VECTOR ℓ_O SPARSE CONDITIONAL INDEPENDENCE GRAPHS

Goran Marjanovic & Victor Solo

School of Electrical Engineering and Telecommunications
University of New South Wales
Sydney, AUSTRALIA
email: v.solo@unsw.edu.au

ABSTRACT

One of the main approaches to system identification of networks of time series or signals is conditional independence graphical (CIG) modeling. In the Gaussian case, the conditional dependence structure of the nodal time series is determined by the location of zeros in the precision matrix (inverse covariance matrix). And this determines the graph structure of the network. Despite the many applications of CIG models, the theory and algorithms have so far only dealt with networks of univariate or scalar signals. But in most applications the nodes carry multivariate or vector signals. Here we extend CIG modeling to handle such data by posing a group l_0 sparse penalised block precision matrix estimation problem. We develop a double cyclic descent algorithm to solve it. And we compare the method with a group l_1 penalised alternative in simulations.

Index Terms— sparsity, l_0, precision matrix

1. INTRODUCTION

Three kinds of graphical modeling of network time series data are currently in play.

Firstly a group of methods which originated in the sociology literature and includes the explorarory techniques of graph analysis involving characteristics such as node degree, centrality, small world networks etc but also the formal confirmatory methods of exponential random graph models (ERGMs) and stochastic block models (SBMs): see [1] and references therein.

Secondly, conditional independence graphical (CIG) models, developed in the statistics literature [2, 3]. In the Gaussian case they are characterised by zeros in the precision or inverse covariance matrix. And these zeros determine the graph structure of the network. This has led to a growing literature on sparsity penalised precision matrix estimation e.g. [5],[10],[12],[23].

Thirdly, graph signal processing which so far emphasizes the extension of Fourer techniques to graphs of univariate signals [4].

In this paper we are concerned with CIGs. There is certainly a well developed CIG literature and many successful applications and CIG models continue to attract considerable attention. However, with a few exceptions, the development so far has been only for univariate CIGs (uvCIGs) i.e. networks of univariate or scalar nodal signals. But in practice most networks carry vector or multivariate signals. We purse then the development of multivariate CIG (mvCIG) modeling.

So far there seem to be only three pieces of work tackling the mvCIG problem. The first seems to be [17], which develops a group l_1 penalised likelihood method. The group sparsity penalty is necessary to handle the multivariate nature of the nodal time series data. This work was followed (and referenced by) [18], which developed a specialised tree-structured penalty approach. Independently of these two works is that of [19], which however does not use sparsity.

In this work we pursue a group l_o penalised approach. The group or vector l_0 penalty delivers much greater sparsity than the l_1 penalty at the cost of non-convexity.

The remainder of the paper is organised as follows. In section 2 we formulate the penalised problem. In section 3 we develop the algorithm. In section 4 we provide comparative simulations. Section 5 contains conclusions.

2. PRELIMINARIES

The mvCIG problem arises as follows. Consider a network with p nodes, with node i having a d_i dimensional time series $x_{i,t}, t = 1, \cdots, T$. Stacking these nodal vector time series together at time t gives a network vector time series x_t of dimension $d = \sum_{i=1}^{p} d_i$. We assume $x_t \sim \mathcal{N}(\boldsymbol{\mu}_{d\times 1}, \boldsymbol{\Sigma}_{d\times d})$, and $\boldsymbol{\mu} = \mathbf{0}$ without loss of generality. $\boldsymbol{\Sigma}$ is thus a block covariance matrix with blocks of dimensions $d_i \times d_j$.

The precision matrix is $\Omega_{d\times d} = \boldsymbol{\Sigma}^{-1}$ with corresponding blocks $\Omega_{ij} \in \mathbb{R}^{d_i \times d_j}$. In this case, if $\Omega_{ij} = \mathbf{0}_{d_i \times d_j}$, where $i \neq j$, then the vector white noise time series $x_{i,t}$ and $x_{j,t}$ are conditionally independent given the all other vector time series at all the other nodes. Hence there is no link between nodes i and j in the corresponding undirected graph. This crucial result, which determines the graph structure of the network, well known in the univariate case [16, 20, 3, 2], can be proved in the multivariate case in a similar way.

