Learning Quasi-Kronecker Product Graphical Models

Mattia Zorzi

Abstract—We consider the problem of learning graphical models where the support of the concentration matrix can be decomposed as a Kronecker product. We propose a method that uses the Bayesian hierarchical learning modeling approach. Thanks to the particular structure of the graph, we use a the number of hyperparameters which is small compared to the number of nodes in the graphical model. In this way, we avoid overfitting in the estimation of the hyperparameters. Finally, we test the effectiveness of the proposed method by a numerical example.

I. INTRODUCTION

Many modern applications are characterized by high-dimensional data sets from which it is important to discover the meaningful interactions among the variables rather than to find an accurate model. An powerful tool to analyze these interrelations is given by graphical models (i.e. Markov networks), [1], [2], [3]. The simplest version of the latter is constituted by a zero mean Gaussian random vector to which we attach an undirected graph: each node corresponds to a component of the random vector and there is an edge between two nodes if and only if the corresponding components are conditionally dependent given the others.

In these applications, there is a large interest of learning sparse graphical models (i.e. graphs with few edges) from data; indeed, these models are characterized by few conditional interdependence relations among the components. Interestingly, a sparse graph corresponds to a covariance matrix whose inverse, say concentration matrix, is sparse. The problem of learning sparse graphical models, sometime called covariance selection problem, can be faced by using regularization techniques, [2], [3], [4], [5]. For instance, [3] proposed a regularized maximum-likelihood (ML) estimator for the covariance matrix where the $\ell_1$ penalty norm on the concentration matrix has been considered. Since the $\ell_1$ norm penalty induces sparsity, the estimated covariance matrix will have a sparse inverse. It is worth noting that these approaches can be extended to dynamic graphical models, [6], [7] as well as factor models [8], [9], [10], [11].

These regularized estimators are known to be sensitive to the choice of the regularization parameter, i.e. the weight on $\ell_1$ penalty, which is typically selected by cross-validation or theoretical derivation. To overcome this issue, a Bayesian hierarchical modeling approach has been considered, [8]. Here, the concentration matrix is modeled as a random matrix whose prior is characterized by a regularization parameter (called hyperparameter). Then, the hyperparameter as well as the covariance matrix are jointly estimated. Since the $\ell_1$ norm shrinks all the entries to zero, and thus introduces a bias, a further improvement is to consider a weighted $\ell_1$ norm, see [9], where the hyperparameter is a matrix whose dimension (in principle) coincides with the number of the nodes in the graph. On the other hand, the introduction of an hyperparameter with many variables could lead to overfitting in the estimation of the hyperparameter matrix.

An important class of graphical models is represented by the so called Kronecker Product (KP) graphical models, [10], [11], [12], [13], [14] wherein it is required that the concentration matrix can be decomposed as a Kronecker product. KP graphical models find application in many fields: spatiotemporal MEG/EEG modeling [15]; recommendation systems like Netflix and gene expression analysis, [16]; face recognition analysis[17]. In these applications the most important feature is the graphical structure, i.e. the fact that the support of the concentration matrix can be decomposed as a Kronecker product.

The contribution of the present paper is to address the problem of learning graphical models where the support of the concentration matrix can be decomposed as a Kronecker product. We call such models Quasi-Kronecker Product (QKP) graphical models. Note that, the assumption that the support can be decomposed as a Kronecker product does not imply that the concentration matrix does. Therefore, QKP graphical models can understood as a weaker version of KP graphical models, making the former class less restrictive than the latter. Adopting the Bayesian hierarchical modeling approach, in the spirit of [9], we introduce two hyperparameter matrices whose total number of variables is small compared to the number of nodes in the graph. In this way, we avoid overfitting in the estimation of the hyperparameter.

The paper is outlined as follows. In Section II we introduce graphical models and the problem of graphical model selection. In Section III we introduce QKP graphical models. In Section IV we propose a Bayesian procedure to learn QKP graphical models from data, while Section V is devoted on how to initialize the procedure. In Section VI we present a numerical example to show the effectiveness of the proposed method. Finally, Section VII draws the conclusions.

We warn the reader that the present paper only reports some preliminary result regarding the Bayesian estimation of QKP graphical models. In particular, all the proofs and most of the technical assumptions needed therein are omitted and will be published afterwards.

