U_k\}.
\]

Thus

\[
\text{Proj}_{K^1}(B) = \sum_{k \in K^1_N} \langle U_k, B \rangle \frac{1}{\|U_k\|_F^2} U_k = I_F \otimes \left[ \sum_{k \in K^1_N} \langle U_k, B \rangle E_k \right],
\]

14
and
\[
\text{Proj}_{N^0}(B) = \sum_{k \in K_N^0} \langle U_k, B \rangle \frac{1}{\|U_k\|^2_F} U_k = I_F \otimes \left[ \sum_{k \in K_N^0} \langle U_k, B \rangle E_k \right].
\]

Then Equations (3.19), (3.20), and (3.21) are equivalent respectively to
\[
\text{Proj}_{DF} \left( \hat{A}_t(0) - \hat{\Gamma}_t(1) \right) = 0,
\]
\[
\text{Proj}_{K_N^0} \left( \hat{A}_t(0) - \hat{\Gamma}_t(1) \right) + \lambda I_F \otimes \left[ \sum_{k \in K_N^0} \text{sign}(\langle E_k, A_N \rangle) E_k \right] = 0,
\]
\[
\text{Proj}_{K_N^0} \left( \hat{A}_t(0) - \hat{\Gamma}_t(1) \right) + \lambda I_F \otimes \left[ \sum_{k \in K_N^0} \partial\langle E_k, A_N \rangle E_k \right] = 0,
\]
where \(A \in K_G\), \(\text{Proj}_{DF} = \text{Proj}_D + \text{Proj}_F\), and \(\partial\langle E_k, A_N \rangle | \in [−1, 1]\). The optimality conditions above are an extension of those for classical Lasso, while the former are furthermore refined to the unpenalized variables versus the penalized variables.

### 3.3.2 Homotopy from \(A(t, \lambda_1)\) to \(A(t, \lambda_2)\)

To develop the homotopy algorithm for the change in \(\lambda\) value, we need to get the formulas of the active variables indexed by \(K_N^1\) in terms of \(\lambda\). To this end, we need to rely on representation (3.19), (3.20), and (3.21), directly in terms of each variable \(\langle U_k, A^0 \rangle\). We firstly reorganize all the model variables into a vector
\[
a^s := \left( \frac{1}{\|U_k\|^2_F} U_k, A^0 \right)_{k \in K} = \left( \frac{1}{\|U_k\|^2_F} U_k, A \right)_{k \in K}.
\]
Note that \(a^s\) is in fact the scaled Lasso solution by the time the variable repeats. Then optimality conditions (3.19), (3.20), and (3.21) are essentially a system of linear equations of unknown \(a^s\), with \(\lambda\) in the coefficients. Thus we aim to firstly represent this linear system in vector form, in order to solve the unknowns. We shall introduce the following notations.

**Notations of Proposition 3.4.** \(\Gamma_0 \in \mathbb{R}^{\left|K\right| \times \left|K\right|}\) is a large matrix defined as
\[
[\Gamma_0]_{k, k'} = \langle U_k, U_{k'} \hat{\Gamma}_t(0) \rangle.
\]
\(\gamma_1 \in \mathbb{R}^{\left|K\right|}\) is a long vector defined as
\[
[\gamma_1]_k = \langle U_k, \hat{\Gamma}_t(1) \rangle.
\]
\( \mathbf{w} \in \mathbb{R}^{\left| K_1 \right|} \) is a long vector where \([\mathbf{w}]_k \) is defined as
\[
\begin{cases}
0, & k \in K_D \cup K_F, \\
\text{sign}[\mathbf{a}^s]_k, & k \in K_1^s, \\
\in [-1, 1], & k \in K_0^N.
\end{cases}
\] (3.25)

We define \( K^1 := K_D \cup K_F \cup K_1^s \), that are all the non-zero variables. Note that except the computational coincidence, the variables in \( K_D \cup K_F \) are usually non-zero. Then we denote the extractions
\[
\Gamma_0^1 = [\Gamma_0]_{K1}^1, \Gamma_0^0 = [\Gamma_0]_{K0}^0, \gamma_1^1 = [\gamma_1]_{K1}, \gamma_0^0 = [\gamma_1]_{K0}^0,
\]
\( a_i^s = [a_i^s]_{K1}, w_1 = [w]_{K1}, w_0 = [w]_{K0}^0 \). (3.26)

We can endow any orders to the elements in \( K^1, K_0^N \) to extract the rows/columns/entries above, only if the orders are used consistently to all the extractions. With these notations, we now can retrieve a system of linear equations from Equations (3.19), (3.21), (3.20) of unknowns \( a_i^s \).

**Proposition 3.4.** A minimizer of Lasso problem (3.15) satisfies the linear system
\[
\begin{cases}
\Gamma_0^1 a_i^s - \gamma_1^1 + \lambda w_1 = 0, \\
\Gamma_0^0 a_i^s - \gamma_0^0 + \lambda w_0 = 0.
\end{cases}
\] (3.27)

The representation of the optimality conditions in Equation (3.27) are similar to those of classical Lasso \([11, 18]\), where \( \Gamma_0, \gamma_1 \) with the embedded structures correspond to \( X^T X, X^T y \) in the optimality conditions of classical Lasso. However in our case, the non-zero and sign pattern are only with respect to the entries of \( A_N \), thus \( w_1 \), which is the equivalent of sign vector, has \( |K_D| + |K_F| \) zeros.

Suppose that \( \mathbf{A}(t, \lambda) \) is the unique solution for a fixed \( \lambda \) of the optimization problem (3.15), then we invert \( \Gamma_0^1 \) in Proposition 3.4 and get the formulas of \( a_i^s \)
\[
\begin{cases}
a_i^s = [\Gamma_0^1]^{-1} (\gamma_1^1 - \lambda w_1) \\
\lambda w_0 = \gamma_0^0 - \Gamma_0^0 a_i^s.
\end{cases}
\] (3.28)

Formula (3.28) is determined by the active set and the sign pattern of the optimal solution at \( \lambda \). It shows that \( a_i^s \) is a piecewise linear function of \( \lambda \), while \( w_0 \) is also a piecewise smooth function.

Therefore, with the assumptions that \( [a_i^s]_{K_0^N} \neq 0 \) (element-wise), and \( |[w]_{K_0^N}| < 1 \) (element-wise), due to continuity properties, there exists a range \( (\lambda_l, \lambda_r) \) containing \( \lambda \), such that for any \( \lambda' \in (\lambda_l, \lambda_r) \), element-wise, \( [a_i^s]_{K_1^s} \) remains nonzero with the signs unchanged, and \( [w]_{K_0^N} \) remains in \((-1, 1)\). Hence, Formula (3.28) is the closed form of all the optimal solutions \( \mathbf{A}(t, \lambda') \), for
\( \lambda' \in (\lambda_l, \lambda_r) \). \( \lambda_l, \lambda_r \) are taken as the closest critical points to \( \lambda \). Each critical point is a \( \lambda \) value which makes either an \([a^\lambda]_k\), \( k \in K^1_N \) become zero, or a \([w]_k\), \( k \in K^0_N \) reach 1 or \(-1\). By letting \([a^\lambda]_k = 0\), \( k \in K^1_N \) and \([w]_k = \pm 1\), \( k \in K^0_N \) in Formula (3.31), we can compute all critical values. We now use \( k_i \) to denote the orders of \( K^1, K^0_N \) that we used in the extraction (3.26). The critical values are then given by

\[
\begin{align*}
\lambda^0_{k_i} &= \left[ (\Gamma^1_0)^{-1} \gamma^1_1 \right]_i / \left[ (\Gamma^0_0)^{-1} w_1 \right]_i, \quad k_i \in K^1, \\
\lambda^+_{k_i} &= \left[ \gamma^0_1 - \Gamma^0_0 (\Gamma^1_0)^{-1} \gamma^1_1 \right]_i / \left[ 1 - \Gamma^0_0 (\Gamma^1_0)^{-1} w_1 \right]_i, \quad k_i \in K^0_N, \\
\lambda^-_{k_i} &= \left[ \gamma^0_1 - \Gamma^0_0 (\Gamma^1_0)^{-1} \gamma^1_1 \right]_i / \left[ -1 - \Gamma^0_0 (\Gamma^1_0)^{-1} w_1 \right]_i, \quad k_i \in K^0_N.
\end{align*}
\] (3.29)

Thus, the closest critical points from both sides are

\[
\begin{align*}
\lambda_l := \max \{ & \max \{ \lambda^0_{k}, k \in K^1_N : \lambda^0_{k} < \lambda \}, \\
& \max \{ \lambda^+_{k}, k \in K^0_N : \lambda^+_{k} < \lambda \}, \\
& \max \{ \lambda^-_{k}, k \in K^0_N : \lambda^-_{k} < \lambda \} \}, \\
\lambda_r := \min \{ & \min \{ \lambda^0_{k}, k \in K^1_N : \lambda^0_{k} > \lambda \}, \\
& \min \{ \lambda^+_{k}, k \in K^0_N : \lambda^+_{k} > \lambda \}, \\
& \min \{ \lambda^-_{k}, k \in K^0_N : \lambda^-_{k} > \lambda \} \}. \quad (3.30)
\end{align*}
\]

If \( \lambda_l \neq \emptyset \) then \( \lambda_l := 0 \), while if \( \lambda_r = \emptyset \) then \( \lambda_r := +\infty \). After \( \lambda' \) leaves the region by adding or deleting one variable to or from the active set, we update in order the corresponding entry in \( w, K^1, K^0_N \), and the solution formula (3.28) (Sherman Morrison formula for one rank update of \([\Gamma^1_0]^{-1}\)) to calculate the boundary of the new region as before. We proceed in this way until we reach the region covering the \( \lambda \) value at which we would like to calculate the Lasso solution, and use Formula (3.28) in this final region to compute the \( a^\ast \) with the desired \( \lambda \) value. Lastly, we retrieve the matrix-form optimal solution based on \( a^\ast \) and the latest \( K^1 \). This completes the first homotopy algorithm. The detailed algorithm see Appendix F.

### 3.3.3 Homotopy from \( \mathbf{A}(t, \lambda) \) to \( \mathbf{A}(t + 1, \lambda) \)

We recall again the classical Lasso in Equation (3.31). We formulate it with vectorial parameter here.

\[
\theta(t, \lambda) = \arg \min_{\theta \in \mathbb{R}^d} \frac{1}{2} \| y - X\theta \|_{\ell_2}^2 + t\lambda \| \theta \|_{\ell_1},
\]

(3.31)

where \( y = (y_1, \ldots, y_t)^T \), \( X = (x_1, \ldots, x_t)^T \), and \( y_\tau \in \mathbb{R}, x_\tau \in \mathbb{R}^d \) are the samples at time \( \tau \). Garrigues & Ghaoui [11] propose to introduce a continuous variable \( \mu \) in Lasso (3.31), leading to
the optimization Problem (3.32)

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{2} \| y - X\theta \|_2^2 + \frac{1}{2} (\mu y_{t+1} - \mu x_{t+1}^T \theta)^2 + t\lambda \| \theta \|_1, \tag{3.32}
\]

in order to let the problem of learning from \( t \) samples evolve to that of learning from \( t + 1 \) samples, as \( \mu \) goes from 0 to 1. Therefore, representing the Lasso solution as a continuous function of \( \mu \) permits the development of homotopy algorithm, which computes the path \( \theta(t, \lambda) \) to \( \theta(t + 1, \frac{t}{t+1} \lambda) \).

