JDSSV

Journal of Data Science, Statistics, and Visualisation

July 2021, Volume I, Issue II. *doi: 10.52933/jdssv.v1i2.11*

A Stepwise Approach for High-Dimensional Gaussian Graphical Models

Ginette Lafit
KU Leuven, Belgium

Francisco Nogales
UC3M, Spain

Marcelo Ruiz
UNRC, Argentina

Ruben Zamar
UBC, Canada

Abstract

We present a stepwise approach to estimate high dimensional Gaussian graphical models. We exploit the relation between the partial correlation coefficients and the prediction errors, and parametrize the model in terms of the Pearson correlation coefficients between the prediction errors of the nodes' best linear predictors. We propose a novel stepwise algorithm for detecting pairs of conditionally dependent variables. We compare the proposed algorithm with existing methods including graphical lasso (Glasso), constrained ℓ_1-minimization (CLIME) and equivalent partial correlation (EPC), via simulation studies and real life applications. In our simulation study we consider several model settings and report the results using different performance measures that look at desirable features of the recovered graph.

Keywords: covariance selection, Gaussian graphical model, forward and backward selection, partial correlation coefficient.

1. Introduction

High-dimensional *Gaussian graphical models* (GGM) are widely used in practice to represent the linear dependency between variables. The underlying idea in GGM is to measure linear dependencies by estimating partial correlations to infer whether there is an association between a given pair of variables, conditionally on the remaining ones. Moreover, there is a close relation between the nonzero partial correlation coefficients and the nonzero entries in the inverse of the covariance matrix. Covariance selection procedures take advantage of this fact to estimate the GGM conditional dependence structure given a sample (Dempster 1972; Lauritzen 1996; Edwards 2000).

When the dimension p is larger than the number n of observations, the sample covariance matrix S is not invertible and the maximum likehood estimate (MLE) of $\mathbf{\Sigma}$ does not exist. When $p/n \leq 1$, but close to 1, S is invertible but ill-conditioned, increasing the estimation error (Ledoit and Wolf 2004). To deal with this problem, several covariance selection procedures have been proposed based on the assumption that the inverse of the covariance matrix, Ω, called *precision matrix*, is sparse.

We present an approach to perform covariance selection in a high dimensional GGM based on a forward-backward algorithm, which we call *StepGraph*. Our procedure takes advantage of the relation between the partial correlation and the Pearson correlation coefficient of the residuals.

Existing methods to estimate the GGM can be classified in three classes: nodewise regression methods, maximum likelihood methods and limited order partial correlations methods. The nodewise regression method was proposed by Meinshausen and Bühlmann (2006). This method estimates a lasso regression for each node in the graph. See for example Peng et al. (2009), Yuan (2010), Liu and Wang (2012), Zhou et al. (2011) and Ren et al. (2015). Penalized likelihood methods include Yuan and Lin (2007), Banerjee et al. (2008), Friedman et al. (2008), Johnson et al. (2011) and Ravikumar et al. (2011) among others. Cai et al. (2011) propose an estimator called CLIME that estimates precision matrices by solving the dual of an ℓ_1 penalized maximum likelihood problem. Limited order partial correlation procedures use lower order partial correlations to test for conditional independence relations. See Spirtes et al. (2000), Kalisch and Bühlmann (2007), Rütimann et al. (2009), Liang et al. (2015) and Huang et al. (2016).

The rest of the article is organized as follows. Section 2 introduces the stepwise approach along with some notation. Section 3 gives simulations results and a real data example. Section 4 presents some concluding remarks. Appendix A reports detailed description of the crossvalidation procedure used to determine the required parameters in our StepGraph algorithm and Appendix B gives additional simulation results.

2. Stepwise Approach to Covariance Selection

2.1. Definitions and Notation

In this section we review some definitions and technical concepts needed later on. Let $\mathcal{G} = (V, E)$ be a graph where $V \neq \emptyset$ is the set of nodes or vertices and $E \subseteq V \times V = V^2$

is the set of edges. For simplicity we assume that $V = \{1, \ldots, p\}$. The graph $\mathcal{G}$ is undirected, that is, $(i, j) \in E$ if and only if $(j, i) \in E$. Two nodes i and j are called connected, adjacent or neighbors if $(i, j) \in E$.