Notation: Given a symmetric matrix $S$, we write $S \succ 0$ ($S \succeq 0$) if $S$ is positive (semi-) definite. $x \sim \mathcal{N}(\mu, \Sigma)$ means $x$ is a Gaussian random vector with mean $\mu$ and covariance $\Sigma$. The paper is outlined as follows. In Section II we introduce graphical models and the problem of graphical model selection. In Section III we introduce QKP graphical models. In Section IV we propose a Bayesian procedure to learn QKP graphical models from data, while Section V is devoted on how to initialize the procedure. In Section VI we present a numerical example to show the effectiveness of the proposed method. Finally, Section VII draws the conclusions.

We warn the reader that the present paper only reports some preliminary result regarding the Bayesian estimation of QKP graphical models. In particular, all the proofs and most of the technical assumptions needed therein are omitted and will be published afterwards.

Notation: Given a symmetric matrix $S$, we write $S \succ 0$ ($S \succeq 0$) if $S$ is positive (semi-) definite. $x \sim \mathcal{N}(\mu, \Sigma)$ means $x$ is a Gaussian random vector with mean $\mu$ and covariance $\Sigma$. The paper is outlined as follows. In Section II we introduce graphical models and the problem of graphical model selection. In Section III we introduce QKP graphical models. In Section IV we propose a Bayesian procedure to learn QKP graphical models from data, while Section V is devoted on how to initialize the procedure. In Section VI we present a numerical example to show the effectiveness of the proposed method. Finally, Section VII draws the conclusions.

We warn the reader that the present paper only reports some preliminary result regarding the Bayesian estimation of QKP graphical models. In particular, all the proofs and most of the technical assumptions needed therein are omitted and will be published afterwards.

Notation: Given a symmetric matrix $S$, we write $S \succ 0$ ($S \succeq 0$) if $S$ is positive (semi-) definite. $x \sim \mathcal{N}(\mu, \Sigma)$ means $x$ is a Gaussian random vector with mean $\mu$ and covariance $\Sigma$. The paper is outlined as follows. In Section II we introduce graphical models and the problem of graphical model selection. In Section III we introduce QKP graphical models. In Section IV we propose a Bayesian procedure to learn QKP graphical models from data, while Section V is devoted on how to initialize the procedure. In Section VI we present a numerical example to show the effectiveness of the proposed method. Finally, Section VII draws the conclusions.

We warn the reader that the present paper only reports some preliminary result regarding the Bayesian estimation of QKP graphical models. In particular, all the proofs and most of the technical assumptions needed therein are omitted and will be published afterwards.
matrix \( \Sigma \). \( \mathbb{E} [ \cdot ] \) denotes the expectation operator. Given two functions \( f(x) \) and \( g(x) \), \( f \propto g \) means that the argmin with respect to \( x \) of \( f \) and \( g \) coincide. Given a matrix \( S \) of dimension \( m \times m \), \( s_{jk} \) denotes its entry in position \((j, k)\). Given a matrix \( S \) of dimension \( m_1m_2 \times m_1m_2 \), \( s_{jk,kl} \) denotes its entry in position \(((j - 1)m_2 + i, (k - 1)m_2 + l)\). Given a matrix \( S \), \( \text{vec}(S) \) denotes the vectorization of matrix \( S \). Given a matrix \( S \) with positive entries, \( \log(S) \) denotes the matrix with entry \( \log(s_{jk}) \) in position \((j, k)\). Given a matrix \( S \), \( \exp(S) \) is the matrix with entry \( \exp(s_{jk}) \) in position \((j, k)\) and \( \text{abs}(S) \) denotes the matrix with entry \( |s_{jk}| \) in position \((j, k)\). \( 1_m \) denotes the \( m \)-dimensional vector of ones.

II. GRAPHICAL MODEL SELECTION

Let \( x = [x_1 \ldots x_m]^T \) be a zero mean Gaussian random vector taking values in \( \mathbb{R}^m \) and with covariance matrix \( \Sigma \succ 0 \). Thus, this random vector is completely characterized by \( \Sigma \). We can attach to \( x \) an undirected graph \( \mathcal{G}(V, \Omega) \) where \( V \) denotes the set of its nodes, and \( \Omega \) denotes the set of its edges. More precisely, each nodes corresponds to a component \( x_j, j = 1 \ldots m \), of \( x \) and there is an edge between nodes \( j, k \) if and only \( x_j \) and \( x_k \) are conditionally dependent given the other components, or equivalently, for any \( j \neq k \):

\[
x_j \perp x_k \mid x_l, l \neq j, k \iff (j, k) \notin \Omega.
\]

Thus, \( \Omega \) describes the conditionally dependent pairs of \( x \). The graph \( \mathcal{G} \) is referred to as graphical model of \( x \); an example is provided in Figure 1. Dempster proved that conditional independence relations are given by the concentration matrix of \( x \), i.e. \( S := \Sigma^{-1} \), [18]:

\[
x_j \perp x_k \mid x_l, l \neq j, k \iff s_{jk} = 0.
\]

Accordingly, sparsity of \( S \), i.e. \( S \) with many entries equal to zero, reflects the fact that the graphical model \( \mathcal{G} \) of \( x \) is sparse, i.e. \( \mathcal{G} \) has few edges.