This homotopy algorithm is derived based on the fact that, the term of new sample will only result in a rank-1 update in the covariance matrix as \( X^T X + \mu^2 x_{t+1} x_{t+1}^T \), because only 1 response variable is present. Thus, the corresponding matrix inverse in the closed form of optimal solution can be still expressed as an explicit function of \( \mu \) using the Sherman Morrison formula, which furthermore allows the calculation of critical points of \( \mu \). However, for the matrix-variate Lasso (3.15), a new sample will cause a rank-\( NF \) update in \( \Gamma_0 \), that is the number of response variables in the Lasso problem. To formally understand this change, we rewrite \( \Gamma_0 \) as the sum of \( t \) reorganized samples analogous to usual

\[
\Gamma_0 = \frac{1}{t} \sum_{\tau=1}^{t} \tilde{X}_{\tau-1} \tilde{X}_{\tau-1}^T, \quad \text{where} \quad \tilde{X}_{\tau-1} \in \mathbb{R}^{K \times NF} \quad \text{with} \quad [\tilde{X}_{\tau-1}]_{k,i} = [U_k]_i x_{\tau-1},
\]

note that a new \( x_{t+1} \) corresponds to the change \( \tilde{X}_t \tilde{X}_t^T \) in \( \Gamma_0 \), which is a rank \( NF \) matrix. Thus it is impossible to express \( [\Gamma_0^\dagger]^{-1} \) as an explicit and simple function of one single \( \mu \). However, note that each column (rank) \( [\tilde{X}_t]_i \) corresponds to introducing new sample of one response variable \( x_{t+1,i} := [x_{t+1}]_i \) at node \( i \) in \( \mathcal{G} \), by rewriting the incremental term of Lasso (3.15)

\[
\| x_{t+1} - Ax_t \|_2^2 = \left\| x_{t+1} - \sum_{k \in K} \langle U_k, A^0 \rangle \frac{1}{\| U_k \|_F} U_k x_t \right\|_2^2 = \sum_{i=1}^{NF} \left( x_{t+1,i} - \sum_{k \in K} \langle U_k, A^0 \rangle \frac{1}{\| U_k \|_F} [U_k]_i, x_t \right)^2 \tag{3.33}
\]

Therefore, we propose to introduce \( NF \) continuous variables \( \mu_1, ..., \mu_{NF} \) in Lasso (3.15), and to consider the following problem

\[
A_{\lambda,t}(\mu_1, ..., \mu_{NF}) = \arg \min_{A \in \mathcal{K}_G} L_{\lambda,t}(\mu_1, ..., \mu_{NF}),
\]

4On the other hand, this implies that \( \Gamma_0 \) will quickly become non-singular from the initial time, as new samples \( x_\tau \) come in.

5More general, a new sample will cause \( NF \) rank change in the corresponding matrix \( I_{NF} \otimes \tilde{\Gamma}_t(0) \) in Lasso (3.14).
where \( L_{\lambda,t}(\mu_1, \ldots, \mu_{NF}) = \frac{1}{2(t+1)} \sum_{i=1}^{t} \| x_{f_i} - A x_{f_i-1} \|_{\ell_2}^2 + \lambda F \| A \|_{\ell_1} \)
\[ + \frac{1}{2(t+1)} \sum_{i=1}^{NF} \mu_i \left( x_{t+1,i} - \sum_{k \in K} \langle U_k, A \rangle \frac{1}{\| U_k \|_F^2} U_k_{k,i} ; x_t \right)^2. \]

Given solution \( A(t, \lambda) \), we first apply the homotopy Algorithm of Section 3.3.2 on it with \( \lambda_1 = \lambda \) and \( \lambda_2 = \frac{t+1}{t} \lambda \) to change the constant before the old data term from \( \frac{1}{t} \) to \( \frac{1}{t+1} \). Then, we have \( A(t, \frac{t+1}{t} \lambda) = A_{\lambda,t}(0, \ldots, 0) \) and \( A(t+1, \lambda) = A_{\lambda,t}(1, \ldots, 1) \). We let evolve the optimization problem (3.15) from time \( t \) to \( t + 1 \) by sequentially varying all \( \mu_i \) from 0 to 1, along the paths
\[ L_{\lambda,t}(0, 0, \ldots, 0) \rightarrow L_{\lambda,t}(1, 0, \ldots, 0) \rightarrow L_{\lambda,t}(1, 1, \ldots, 1) = L_{\lambda,t+1}. \]

**Proposition 3.5.** A minimizer \( A_{\lambda,t}(\ldots, 1, \mu_i, 0, \ldots) \) of \( \min_{A \in \mathbb{K}_q} L_{\lambda,t}(\ldots, 1, \mu_i, 0, \ldots) \) satisfies the linear system
\[
\begin{cases}
\Gamma_0^1(\mu_i) a_i^s - \gamma_1^1(\mu_i) + (1 + \frac{1}{t}) \lambda w_1 = 0 \\
\Gamma_0^0(\mu_i) a_i^s - \gamma_1^0(\mu_i) + (1 + \frac{1}{t}) \lambda w_0 = 0,
\end{cases}
\]

where \( a_i, K_{q}^0, K_{N}^1, K^1, w \) are with respect to \( A = A_{\lambda,t}(\ldots, 1, \mu_i, 0, \ldots) \), defining furthermore the extractions through (3.26),
\[ \Gamma_0(\mu_i) = \Gamma_0 + \frac{1}{t} \sum_{n=1}^{i-1} [\tilde{X}_t]_{n, \mu_i} [\tilde{X}_t]_{n, i}^\top + \frac{\mu_i}{t} [\tilde{X}_t]_{i, \mu_i} [\tilde{X}_t]_{i, i}^\top, \]

and
\[ \gamma_1(\mu_i) = \gamma_1 + \frac{1}{t} \sum_{n=1}^{i-1} x_{t+1,n} [\tilde{X}_t]_{n, i} + \frac{\mu_i}{t} x_{t+1,i} [\tilde{X}_t]_{i, i}, \]

with \( \Gamma_0, \gamma_1 \) are the same ones as in Proposition 3.4.

The optimal conditions given in Proposition 3.5 show that, each path only relates to the one rank change: \( \frac{\mu_i}{t} [\tilde{X}_t]_{i, \mu_i} [\tilde{X}_t]_{i, i}^\top \), for the latest updated \( \Gamma_0 \). Thus we can apply the Sherman Morrison formula on \([\Gamma_0^1(\mu_i)]^{-1}\) to retrieve the smooth function of \( \mu_i \), and express \( a_i^s \) and \( w_0 \) as smooth functions of \( \mu_i \), which furthermore makes the calculation of the critical points of \( \mu_i \) explicit. To leverage these continuity properties, we still assume \([a_i^s]_{K_{q}^0} \neq 0 \) (element-wise), and \(|[w]_{K_{q}^0}| < 1 \) (element-wise). For the algorithm of path \( A_{\lambda,t}(0, \ldots, 0) \) to \( A_{\lambda,t}(1, \ldots, 1) \), it is sufficient to impose such assumption only on \( A_{\lambda,t}(0, \ldots, 0) \). By arguing as in Section 3.3.2, we can derive the homotopy algorithm for the whole data path. For details, see Algorithm 5 in the appendices.
3.3.4 Update from $\lambda_t$ to $\lambda_{t+1}$

Given the previous solution $A(t, \lambda_t)$, one way to select the hyperparameter value $\lambda$ is to introduce the empirical objective function [11, 21], which takes the form

$$f_{t+1}(\lambda) = \frac{1}{2} \|x_{t+1} - A(t, \lambda)x_t\|_{\ell_2}^2,$$

and to employ the updating rule

$$\lambda_{t+1} = \lambda_t - \eta \frac{df_{t+1}(\lambda)}{d\lambda} |_{\lambda=\lambda_t},$$

(3.39)

where $\eta$ is the step size. For convenience, we write $\frac{df_{t+1}(\lambda)}{d\lambda} |_{\lambda=\lambda_t}$ as $d f_{t+1}$. Analogously, we adopt the notation $d A(p, \lambda_t q)$ to denote the derivative with respect to $\lambda$, taken at value $\lambda_{t+1}$.

The objective function $f_{t+1}$ can be interpreted as a one-step prediction error on unseen data. Since the Lasso solution is piece-wise linear with respect to $\lambda$, it follows that when $\lambda$ is not a critical point, the derivative can be calculated as

$$d f_{t+1} = \langle Gr(x), \frac{d A(t, \lambda)}{d\lambda} \rangle = -\left[ a^G_1 \right]^\top \Gamma_0^{-1} w_1,$$

(3.40)

where $a^G_1 \in \mathbb{R}^{|K^1|}$ is defined as $\langle U_k, G(t) \rangle = \langle U_k, G^i \rangle$, $k_i \in K^1$, with $K^1$, $w_1$, $\Gamma_0^{-1}$ associated with $A(t, \lambda_t)$, and

$$G(t) = (A(t, \lambda_t)x_t - x_{t+1}) x_t^\top.$$

The derivatives of the entries of $A(t, \lambda)$ indexed by $K^1$ at $\lambda_t$ can be calculated through the formula (3.28) of $a^*_1$. By contrast, the derivatives of the entries of $A(t, \lambda)$ indexed by $K^0_N$ all equal zero. To obtain the non-negative parameter value, we project $\lambda_{t+1}$ onto interval $[0, +\infty)$ by taking max$\{\lambda_{t+1}, 0\}$, whenever the result from Equation (3.39) is negative.

Note that $\lambda_{t+1}$ defined in Equation (3.39) can be interpreted as the online solution from the projected stochastic gradient descent derived for the batch problem

$$\lambda^*_n = \arg\min_{\lambda \geq 0} \frac{1}{2n} \sum_{t=1}^n \|x_{t+1} - A(t, \lambda)x_t\|_{\ell_2}^2.$$

(3.41)

Therefore, the sublinear regret property of projected stochastic gradient descent implies that, when $\eta$ is given as $O\left(\frac{1}{\sqrt{n}}\right)$, we have

$$\frac{1}{2n} \sum_{t=1}^n \|x_{t+1} - A(t, \lambda_t)x_t\|_{\ell_2}^2 - \frac{1}{2n} \sum_{t=1}^n \|x_{t+1} - A(t, \lambda^*_n)x_t\|_{\ell_2}^2 = O\left(\frac{1}{\sqrt{n}}\right).$$

(3.42)
Equation (3.42) implies that in the sense of average one step prediction error defined as Equation (3.41), the adaptive hyperparameter sequence \( \{\lambda_t\}_t \) will perform almost as well as the best parameter \( \lambda^*_n \), for a large number of online updates, with sufficiently small step size \( \eta \). This completes the online procedure in the high dimensional domain, which we conclude in Algorithm 1.

**Algorithm 1** Online Structured matrix-variate Lasso

**Input:** \( A(t, \lambda_t), \Gamma_0, \gamma_1, K_N^1 \) (ordered list), \( w_N^1, \lambda_t, [\Gamma^1_0]^{-1}, x_{t+1}, X_t, t \), where \( K_N^1, w_N^1, [\Gamma^1_0]^{-1} \) are associated with \( A(t, \lambda_t) \), and \( w_N^1 = [w]_{K_N^1} \).

Select \( \lambda_{t+1} \) according to Section 3.3.4.

Update \( A(t, \lambda_t) \rightarrow A(t, \frac{t+1}{t} \lambda_{t+1}) \) using Algorithm 4.

Update \( A(t, \frac{t+1}{t} \lambda_{t+1}) \rightarrow A(t + 1, \lambda_{t+1}) \) using Algorithm 5.

**Output:** \( A(t + 1, \lambda_{t+1}), \Gamma_0, \gamma_1, K_N^1, w_N^1, \lambda_{t+1}, [\Gamma^1_0]^{-1} \).

4 Augmented Model for Periodic Trends

The online methods derived previously are based on the data process (2.5), which assumes the samples \( (x_\tau)_{\tau \in \mathbb{N}} \) have the time-invariant mean zero. In this section, we propose a more realistic data model which considers the trends, and adapt the online methods for stationary data to this augmented model.

In literature of time series analysis, stationarity is very often taken as part of model assumptions due to its analytic advantage. Meanwhile the raw data usually is not stationary, for example Figure 1. In Figure 4, we show moreover a comparison of stationary time series and non-stationary time series. Thus in offline learning, to fit the models on data, a *detrend* step is needed, which

![Figure 4](image-url)

**Figure 4**: Top is the stationary time series from Model (2.6) at one component, bottom is the time series from the augmented Model (4.1) in the same realisation.
approximates the trend function using the entire data set, then removes it from the raw data. However, since the principle of online learning does not require the presence of all data, such pre-processing step is forbidden. Thus, we need to consider the trend as the explicit parameters additional to the graph parameters $A_N, A_F$ in the online model. We propose the following augmented model

$$\begin{align*}
\begin{cases}
x_t &= b^0_t + x'_t, \\
x'_t &= A x'_{t-1} + z_t,
\end{cases}
\end{align*}$$

(4.1)

where $x_t = \text{vec}(X_t)$, $b^0_t \in \mathbb{R}^{NF}$, $z_t \in \mathbb{R}^{NF} \sim \text{IID} (0, \Sigma)$ with non-singular covariance matrix $\Sigma$ and bounded fourth moments, and $x'_0 = \sum_{j=0}^{\infty} A^j z_{t-j}$. The observations of Model (4.1) is $x_t$, while $x'_t$ has the similar role as the unobserved state in the state space models however the observation equation here is much simplified. Therefore the estimators are built on the series $x_t$. Note that Model (4.1) admits another reparameterization with intercept

$$x_t = b_t + A x_{t-1} + z_t, \quad b_t = b^0_t - A b^0_{t-1}. \quad (4.2)$$

The augmented model assumes that the non-stationarity of the observations $x_t$ is caused by the trend $b^0_t$. We consider in particular in this work, the periodic trend

$$b^0_t = b^0_m, \quad m = 0, ..., M - 1, \quad m = t \mod M^6, \quad (4.3)$$

where $M$ is the length of period and it is a hyperparameter to be preassigned. This type of trend is frequently encountered in practice. For example, an annual recurrence ($M = 12$) can be found in many monthly data sets recorded over years, such as the weather data in Figure 1. In the following sections, we will adapt the two learning frameworks presented in Section 3 to the augmented model for the periodic trends, in order to infer the trends and graphs simultaneously from non-stationary time series $x_t$, in an online fashion.