A *graphical model* (GM) is a graph such that V indexes a set of variables $\{X_1, \ldots, X_p\}$ and E is defined by:

$$(i, j) \notin E \text{ if and only if } X_i \perp\!\!\!\perp X_j \mid X_{V \setminus \{i,j\}}. \tag{1}$$

Here $\perp\!\!\!\perp$ denotes *conditional independence*.

Given a node $i \in V$, its neighborhood $\mathcal{A}_i$ is defined as

$$\mathcal{A}_i = \{l \in V \setminus \{i\} : (i, l) \in E\}. \tag{2}$$

Notice that $\mathcal{A}_i$ gives the nodes directly connected with i and therefore a GM can be effectively described by giving the system of neighborhoods $\{\mathcal{A}_i\}_{i=1}^p$.

We further assume that $(X_1, \ldots, X_p)^\top \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = (\sigma_{ij})_{i,j=1\ldots,p}$ is a positive-definite covariance matrix. In this case the graph is called a *Gaussian graphical model* (GGM). The matrix $\mathbf{\Omega} = (\omega_{ij})_{i,j=1\ldots,p} = \mathbf{\Sigma}^{-1}$ is called *precision matrix*.

There exists an extensive literature on GM and GGM. For a detailed treatment of the theory see for instance Lauritzen (1996), Edwards (2000), and Bühlmann and Van De Geer (2011).

2.2. Conditional Dependence in a GGM

In a GGM the set of edges E represents the conditional dependence structure of the vector $(X_1, \ldots, X_p)$. To represent this dependence structure as a statistical model it is convenient to find a parametrization for E.

In this subsection we introduce a convenient parametrization of E using well known results from classical multivariate analysis. For an exhaustive treatment of these results see, for instance, Anderson (2003), Cramér (1999), Lauritzen (1996) and Eaton (2007).

Given a subset $\mathcal{A}$ of V, $\mathbf{X}_\mathcal{A}$ denotes the vector of variables with subscripts in $\mathcal{A}$ in increasing order. For a given pair of nodes (i, l), set $\mathbf{X}_1^\top = (X_i, X_l)$, $\mathbf{X}_2 = \mathbf{X}_{V \setminus \{i,l\}}$ and $\mathbf{X} = \left(\mathbf{X}_1^\top, \mathbf{X}_2^\top\right)^\top$. Note that $\mathbf{X}$ has multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix

$$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \tag{3}$$

such that Σ_{11} has dimension 2×2, Σ_{12} has dimension $2 \times (p-2)$ and so on. The matrix in (3) is a partition of a permutation of the original covariance matrix Σ, and will be also denoted by Σ, after a small abuse of notation.

Moreover, we set

$$\Omega = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}^{-1} = \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{pmatrix}.$$

Then, by (B.2) of Lauritzen (1996), the blocks Ω_{kj} can be written explicitly in terms of Σ_{kj} and Σ_{kk}^{-1} with $k, j = 1, 2$. In particular

$\Omega_{11} = \left(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)^{-1}$ where

$$\Omega_{11} = \begin{pmatrix} \omega_{ii} & \omega_{il} \\ \omega_{li} & \omega_{ll} \end{pmatrix}$$

is the submatrix of Ω (with rows i and l and columns i and l). Hence,

$$\begin{aligned} \text{COV}\left(\mathbf{X}_1|\mathbf{X}_2\right) &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\ &= \Omega_{11}^{-1} \\ &= \frac{1}{\omega_{ii}\omega_{ll} - \omega_{il}\omega_{li}} \begin{pmatrix} \omega_{ll} & -\omega_{il} \\ -\omega_{li} & \omega_{ii} \end{pmatrix} \end{aligned} \tag{4}$$

and, in consequence, the conditional correlation between X_i and X_l can be expressed as

$$\text{CORR}\left(X_i, X_l|\mathbf{X}_{V\setminus\{i,l\}}\right) = -\frac{\omega_{il}}{\sqrt{\omega_{ii}\omega_{ll}}}. \tag{5}$$