In many applications, it is required to learn a sparse graphical model \( \mathcal{G} \) from data. More precisely, given a sequence of data \( x^N := \{x_1 \ldots x_N\} \) generated by \( x \), find a sparse graphical model \( \mathcal{G}(V, E) \) for \( x \) where \( \mathcal{G} \) is sparse. The simplest idea is to compute the sample covariance from the data

\[
\hat{\Sigma} = \frac{1}{N} \sum_{k=1}^{N} x_kx_k^T;
\]

then the graphical model is given by the support of \( \hat{\Sigma}^{-1} \). However, the resulting graph is full even in the case that the underlying system is well described by a sparse graphical model. In [8], a procedure based on a Bayesian hierarchical model has been proposed. More precisely, the entries of \( S \) are assumed to be i.i.d. and Laplace distributed with hyperparameter \( \gamma \geq 0 \), i.e. the probability density function (pdf) of \( s_{jk} \) is \( p(s_{jk}) = \gamma / 2 \exp (-\gamma |s_{jk}|) \). The resulting procedure is described in Algorithm 1. The main drawback of this approach is that it assigns a priori the same level of sparsity to each entry of \( S \). This method has been extended to the case wherein only the entries in the same column of \( S \) have the same distribution, [9], allowing different levels of sparsity in the prior. A further extension is to assume that all the entries of \( S \) may be distributed in a different way, but respecting the symmetry, that is \( p(s_{jk}) = p(s_{kj}) = \gamma_{jk}/2 \exp (-\gamma_{jk}|s_{jk}|) \) with \( \gamma_{jk} \geq 0 \). Using arguments similar to the ones in [9], it is not difficult to find that the procedure described in Algorithm 2.

Algorithm 1 Algorithm for sparse graphical models in [8]

1: Set \( \varepsilon > 0 \), \( \gamma^{(0)} > 0 \), \( \epsilon_{\text{STOP}} > 0 \)
2: \( h \leftarrow 1 \)
3: repeat
4: Solve
\[
\hat{S}_{(h)} = \arg \min_{S \succ 0} \frac{N}{2} \log |S| + \frac{N}{2} \text{tr}(S\hat{\Sigma}) + \gamma^{(h-1)} \sum_{j,k=1}^{m} |s_{jk}|
\]
5: Update hyperparameter
\[
\gamma^{(h)} = \frac{m^2}{\sum_{j,k=1}^{m} |s_{jk}^{(h)}| + \varepsilon}
\]
6: \( h \leftarrow h + 1 \)
7: until \( \|\hat{S}^{(h)} - \hat{S}^{(h-1)}\| \leq \epsilon_{\text{STOP}} \)

III. QKP GRAPHICAL MODELS

Consider the undirected graph \( \mathcal{G}(V, \Omega) \) and let \( m \) denote the number of its nodes. Let \( E_{\Omega} \) be the \( m \times m \) binary matrix defined as follows:

\[
(E_{\Omega}) = \begin{cases} 
1, & \text{if } (j, k) \in \Omega \\
0, & \text{otherwise}
\end{cases}.
\]

We say that \( \mathcal{G}(V, \Omega) \) is a Kronecker Product graph if there exist two graphs \( \mathcal{G}_1(V_1, \Omega_1) \) and \( \mathcal{G}_1(V_2, \Omega_2) \) with \( m_1 \) and \( m_2 \) nodes, respectively, such that:

\[
E_{\Omega} = E_{\Omega_1} \otimes E_{\Omega_2}
\]

where \( m = m_1m_2 \). In shorthand notation we will write \( \mathcal{G} = \mathcal{G}_1 \otimes \mathcal{G}_2 \). In practice, in this graph \( \mathcal{G} \) we can recognize
depends on the hyperparameters

symmetric random matrix with nonnegative entries and

model
dimension.