### 4.1 New OLS Estimators and Asymptotic Distributions

For the augmented model (4.1), we propose a new OLS estimator of $A$, which is based on the new sample auto-covariances, together with the OLS estimator of $b^0_m$. Because two crucial properties to derive the Wald tests in Section 3.2 are the consistency of sample auto-covariances $\hat{\Gamma}_t(0), \hat{\Gamma}_t(1)$, and the CLT of OLS estimator $\hat{A}_t$, we derive the corresponding asymptotic results for the new estimators, and show that these asymptotics are exactly the same as in the stationary case. Therefore, all the results and procedures presented in Section 3.2 can be applied directly on the

---

$^6$The modulo of a negative integer is defined by the positive reminder in this case, for example, $-1 \mod M = M - 1$. 

22
new estimators. We first define the estimator of $A$, still denoted as $\hat{A}_t$, using general least squares (GLS) method

$$\hat{A}_t, \hat{b}_{m,t} = \arg\min_{A,b_m} \sum_{m=0}^{M-1} S_m(A,b_m),$$

where

$$S_m = \sum_{\tau \in I_{m,t}} \tilde{z}_\tau \Sigma^{-1} \tilde{z}_\tau, \quad \tilde{z}_\tau = x_\tau - b_m - A x_{\tau-1},$$

with $I_{m,t} = \{\tau = 1, ..., t : \tau \mod M = m\}$, and $\Sigma^{-1}$ the true white noise covariance given in Model (2.5). Note that $\tilde{z}_\tau$ represents the residual of the prediction of sample $x_\tau$. The explicit forms of $\hat{A}_t, \hat{b}_{m,t}$ can be found through straightforward calculation, which yields new sample auto-covariances, denoted still as $p_{\Gamma t q}^0, p_{\Gamma t q}^1$, and the estimator of trend $\hat{b}_{0 m,t}^0$. Specifically, we have

$$\begin{align*}
\hat{A}_t &= \hat{\Gamma}_t(1) \left[ \hat{\Gamma}_t(0) \right]^{-1}, \\
\hat{b}_{m,t} &= \hat{x}_{m,t} - \hat{A}_t \hat{x}_{m-1,t} \Rightarrow \hat{b}_{0 m,t}^0 = \hat{x}_{m,t} \text{(or } \bar{x}_{m,t}),
\end{align*}$$

with

$$\begin{align*}
\hat{\Gamma}_t(0) &= \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left( \frac{\sum_{\tau \in I_{m,t}} x_{\tau-1} x_{\tau-1}^T}{p_{m,t}} - x_{m-1,t} x_{m-1,t}^T \right), \\
\hat{\Gamma}_t(1) &= \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left( \frac{\sum_{\tau \in I_{m,t}} x_{\tau} x_{\tau-1}^T}{p_{m,t}} - \bar{x}_{m,t} \bar{x}_{m-1,t}^T \right), \\
p_{m,t} &= |I_{m,t}|, \quad \bar{x}_{m,t} = \sum_{\tau \in I_{m,t}} \frac{x_\tau}{p_{m,t}}, \quad m = 0, ..., M - 1, \\
\bar{x}_{m-1,t} &= \sum_{\tau \in I_{m,t}} \frac{x_{\tau-1}}{p_{m,t}}, \quad m = 0, ..., M - 1.
\end{align*}$$

Note that $\bar{x}_{-1,t}$ denotes $\bar{x}_{M-1,t}$. It is also straightforward to understand the new auto-covariance estimators. Each $S_m(A,b_m)$ leads to an OLS problem of regression equation (4.2). Its minimization introduces two sample covariance matrices. The weighted average of all such sample auto-covariance matrices for $m = 0, \ldots, M - 1$ is the new sample auto-covariance for Model (4.1). $p_{m,t}$ denotes the number of times that the samples from the $m$-th state point in the period have been predicted in the sense of Equation (4.4). As $t$ grows, $\bar{x}_{m,t}$ becomes $\bar{x}_{m,t}$ quickly, and $p_{m,t}$ becomes $\frac{t}{M}$. For the augmented model, GLS and OLS estimators are still identical, with the latter defined as

$$\arg\min_{A,b_m} \sum_{m=0}^{M-1} \sum_{\tau \in I_{m,t}} \tilde{z}_\tau \tilde{z}_\tau.$$
The estimators given by Formula (4.4) enjoy the asymptotic properties in Proposition 4.1.

**Proposition 4.1.** The following asymptotic properties hold for the estimators \( \hat{\Gamma}_t(0), \hat{\Gamma}_t(1), \bar{A}_t, \bar{b}_{m,t}^0 \), as \( t \to +\infty \),

1. \( \hat{\Gamma}_t(0) \xrightarrow{p} \Gamma(0), \hat{\Gamma}_t(1) \xrightarrow{p} \Gamma(1) \),
2. \( \bar{b}_{m,t}^0 \xrightarrow{p} b_m^0, \bar{A}_t \xrightarrow{p} A \),
3. \( \sqrt{t} \text{vec}(\bar{A}_t - A) \overset{d}{\rightarrow} \mathcal{N}(0, [\Gamma(0)]^{-1} \otimes \Sigma) \),

where \( \Gamma(h) = \mathbb{E} \left( \left[ x'_t [x'_{t-h}]' \right]' \right), h \geq 0, \Sigma = \mathbb{E} \left( z_t z_t' \right) \).

The proofs of the above results are given in Appendix B. Thus, Theorem 3.3 and the bisection Wald test procedure are still valid using \( \text{Proj}_{\hat{\mathcal{G}}_N}(\bar{A}_t) \) and \( \hat{\Gamma}_t(0), \hat{\Gamma}_t(1) \) defined in this section. On the other hand, \( \hat{\Gamma}_t(0) \) and \( \hat{\Gamma}_t(1) \) satisfy the one rank update formulas

\[
\hat{\Gamma}_{t+1}(0) = \frac{t}{t+1} \hat{\Gamma}_t(0) + \frac{1}{t+1} \left[ \frac{p_{m,t}}{p_{\bar{m},t}} + 1 \right] (x_t - \bar{x}_{\bar{m}-1,t})(x_t - \bar{x}_{\bar{m}-1,t}'){\left[ \frac{p_{m,t}}{p_{\bar{m},t}} + 1 \right]}^{-1}, \\
\hat{\Gamma}_{t+1}(1) = \frac{t}{t+1} \hat{\Gamma}_t(1) + \frac{1}{t+1} \left[ \frac{p_{m,t}}{p_{\bar{m},t}} + 1 \right] (x_{t+1} - \bar{x}_{\bar{m}-1,t})(x_t - \bar{x}_{\bar{m}-1,t}'){\left[ \frac{p_{m,t}}{p_{\bar{m},t}} + 1 \right]}^{-1},
\]

(4.6)

where \( \bar{m} = (t + 1) \text{ mod } M \). Thus, when new sample comes, \( [\hat{\Gamma}_{t+1}(0)]^{-1} \) can still be calculated efficiently given the matrix inverse at the previous time. The details of the extended low dimensional learning procedure see Algorithm 3 in the appendices.

### 4.2 Augmented Structured Matrix-variate Lasso and the Optimality Conditions

To adapt the Lasso-based approach to Model (4.2), the corresponding trend and graph estimators can be obtained by minimizing the augmented Matrix-variate Lasso problem

\[
\mathbf{A}(t, \lambda), \mathbf{b}_m(t, \lambda) = \arg \min_{\mathbf{A} \in \mathcal{K}_A, \mathbf{b}_m} \frac{1}{2t} \sum_{m=0}^{M-1} \sum_{\tau \in I_{m,t}} \| x_{t+1} - \mathbf{b}_m - A x_{t+1} \|_2^2 + \lambda F \| A_N \|_{\ell_1}.
\]

(4.7)

As in the extension of our first approach, the extra bias terms \( b_m, m = 0, \ldots, M - 1 \), do not affect the core techniques, rather they force the methods to consider the \( M \) means in the sample autocovariances. Since \( b_m \) only appear in the squares term, the minimizers \( \mathbf{b}_m(t, \lambda) \) have the same dependency with \( \mathbf{A}(t, \lambda) \) as in Equation (4.5). Thus the trend \( \mathbf{b}_m^0 \) can still be estimated by \( \bar{x}_{m,t} \), and we extend the algorithms in Section 3.3 to update the batch solution of augmented Lasso (4.7) from \( \mathbf{A}(t, \lambda_t) \) to \( \mathbf{A}(t+1, \lambda_{t+1}) \), given new sample \( x_{t+1} \). To compute the regularization path \( \mathbf{A}(t, \lambda_t) \to \mathbf{A}(t, (1 + \frac{1}{t})\lambda_{t+1}) \), Proposition 4.2 implies that Algorithm 4 can still be used, with the adjusted definitions of \( \Gamma_0 \) and \( \gamma_1 \).
Proposition 4.2. A minimizer $A(t, \lambda)$ of Lasso problem (4.7) satisfies the linear system

$$ \begin{cases} \Gamma_0^1 a_1^s - \gamma_1^1 + \lambda w_1 = 0 \\ \Gamma_0^0 a_1^s - \gamma_0^0 + \lambda w_0 = 0, \end{cases} $$

(4.8)

where $a^s$ is the vectorized scaled Lasso solution $A(t, \lambda)$, $w, K^1, K_N^0$ are also defined analogously from $A(t, \lambda)$, while $\hat{\Gamma}_t(0)$ and $\hat{\Gamma}_t(1)$ used in the definitions of $\Gamma_0$ and $\gamma_1$ are the new sample auto-covariance matrices in Equation (4.5).

For the data path $A(t, (1 + \frac{1}{t}) \lambda_{t+1}) \rightarrow A(t + 1, \lambda_{t+1})$, in the same spirit of Problem (3.34), we introduce variables $\mu_1, ..., \mu_{NF}$ to let evolve Lasso problem (4.7) from time $t$ to $t + 1$ through the following variational problem

$$ A_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF}), b_{m,\lambda_{t+1},t}(\mu_1, ..., \mu_{NF}) = \arg \min_{A \in K_0, b_m} L_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF}), $$

where $L_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF}) = \frac{1}{2(t+1)} \sum_{m=0}^{M-1} \sum_{r \in \Omega_{m,t}} \| \bar{x}_r - b_m - A \bar{x}_{t-1} \|_2^2$

$$ + \lambda_{t+1} F_{\lambda} \| \bar{x}_{t+1} \|_1 + \frac{1}{2(t+1)} \sum_{i=1}^{NF} \mu_i (\bar{x}_{t+1;i} - \bar{m}_i) - \sum_{k \in K} \langle U_k, A^0 \rangle \frac{1}{\| U_k \|_F^2} \langle U_k, \bar{x}_{t} \rangle^2, $$

(4.9)

where $b_{\bar{m},i} = [b_{\bar{m}}]$, $\bar{m} = (t + 1) \mod M$. To extend the homotopy Algorithm of data path, we first calculate the optimality conditions of Lasso $L_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF})$ with respect to the constraint-free $A^0$ in Equation (4.10), then extract its vector representation in terms of $a^s$.

A minimizer $A^0$ such that $A = \text{Proj}_G(A^0)$ of $L_{\lambda_{t+1},t}(\mu_1, ..., \mu_{NF})$ satisfies

$$ 0 \in \frac{\partial L_{t+1}(\mu_1, ..., \mu_{NF})}{\partial A^0} $$

$$ = \frac{t}{t+1} \left[ \sum_{k,k' \in K} \langle U_{k}, U_{k'} \hat{\Gamma}_t(0) \rangle \langle \frac{1}{\| U_k \|_F^2} U_{k'}, A^0 \rangle \frac{1}{\| U_k \|_F^2} U_k - \sum_{k \in K} \langle U_{k}, \hat{\Gamma}_t(1) \rangle \langle \frac{1}{\| U_k \|_F^2} U_{k'} \rangle \frac{1}{\| U_k \|_F^2} U_k \right] $$

$$ + \frac{1}{t+1} \sum_{i=1}^{NF} \mu_i \frac{p_{m,t}}{p_{m,t} + \mu_i} \left[ \sum_{k,k' \in K} (\bar{x}_{t} - \bar{x}_{m-1;t})^T [U_k]_{t;} [U_k]_{t;} (\bar{x}_{t} - \bar{x}_{m-1;t}) \langle \frac{1}{\| U_k \|_F^2} U_{k'} \rangle \frac{1}{\| U_k \|_F^2} U_k \right] $$

$$ - \frac{1}{t+1} \sum_{i=1}^{NF} \mu_i \frac{p_{m,t}}{p_{m,t} + \mu_i} \left[ \sum_{k \in K} (\bar{x}_{t+1;i} - \bar{x}_{\bar{m},t;i})^T [U_k]_{t;} [U_k]_{t;} (\bar{x}_{t} - \bar{x}_{m-1;t}) \langle \frac{1}{\| U_k \|_F^2} U_{k'} \rangle \frac{1}{\| U_k \|_F^2} U_k \right] $$

$$ + \lambda_{t+1} \sum_{k \in K_N} \partial \langle U_{k}, A^0 \rangle \langle \frac{1}{\| U_k \|_F^2} U_{k} \rangle, $$

(4.10)
where \( \hat{f}_t(0), \hat{f}_t(1) \) are the ones defined in Equation (4.5), and \( \mathbf{x}_{\hat{m},t} = \begin{bmatrix} \mathbf{x}_m,t \end{bmatrix}_i \). Note that, when \((t \mod M) \neq m\), \( \bar{x}_m,t = \bar{x}_m,t \).