In order to obtain a block sparse precision matrix $\Omega_{d\times d}$ we posit the following group l_0 Penalized Log-Likelihood (l_0-PLL) criterion:

$$F(\Omega) = -\log\det(\Omega) + \mathrm{tr}(\mathbf{S}\Omega) + \lambda \sum_{i\neq j} I(\Omega_{ij} \neq \mathbf{0}) \qquad (2.1)$$

where $\mathbf{S}_{d\times d} = \frac{1}{T}\Sigma_1^T x_t x_t^T$ is the sample covariance matrix and $I(\cdot)$ is the indicator function, i.e.

$$I(\Omega_{ij} \neq 0) = \begin{cases} 0 & if \quad \Omega_{ij} = 0 \\ 1 & if \quad otherwise \end{cases}$$

and $\lambda > 0$ is a scalar (tuning) parameter. *It is crucial to understand that all entries in matrix Ω_{ij} will be zeroed together.*

The group l_0 penalty is a sparsity inducing norm and the log-likelihood promotes goodness-of-fit.

This work was partly supported by the Australian Research Council

2.1. l_0 vs. l_1 Sparsity Penalty

The penalty in (2.1) is known as the group l_0 penalty because it penalises groups (blocks) of the matrix variable and so causes all the entries in a block to be zeroed. It is non-convex so that (2.1) is non-convex. A convex approximation of the l_0 function is the l_1 norm. In this setting, we have the corresponding group l_1 penalty, defined as $\sum_{i\neq j}\|\Omega_{ij}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm [17, 18].

In the sparse univariate precision matrix literature, e.g., [12, 13, 14, 15], it was shown that the l_0 penalty results in a less biased estimator than the l_1 penalty. We will demonstrate the same behaviour in the multivariate case.

3. ALGORITHM DEVELOPMENT

The algorithm for solving either the group l_1-PLL or l_0-PLL problem proceeds in two stages. The first stage is a Block-wise cyclic descent (BCD) procedure, where one optimises over one block of the matrix Ω at a time. The BCD reduces the problem of optimising over a matrix in (2.1) to a problem of optimising over a vector. The second stage does the latter optimisation also using CD. Our algorithm differs from that in [17] precisely because we use a second stage of CD. We now describe the two stages in detail.

3.1. The Block-wise Cyclic Descent (BCD) Stage

Letting Ω denote the current iterate, the new blocks to be updated are identified and the rest of the matrix entries are fixed. To describe the updating procedure we have to deal with permutations of Ω and $\mathbf{S}$, which have the identified target block Ω_a placed at the end i.e.

$$\Omega_\pi = \begin{bmatrix} \Omega_o & \mathbf{B}_a \\ \mathbf{B}_a^T & \Omega_a \end{bmatrix}, \mathbf{S}_\pi = \begin{bmatrix} \mathbf{S}_o & \mathbf{S}_{oa} \\ \mathbf{S}_{oa}^T & \mathbf{S}_a \end{bmatrix} \tag{3.1}$$

where $\mathbf{B}_a$ and $\mathbf{S}_{oa}$ are column matrices. Matrix Ω_a has dimension $d_a \times d_a$, and $\mathbf{B}_a$ and has dimension $d_{-a} \times d_a$, where $d_{-a} = d - d_a$. In particular

$$\mathbf{B}_a = \begin{pmatrix} \mathbf{B}_{1a} \\ \mathbf{B}_{2a} \\ \vdots \\ \mathbf{B}_{(p-1)a} \end{pmatrix}$$

where $\mathbf{B}_{ia}$ has dimensions $d_i \times d_a$ for $1 \leq i \leq p-1$.

By partitioned determinants we get

$$\det(\Omega_\pi) = \det(\Omega_o)\det(\Omega_a - \mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a)$$

and also

$$\text{tr}(\mathbf{S}\Omega_\pi) = \text{tr}(\mathbf{S}_o\Omega_o) + 2\,\text{tr}(\mathbf{S}_{oa}^T\mathbf{B}_a) + \text{tr}(\mathbf{S}_a\Omega_a)$$