Thus, in our case we require that

the interaction among the variables involved in the applica-

tion. Thus, in our case we require that

the observed data are explained by a sparse graphical model

in the latter the support of the concentration matrix and

(QKP) graphical model to distinguish from the Kronecker

graphical model is referred to as Quasi-Kronecker Product

Ω

modules, while

matrix (i.e. concentration matrix)

Ω

structure, described by

Ω

1

2

1

2

modules containing m2 nodes sharing the same graphical

structure, described by Ω2: the interaction among those m1

modules is described by Ω1. An illustrative example is given

in Figure 2.

Let x = [x1 . . . xm]T be a zero mean Gaussian random

vector taking values in Rm and with inverse covariance matrix
(i.e. concentration matrix) S > 0 whose support is

E1 = E11 ⊕ E12. We can attach to x a Kronecker Product

graph G(V, Ω) = G1(V1, Ω1) ⊗ G2(V2, Ω2) Accordingly, Ω1

characterizes the conditional dependence relations among

the modules, while Ω2 characterizes the recurrent conditional

dependence relations among the nodes in each module. This

 graphical model is referred to as Quasi-Kronecker Product
(QKP) graphical model to distinguish from the Kronecker

Product (KP) graphical model proposed in [13]. Indeed,

in the latter the support of the concentration matrix and

its support admit a Kronecker decomposition, i.e. E11 =

E11 ⊕ E12 and S = S1 ⊗ S2. In our graphical model, even if

the support of the concentration matrix admits a Kronecker

product decomposition, the concentration matrix does not.

IV. LEARNING QKP GRAPHICAL MODELS

We address the problem of learning a QKP graphical model

G = G1 ⊗ G2 from data. In many real applications

the observed data are explained by a sparse graphical model

because the latter allows a straightforward interpretation of

the interaction among the variables involved in the applica-

tion. Thus, in our case we require that G1 and G2 are sparse.

The problem can be formalized as follows.

Problem 1: Let x be a zero mean random vector of

dimension m = m1m2, m1, m2 ∈ N, with zero mean

and inverse covariance matrix S. Given a sequence of data

xN := {x1T . . . xN T} generated by x, find a QKP graphical

model G(V, E) = G1(V1) ⊗ G2(V2, E2) for x where G1 and

G2 are sparse and have m1, m2 nodes, respectively.

To solve Problem 1 we adopt the Bayesian hierarchical

modeling. S is modeled as a random matrix whose prior

depends on the hyperparameters Λ and Γ. Λ is a m1 × m1

symmetric random matrix with nonnegative entries and Γ

is a m2 × m2 symmetric random matrix with nonnegative

entries. The hyperprior for Λ and Γ depends on ε1 and ε2,

respectively. The latter are deterministic positive quantities.

We proceed to characterize the Bayesian model in detail.
The conditional pdf of xN under model x ∼ N(0, S−1) is:

p(xN | S) = \frac{1}{\sqrt{(2\pi)^m |S−1|}} \exp\left(-\frac{1}{2} x^T S^{-1} x\right)

\propto |S|^{N/2} \exp\left(-\frac{1}{2} \text{tr}(S\hat{\Sigma})\right)

(6)

where the neglected terms do not depend on S. In what

follows we assume that \hat{\Sigma} ≻ 0. We model the entries of

S as independent random variables, so that the prior for S is:

p(S | Λ, Γ) = \prod_{j,k=1}^{m1} \prod_{i,l=1}^{m2} p(s_{jk,il} | λ_{jk}, γ_{il})

(7)

where s_{jk,il} is Laplace distributed

p(s_{jk,il} | λ_{jk}, γ_{il}) = \frac{λ_{jk}γ_{il}}{2} \exp\left(-λ_{jk}γ_{il} | s_{jk,il}\right).