The following remarks can then be made.

**Remark 2.** Subdifferential formula (4.10) is almost the same as its stationary counterpart except that the former uses centered data, as well as the appearance of term \( \frac{p_{m,t}}{p_{n,t} + \mu_i} \).

**Remark 3.** Equation (4.10) implies the update formula (4.6). Since

\[ L_{\lambda_{t+1},t}(1, ..., 1) = L_{\lambda_{t+1},t+1}, \]

and \( \frac{\partial L_{\lambda_{t+1},t+1}}{\partial A_0} \) is given in Equation (3.18), with \( \lambda = \lambda_{t+1}, \hat{g}_{t+1}(0), \hat{g}_{t+1}(1) \) defined alternatively in Equation (4.5). Thus, by equating the quantities in \( \langle U_k, U_k' \rangle \) and \( \langle U_k, \cdot \rangle \) in the corresponding subdifferential formulas respectively, update formula (4.6) can be induced.

We recall that we update the solution along the path

\[ L_{\lambda_{t+1},t}(0, 0, ..., 0) \rightarrow L_{\lambda_{t+1},t}(1, 0, ..., 0) \rightarrow L_{\lambda_{t+1},t}(1, 1, ..., 1) = L_{\lambda_{t+1},t+1}. \]

At each step \( L_{\lambda_{t+1},t}(\ldots, 1, \mu_i, 0, \ldots), \mu_i \in [0, 1] \), the optimal solution \( A_{\lambda_{t+1},t}(\ldots, 1, \mu_i, 0, \ldots) \) is piecewise smooth with respect to \( \mu_i \), element-wise. We retrieve the linear system of \( a_1^s \) in terms of \( \mu_i \) for each \( A_{\lambda_{t+1},t}(\ldots, 1, \mu_i, 0, \ldots) \) in Proposition 4.3.

**Proposition 4.3.** A minimizer \( A_{\lambda_{t+1},t}(\ldots, 1, \mu_i, 0, \ldots) \) of Lasso \( L_{\lambda_{t+1},t}(\ldots, 1, \mu_i, 0, \ldots) \) satisfies the linear system

\[
\begin{cases}
\Gamma_0^t(\mu_i)a^s_1 - \gamma_1^0(\mu_i) + (1 + \frac{1}{t})\lambda_{t+1}w_1 = 0 \\
\Gamma_0^t(\mu_i)a^s_1 - \gamma_0^0(\mu_i) + (1 + \frac{1}{t})\lambda_{t+1}w_0 = 0,
\end{cases}
\]

(4.11)

where \( a^s, w, K^1, K_0^0 \) are with respect to \( A_{\lambda_{t+1},t}(\ldots, 1, \mu_i, 0, \ldots) \), defining the extractions through (3.26),

\[
\Gamma_0(\mu_i) = \Gamma_0 + \frac{1}{t} \sum_{n=1}^{i-1} \frac{p_{m,t}}{p_{m,t} + 1} [\bar{X}_t - \bar{X}_{\hat{m} - 1,t};n] [\bar{X}_t - \bar{X}_{\hat{m} - 1,t}];n
\]

\[
+ \frac{\mu_i}{t} \frac{p_{m,t}}{p_{m,t} + \mu_i} [\bar{X}_t - \bar{X}_{\hat{m} - 1,t}]_i [\bar{X}_t - \bar{X}_{\hat{m} - 1,t}];i,
\]

\[
\gamma_1(\mu_i) = \gamma_1 + \frac{1}{t} \sum_{n=1}^{i-1} \frac{p_{m,t}}{p_{m,t} + 1} (x_{t+1,n} - (x_{\hat{m},t})_n) [\bar{X}_t - \bar{X}_{\hat{m} - 1,t}];n
\]

\[
+ \frac{\mu_i}{t} \frac{p_{m,t}}{p_{m,t} + \mu_i} (x_{t+1,i} - (x_{\hat{m},t})_i) [\bar{X}_t - \bar{X}_{\hat{m} - 1,t}];i,
\]

with \( [\bar{X}_{\hat{m} - 1,t}]_{k,i} = [U_k]_{i,:} X_{\hat{m} - 1,t}, p_{m,t} := t \mod M, \Gamma_0, \gamma_1 \) the same as Proposition 4.2.
Therefore, the derived homotopy algorithm is essentially the previous homotopy Algorithm 5 with minor changes. For details, see Algorithm 6 in the appendices.

Lastly, we derive the updating rule for the regularization parameter. We still consider the one step prediction error, which writes as the following objective function in the case of Model (4.2)

\[
f_t(\lambda) = \frac{1}{2} \| x_{t+1} - b_{\bar{m}}(t, \lambda) - A(t, \lambda)x_t \|_2^2.
\]  

(4.12)

Given the previous solution \( A(t, \lambda_t) \) and \( b_{\bar{m}}(t, \lambda_t) \), we assume that \( \lambda_t \) is not a critical point. Then the derivative of \( f_t \) with respect to \( \lambda \) is calculated as

\[
\frac{df_t(\lambda_t)}{d\lambda} = \left< \frac{df_t(\lambda)}{db_{\bar{m}}(t, \lambda)} \right|_{\lambda=\lambda_t} db_{\bar{m}}(t, \lambda_t) + \left< \frac{df_t(\lambda)}{dA(t, \lambda)} \right|_{\lambda=\lambda_t} dA(t, \lambda_t)
\]

\[
= \left< G^b_t - \frac{dA(t, \lambda_t)}{d\lambda} \bar{x}_{n-1,t} \right> + \left< G_t, \frac{dA(t, \lambda_t)}{d\lambda} \right>
\]

\[
= \left< A(t, \lambda_t)x_t - x_{t+1} + b_{\bar{m}}(t, \lambda_t), [x_t - \bar{x}_{n-1,t}]^\top \right>,
\]

(4.13)

where \( G^b_t = b_{\bar{m}}(t, \lambda_t) - x_{t+1} + A(t, \lambda_t)x_t \), \( G_t = [A(t, \lambda_t)x_t - x_{t+1} + b_{\bar{m}}(t, \lambda_t)]x_t^\top \). Analogous to Section 3.3.4, we have \( \left< G_t, \frac{dA(t, \lambda_t)}{d\lambda} \right> = -\left[ a_1^G_t \right]^\top \left[ \Gamma_0 \right]^{-1} w_1 \). Using the same updating rules of the projected stochastic gradient descent presented in Section 3.3.4, we can compute the online solution \( \lambda_{t+1} \). We can see that, the introduction of bias terms \( b_m \) into the original model makes them center the raw data automatically during the model fitting. This enables the direct learning over raw time series, while maintaining the performance of methods comparable to the stationarity-based ones.

We summarize the complete learning procedure of this subsection in Algorithm 6 in the appendices.

5 Experiments

We test the two proposed approaches for the online graph and trend learning on both synthetic and real data sets.

5.1 Synthetic Data

5.1.1 Evaluation Procedures

We now present the evaluation procedure for the augmented model approaches. In each simulation, we generate a true graph \( A \) with the structure indicated by \( K_G \). In particular, we impose sparsity on its spatial graph \( A_N \) by randomly linking a subset of node pairs. The values of non-zero entries in \( A_N \), and the entries in \( A_F \), \( \text{diag}(A) \) are generated in a random way. Additionally, we generate
a trend over a period of \( M \) time points for each node and each feature. Therefore, the true
\( b^0_m, m = 0, ..., M - 1 \), consists in these \( NF \) trend vectors, each containing \( M \) elements. Then,
we synthesize very few samples \( x_t \) from Model (4.1) until time \( t_0 \). Figure 4 shows an example of
the synthetic time series \( x_t \), compared with its stationary source \( x'_t \) before adding the periodic
trend. The graph and trend estimators proposed in Section 4 only use \( x_t \). We then set up the
batch Lasso problem (4.7) with the generated samples and use its solution \( A(t_0, \lambda_0) \) to start the
high-dimensional online procedure since the next synthetic sample. The batch problem is solved
via the accelerated proximal gradient descent with the backtracking line search [23, Section 3.2.2].
We especially set \( \lambda_0 \) as a large number so as to have an over sparse initial solution. Therefore, we
expect to see a decreasing \( \lambda_t \), together with a more accurate estimate \( A(t, \lambda_t) \) as \( t \) grows. The
updating of \( \hat{\Gamma}_t(0) \) and \( \hat{\Gamma}_t(1) \) starts from \( t = 1 \), using Formula (4.6), since they are the only inputs
of the proximal gradient descent algorithm. However, we wait until there are enough samples
for \( \hat{\Gamma}_t(0) \) to be invertible, then we start the low-dimensional online procedure at time \( t \) with
\( [\hat{\Gamma}_t(0)]^{-1} \). To analyse the performance of the proposed approaches, we define the average one
step prediction error metric as,
\[
\sum_{\tau=1}^{t} \frac{\|x_{\tau+1} - \hat{b}_{m(\tau+1)},\tau - \hat{A}_\tau x_{\tau}\|_2}{t\|x_{\tau+1}\|_2}, \quad m(\tau + 1) = (\tau + 1) \mod M, \tag{5.1}
\]
and root mean square deviation (RMSD) as,
\[
\frac{\|\hat{A}_t - A\|_F}{\|A\|_F}, \tag{5.2}
\]
where \( \hat{A}_\tau \) denotes the online estimates from either approach at time \( \tau \), and
\[
\hat{b}_{m(\tau+1),\tau} = \begin{cases} \hat{x}_{m(\tau+1),\tau} - \hat{A}_\tau x_{m(\tau+1)-1,\tau}, & \text{in low-dimensional}, \\ \hat{x}_{m(\tau+1),\tau} - A(\tau, \lambda_\tau) x_{m(\tau+1)-1,\tau}, & \text{in high-dimensional}. \end{cases} \tag{5.3}
\]
We collect the metric values along time. We perform such simulation multiple times to obtain
furthermore the means and the standard deviations of error metrics at each iteration (when the
estimators are available) to better demonstrate the performance. The true graph \( A \) and trends
\( b^0_m \) are generated independently across these simulations.

5.1.2 Simulation Results

We first visualize the representative estimates in heatmap with \( N = 10, F = 4, \) and \( M = 12 \) for
illustration purpose. Then we plot the evolution of error metrics and regularization parameter
of 30 simulations for \( N = 20, F = 5, \) and \( M = 12 \). Lastly, we report the running time. The
hyperparameter settings are given in the captions of figures of corresponding results.
Figures 5 and 6 show the estimated graphs of two approaches when their corresponding online procedures start. In Figure 5, we can see that the batch solution which starts the high-dimensional procedure is over sparse due to the large $\lambda_0$. We can notice from Figure 6 that the two initial estimations of $A_F$ are already satisfactory, especially the Lasso solution which uses only 20 samples. Actually, estimations of $A_F$ and $\text{diag}(A)$ converge to the truth very quickly in both cases when $N$ is significantly larger than $F$. Figures 7 and 8 show that the estimations of $A_N$ of both approaches tend to the true values as more samples are received. Meanwhile, Figure 9 shows the effectiveness of trend estimator $X_{m,t}$ defined in Equation (4.5).

We now show the numeric results of 30 simulations, with $N = 20$, $F = 5$, and $M = 12$. We test three different step sizes $\eta$, $5 \times 10^{-7}$, $1 \times 10^{-6}$, and $5 \times 10^{-6}$. With each value we perform 10 independent simulations. Figure 10 and 11 plot the evolution of error metrics (5.1) and (5.2), respectively. For better visualization effect, since the performance of the low-dimensional
Figure 7: Spatial graph estimated at the arrival of the 182-th sample. True $A_N$ (left), $\hat{A}_{N,182}$ of the low-dimensional procedure (middle), and $A_N(182,0.0286)$ of the high-dimensional procedure (right) are represented by heatmaps. Simulation settings: $N = 10$, $F = 4$, number of model parameters = 571, significance level of $\chi^2$ test = 0.1, $\eta = 5 \times 10^{-6}$.