Since $F|_{\Omega,\mathbf{S}} = F|_{\Omega_\pi,\mathbf{S}_\pi}$, we have

$$\begin{aligned} F(\Omega) &= -\log\det(\Omega_o) - \log\det(\Omega_a - \mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a) \\ &+ \text{tr}(\mathbf{S}_o\Omega_o) + 2\,\text{tr}(\mathbf{S}_{oa}^T\mathbf{B}_a) + \text{tr}(\mathbf{S}_a\Omega_a) \\ &+ \lambda\sum_{i\neq j} I(\Omega_{o,ij} \neq \mathbf{0}) + 2\lambda\sum_i I(\mathbf{B}_{ia} \neq \mathbf{0}) \end{aligned}$$

where $\Omega_{o,ij}$ is the ij-th block in sub-matrix Ω_o. Therefore, to update the a-th block we fix Ω_o and need to optimise

$$\begin{aligned} F_a &= -\log\det(\Omega_a - \mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a) + 2\,\text{tr}(\mathbf{S}_{oa}^T\mathbf{B}_a) \\ &+ \text{tr}(\mathbf{S}_a\Omega_a) + 2\lambda\sum_i I(\mathbf{B}_{ia} \neq \mathbf{0}) \end{aligned} \tag{3.2}$$

We now compute the perturbation in F_a induced by a perturbation in Ω_a

$$\begin{aligned} \delta F_a &= -\text{tr}(\mathbf{\Delta}_a^{-1}\delta\Omega_a) + \text{tr}(\mathbf{S}_a\delta\Omega_a) \\ \mathbf{\Delta}_a &= \Omega_a - \mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a \end{aligned}$$

The perturbation in F_a must vanish for arbitrary $\delta\Omega_a$, leading to the Euler equation

$$\mathbf{\Delta}_a^{-1} = \mathbf{S}_a \;\Rightarrow\; \mathbf{\Delta}_a = \mathbf{S}_a^{-1}$$

which implies

$$\Omega_a = \mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a + \mathbf{S}_a^{-1} \tag{3.3}$$

If $\Omega \succ \mathbf{0}$, then we must have $\Omega_o \succ \mathbf{0}$, which in turn implies $\Omega_o^{-1} \succ \mathbf{0}$. Then, it can easily be shown that $\mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a \succeq \mathbf{0}$. Hence, due to the fact that $\mathbf{S} \succeq \mathbf{0}$ and $\mathbf{S}_a^{-1}$ exists, we must have $\mathbf{S}_a^{-1} \succ \mathbf{0}$. As a result, $\Omega_a \succ \mathbf{0}$.

Notice that in (3.3) we need to invert matrices Ω_o and $\mathbf{S}_a$. This can be done in low dimensions using Cholesky factorization. For large dimensions, iterative methods such as Conjugate Gradients (CG) are preferred, because the cost of each iteration is proportional to the number of non-zeros in the matrix. Thus the block sparse structure of Ω_o guarantees a smaller number of non-zeros.

To update Ω_a in (3.3), function F_a now needs to be minimized with respect to $\mathbf{B}_a$.

3.2. The Cyclic Descent (CD) Stage for l_0-PLL

Substituting (3.3) back in F_a in (3.2) gives (after dropping superfluous terms)

$$F_a = \text{tr}(\mathbf{S}_a\mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a) + 2\,\text{tr}(\mathbf{S}_{oa}^T\mathbf{B}_a) + 2\lambda\sum_i I(\mathbf{B}_{ia} \neq \mathbf{0})$$

To carry out the optimization we convert the above to vector form, via the two well known identities

$$\begin{aligned} \text{tr}(\mathbf{AB}^T) &= vec^T(\mathbf{A})vec(\mathbf{B}) \\ \text{tr}(\mathbf{ABCD}) &= vec^T(\mathbf{B})(\mathbf{C}\otimes\mathbf{A})vec(\mathbf{D}^T) \end{aligned}$$

where $vec(\mathbf{M})$ converts matrix $\mathbf{M}$ into a column vector, and $\otimes$ is the Kronecker product. So, introduce vectors

$$\boldsymbol{\beta}_a = vec(\mathbf{B}_a) \;\text{ and }\; \sigma_{oa} = vec(\mathbf{S}_{oa})$$

then

$$\begin{aligned} \text{tr}(\mathbf{B}_a\mathbf{S}_{oa}^T) &= \boldsymbol{\beta}_a^T\sigma_{oa} \\ \text{tr}(\mathbf{S}_a\mathbf{B}_a^T\Omega_o^{-1}\mathbf{B}_a) &= \text{tr}(\Omega_o^{-1}\mathbf{B}_a\mathbf{S}_a\mathbf{B}_a^T) \\ &= \boldsymbol{\beta}_a^T(\mathbf{S}_a\otimes\Omega_o^{-1})\boldsymbol{\beta}_a \end{aligned}$$