(8)

Λ and Γ are independent random matrices, i.e. p(Λ, Γ) =
p(Λ)p(Γ). The entries of Λ and Γ are assumed to be

independent

p(Λ) = \prod_{j,k=1}^{m1} p(λ_{jk}), p(Γ) = \prod_{i,l=1}^{m2} p(γ_{il})

(9)

and with exponential distribution

p(λ_{jk}) = ε1 \exp(−ε1 λ_{jk}), p(γ_{il}) = ε2 \exp(−ε2 γ_{il})

(10)

with ε1, ε2 deterministic and positive quantities. At this

point, some comment regarding the choice of the prior on S

and the hyperprior on Λ, Γ is required. From (6) it is clear

that s_{jk,il} takes value close to zero with high probability if

the product λ_{jk}γ_{il} is large. Moreover, if γ_{il} is very large

for some (i, l) and λ_{jk} ≥ 0, for all j, k = 1 . . . m1, such

that γ_{il}e is large then s_{jk,il} with j, k = 1 . . . m1 take

values close to zero with high probability. Accordingly,

different modules in the graph will have a similar sparsity

pattern with high probability. Accordingly, prior (9) assigns

high probability to QKP graphical models. The hyperprior

in (10) guarantees that λ_{jk} and γ_{il} diverge with probability

zero. As we will see, this assumption guarantees that the

optimization procedure that we propose is well-posed.

Next, we characterize the maximum a posteriori (MAP)

estimator of S (and thus also the MAP estimator of the

covariance matrix by the invariance principle). The latter

minimizes the negative log-likelihood

ℓ(xN ; S, Λ, Γ) = − \log p(xN, S, Λ, Γ)

(11)

where p(xN, S, Λ, Γ) is the joint pdf of xN, S, Λ and Γ.

Note that,

p(xN, S, Λ, Γ) = p(xN | S)p(S | Λ, Γ)p(Λ)p(Γ),

(12)
At the hard problem, we propose an iterative three-step procedure. Since the joint optimization of the three variables is still an open problem, we propose an iterative three-step procedure. At the $h$-th iteration we solve the following three optimization problems:

\[
\hat{S}^{(h)} = \underset{\hat{S}, \hat{\Lambda}, \hat{\Gamma}}{\text{argmin}} \ell(x^N; S, \hat{\Lambda}^{(h-1)}, \hat{\Gamma}^{(h-1)}) \quad (15)
\]

\[
\hat{\Lambda}^{(h)} = \underset{\Lambda}{\text{argmin}} \ell(x^N; \hat{S}^{(h)}, \Lambda, \hat{\Gamma}^{(h-1)}) \quad \text{s.t. } \lambda_{jk} \geq 0 \quad (16)
\]

\[
\hat{\Gamma}^{(h)} = \underset{\Gamma}{\text{argmin}} \ell(x^N; \hat{S}^{(h)}, \hat{\Lambda}^{(h)}, \Gamma) \quad \text{s.t. } \gamma_{il} \geq 0 \quad (17)
\]

It is possible to prove that Problems (15), (16) and (17) do admit a unique solution. Moreover, the resulting procedure is illustrated in Algorithm 3. In the aforementioned algorithm the hyperparameters selection is performed iteratively through Step 5. It is worth noting that Algorithm 3 is similar to Algorithm 1 and Algorithm 2: the main difference is that in the proposed algorithm we have two types of hyperparameters. As a consequence, in the proposed algorithm we have three optimization steps instead of two.

**Algorithm 3 Proposed algorithm**

1: Set $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, $\hat{\Lambda}^{(0)}$ and $\hat{\Gamma}^{(0)}$ positive entrywise, $\varepsilon_{STOP} > 0$
2: $h \leftarrow 1$
3: repeat
4: Solve
\[
\hat{S}^{(h)} = \underset{\hat{S}}{\text{argmin}} \frac{N}{2} \log |S| + \frac{N}{2} \text{tr}(S\hat{\Sigma}) + \sum_{j,k=1}^{m_1} \sum_{i,l=1}^{m_2} \lambda_{jk} \gamma_{il} |s_{jk,il}| \quad \text{s.t. } S > 0, \lambda_{jk} \geq 0, \gamma_{il} \geq 0
\]
5: Update hyperparameters
\[
\hat{\lambda}_{jk}^{(h)} = \frac{m_2^2}{\sum_{i,l=1}^{m_2} \hat{\lambda}_{jk}^{(h-1)} |s_{jk,il}| + \varepsilon_1} \quad j, k = 1 \ldots m_1
\]
\[
\hat{\gamma}_{il}^{(h)} = \frac{m_1^2}{\sum_{j,k=1}^{m_1} \hat{\lambda}_{jk}^{(h)} |s_{jk,il}| + \varepsilon_2} \quad i, l = 1 \ldots m_2
\]
6: $h \leftarrow h + 1$
7: until $\|\hat{S}^{(h)} - \hat{S}^{(h-1)}\| < \varepsilon_{STOP}$