Figure 8: Spatial graph estimated at the arrival of the 591-th sample. True $A_N$ (left), $\hat{A}_{N,591}$ of the low-dimensional procedure (middle), and $A_N(591,0.0130)$ of the high-dimensional procedure (right) are represented by heatmaps. Simulation settings: $N = 10$, $F = 4$, number of model parameters = 571, significance level of $\chi^2$ test = 0.1, $\eta = 5 \times 10^{-6}$.

procedure does not depend on $\eta$, we only show one mean metric curve instead of 3, in the two figures, which is calculated from the results of these 30 simulations.

Figures 10 and 11 show the convergence of $A_N$ estimations of both procedures. Moreover, for the high-dimensional procedure, the step size $\eta$ determines the convergence speed. Especially, we can see from Figure 10 that the RMSD of the high-dimensional procedure with $\eta = 5 \times 10^{-6}$ decreases the most quickly for the first 100 iterations, after which it starts to slow down and decrease more slowly than the RMSDs of the other two step sizes. For the low-dimensional procedure, when its estimator is available, the RMSD decreases very fast, and it shows the trend to keep decreasing for larger sample size. Nevertheless, the estimator of the low-dimensional procedure performs worse than the Lasso estimators in the sense of the prediction of unseen data, as shown in Figure 11. This is likely linked with the fact that the selection procedure updates
Figure 9: *Trend of the first node, first feature, estimated at different times.* Estimation at $t = 182$ (left), $t = 273$ (middle), $t = 591$ (right). Simulation settings: $N = 10$, $F = 4$, $M = 12$, number of model parameters = 571.

The regularization parameter of the Lasso estimator towards the direction that minimizes the one step prediction error (5.1). The larger standard deviation is due to the larger magnitude of low-dimensional estimator, contrast to the Lasso estimator which is regularized by the $\ell_1$ norm. This can also be observed in the scales of the $y$–axis in Figures 5 - 8. For synthetic data, it is

Figure 10: *Root mean square deviation.* The red curves are the mean RMSD of the high-dimensional procedure, taken over 10 simulations each. The blue curve is the mean RMSD of the low-dimensional procedure, taken over the same 30 simulations. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters = 1500, significance level of $\chi^2$ test = 0.1, $t_0 = 20$, $\lambda_0 = 0.03$. In the first high dimensional phase, the accurate estimator of the low-dimensional procedure is not available.
Figure 11: Average one step prediction error. The red curves are the mean prediction error of the high-dimensional procedure, taken over 10 simulations each. The blue curve is the mean prediction error of the low-dimensional procedure, taken over the same 30 simulations. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters = 1500, significance level of $\chi^2$ test = 0.1, $t_0 = 20$, $\lambda_0 = 0.03$.

It is not surprising that the RMSD from the low-dimensional procedure will tend toward zero, because these data are precisely sampled from the model used in the method derivation. On the other hand, at each online iteration, the OLS estimation is calculated accurately. In contrast, for the homotopy algorithms, they still introduce small errors, possibly due to the following assumptions used in the derivation of the method: 1. the active elements of $K^1_N$ of the algorithm inputs are not zero\(^7\); 2. the sub-derivatives of those zero elements are strictly within $(-1, 1)$; 3. every $\lambda_t$ at which we calculate the derivative as in Section 3.3.4 is not a critical point. Thus, for example, small non-zero entry values in the inputs may cause the numerical errors. However, in real applications, the only available metric which allows the performance comparison is the prediction error (5.1).

Figure 12 demonstrates the performance of the updating method of the regularizing parameter $\lambda$, and the impact from different step size values $\eta$. The curves emphasize the convergence of the estimation updated by the high-dimensional procedure. Moreover, we can observe that $\lambda_t$ are decreasing, which was expected from the experiment design. On the other hand, the results show that a larger step size will make the convergence faster, yet more affected by the noise, especially when the solution has converged.

We also compare the running time of a single online update for the two methods in Figure 13. Firstly, it is clear that updating the Lasso solutions by the homotopy algorithms saves considerable

---

\(^7\)This hypothesis means that, some zero $(a^\dagger)^{(k)}$, $k \in K^1_N$ should not satisfy the first equations of the optimality condition (3.27) and (3.35), due to the computation coincidence.
time, which is on average 0.20 seconds for the graph size $N = 20$, $F = 5$. The running time of the accelerated proximal gradient descent performed in the beginning of these simulations costs more than 3 seconds. By contrast, an update using the low-dimensional procedure takes 25 seconds on average. We can also notice that the high-dimensional procedure with larger step size runs slower, because the updated regularization parameter is quite different from the preceding one, as evidenced by the results in Figure 12.

Lastly, it is worthwhile to point out that, because the true $A_N$ has a high level of sparsity, the Wald test will accept $H_0 : \alpha = 0$ more easily with lower significance levels, and we can observe the Wald estimator $\hat{A}_{N,t}$ rejects those false non-zero entries faster. Nevertheless, since we do not know the true graph sparsity for real data, the significance level can be regarded as the hyperparameter which controls the desired sparsity as well for the first approach.

5.2 Climatology Data

We use the U.S. Historical Climatology Network (USHCN) data\(^8\) to test our proposed approaches. The data set contains monthly averages of four climatology features, recorded at weather stations located across the United States, over years. The four features are: minimal temperature, maximal temperature, mean temperature, and precipitation. A snippet of the data set has been given in Figure 1, which illustrates these feature time series observed from a certain spatial location. A clear periodic trend can be seen from each scalar time series, with period length equal to 12

\(^8\)The data set is available at https://www.ncdc.noaa.gov/ushcn/data-access
Figure 13: Running time of each online update. The red curves are the mean running time of the high-dimensional procedure, taken over 10 simulations each. The blue curve is the mean running time of the low-dimensional procedure, taken over the same 30 simulations. The shaded areas represent the corresponding one standard deviations. Other simulation settings: $N = 20$, $F = 5$, $M = 12$, number of model parameters = 1500, significance level of $\chi^2$ test = 0.1, $t_0 = 20$, $\lambda_0 = 0.03$.

months. We can also notice that some observations are missing in the data set; to focus on the evaluation of learning approaches, we do not consider the stations with incomplete time series. Geographically, we picked data only from California and Nevada for this experiment. The summary of experiment setting thus is: $N = 27$, $F = 4$, $M = 12$, total number of time points = 1523 months (covering the years from 1894 to 2020). We apply the approaches from Section 4 on the raw time series to learn the weather graph of the region. The testing procedure using the real data is identical to the evaluation procedure with the synthetic data. We use the first $t_0 + 1$ observations to set up the corresponding batch Lasso problem, and use its solution to start the high-dimensional procedure. The low-dimensional procedure will be started once $\hat{\Gamma}_{t_0}(0)$ becomes invertible. The average one step prediction error is calculated along online iterations. $t_0$ and $\lambda_0$ are always set as 20 and 0.03, respectively. Their values do not affect the methods’ performance much, because of the adaptive tuning procedures of the regularization parameter.

Figure 14 and 15 show the spatial graphs learned by the two proposed approaches in Section 4 updated at different times. Figure 16 plots the evolution of regularization parameter value.
We can see that, for the high-dimensional procedure, when more observations are received, it finds that more location pairs actually have a Granger causal effect on each other. On the other hand, compared to the estimated graphs from the high-dimensional procedure, those from the low-dimensional procedure vary more along time, which can be caused by the following facts: 1. in the early stage, $\hat{\Gamma}_t(0)$ is still ill-conditioned, therefore its inverse brings unstable OLS solutions; 2. the low-dimensional procedure relies on large sample properties of the designed estimators. These points are also supported by the average one step prediction curve given in Figure 17, where it is shown that, the prediction error of the low-dimensional procedure is significantly larger than the high-dimensional procedure, especially when the sample size is around 500 to 800.

Figure 14: Updated spatial graph by the low-dimensional procedure at different times. $t = 507$ (left), $t = 1015$ (middle), and $t = 1522$ (right). Experiment settings: $N = 27$, $F = 4$, $M = 12$, number of model parameters = 1761, significance level of $\chi^2$ test = 0.1, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$. The row labels are the 6-digit Cooperative Observer Identification Number of the corresponding weather stations.

Figure 15: Updated spatial graph by the high-dimensional procedure at different times. $t = 507$ (left), $t = 1015$ (middle), and $t = 1522$ (right). Experiment settings: $N = 27$, $F = 4$, $M = 12$, number of model parameters = 1761, significance level of $\chi^2$ test = 0.1, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$. The rows and columns correspond to the weather stations whose 6-digit Cooperative Observer Identification Number are given by the row labels.
Next we show the last updated feature graphs in Figure 18. We can see that the estimated feature relationships from the two approaches coincide in \( t_{\min} \) and \( t_{\max} \), \( t_{\min} \) and \( t_{\avg} \), \( t_{\min} \) and \( \text{prcp} \). However, the relationship between \( t_{\avg} \) and \( \text{prcp} \) is very weak in the Lasso estimation, while strong in the projected OLS estimation.

In particular, Figure 19 reports the evolution of estimated trends from one representative spatial location along time, where we can observe the increase of temperature from the past to the present.

Lastly, in Figure 20, we plot the edge overlap (considering the signs of weights) of the two last updated spatial graphs, where we also visualize this spatial graph superimposed on the actual geographical graph. We can see that the remote weather stations have less dependency with other stations, while more edges appear within the area where lots of stations are densely located together. These observations imply that the inferred graphs provide the consistent weather patterns with geographical features. Furthermore, they validate the legitimacy of Models (2.5) and (4.1), as well as the effectiveness of the proposed learning methods.
Figure 17: 
$\textit{Average one step prediction error of raw time series.}$ the low-dimensional procedure (top), and the high-dimensional procedure (bottom).

Figure 18: 
$\textit{Updated feature graph at } t = 1522.$ the low-dimensional procedure (left), and the high-dimensional procedure (right). Experiment settings: $N = 27, F = 4, M = 12$, number of model parameters $= 1761$, significance level of $\chi^2$ test $= 0.1$, $\eta = 10^{-5}$, $t_0 = 20$, $\lambda_0 = 0.03$.

6 Conclusion

In this paper, we proposed a novel auto-regressive model for matrix-variate time series with periodic trends. We devised two online learning frameworks respectively for low and high dimensions. Especially in the high dimensional regime, we introduce the novel Lasso type (3.15) and extend the classical homotopy algorithms. Lasso (3.15) differs from the classical Lasso
Figure 19: *Estimated trends along years.* On the left, middle, right are the estimated trends at different years of Station USH00040693 for minimal temperature, average temperature, and precipitation respectively. Experiment settings: $N = 27$, $F = 4$, $M = 12$.

Figure 20: *Overlap spatial graph.* On the left is the adjacency matrix of an unweighed undirected graph which is the overlap of the two last updated spatial graphs in Figure 15, with the colors reporting the common edge signs. On the right is the visualization of this overlap spatial graph on the actually geographical map. The nodes with bigger sizes connect with more nodes.

by requiring the coefficient matrix $A$ to have the desired structure indicated by $K_G$ and the partial sparsity penalized by $\|A_N\|_{\ell_1}$. Thus the derivation of the homotopy algorithms provides useful techniques to address the structure constraint. Moreover, this derivation does not rely
on the specific structure, nor on the particular partial sparsity regularization. Therefore, they can be applied to other model designs. Other model extensions are possible, for example, from matrix-variate to tensor-variate time series by using the multi-way Kronecker sum notion, or considering more time lag terms in the matrix-variate AR model, and accordingly replacing Lasso penalty with group Lasso penalty in (3.15). We tested our approaches using both synthetic and real data, and observed very promising results.

Acknowledgments

Jérémie Bigot is a member of Institut Universitaire de France (IUF), and this work has been carried out with financial support from the IUF.

References

[1] Bach, F. R. and Jordan, M. I. Learning graphical models for stationary time series. IEEE transactions on signal processing, 52(8):2189–2199, 2004.

[2] Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1):183–202, 2009.

[3] Bolstad, A., Van Veen, B. D., and Nowak, R. Causal network inference via group sparse regularization. IEEE transactions on signal processing, 59(6):2628–2641, 2011.

[4] Bonilla, E. V., Chai, K., and Williams, C. Multi-task gaussian process prediction. Advances in neural information processing systems, 20, 2007.

[5] Chen, R., Xiao, H., and Yang, D. Autoregressive models for matrix-valued time series. Journal of Econometrics, 222(1):539–560, 2021.