So, the above F_a becomes

$$F_a = \boldsymbol{\beta}_a^T\mathbf{M}_o^{-1}\boldsymbol{\beta}_a + 2\boldsymbol{\beta}_a^T\sigma_{oa} + 2\lambda\sum_i I(\boldsymbol{\beta}_{ia} \neq \mathbf{0}) \tag{3.4}$$

where $\mathbf{M}_o = \mathbf{S}_a^{-1}\otimes\Omega_o \succ \mathbf{0}$.

F_a in (3.4) can now be optimised with respect to $\boldsymbol{\beta}_a$ with the CD algorithm, where the components $\boldsymbol{\beta}_{ia}$ in $\boldsymbol{\beta}_a$ are updated for each $i = 1,\ldots,p-1,1,\ldots,p-1,\ldots$. So, to obtain the update for $\boldsymbol{\beta}_{ia}$ we partition $\boldsymbol{\beta}_a$ into two vector components, i.e. we let

$$\boldsymbol{\beta}_a = \begin{bmatrix} \boldsymbol{\beta}_{ia} \\ \boldsymbol{\beta}_{-ia} \end{bmatrix} \tag{3.5}$$

where vector $\boldsymbol{\beta}_{-ia}$ represents all components in $\boldsymbol{\beta}_a$ which do not belong to $\boldsymbol{\beta}_{ia}$. Then, we need to correspondingly partition $\mathbf{M}_o^{-1} = \mathbf{S}_a \otimes \Omega_o^{-1}$. Since $\mathbf{M}_o^{-1}$ is a Kronecker product of two positive definite matrices it is itself positive definite. Hence, we can consider the Cholesky decomposition of $\mathbf{M}_o^{-1}$, in which case $\mathbf{M}_o^{-1} = \mathbf{L}^T\mathbf{L}$, where $\mathbf{L}$ is lower triangular. We can now partition $\mathbf{L}$ according to (3.5) by letting $\mathbf{L}_{ia}$ denote all the columns in $\mathbf{L}$ that correspond to the entries in $\boldsymbol{\beta}_a$ that form $\boldsymbol{\beta}_{ia}$. In this case, we let $\mathbf{L}_{-ia}$ denote the rest of the columns in $\mathbf{L}$. Then, note that

$$\begin{aligned}\beta_a^T\mathbf{M}_o^{-1}\boldsymbol{\beta}_a &= [\boldsymbol{\beta}_{ia}^T, \boldsymbol{\beta}_{-ia}^T]\begin{bmatrix}\mathbf{L}_{ia}^T\\ \mathbf{L}_{-ia}^T\end{bmatrix}[\mathbf{L}_{ia}, \mathbf{L}_{-ia}]\begin{bmatrix}\boldsymbol{\beta}_{ia}\\ \boldsymbol{\beta}_{-ia}\end{bmatrix}\\ &= [\boldsymbol{\beta}_{ia}^T, \boldsymbol{\beta}_{-ia}^T]\begin{bmatrix}\mathbf{L}_{ia}^T\mathbf{L}_{ia} & \mathbf{L}_{ia}^T\mathbf{L}_{-ia}\\ \mathbf{L}_{-ia}^T\mathbf{L}_{ia} & \mathbf{L}_{-ia}^T\mathbf{L}_{-ia}\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_{ia}\\ \boldsymbol{\beta}_{-ia}\end{bmatrix}\end{aligned}$$

in which case (3.4) reduces to (after dropping terms that do not depend on $\boldsymbol{\beta}_{ia}$)

$$F_{ia} = \boldsymbol{\beta}_{ia}^T\mathbf{L}_{ia}^T\mathbf{L}_{ia}\boldsymbol{\beta}_{ia} + 2\mathbf{z}_{ia}^T\boldsymbol{\beta}_{ia} + 2\lambda I(\boldsymbol{\beta}_{ia} \neq \mathbf{0}) \tag{3.6}$$