This gives the standard parametrization of E in terms of the support of the precision matrix

$$\text{SUPP}\left(\Omega\right) = \{(i,l) \in V^2 : \, i \neq l, \omega_{i,l} \neq 0\}. \tag{6}$$

We now introduce another parametrization of E, which we need to define and implement our proposed method. We consider the regression error for the regression of $\mathbf{X}_1$ on $\mathbf{X}_2$,

$$\boldsymbol{\varepsilon} = \mathbf{X}_1 - \widehat{\mathbf{X}}_1 = \mathbf{X}_1 - \boldsymbol{\beta}^\top \mathbf{X}_2$$

where $\boldsymbol{\beta}$ is the matrix of regression coeficients and and let ε_i and ε_l denote the entries of $\boldsymbol{\varepsilon}$ (i.e. $\boldsymbol{\varepsilon}^\top = (\varepsilon_i, \varepsilon_l)$). The regression error $\boldsymbol{\varepsilon}$ is independent of $\widehat{\mathbf{X}}_1$ and has normal distribution with mean $\mathbf{0}$ and covariance matrix Ψ_{11} with elements denoted by

$$\Psi_{11} = \begin{pmatrix} \psi_{ii} & \psi_{il} \\ \psi_{li} & \psi_{ll} \end{pmatrix}. \tag{7}$$

A straightforward calculation shows that

$$\begin{aligned} \Psi_{11} &= \text{COV}\left(\mathbf{X}_1\right) + \text{COV}\left(\widehat{\mathbf{X}}_1\right) - 2\text{COV}\left(\mathbf{X}_1, \widehat{\mathbf{X}}_1\right) \\ &= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\ &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = \Omega_{11}^{-1}. \end{aligned}$$

See Cramér (1999, Section 23.4).

Therefore, by this equality, (4) and (5), the partial correlation coefficient and the conditional correlation are equal

$$\rho_{il\cdot V\setminus\{i,l\}} = \text{CORR}\left(X_i, X_l|\mathbf{X}_{V\setminus\{i,l\}}\right) = \frac{\psi_{il}}{\sqrt{\psi_{ii}\psi_{ll}}}.$$

Summarizing, the problem of determining the conditional dependence structure in a GGM (represented by E) is equivalent to finding the pairs of nodes of V that belong to the set

$$\{(i,l) \in V^2 : i \neq l, \psi_{il} \neq 0\} \tag{8}$$

which is equal to the support of the precision matrix, $\text{SUPP}(\Omega)$, defined by (6).

Remark 1 *As noticed above, under normality, partial and conditional correlation are the same. However, in general they are different concepts (Lawrance 1976).*

Remark 2 *Let $\beta_{i,l}$ be the regression coefficient of X_l in the regression of X_i versus $\mathbf{X}_{V\setminus\{i\}}$ and, similarly let $\beta_{l,i}$ be the regression coefficient of X_i in the regression of X_l versus $\mathbf{X}_{V\setminus\{l\}}$. Then it follows that $\rho_{il\cdot V\setminus\{i,l\}} = sign(\beta_{l,i})\sqrt{\beta_{l,i}\beta_{i,l}}$. This allows for another popular parametrization for E. Moreover, let ϵ_i be the error term in the regression of the i^{th} variable on the remaining ones. Then by Lemma 1 in Peng et al. (2009) we have that $\text{COV}(\epsilon_i, \epsilon_l) = \omega_{il}/\omega_{ii}\omega_{ll}$ and $\text{VAR}(\epsilon_i) = 1/\omega_{ii}$.*

2.3. The Stepwise Algorithm

Conditionally on its neighbors, X_i is independent of all the other variables. Therefore, given a system of neighborhoods $\{\mathcal{A}_i\}_{i=1}^p$ and $l \notin \mathcal{A}_i$ (and so $i \notin \mathcal{A}_l$), the partial correlation between X_i and X_l can be obtained by the following procedure based on Lemma 1 of Peng et al. (2009) described in Remark 2: (i) regress X_i on $\mathbf{X}_{\mathcal{A}_i}$ and compute the regression residual ε_i; (ii) regress X_l on $\mathbf{X}_{\mathcal{A}_l}$ and compute the regression residual ε_l; (iii) calculate the Pearson correlation between ε_i and ε_l.