[6] Chen, S. and Chen, X. Weak connectedness of tensor product of digraphs. Discrete Applied Mathematics, 185:52–58, 2015.

[7] Dong, X., Thanou, D., Rabbat, M., and Frossard, P. Learning graphs from data: A signal representation perspective. IEEE Signal Processing Magazine, 36(3):44–63, 2019.

[8] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. Least angle regression. Annals of statistics, 32(2):407–499, 2004.

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

[10] Friedman, J., Hastie, T., and Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1):1, 2010.
[11] Garrigues, P. and Ghaoui, L. An homotopy algorithm for the lasso with online observations. Advances in neural information processing systems, 21:489–496, 2008.

[12] Greenewald, K., Zhou, S., and Hero III, A. Tensor graphical lasso (teralasso). Journal of the Royal Statistical Society: Series B (Statistical Methodology), 81(5):901–931, 2019.

[13] Hammack, R. H., Imrich, W., Klavžar, S., Imrich, W., and Klavžar, S. Handbook of product graphs, volume 2. CRC press Boca Raton, 2011.

[14] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

[15] Imrich, W. and Peterin, I. Cartesian products of directed graphs with loops. Discrete Mathematics, 341(5):1336–1343, 2018.

[16] Kalaitzis, A., Lafferty, J., Lawrence, N. D., and Zhou, S. The bigraphical lasso. In International Conference on Machine Learning, pp. 1229–1237. PMLR, 2013.

[17] Lütkepohl, H. New introduction to multiple time series analysis. Springer Science & Business Media, 2005.

[18] Malioutov, D. M., Cetin, M., and Willsky, A. S. Homotopy continuation for sparse signal representation. In Proceedings.(ICASSP’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., volume 5, pp. v–733. IEEE, 2005.

[19] Mei, J. and Moura, J. M. Signal processing on graphs: Causal modeling of unstructured data. IEEE Transactions on Signal Processing, 65(8):2077–2092, 2016.

[20] Meinshausen, N., Bühlmann, P., et al. High-dimensional graphs and variable selection with the lasso. Annals of statistics, 34(3):1436–1462, 2006.

[21] Monti, R. P., Anagnostopoulos, C., and Montana, G. Adaptive regularization for lasso models in the context of nonstationary data streams. Statistical Analysis and Data Mining: The ASA Data Science Journal, 11(5):237–247, 2018.

[22] Osborne, M. R., Presnell, B., and Turlach, B. A. A new approach to variable selection in least squares problems. IMA journal of numerical analysis, 20(3):389–403, 2000.

[23] Parikh, N. and Boyd, S. Proximal algorithms. Foundations and Trends in optimization, 1(3):127–239, 2014.

[24] Sandryhaila, A. and Moura, J. M. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. IEEE Signal Processing Magazine, 31(5):80–90, 2014.
[25] Songsiri, J. and Vandenberghe, L. Topology selection in graphical models of autoregressive processes. *The Journal of Machine Learning Research*, 11:2671–2705, 2010.

[26] Wang, Y., Jang, B., and Hero, A. The sylvester graphical lasso (syglasso). In *International Conference on Artificial Intelligence and Statistics*, pp. 1943–1953. PMLR, 2020.

[27] Zaman, B., Ramos, L. M. L., Romero, D., and Beferull-Lozano, B. Online topology identification from vector autoregressive time series. *IEEE Transactions on Signal Processing*, 69:210–225, 2020.

A Proof of Results in Section 3.2 and the CLT for $\hat{A}_t$

**Proof of Theorem 3.3.** By Cramér-Wold theorem, $\sqrt{t} \text{vec}(\mathbf{A}_t - A) \overset{d}{\rightarrow} \mathcal{N}(0, \Sigma_{ols})$ is equivalent to

$$
\langle \Lambda, \sqrt{t} \left( \mathbf{A}_t - A \right) \rangle \overset{d}{\rightarrow} \mathcal{N}(0, \text{vec}(\Lambda)^\top \Sigma_{ols} \text{vec}(\Lambda)), \quad \forall \Lambda \in \mathbb{R}^{NF \times NF}.
$$

(A.1)

On the other hand, we can express the entries of $\text{vec} \left( \sqrt{t} \left( \mathbf{A}_N - A_N \right) \right)$ as a linear function of $\mathbf{A}_t$

$$
\text{vec} \left( \sqrt{t} \left( \mathbf{A}_N - A_N \right) \right) = \sum_{k \in K_N} \langle U_k, \sqrt{t} \left( \mathbf{A}_t - A \right) \rangle \text{vec}(E_k).
$$

(A.2)

Then for all $\lambda \in \mathbb{R}^{N(N-1)/2}$, we have

$$
\lambda^\top \text{vec} \left( \sqrt{t} \left( \mathbf{A}_N - A_N \right) \right) = \langle \sum_{k \in K_N} \lambda^\top \text{vec}(E_k)U_k, \sqrt{t} \left( \mathbf{A}_t - A \right) \rangle.
$$

Let $\Lambda$ in Equation (A.1) be $\sum_{k \in K_N} \lambda^\top \text{vec}(E_k)U_k$, then we have

$$
\lambda^\top \text{vec} \left( \sqrt{t} \left( \mathbf{A}_{N,t} - A_N \right) \right) \overset{d}{\rightarrow} \mathcal{N}(0, \text{vec}(\Lambda)^\top \Sigma_{ols} \text{vec}(\Lambda)).
$$

Note that, $\text{vec}(\Lambda) = \sum_{k \in K_N} \lambda^\top \text{vec}(E_k)\text{vec}(U_k)$. Thus $\text{vec}(\Lambda)^\top \Sigma_{ols} \text{vec}(\Lambda) = \lambda^\top \Sigma_N \lambda$. Use Cramér-Wold theorem again, we can get the theorem result.

**Theorem A.1.** (CLT for $\hat{A}_t$)

$$
\sqrt{t} \text{vec}(\hat{A}_t - A) \overset{d}{\rightarrow} \mathcal{N}(0, \Sigma_{\hat{A}})
$$

(A.3)

where $\Sigma_{\hat{A}} = \sum_{k,k' \in K} \text{vec}(U_k)^\top \Sigma_{ols} \text{vec}(U_{k'}) \left[ \text{vec}(U_k)\text{vec}(U_{k'}) \right]^\top$. 

41
Proof: The proof is similar as before. Because, we can express any entries of $\mathbf{A}_t$ as a linear function of $\mathbf{A}_t$:

$$\mathbf{A}_t = \sum_{k \in K} \langle U_k, \mathbf{A}_t \rangle U_k.$$  \hfill (A.4)

Thus, for all $\Lambda' \in \mathbb{R}^{NF \times NF}$, we have

$$\langle \Lambda', \sqrt{t} (\mathbf{A}_t - A) \rangle = \sum_{k \in K} \langle \Lambda', U_k \rangle U_k, \sqrt{t} (\mathbf{A}_t - A) \rangle.$$  

Let $\Lambda$ in Equation (A.1) be $\sum_{k \in K} \langle \Lambda', U_k \rangle U_k$, then

$$\langle \Lambda', \sqrt{t} (\mathbf{A}_t - A) \rangle \overset{d}{\rightarrow} \mathcal{N}(0, \text{vec}(\Lambda')^\top \Sigma \text{vec}(\Lambda')).$$

Use Cramér-Wold theorem again, we can get the theorem result. The distribution in this theorem is degenerate. ■

**Proof of Corollary 3.3.1.** The proof is an adaption of Lütkepohl [17, Section 3.6]. We first construct the following matrix:

$$C = \begin{pmatrix} \vdots \\ \text{svec}(E_{h_k})^\top \end{pmatrix} \in \mathbb{R}^{P \times \frac{N(N-1)}{2}}.$$  \hfill (A.5)

Then test $H_0$ versus $H_1$ equals to

$$H_0': C \text{svec}(A_N) = 0 \text{ versus } H_1': C \text{svec}(A_N) \neq 0.$$  

Following CLT 3.3, we have

$$\sqrt{t} C \text{svec}(\mathbf{A}_{N,t} - A_N) \overset{d}{\rightarrow} \mathcal{N}(0, C \Sigma N C^\top).$$

Hence, when $H_0'$ holds,

$$\sqrt{t} C \text{svec}(\mathbf{A}_{N,t}) \overset{d}{\rightarrow} \mathcal{N}(0, C \Sigma N C^\top).$$

Then by Proposition C.2 (4) in [17], we have

$$\sqrt{t} \left[ C \hat{\Sigma}_{N,t} C^\top \right]^{-\frac{1}{2}} C \text{svec}(\mathbf{A}_{N,t}) \overset{d}{\rightarrow} \mathcal{N}(0, I_P),$$

where $\hat{\Sigma}_{N,t} = \sum_{k,k' \in K_N} \text{vec}(U_k)^\top \hat{\Sigma}_{ols,t} \text{vec}(U_{k'}) (\text{vec}(E_k) \text{vec}(E_{k'})^\top)$ is the consistent estimator of $\Sigma_N$. Then by continuous mapping theorem:

$$t \hat{\alpha}_t^\top \left[ C \hat{\Sigma}_{N,t} C^\top \right]^{-1} \hat{\alpha}_t \overset{d}{\rightarrow} \chi^2(P).$$

Note that $C \text{svec}(\mathbf{A}_{N,t}) = \hat{\alpha}_t$, and $(\text{vec}(E_k))_{k \in K_N}$ are orthonormal basis in $\mathbb{R}^{\frac{N(N-1)}{2}}$, thus we have $C \hat{\Sigma}_{N,t} C^\top = \hat{\Sigma}_{W,t}$. ■
B Proof of Proposition 4.1

From Definition (4.5), we have

\[
\hat{\Gamma}_t(0) = \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left( \frac{\sum_{\tau \in I_{m,t}} x_{\tau-1} x_{\tau-1}^\top}{p_{m,t}} - \frac{x_{m-1,t} x_{m-1,t}^\top}{p_{m,t}} \right) \\
= \frac{\sum_{\tau=1}^t x_{\tau-1} x_{\tau-1}^\top}{t} - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left( \frac{x_{m-1,t} x_{m-1,t}^\top}{p_{m,t}} \right).
\]

Plug \( x_t = b_t^0 + x_t' \) in the last equation above, we can get the formula only with respect with \( x_t' \)

\[
\hat{\Gamma}_t(0) = \hat{\Gamma}_t(0)' - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left( \frac{x_{m-1,t} x_{m-1,t}^\top}{p_{m,t}} \right),
\]

where \( x_{m-1,t} = \sum_{\tau \in I_{m,t}} \frac{x_{\tau-1}}{p_{m,t}}, \) \( m = 0, ..., M - 1, \) and \( \hat{\Gamma}_t(0)' := \frac{\sum_{\tau=1}^t x_{\tau-1} x_{\tau-1}^\top}{t} \). Note that \( x_{-1,t}' = x_{M-1,t}' \).

Similarly, denote \( \frac{\sum_{\tau=1}^t x_{\tau} x_{\tau-1}^\top}{t} \) by \( \hat{\Gamma}_t(1)' \), we have

\[
\hat{\Gamma}_t(1) = \hat{\Gamma}_t(1)' - \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} \left( \frac{x_{m,t} x_{m-1,t}^\top}{p_{m,t}} \right),
\]

with \( x_{m,t} = \sum_{\tau \in I_{m,t}} \frac{x_{\tau}}{p_{m,t}}, \) \( m = 0, ..., M - 1. \) Since \( (x_t')_t \) is the causal solution of VAR (2.5), we have

\begin{align*}
(a') \; & \hat{\Gamma}_t(0)' \overset{p}{\to} \Gamma(0), \hat{\Gamma}_t(1)' \overset{p}{\to} \Gamma(1), \\
(b') \; & \hat{\Gamma}_t(1)' \left[ \hat{\Gamma}_t(0)' \right]^{-1} \overset{p}{\to} \Gamma(1) \left[ \Gamma(0) \right]^{-1} = A, \\
(c') \; & \sqrt{t} \; vec \left( \hat{\Gamma}_t(1)' \left[ \hat{\Gamma}_t(0)' \right]^{-1} - A \right) \overset{d}{\to} \mathcal{N}(0, \Sigma). \end{align*}

Thus, to reach the results in Proposition 4.1, we need additionally the asymptotic properties of sample mean \( \bar{x}_{m,t}' \), which are given in Lemma B.1.

**Lemma B.1.** (CLT of \( \bar{x}_{m,t}' \))

\[
\sqrt{p_{m,t}} \bar{x}_{m,t}' \overset{d}{\to} \mathcal{N}(0, \Phi \Sigma_M \Phi^\top), \quad \forall m = 0, ..., M - 1,
\]

43
where \( \Phi = (I_N - A^M)^{-1} \), and \( \Sigma_M = \sum_{h=0}^{M-1} A^h \Sigma (A^h)^\top \). Therefore, \( \tilde{x}'_{m,t} \xrightarrow{p} 0 \).