where $\mathbf{z}_{ia} = \mathbf{L}_{ia}^T\mathbf{L}_{-ia}\boldsymbol{\beta}_{-ia} + \sigma_{oa,i}$ and $\sigma_{oa,i}$ is a sub-vector of σ_{oa} whose entries correspond to the entries in $\boldsymbol{\beta}_a$ that form $\boldsymbol{\beta}_{ia}$. The minimiser of F_{ia} is either $\mathbf{0}$, in which case, $F_{ia} = F_{ia}^0 = \mathbf{0}$, or is not $\mathbf{0}$ and hence is the minimiser of

$$\boldsymbol{\beta}_{ia}^T\mathbf{L}_{ia}^T\mathbf{L}_{ia}\boldsymbol{\beta}_{ia} + 2\mathbf{z}_{ia}^T\boldsymbol{\beta}_{ia} + 2\lambda \tag{3.7}$$

So, by differentiating (3.7) and equating to $\mathbf{0}$ gives $\mathbf{L}_{ia}^T\mathbf{L}_{ia}\boldsymbol{\beta}_{ia} + \mathbf{z}_{ia} = \mathbf{0}$. Solving for $\boldsymbol{\beta}_{ia}$ gives the minimiser

$$\boldsymbol{\beta}_{ia}^\star = -(\mathbf{L}_{ia}^T\mathbf{L}_{ia})^{-1}\mathbf{z}_{ia} \tag{3.8}$$

Substituting (3.8) back into (3.7) gives

$$F_{ia}^\star = -\mathbf{z}_{ia}^T(\mathbf{L}_{ia}^T\mathbf{L}_{ia})^{-1}\mathbf{z}_{ia} + 2\lambda$$

Therefore, the minimiser of F_{ia} in (3.6), denoted by $\boldsymbol{\beta}_{ia}^+$ is $\mathbf{0}$ if $F_{ia}^0 \leq F_{ia}^\star$ or $\boldsymbol{\beta}_{ia}^\star$ otherwise, i.e.

$$\boldsymbol{\beta}_{ia}^+ = \boldsymbol{\beta}_{ia}^\star I(\mathbf{z}_{ia}^T\left(\mathbf{L}_{ia}^T\mathbf{L}_{ia}\right)^{-1}\mathbf{z}_{ia} > 2\lambda\Big) \tag{3.9}$$

CD Algorithm for Minimising F_a in (3.4)

Input: Initial $\boldsymbol{\beta}_a$, $\lambda > 0$, and let $i = 1$.

01. Compute the optimal $\boldsymbol{\beta}_{ia}^+$ in (3.9).

02. Update $\boldsymbol{\beta}_a$ by updating $\boldsymbol{\beta}_{ia}$ with $\boldsymbol{\beta}_{ia}^+$.

03. If $i < p-1$, then $i = i+1$. Otherwise $i = 1$. Go to 01.

3.3. Algorithm Statement

The algorithm for minimising the group l_0 penalised log-likelihood criterion F in (2.1) is now given below.

Algorithm for Minimising $F(\Omega)$ in (2.1)

Input: Dimensions $d_1, \ldots, d_m$, initial $\Omega_{d\times d} \succ \mathbf{0}$, where $d = \sum_{i=1}^m d_i$, and $\lambda > 0$. Let $a = 1$.

01. Partition Ω as shown in (3.1), and identify the new target blocks in Ω, denoting them by

$$\Omega_a \in \mathbb{R}^{d_a\times d_a} \text{ and } \mathbf{B}_a \in \mathbb{R}^{d_{-a}\times d_a}$$

02. Using the current $\boldsymbol{\beta}_a = vec(\mathbf{B}_a)$ as the initialiser, apply the CD algorithm from Section 3.2 to minimise F_a in (3.4). Denote the minimiser by $\boldsymbol{\beta}_a^+$, and convert it to matrix form $\mathbf{B}_a^+ = mat(\boldsymbol{\beta}_a^+)$. Replace $\mathbf{B}_a$ in Ω with the update $\mathbf{B}_a^+$.