This reasoning motivates the StepGraph algorithm. At each step k of StepGraph, we have a working system of neighborhoods $\widehat{\mathcal{A}}_1^k, ..., \widehat{\mathcal{A}}_p^k$. Then, if $l \notin \widehat{\mathcal{A}}_j^k$ one would expect, under this working assumption, that the empirical partial correlation coefficient $\widehat{\rho}_{jl\cdot\widehat{\mathcal{A}}_j^k}$ is close to zero. If the maximum absolute partial correlation computed this way is large, then we conclude that the working system of neighborhoods needs to be updated. We then add the most likely new edge, the one with the largest partial correlation. This constitutes the forward step. In the backward step, if the minimum absolute partial correlation coefficient between presently connected nodes, j and l, is too small, then this edge is removed.

A step by step description of StepGraph is given below:

Graphical Stepwise Algorithm

> **Input** The (centered) data $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$, and the forward and backward thresholds α_f and α_b.
>
> **Initialization** $k = 0$: set $\widehat{\mathcal{A}}_1^0 = \widehat{\mathcal{A}}_2^0 = \cdots = \widehat{\mathcal{A}}_p^0 = \phi$.
>
> **Iteration Step** Given $\widehat{\mathcal{A}}_1^k, \widehat{\mathcal{A}}_2^k, ..., \widehat{\mathcal{A}}_p^k$ we compute $\widehat{\mathcal{A}}_1^{k+1}, \mathcal{A}_2^{k+1}, ..., \widehat{\mathcal{A}}_p^{k+1}$ as follows.
>
> > **Forward** For each $j = 1, ..., p$ do the following.
> > For each $l \notin \widehat{\mathcal{A}}_j^k$ calculate the partial correlations f_{jl}^k as follows.

(a) Regress the j^{th} variable on the variables with subscript in the set $\widehat{\mathcal{A}}_j^k$ and compute the regression residuals $\mathbf{e}_j^k = \left(e_{1j}^k, e_{2j}^k, ..., e_{nj}^k\right)$.

(b) Regress the l^{th} variables on the variables with subscript in the set $\widehat{\mathcal{A}}_l^k$ and compute the regression residuals $\mathbf{e}_l^k = \left(e_{1l}^k, e_{2l}^k, ..., e_{nl}^k\right)$.

(c) Obtain the partial correlation f_{jl}^k by calculating the Pearson correlation between $\mathbf{e}_j^k$ and $\mathbf{e}_l^k$.

If

$$\max_{l \notin \widehat{\mathcal{A}}_j^k, j \in V} \left|f_{jl}^k\right| = \left|f_{j_0 l_0}^k\right| \geq \alpha_f$$

set $\widehat{\mathcal{A}}_{j_0}^{k+1} = \widehat{\mathcal{A}}_{j_0}^k \cup \{l_0\}$, $\widehat{\mathcal{A}}_{l_0}^{k+1} = \widehat{\mathcal{A}}_{l_0}^k \cup \{j_0\}$, $\widehat{\mathcal{A}}_l^{k+1} = \widehat{\mathcal{A}}_l^k$ for $l \neq j_0, l_0$

If

$$\max \left|f_{jl}^k\right| = \left|f_{j_0 l_0}^k\right| < \alpha_f, \text{ stop.}$$

Backward For each $j = 1, ..., p$ do the following.
For each $l \in \widehat{\mathcal{A}}_j^{k+1}$ calculate the partial correlation b_{jl}^k as follows.

(a) Regress the j^{th} variables on the variables with subscript in the set $\widehat{\mathcal{A}}_j^{k+1} \setminus \{l\}$ and compute the regression residuals $\mathbf{r}_j^k = \left(r_{1j}^k, r_{2j}^k, ..., r_{nj}^k\right)$.

(b) Regress the l^{th} variable on the variables with subscript in the set $\widehat{\mathcal{A}}_l^{k+1} \setminus \{j\}$ and compute the regression residuals $\mathbf{r}_l^k = \left(r_{1l}^k, r_{2l}^k, ..., r_{nl}^k\right)$.

(c) Compute the partial correlation b_{jl}^k by calculating the Pearson correlation between $\mathbf{r}_j^k$ and $\mathbf{r}_l^k$.