Where the neglected terms do not depend on $S$, $\Lambda$ and $\Gamma$. It is clear that the MAP estimator of $S$ depends on $\Lambda$, $\Gamma$, $\varepsilon_1$ and $\varepsilon_2$, in what follows we assume $\varepsilon_1$ and $\varepsilon_2$ fixed. Then, a way to estimate $\Lambda$ and $\Gamma$ from the data is provided by the empirical Bayes approach: $\Lambda$ and $\Gamma$ are given by maximizing the marginal likelihood of $x^N$ which is obtained by integrating out $S$ in [12], [19]. However, it is not easy to find an analytical expression for the marginal likelihood in this case. An alternative simplified approach for optimizing $\Lambda$, $\Gamma$ is the generalized maximum likelihood (GML) method, [20]. According to this method, $S$, $\Lambda$ and $\Gamma$ minimize jointly (13):

\[
(\hat{S}, \hat{\Lambda}, \hat{\Gamma}) = \underset{S, \Lambda, \Gamma}{\text{argmin}} \ell(x^N; S, \Lambda, \Gamma) \quad \text{s.t. } S > 0, \lambda_{jk} \geq 0, \gamma_{il} \geq 0 \quad (14)
\]

Fig. 2: Panel (a). Example of a Kronecker Product graph: we have three modules (yellow, blue, green): each module contains four nodes. Black edges describe conditional dependence relations among the nodes of the same module. Red edges describe conditional dependence relations among the modules. Panel (b). The corresponding matrix $E_\Omega$. Panel (c). Graphs $G_1(V_1, \Omega_1)$ and $G_2(V_2, \Omega_2)$ generating the Kronecker Product graph $G_1 \otimes G_2$ in Panel (a).
V. INITIAL CONDITIONS

In Algorithm 3 we have to fix the initial conditions for the hyperparameters, that is $\tilde{\Lambda}^{(0)}$ and $\tilde{\Gamma}^{(0)}$. The idea is to approximate $\Sigma^{-1}$ through a Kronecker product, then the two matrices of this product are used to initialize $\Lambda$ and $\Gamma$.

Given $\Sigma$, we want to find $W$ and $\hat{Y}$ of dimension $m_1$ and $m_2$, respectively, with positive entries such that $W \otimes Y \approx \text{abs}(\hat{\Sigma}^{-1}) + \epsilon 1_{m_1,m_2} 1_{m_1,m_2}^T$ where $\epsilon > 0$ is chosen sufficiently small. The presence of the term $\epsilon 1_{m_1,m_2} 1_{m_1,m_2}^T$ allows to take the entrywise logarithm on both sides, obtaining

$$W \otimes I_{m_2} + I_{m_1} \otimes Y \approx \text{log}(\text{abs}(\hat{\Sigma}^{-1}) + \epsilon 1_{m_1,m_2} 1_{m_1,m_2}^T)$$

where $W = \log W$ and $Y = \log \hat{Y}$. It is not difficult to see that the following relations hold:

$$\text{vec}(W \otimes I_{m_2}) = (I_{m_1} \otimes 1_{m_2} \otimes I_{m_1} \otimes 1_{m_2}) \text{vec}(W)$$
$$\text{vec}(I_{m_1} \otimes Y) = (I_{m_2} \otimes I_{m_1} \otimes I_{m_1} \otimes I_{m_2}) \text{vec}(Y).$$

Then, we can write (18) as $Az \approx b$ where

$$A = \begin{bmatrix} I_{m_1} \otimes 1_{m_2} \otimes I_{m_1} \otimes 1_{m_2} & I_{m_1} \otimes I_{m_2} \otimes I_{m_1} \otimes I_{m_2} \\ \text{vec}(W)^T & \text{vec}(Y)^T \end{bmatrix}$$
$$z = \begin{bmatrix} \text{vec}(W) \\ \text{vec}(Y) \end{bmatrix}$$
$$b = \text{vec}(\text{log}(\text{abs}(\hat{\Sigma}^{-1}) + \epsilon 1_{m_1,m_2} 1_{m_1,m_2}^T)).$$