03. Replace Ω_a in Ω with the update

$$\Omega_a^+ = {\mathbf{B}_a^+}^T\Omega_o^{-1}\mathbf{B}_a^+ + \mathbf{S}_a^{-1}$$

04. If $a = m$, then let $a = 1$. Otherwise let $a = a+1$.

05. Go to 01.

Theorem 1 Let $\Omega_{d\times d} \succ \mathbf{0}$ denote the current iterate of the inverse covariance in the above algorithm, and let Ω_+ denote the new iterate with updated blocks Ω_a and $\mathbf{B}_a$. Then

(a) $\Omega_+ \succ \mathbf{0}$

(b) $F(\Omega_+) \leq F(\Omega)$

Proof. (a): since the update $\Omega_a^+ > \mathbf{S}_a^{-1}$ is positive definite, so is Ω_+. (b): follows since each cycle involves a minimization.

Remark 1 (Regarding the group l_1-PLL algorithm) The algorithm for minimising the group l_1-PLL criterion is very similar to the one above. The difference is in the CD procedure for updating $\boldsymbol{\beta}_a$. Namely, the update $\boldsymbol{\beta}_{ia}$ in the CD method is a minimiser of F_{ia} in (3.6), where the indicator function $I(\boldsymbol{\beta}_{ia} \neq \mathbf{0})$ is replaced with $\|\boldsymbol{\beta}_{ia}\|_2$. As a result, the CD update in (3.9) is replaced with

$$\boldsymbol{\beta}_{ia}^+ = \boldsymbol{\beta}_{ia}^\bullet I\left(\frac{-\mathbf{z}_{ia}^T\boldsymbol{\beta}_{ia}^\bullet}{\|\boldsymbol{\beta}_{ia}^\bullet\|_2} > \lambda\right)$$

where $\boldsymbol{\beta}_{ia}^\bullet \neq \mathbf{0}$ is a solution of the equation[1]

$$\mathbf{L}_{ia}^T\mathbf{L}_{ia}\mathbf{u} + \mathbf{z}_{ia} + \lambda\frac{\mathbf{u}}{\|\mathbf{u}\|_2} = \mathbf{0}$$

which can be solved, for example, using a fixed point method, i.e. $\mathbf{u}_{k+1} = -(\mathbf{L}_{ia}^T\mathbf{L}_{ia})^{-1}(\mathbf{z}_{ia} + \lambda\frac{\mathbf{u}_k}{\|\mathbf{u}_k\|_2})$, where $\mathbf{u}_0 = -(\mathbf{L}_{ia}^T\mathbf{L}_{ia})^{-1}\mathbf{z}_{ia}$.

4. SIMULATIONS

Here we use a simulation to compare the group l_0 and l_1 PLL inverse covariance estimators.

Five blocks (groups) of different size are considered i.e. $[d_1, d_2, d_3, d_4, d_5] = [6, 9, 15, 3, 9]$. As a result, $d = \sum_i d_i = 42$. The ground truth $\Omega_{d\times d} \succ \mathbf{0}$ is constructed as follows: for each d_i we generate a sparse matrix $\mathbf{U}_{d_i\times d_i}$ with non-zeros equal to ± 1, set Ω_{ii} to be $\mathbf{U}^T\mathbf{U}$ and then add a diagonal term to ensure $\Omega_{ii} \succ \mathbf{0}$. In the resulting Ω we then insert sparse off-diagonal blocks Ω_{31}, Ω_{42} and Ω_{51}, as well as their corresponding transposes. A diagonal term is added to the final Ω to ensure positive definiteness. The condition number of Ω is approximately 65, so Ω is well conditioned.

Given Ω, we draw $n = 10 \times d$ samples $\sim \mathcal{N}(\mathbf{0}, \Omega^{-1})$ to construct 30 instances of the sample covariance matrix $\mathbf{S}$.

Since in practice we do not know Ω, the tuning parameter $\lambda > 0$ must be selected using a model selection technique. Here we choose

[1] obtained by differentiating the new F_{ia} (with the l_2 penalty) and setting the result to zero.

λ such that the corresponding algorithm solution $\widehat{\Omega}$ minimises the BIC criterion [17]

$$\begin{aligned} \text{BIC}(\lambda) &= \text{tr}(\mathbf{S}\widehat{\Omega}) - \log\det(\widehat{\Omega}) \\ &+ \sum_{i<j} I(\widehat{\Omega}_{ij} \neq \mathbf{0})\, d_i\, d_j \frac{\log(n)}{n} \end{aligned}$$

In our simulations we found that the λ minimising BIC was very close to the λ that minimising the Kullback-Leibler (KL) loss – e.g. [21, 22, 23, 24, 25, 13, 14, 26, 15] used when computing the *oracle* precision matrix estimator.