**Proof of Lemma B.1.** Because of the periodicity, \((x'_t)_{t \in I_m, \infty}\) is also a stationary process from VAR: \( \tilde{X}_t = A^M \tilde{X}_{t-1} + \tilde{z}_t' \), with \( \tilde{z}_t' \sim \text{IID}(0, \Sigma_M) \), for all \( m = 0, \ldots, M - 1 \). Thus, apply Proposition 3.3 in Lütkepohl [17], we get the result.

**Proof of Proposition 4.1.**

(a) When \( t \to \infty \), \( \hat{\Gamma}_t(0) = \hat{\Gamma}_t(0)' - \sum_{m=0}^{M-1} \frac{1}{M} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \xrightarrow{p} \Gamma(0) - 0 = \Gamma(0) \), and \( \hat{\Gamma}_t(1) = \hat{\Gamma}_t(1)' - \sum_{m=0}^{M-1} \frac{1}{M} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \xrightarrow{p} \Gamma(1) \), with \( \tilde{x}'_{-1,t} := \tilde{x}'_{M-1,t} \).

(b) \( \bar{x}_{m,t} = \frac{\sum_{m=0}^{M-1} b_m^0 + x'_t}{pm_t} = b_m^0 + \bar{x}_{m,t} \xrightarrow{p} b_m^0, \forall m = 0, \ldots, M - 1 \). Since asymptotically, \( \bar{x}_{m,t} = \bar{x}_{m,t} \), thus both means can be used to estimate \( b_m^0 \). On the other hand, based on (a), using continuous mapping theorem on the matrix inverse, we have \( \bar{A}_t = \hat{\Gamma}_t(1) \left[ \hat{\Gamma}_t(0) \right]^{-1} \xrightarrow{p} A \).

(c) When \( t \to \infty \), \( \bar{A}_t \) equals

\[
\left[ \hat{\Gamma}_t(1)' - \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \right] \left[ \hat{\Gamma}_t(0)' - \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m-1,t} [\tilde{x}'_{m-1,t}]^\top \right) \right]^{-1}
\]

Use Woodbury formula on the matrix inverse, we have

\[
\sqrt{t}(\bar{A}_t - A) = \sqrt{t}(\hat{\Gamma}_t(1)' \left[ \hat{\Gamma}_t(0) \right]^{-1} - A) - \sqrt{t} \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \left[ \hat{\Gamma}_t(0) \right]^{-1}
\]

\[
+ \frac{\sqrt{t}}{1 - g} \hat{\Gamma}_t(1)' \left[ \hat{\Gamma}_t(0) \right]^{-1} \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m-1,t} [\tilde{x}'_{m-1,t}]^\top \right) \left[ \hat{\Gamma}_t(0) \right]^{-1}
\]

\[
- \frac{\sqrt{t}}{1 - g} \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \left[ \hat{\Gamma}_t(0) \right]^{-1} \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m-1,t} [\tilde{x}'_{m-1,t}]^\top \right) \left[ \hat{\Gamma}_t(0) \right]^{-1}
\]

where, \( g = \text{tr}(\sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m-1,t} [\tilde{x}'_{m-1,t}]^\top \right) \left[ \hat{\Gamma}_t(0) \right]^{-1}) \). Based on the result of (c’), to reach the same asymptotic distribution, we only need to show that, the reminder terms, namely from the second term to the last term above, all converge to 0 in probability.

From Slutsky’s theorem and Lemma B.1, we have the asymptotic result:

\[
\forall m, \frac{pm_t}{\sqrt{t}} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) = \frac{1}{\sqrt{M}} \left( \sqrt{pm_t} \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \xrightarrow{p} 0.
\]

Thus, \( \sqrt{t} \sum_{m=0}^{M-1} \frac{pm_t}{t} \left( \tilde{x}'_{m,t} [\tilde{x}'_{m-1,t}]^\top \right) \left[ \hat{\Gamma}_t(0) \right]^{-1} \xrightarrow{p} 0 \).

44
Similarly, \( \sqrt{t} \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} (\mathbf{x}'_{m-1,t},[\mathbf{x}'_{m-1,t}])^{-1}_t 0 \). Since, it is obvious that \( \sum_{m=0}^{M-1} \frac{p_{m,t}}{t} (\mathbf{x}'_{m-1,t},[\mathbf{x}'_{m-1,t}])^{-1}_t \xrightarrow{P} 0 \), then use the properties of convergence in probability and continuous mapping theorem, we can show the reminder terms all converge to 0 in probability.

\[ \blacksquare \]
Algorithm 2

Input: $x_{t+1}, x_t, \hat{\Gamma}_t(0), \hat{\Gamma}_t(1), [\hat{\Gamma}_t(0)]^{-1}, t.$

# Update:
$\hat{\Gamma}_{t+1}(1) = \frac{t+1}{t} \hat{\Gamma}_t(1) + \frac{1}{t+1} x_{t+1} x_t^\top, \quad \hat{\Gamma}_{t+1}(0) = \frac{t}{t+1} \hat{\Gamma}_t(0) + \frac{1}{t+1} x_{t+1} x_t^\top,$
$[\hat{\Gamma}_{t+1}(0)]^{-1} = \frac{t+1}{t} [\hat{\Gamma}_t(0)]^{-1} - \frac{t+1}{t} [\hat{\Gamma}_t(0)]^{-1} x_{t+1} x_t^\top [\hat{\Gamma}_t(0)]^{-1},$
$\hat{\Sigma}_{t+1} = \hat{\Gamma}_{t+1}(0) - \hat{\Gamma}_{t+1}(1) \hat{\Gamma}_{t+1}(0)^{-1} \hat{\Gamma}_{t+1}(1)^\top,$
$\hat{\Lambda}_{t+1} = \hat{\Gamma}_{t+1}(1)[\hat{\Gamma}_{t+1}(0)]^{-1}.$

# Projection:
$\hat{\Lambda}_{t+1} = \text{Proj}_G(\hat{\Lambda}_{t+1}),$ retrieve $\hat{\Lambda}_{D,t+1}, \hat{\Lambda}_{F,t+1}, \hat{\Lambda}_{N,t+1}$ using Equation (3.10).

Sort such that: $|\{(\hat{\Lambda}_{N,t+1})_{i_1,j_1} \leq \ldots \leq |(\hat{\Lambda}_{N,t+1})_{i_{|K_N|},j_{|K_N|}}|. $

# Bisection Wald test procedure:
Initialize $p_l = 1, p_r = |K_N|, p_m = \text{Floor}(\frac{p_l + p_r}{2}).$

Construct the corresponding test statistic $\hat{\lambda}_{W,t+1}$ or $\hat{\lambda}_{F,t+1}$ using Equation (3.13).

Perform tests $H(1)$ and $H(|K_N|)$ based on Corollary 3.3.1.

if $H(1), H(|K_N|)$ are not rejected then $\hat{\Lambda}_{N,t+1} = 0,$
else
  if $H(1), H(|K_N|)$ are both rejected then
    No changes are made to $\hat{\Lambda}_{N,t+1},$
  else
    while $p_l + 1 < p_r$ do
      $p_m \leftarrow \text{Floor}(\frac{p_l + p_r}{2})$, perform $H(p_m)$.
      if $H(p_m)$ is not rejected then
        $p_l \leftarrow p_m,$
      else
        $p_r \leftarrow p_m.$
      end if
    end while
    Let $(\hat{\Lambda}_{N,t+1})_{i_1,j_1} = \ldots = (\hat{\Lambda}_{N,t+1})_{i_{p_l},j_{p_l}} = 0.$
  end if
end if
$\hat{\Lambda}_{t+1} = \hat{\Lambda}_{D,t+1} + \hat{\Lambda}_{F,t+1} \otimes \hat{\Lambda}_{N,t+1}.$
$t \leftarrow t + 1.$

Output: $\hat{\Lambda}_{t+1}, \hat{\Gamma}_{t+1}(0), \hat{\Gamma}_{t+1}(1), [\hat{\Gamma}_{t+1}(0)]^{-1}, t.$

C  Bisection Wald Test for the Identification of Sparsity Structure of $\hat{\Lambda}_N$
Algorithm 3

Extended Algorithm 2 for the Augmented Model

Let

\[
\Gamma_{ij} = \begin{pmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ij}^T & \Sigma_{jj} \end{pmatrix} = \begin{pmatrix} \hat{\Gamma}_t(0)^{-1} & \hat{\Gamma}_t(0)^{-1} \\ \hat{\Gamma}_t(0)^{-1} \end{pmatrix}_{I_k,I_k'},
\]

where

\[
\hat{\Gamma}_t(0)^{-1} = \left( \hat{\Gamma}_t(0) \right)^{-1} = \left( \hat{\Gamma}_t(0) \right)^{-1}
\]

with order indices

\[
I_k = \{ i_k, i_k + F, \ldots, i_k + (N-1)F \}, \quad I_k' = \{ i_k', i_k' + F, \ldots, i_k' + (N-1)F \},
\]

D Extended Algorithm 2 for the Augmented Model

Algorithm 3

Input: \( \mathbf{x}_{t+1}, \mathbf{x}_t, \hat{\Gamma}_t(0), \hat{\Gamma}_t(1), \hat{\Gamma}_t(0)^{-1}, \hat{\Gamma}_t(1)^{-1}, t, \{ \mathbf{p}_m,t \}_{m=0}^{M-1}, \{ \mathbf{x}_m,t \}_{m=0}^{M-1} \).

Update \( \hat{\Gamma}_{t+1}(0), \hat{\Gamma}_{t+1}(1) \) from Equation (4.6).

\[
\hat{\Gamma}_{t+1}(0)^{-1} = \frac{t+1}{t} \hat{\Gamma}_t(0)^{-1} - \frac{t+1}{t} \left( \hat{\Gamma}_t(0)^{-1} \mathbf{x}_t - \mathbf{x}_{t-1} \mathbf{x}_t^T \hat{\Gamma}_t(0)^{-1} \right)
\]

\[
\hat{\Sigma}_{t+1} = \hat{\Gamma}_{t+1}(0) \hat{\Gamma}_t(1) (\hat{\Gamma}_{t+1}(0)^{-1} \hat{\Gamma}_t(0))^{-1}.
\]

Step Projection to Bisection Wald test procedure are identical to Algorithm 2.

Let \( \hat{\mathbf{A}}_{t+1} = \hat{\mathbf{A}}_{D,t+1} + \hat{\mathbf{A}}_{F,t+1} \otimes \hat{\mathbf{A}}_{N,t+1} \).

Update: \( \mathbf{x}_{m-1,t+1} = \frac{1}{p_{m,t+1}} \mathbf{x}_t, \mathbf{x}_{m,t+1} = \mathbf{x}_{m,t}, \forall m \neq \bar{m} - 1 \),

\[
\mathbf{p}_{m,t+1} = \mathbf{p}_{m,t} + 1, \quad \mathbf{p}_{m,t+1} = \mathbf{p}_{m,t}, \forall m \neq \bar{m},
\]

\[
t \leftarrow t + 1.
\]

Output: \( \hat{\mathbf{A}}_{t+1}, \hat{\Gamma}_{t+1}(0), \hat{\Gamma}_{t+1}(1), \hat{\Gamma}_{t+1}(0)^{-1}, t, \{ \mathbf{p}_m,t \}_{m=0}^{M-1}, \{ \mathbf{x}_m,t \}_{m=0}^{M-1} \).
E  Proximal Gradient Descent for Lasso (3.15)

The implementation of proximal gradient descent for Lasso (3.15) is given as follows.

\[ A^{k+1} = \text{prox}(A^k - \eta^k \nabla f(A^k)), \]

\[ = \arg \min_{A \in \mathcal{G}} \frac{1}{2\eta^k} \left\| A - \left( A^k - \eta^k \nabla f(A^k) \right) \right\|_{\ell^2}^2 + \lambda \ell_1 F \| A_N \|_{\ell_1} \]

\[ = \arg \min_{A \in \mathcal{G}} \frac{1}{2\eta^k} \left\| A - \text{Proj}_G \left( A^k - \eta^k \nabla f(A^k) \right) \right\|_{\ell^2}^2 + \lambda \ell_1 F \| A_N \|_{\ell_1} \]  

\[ \iff \begin{cases} 
A_N^{k+1} = \min_{A_N} \| A_N - \text{Proj}_{\mathcal{G}_N} \left( A^k - \eta^k \nabla f(A^k) \right) \|_{\ell^2}^2 + 2\eta^k \lambda \ell_1 \| A_N \|_{\ell_1}, \\
A_F^{k+1} = \text{Proj}_{\mathcal{G}_F} \left( A^k - \eta^k \nabla f(A^k) \right), \\
\text{diag}(A^{k+1}) = \text{Proj}_{\mathcal{D}} \left( A^k - \eta^k \nabla f(A^k) \right),
\end{cases} \tag{E.1} \]

where \( \nabla f(A^k) = A^k \hat{\Gamma}_t(0) - \hat{\Gamma}_t(1) \), we denote \( A^{k+1}(t, \lambda_t) \) by \( A^{k+1} \) to avoid the heavy notation.