Accordingly, $z$ can be found by solving the least squares problem $\hat{z} = \arg\min_{z} \| Az - b \|_2$, therefore $\hat{z} = (A^T A)^{-1} A^T b$. From $z$ we recover $W$ and $Y$. Note that, $W$ and $Y$ computed from $z$ are not symmetric matrices. Thus, to compute $\tilde{W}$ and $\tilde{Y}$ from $W$ and $Y$ we force the symmetric structure: $\tilde{W} = (\exp(W) + \exp(W)^T)/2$ and $\tilde{Y} = (\exp(Y) + \exp(Y)^T)/2$. At this point, it is worth noting that $w_{jk}y_{il}$ provides roughly the order of magnitude of $s_{jk,il}$. On the other hand, the hyperparameters $\lambda_{jk}$ and $\gamma_{il}$ provides the prior order the order the order of magnitude of $s_{jk,il}$, the larger $\lambda_{jk}\gamma_{il}$ is, the more $s_{jk,il}$ is close to zero. Accordingly, we choose $\tilde{\Lambda}^{(0)}$ and $\tilde{\Gamma}^{(0)}$ such that:

$$\hat{\lambda}_{jk}^{(0)} = \frac{1}{w_{jk}}, \quad j,k = 1 \ldots m_1$$
$$\hat{\gamma}_{il}^{(0)} = \frac{1}{y_{il}}, \quad i,l = 1 \ldots m_2.$$

VI. SIMULATION RESULTS

We consider a Monte Carlo experiment structured as follows:

- We generate 60 QKP graphical models with $m_1 = 6$ modules and each module contains $m_2 = 10$ nodes. For each model, $\Omega_1$ and $\Omega_2$ are generated randomly. The fraction of edges is set equal to 20% for both $\Omega_1$ and $\Omega_2$.
- For each model we generate a finite-length realization $x_N := \{x_1 \ldots x_N\}$, with $N = 1000$, and we compute the sample covariance $\hat{\Sigma}$.
- For each realization we consider the following estimators:
  - S1 estimator: it computes a sparse graphical model by using Algorithm 1 in this case we have one scalar hyperparameter;
  - S2 estimator: it computes a sparse graphical model by using Algorithm 2 in this case we have $m_1m_2(m_1m_2 + 1)/2 = 1830$ variables in the hyperparameter matrix;
  - QKP estimator: it computes a QKP graphical model with $m_1 = 6$ modules and each module has $m_2 = 10$ nodes, in this case the total number of variables in the hyperparameters matrices is $m_1(m_1 + 1)/2 + m_1(m_2 + 1)/2 = 76$.
- For each realization, we compute the relative error in reconstructing the concentration matrix and the relative error in reconstructing the sparsity pattern using S1, S2 and QKP. For instance, the relative error in reconstructing concentration matrix using QKP estimator is

$$e_{QKP} = \frac{\|S_{true} - \hat{S}_{QKP}\|}{\|S_{true}\|}$$

where $S_{true}$ denotes the concentration matrix of the true model, while $\hat{S}_{QKP}$ is the estimated concentration matrix; here, $\| \cdot \|$ denotes the Frobenius norm. The relative error in reconstructing the sparsity pattern using QKP estimator is

$$e_{SP,QKP} = \frac{\|E_{true} - E_{\Omega_{QKP}}\|}{(m_1m_2(m_1m_2 + 1))/2}$$

where $\Omega_{set}$ and $\Omega_{QKP}$ denote the set of edges of the true graphical model and the one estimated, respectively.

Figure 4 depicts the set of edges estimated using the three estimators in a realization of the Monte Carlo experiment. As we see, only QK provides a structure similar to the true model. Figure 4 shows the boxplot of the relative error in reconstructing the sparsity pattern (left panel) and the concentration matrix (right panel). As we can see, the worst performance is given by S1, while the best performance is achieved by QKP. In particular, the relative error of the estimated sparsity pattern for QKP is very small compared to the other two methods. The poor performance of S1 is due by the fact that only a scalar hyperparameter is not sufficient to capture the correct structure of the graph. On the contrary, the poor performance of S2 is due by the fact that there is overfitting in the estimation of the hyperparameter matrix.

VII. CONCLUSIONS

We have introduced Quasi-Kronecker Product graphical models wherein the nodes are regrouped in modules having the same number of nodes. The interactions among the nodes of the same module as well as the interaction among the nodes of two modules follow a common structure. Then, we have addressed the problem of learning QKP graph models from data using a Bayesian hierarchical model. Finally, we have compared the proposed procedure with Bayesian learning techniques for estimating sparse graphical models: simulation evidence showed the effectiveness of the proposed method.
Fig. 3: Set of edges estimated in a realization of the Monte Carlo experiment. First picture. $E_{\hat{\Omega}}$ where $\Omega$ is the set of edges of the true model. Second picture. $E_{\hat{\Omega}_{S1}}$ where $\Omega_{S1}$ is the set of edges estimated using S1. Third picture. $E_{\hat{\Omega}_{S2}}$ where $\Omega_{S2}$ is the set of edges estimated using S2. Fourth picture. $E_{\hat{\Omega}_{QKP}}$ where $\Omega_{QKP}$ is the set of edges estimated using QKP.