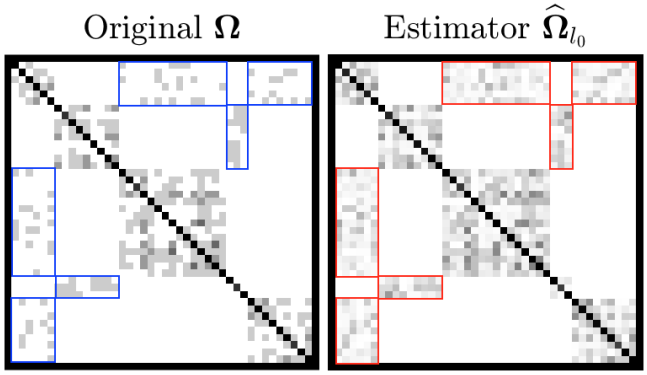

Fig. 1. Comparing Ω and the BIC optimal $\widehat{\Omega}$ obtained by minimising the group l_0-PLL criterion (using one of the 30 sample $\mathbf{S}$).

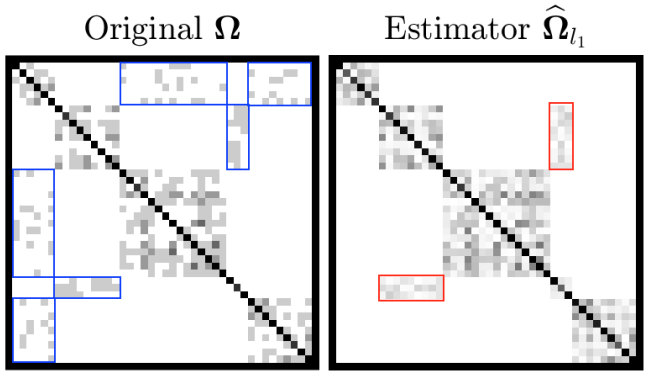

Fig. 2. Comparing Ω and the BIC optimal $\widehat{\Omega}$ obtained by minimising the group l_1-PLL criterion (using the same $\mathbf{S}$ as for figure 1).

In figure 1 we see that all of the off-diagonal blocks (blue) in Ω have been correctly identified. Also the non-zeros in the estimated off-diagonal blocks seem to match the corresponding non-zeros in Ω.

In figure 2 we see that only the off-diagonal block Ω_{42} has been identified. However, the non-zeros in $\widehat{\Omega}_{42}$ poorly match the corresponding non-zeros in Ω_{42}.

Note that the reconstructions in figure 1 and 2 represent the general result (over the 30 instances considered).

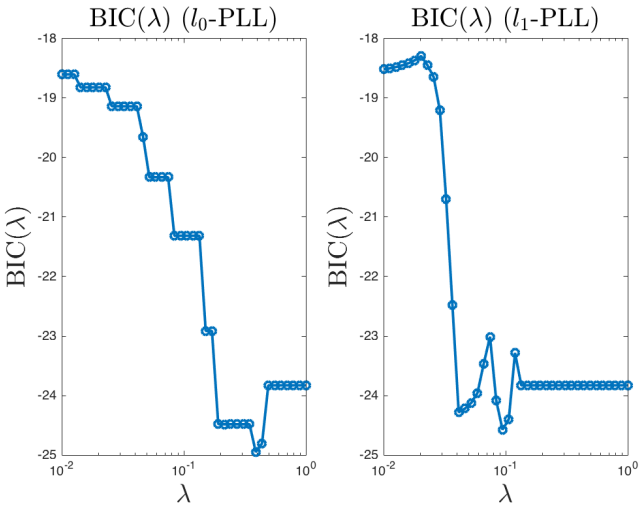

Fig. 3. Plotting the BIC criterion vs. λ. The group l_0 and l_1 estimators in figure 1 and 2 are obtained with the λ that respectively gives the minimum BIC value in this figure.