The forward step requires to calculate the gradient only in \( \mathbb{R}^{NF \times NF} \), then the backward step amounts to a classical Lasso after projecting the gradient onto \( \mathcal{G} \). Thus the structure constraint and the partial sparsity do not pose additional difficulties.
F  Homotopy Algorithm for Regularization Path $A(t, \lambda_1)$ to $A(t, \lambda_2)$

Algorithm 4

**Input:** $N, F, \Gamma_0, \gamma_1, K_1^0$ (ordered list), $w_N^1, \lambda_1, \lambda_2, [\Gamma_0^1]^{-1}$, where $K_1^0, w_N^1, [\Gamma_0^1]^{-1}$ are associated with $A(t, \lambda_1)$, and $w_N^1 = [w]_{K_1^0}$.

**Initialization:** $\lambda \leftarrow \lambda_1, K_N^0 \leftarrow K_N \setminus K_1^1, K_1^1 \leftarrow K_D + K_F + K_N^1$, where + is the ordered append of two lists.

#Computing the regularization path (the steps in parentheses are the modifications for the case $\lambda_1 > \lambda_2$):

**while** $\lambda < \lambda_2$ (or $\lambda > \lambda_2$) **do**

  **Generate** $\Gamma_0^0, \gamma_1^0, \gamma_0^1, w_1$, based on Proposition 3.4.
  **Compute** $\lambda_r$ (or $\lambda_l$), based on Equations (3.29) and (3.30).
  **if** $\lambda_r < \lambda_2$ (or $\lambda_l > \lambda_2$) **then**

    $\lambda = \lambda_r$ (or $\lambda = \lambda_l$),

    #Update the active set and the sign vector:

    **if** $[a_k^1]$, becomes zero for some $k_i \in K_1^1$ and $k_i \in K_1^1$, namely, $\lambda$ comes from $\{\lambda_0^k\}_k$ **then**

    $K_N^0 \leftarrow K_N^0 \setminus \{k\}, K_1^1 \leftarrow K_1^1 \setminus \{k\}, K_N^0 \leftarrow K_N^0 + \{k\}$.

    Remove $[w_N^1]_{i-\vert K_D\vert -\vert K_F\vert}$ from $w_N^1$.

    Remove the $i$-th row together with the $i$-th column from $\Gamma_0^1$, and use Sherman Morrison formula to update $[\Gamma_0^1]^{-1}$.

  **else if** $[w_0]$ reaches 1 for some $k_i \in K_N^0$, namely, $\lambda$ comes from $\{\lambda^+_k\}_k$ **then**

    $K_N^0 \leftarrow K_N^0 \setminus \{k\}, K_N^1 \leftarrow K_N^1 + \{k\}, K_1^1 \leftarrow K_1^1 + \{k\}$.

    Append 1 to the end of sign vector $w_1$.

    Append row $[\Gamma_0]_{k,K^1}$, column $[\Gamma_0]_{k,K^1}$ after the last row and last column $\Gamma_0^1$, respectively, and use Sherman Morrison formula to update $[\Gamma_0^1]^{-1}$.

  **else if** $[w_1]_k^0$ reaches $-1$ for some $k_i \in K_N^0$, namely, $\lambda$ comes from $\{\lambda^-_k\}_k$ **then**

    $K_N^0 \leftarrow K_N^0 \setminus \{k\}, K_N^1 \leftarrow K_N^1 + \{k\}, K_1^1 \leftarrow K_1^1 + \{k\}$.

    Append $-1$ to the end of sign vector $w_1$.

    Append row $[\Gamma_0]_{k,K^1}$, column $[\Gamma_0]_{k,K^1}$ after the last row and last column $\Gamma_0^1$, respectively, and use Sherman Morrison formula to update $[\Gamma_0^1]^{-1}$.

  **end if**

**else**

  $\lambda = \lambda_2$.

**end if**

**end while**

Compute $a_1^0$, using Equation (3.28) and the last updated $[\Gamma_0^1]^{-1}, \gamma_1^1, w_1$. Retrieve $A(t, \lambda_2)$ from this $a_1^0$.

**Output:** $A(t, \lambda_2), K_N^1, w_N^1, [\Gamma_0^1]^{-1}$.
G  Homotopy Algorithm for Data Path $A(t, \frac{t+1}{t}\lambda)$ to $A(t+1, \lambda)$

Algorithm 5

1: **Input:** $N, F, \Gamma_0, \gamma_1, K_N^1$ (ordered list), $w_N^k, \lambda, [\Gamma_0^1]^{-1}$, $x_{t+1}, \bar{X}_t, t$, where $K_N^1, w_N^1, [\Gamma_0^1]^{-1}$ are associated with $A(t, \frac{t+1}{t}\lambda)$, and $w_N^i = [w]_{K_N^1}$.

2: **Initialization:** $\lambda \leftarrow \lambda_1$, $K_N^0 \leftarrow K_N^1 \backslash K_N^0$, $K^1 \leftarrow K_D + K_F + K_N^1$, where + is the ordered append of two lists.

3: for $i = 1, \ldots, NF$ do
4: $\mu \leftarrow 0.$
5: while $\mu < 1$ do
6: Generate $\Gamma_0^0, \gamma_0, \gamma_1^0, w_1$, based on Proposition 3.4.
7: $a_i^s = [\Gamma_0^1]^{-1} \left( (\frac{t}{t+1}) \lambda w_1 \right)$.
8: $e = x_{t+1,i} - ((\bar{X}_t[i], K^1_i, i) \top a_i^s) \; \; \; u = [\Gamma_0^1]^{-1} (\bar{X}_t[K^1,i])^\top u.$
9: $\mu_k^0 = -\frac{T(a_i^0)/\alpha (a_i^0) + e(u_i)}{\alpha (a_i^0)}$, $k_i \in K^1$ such that $k_i \in K_N^1$.
10: $\mu_\pm = \frac{-\mu_+}{\min(e(u_i), e(\bar{X}_t[k_i], a_i^0) + \alpha \gamma_1)}$, $k_i \in K_N^0$, $\mu_+ = [\gamma_0^1 a_i^0 - \gamma_1^0 \pm (1 + \frac{1}{t}) \lambda]$.
11: $\mu' = \min \{ \min \{ \mu_0^k, k \in K_N^1 : \mu_0^k > \mu \}, \mu_+^0, k \in K_N^0 : \mu_+^0 > \mu \}, \min \{ \mu_0^k, k \in K_N^0 : \mu_0^k > \mu \}, \min \{ \mu_0^k, k \in K_N^0 : \mu_0^k > \mu \}$.
12: if $\mu' = \emptyset$, $\mu' \leftarrow +\infty$.
13: if $\mu' < 1$ then
14: $\mu = \mu'$.
15: if $\mu'$ is some $\mu_0^k$ then
16: $K_N^0 \leftarrow K_N^0 \backslash \{ k \}$, $K_N^0 \leftarrow K_N^0 \backslash \{ k \}$, $K_N^0 \leftarrow K_N^0 \backslash \{ k \}$.
17: Remove $[w_N^1][i \in |K_D| - |K_F|]$ from $w_N^1$.
18: Remove the $i$-th row, the $i$-th column from $\Gamma_0^1$, use Sherman Morrison formula to update $[\Gamma_0^1]^{-1}$.
19: else if $\mu'$ is some $\mu_+^k$ (or $\mu_-^k$) then
20: $K_N^0 \leftarrow K_N^0 \backslash \{ k \}$, $K_N^0 \leftarrow K_N^0 \backslash \{ k \}$, $K_N^0 \leftarrow K_N^0 \backslash \{ k \}$.
21: Append 1 (or $-1$) to the end of sign vector $w_N^1$.
22: Append row $[\Gamma_0^1]_{k,K^1}$, column $[\Gamma_0^1]_{K^1,k}$ after the last row and last column $\Gamma_0^1$, respectively, and use Sherman Morrison formula to update $[\Gamma_0^1]^{-1}$.
23: end if
24: else
25: $\mu \leftarrow 1.$
26: end if
27: end while
28: $[\Gamma_0^1]^{-1} \leftarrow \frac{1}{t} \left[ \frac{1}{t} [\Gamma_0^1 + \frac{1}{t} (\bar{X}_t[i], [\bar{X}_t[K^1,i]]^\top)^{-1} \right.$
29: $\Gamma_0 \leftarrow \Gamma_0 + \frac{1}{t} [\bar{X}_t[i], [\bar{X}_t[i], [\bar{X}_t[K^1,i]]^\top, \gamma_1 \leftarrow \gamma_1 + \frac{1}{t} x_{t+1,i} [\bar{X}_t[i, i]$
30: end for
31: $a_i^s = a_i^s + \epsilon u_i / (t + \alpha)$. Retrieve $A(t + 1, \lambda)$ based on $K^1$ and $a_i^s$.
32: $[\Gamma_0^1]^{-1} \leftarrow \frac{t+1}{t} [\Gamma_0^1]^{-1}, \Gamma_0 \leftarrow \frac{t}{t+1} \Gamma_0, \gamma_1 \leftarrow \frac{t}{t+1} \gamma_1.$
33: **Output:** $A(t + 1, \lambda), K_N^0, w_N^1, [\Gamma_0^1]^{-1}$, $\Gamma_0, \gamma_1.$
Algorithm 6

**Input:** $A(t, \lambda_t), \Gamma_0, \gamma_1, K_N$ (ordered list), $w_N^1, \lambda_t, [\Gamma_0^1]^{-1}, x_{t+1}, \hat{X}_t, \bar{m}, t, M, (p_m,t)_{m=0}^{M-1}$, $(x_{m,t})_{m=0}^{M-1}, b_{m,t}$, where $K_N^1, w_N^1, [\Gamma_0^1]^{-1}$ are associated with $A(t, \lambda_t)$.

Select $\lambda_{t+1}$ according to the end of Section 4.2.

Update $A(t, \lambda_t) \rightarrow A(t, \frac{t+1}{t}\lambda_{t+1})$ using algorithm 4.

Center $x_{t+1} \leftarrow x_t + x_{m,t}$. Compose $\hat{X}_{m-1,t}$ as $[\hat{X}_{m-1,t}]_{k,i} = [U_k]_t x_{m-1,t}$, and center $\hat{X}_t \leftarrow \hat{X}_t - \hat{X}_{m-1,t}$.

Update $A(t, \frac{t+1}{t}\lambda_{t+1}) \rightarrow A(t + 1, \lambda_{t+1})$ using algorithm 5, with modifications:

- Line 8 change to $\alpha = [\hat{X}_t]_{K^1,i} u + p_m,t$,
- Line 28, 29 change respectively to:
  
  $[\Gamma_0^1]^{-1}$ rank 1 update $[\Gamma_0^1 + \frac{p_m,t}{l(p_m,t+1)}[\hat{X}_t]_{K^1,i}[\hat{X}_t]_{K^1,i}^T]^{-1}$
  
- $\Gamma_0 \leftarrow \Gamma_0 + \frac{p_m,t}{l(p_m,t+1)}[\hat{X}_t]_{K^1,i}[\hat{X}_t]_{K^1,i}^T$
  
- $\gamma_1 \leftarrow \gamma_1 + \frac{p_m,t}{l(p_m,t+1)}x_{t+1,i}[\hat{X}_t]_{K^1,i}$

Update $x_{m-1,t+1} \leftarrow \frac{p_m,t+1}{p_m,t+1}x_{m-1,t} + \frac{1}{p_m,t+1}x_t$, and $x_{m,t+1} \leftarrow x_{m,t}$, $\forall m \neq \bar{m} - 1$.

$p_{\bar{m}, t+1} \leftarrow p_m, t + 1$, and $p_{m,t+1} \leftarrow p_m, t$, $\forall m \neq \bar{m}$,

$\bar{m} \leftarrow (t + 2) \mod M$.

$b_{\bar{m}', t+1} \leftarrow x_{\bar{m}', t+1} - A(t + 1, \lambda_{t+1})x_{\bar{m}', t+1}$,

$t \leftarrow t + 1$.

**Output:** $A(t + 1, \lambda_{t+1}), \Gamma_0, \gamma_1, K_N^1, w_N^1, \lambda_{t+1}, [\Gamma_0^1]^{-1}, t, (p_m,t+1)_{m=0}^{M-1}$, $(x_{m,t+1})_{m=0}^{M-1}$, $b_{\bar{m}', t+1}$.