Fig. 4: Monte Carlo experiment with realizations of length $N = 1000$. Left panel. Boxplot of the relative error in reconstructing the sparsity pattern. Right panel. Boxplot of the relative error in reconstructing the concentration matrix.

REFERENCES

[1] S. Lauritzen, Graphical Models. Oxford: Oxford University Press, 1996.
[2] J. Z. Huang, N. Liu, M. Pourahmadi, and L. Liu, “Covariance matrix selection and estimation via penalised normal likelihood,” Biometrika, vol. 93, no. 1, pp. 83–98, 2006.
[3] O. Banerjee, L. E. Ghaoui, and A. d’Aspremont, “Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data,” Journal of Machine learning research, vol. 9, pp. 485–516, 2008.
[4] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” Biostatistics, vol. 9, no. 3, pp. 432–441, 2008.
[5] A. d’Aspremont, O. Banerjee, and L. El Ghaoui, “First-order methods for sparse covariance selection,” SIAM Journal on Matrix Analysis and Applications, vol. 30, no. 1, pp. 56–66, 2008.
[6] J. Songsiri and L. Vandenberghe, “Topology selection in graphical models of autoregressive processes,” J. Mach. Learning Res., vol. 11, pp. 2671–2705, 2010.
[7] M. Zorzi and R. Sepulchre, “AR identification of latent-variable graphical models,” IEEE Transactions on Automatic Control, vol. 61, pp. 2327–2340, Sep. 2016.
[8] N. B. Asadi, I. Rish, K. Scheinberg, D. Kanovsky, and B. Ramabhadran, “Map approach to learning sparse gaussian markov networks,” in Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pp. 1721–1724, 2009.
[9] K. Scheinberg, I. Rish, and N. B. Asadi, “Sparse markov net learning with priors on regularization parameters.” in ISAIM, 2010.
[10] T. Tsiligkaridis and A. O. Hero, “Covariance estimation in high dimensions via kronecker product expansions,” IEEE Transactions on Signal Processing, vol. 61, no. 21, pp. 5347–5360, 2013.
[11] K. Werner, M. Jansson, and P. Stoica, “On estimation of covariance matrices with kronecker product structure,” IEEE Transactions on Signal Processing, vol. 56, pp. 478–491, Feb. 2008.
[12] P. Dutilleul, “The mile algorithm for the matrix normal distribution,” Journal of statistical computation and simulation, vol. 64, no. 2, pp. 105–123, 1999.
[13] T. Tsiligkaridis, A. O. Hero III, and S. Zhou, “On convergence of kronecker graphical lasso algorithms,” IEEE transactions on signal processing, vol. 61, no. 7, pp. 1743–1755, 2013.
[14] B. Sinapius and M. Verhaegen, “Kronecker-ax models in identifying (2d) spatial-temporal systems,” 20th IFAC World Congress, vol. 50, no. 1, pp. 14131 – 14136, 2017.
[15] F. Bijn, J. C. De Munck, and R. M. Heethaar, “The spatiotemporal meg covariance matrix modeled as a sum of kronecker products,” NeuroImage, vol. 27, no. 2, pp. 402–415, 2005.
[16] G. I. Allen and R. Tibshirani, “Transposable regularized covariance models with an application to missing data imputation,” The Annals of Applied Statistics, vol. 4, no. 2, p. 764, 2010.
[17] Y. Zhang and J. G. Schneider, “Learning multiple tasks with a sparse matrix-normal penalty,” in Advances in Neural Information Processing Systems, pp. 2550–2558, 2010.
[18] Y. Zhang and J. G. Schneider, “Learning multiple tasks with a sparse matrix-normal penalty,” in Advances in Neural Information Processing Systems, pp. 2550–2558, 2010.
[19] C. E. Rasmussen, “Gaussian processes in machine learning,” in Advanced lectures on machine learning, pp. 63–71, Springer, 2004.
[20] Z. Zhou, R. Leahy, and J. Qi, “Approximate maximum likelihood hyperparameter estimation for gibbs priors,” IEEE transactions on image processing, vol. 6, no. 6, pp. 844–861, 1997.