5. CONCLUSION

We addressed the multivariate (multi-attribute) conditional independence graphical modelling problem by proposing a novel block cyclic descent algorithm for minimising the group l_0 penalized log-likelihood. We compared the method with a group l_1 penalised alternative in simulations, and found that the group l_0 penalisation can yield superior estimates.

6. REFERENCES

[1] E. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Berlin: Springer, 2009.

[2] S. L. Lauritzen, *Graphical Models*. Oxford: Oxford University Press, 1991.

[3] J. Whittaker, *Graphical Models in Applied Mathematical Analysis*. New York: Wiley, 1990.

[4] A. Sandryhaila and JMF. Moura, "Big Data Analysis with Signal Processing on Graphs," *IEEE Sig. Proc. Mag.*, vol. 34, pp. 80–90, 2014.

[5] N. Meinshausen and P. Buhlmann, "High dimensional graphs and variable selection with the LASSO," *Ann. Stat.*, vol. 34, no. 3, pp. 1436–1462, 2006.

[6] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, "Sparse inverse covariance matrix estimation using quadratic approximation," 2011.

[7] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. Ravikumar, and R. Poldrack, "Big & Quic: Sparse inverse covariance estimation for a million variables," pp. 2339–2347, 2013.

[8] E. Treister and J. Turek, "A block–coordinate descent approach for large–scale sparse inverse covariance estimation," *NIPS*, vol. 27, 2014.

[9] P. A. Olsen, F. Oztoprak, J. Nocedal, and S. J. Rennie, "Newton-like methods for sparse inverse covariance estimation," *NIPS*, 2012.

[10] D. Guillot, B. Rajaratnam, B. T. Rolfs, A. Maleki, and I. Wong, "Iterative thresholding algorithm for sparse inverse covariance estimation," *NIPS*, 2012.

[11] G. Marjanovic, M. Ulfarsson, and V. Solo, "Large-scale l_0 sparse inverse covariance estimation," *IEEE ICASSP*, 2016.

[12] G. Marjanovic and V. Solo, "l_0 sparse graphical modeling," *IEEE ICASSP*, pp. 2084–2087, 2011.

[13] ——, "On l_q optimization and sparse inverse covariance selection," *IEEE T. Signal Proces.*, vol. 62, no. 7, 2014.

[14] G. Marjanovic and A. O. Hero III, "On l_q estimation of sparse inverse covariance," *IEEE ICASSP*, 2014.

[15] ——, "l_0 Sparse Inverse Covariance Estimation," *IEEE T. Signal Proces.*, vol. 63, no. 12, pp. 3218–3231, 2015.

[16] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *J. Mach. Learn. Res.*, vol. 9, pp. 485–516, 2008.

[17] M. Kolar, H. Liu, and E. Xing, "Markov network estimation from multi-attribute data," *ICML*, pp. 73–81, 2013.

[18] S. Yang, Q. Sun, S. Ji, P. Wonka, I. Davidson, and J. Ye, "Structural Graphical Lasso for Learning Mouse Brain Connectivity," *Proc. 21st ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 1385–1394, 2015.

[19] M. Luessi, M. Bianciardi, M. S. Hamalainen, and V. Solo, "Mutual Information Based Multivariate Connectivity Analysis Methods for fMRI," 2014, abstracts, Human Brain Mapping, Hamburg, June.

[20] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, pp. 157–175, 1972.

[21] F. Wong, C. K. Carter, and R. Kohn, "Efficient estimation of covariance selection models," *Biometrika*, vol. 90, no. 4, pp. 809–830, 2003.

[22] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[23] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu, "Sparse permutation invariant covariance estimation," *Electron. J. Stat.*, vol. 2, pp. 494–515, 2008.

[24] J. Fan, Y. Feng, and Y. Wu, "Network exploration via the adaptive LASSO and SCAD penalties," *Ann. Appl. Stat.*, vol. 3, no. 2, pp. 521–541, 2009.

[25] E. Levina, A. J. Rothman, and J. Zhu, "Sparse estimation of large covariance matrices via a nested LASSO penalty," *Ann. Appl. Stat.*, vol. 2, pp. 245–263, 2008.

[26] G. Marjanovic and A. O. Hero III, "l_0 sparse inverse covariance estimation," 2014, arXiv: http://arxiv.org/abs/1408.0850.