If

$$\min_{l \in \widehat{\mathcal{A}}_j^k, j \in V} \left|b_{jl}^k\right| = \left|b_{j_0 l_0}^k\right| \leq \alpha_b$$

set $\widehat{\mathcal{A}}_{j_0}^{k+1} \to \widehat{\mathcal{A}}_{j_0}^{k+1} \setminus \{l_0\}$, $\widehat{\mathcal{A}}_{l_0}^{k+1} \to \widehat{\mathcal{A}}_{l_0}^{k+1} \setminus \{j_0\}$.

Output

1. A collection of estimated neighborhoods $\widehat{\mathcal{A}}_j$, $j = 1, \ldots, p$.
2. The set of estimated edges $\widehat{E} = \left\{(i, l) \in V^2 : i \in \widehat{\mathcal{A}}_l\right\}$.
3. An estimate of $\mathbf{\Omega}$, $\widehat{\mathbf{\Omega}} = (\widehat{\omega}_{il})_{i,l=1}^p$ with $\widehat{\omega}_{il}$ defined as follow: in the case $i = l$, $\widehat{\omega}_{ii} = n/(\boldsymbol{e}_i^T \boldsymbol{e}_i)$ for $i = 1, ..., p$, where $\boldsymbol{e}_i$ is the vector of the prediction errors in the regression of the i^{th} variable on $\boldsymbol{X}_{\widehat{\mathcal{A}}_i}$. In the case $i \neq l$ we must distinguish two cases, if $l \notin \widehat{\mathcal{A}}_i$ then $\widehat{\omega}_{il} = 0$, otherwise $\widehat{\omega}_{il} = n\left(\boldsymbol{e}_i^T \boldsymbol{e}_l\right) / \left[\left(\boldsymbol{e}_i^T \boldsymbol{e}_i\right)\left(\boldsymbol{e}_l^T \boldsymbol{e}_l\right)\right]$ (see Remark 2).

2.4. Thresholds Selection by Cross-Validation

Let $\boldsymbol{X}$ be the $n \times p$ matrix with rows $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, n$, corresponding to n observations. We randomly partition the dataset $\{\mathbf{x}_i\}_{1 \leq i \leq n}$ into K disjoint subsets of

approximately equal sizes, the t^{th} subset being of size $n_t \geq 2$ and $\sum_{t=1}^{K} n_t = n$. For every t, let $\{\mathbf{x}_i^{(t)}\}_{1 \leq i \leq n_t}$ be the t^{th} *validation subset*, and its complement $\{\widetilde{\mathbf{x}}_i^{(t)}\}_{1 \leq i \leq n-n_t}$, the t^{th} *training subset*. For every t and for every pair (α_f, α_b) of threshold parameters let $\widehat{\mathcal{A}}_1^{(t)}, \ldots, \widehat{\mathcal{A}}_p^{(t)}$ be the estimated neighborhoods given by StepGraph using the t^{th} training subset. For every $j = 1, \ldots, p$ let $\widehat{\beta}_{\widehat{\mathcal{A}}_j^{(t)}}$ be the estimated coefficient of the regression of the variable X_j on the neighborhood $\widehat{\mathcal{A}}_j^{(t)}$.

Consider now the t^{th} validation subset. So, for every j, using $\widehat{\beta}^{(t)}_{\mathcal{A}_j^{(t)}}$, we obtain the vector of predicted values $\widehat{\mathbf{X}}_j^{(t)}(\alpha_f, \alpha_b)$. If $\mathcal{A}_j^{(t)} = \emptyset$ we predict each observation of X_j by the sample mean of the observations in the t^{th} dataset of this variable.

Then, we define the K–fold cross–validation function as

$$CV(\alpha_f, \alpha_b) = \frac{1}{n} \sum_{t=1}^{K} \sum_{j=1}^{p} \left\| \mathbf{X}_j^{(t)} - \widehat{\mathbf{X}}_j^{(t)}(\alpha_f, \alpha_b) \right\|^2 \tag{9}$$

where $\|\cdot\|$ denotes the L2-norm or euclidean distance in $\mathbb{R}^p$. Hence the K–fold cross–validation forward–backward thresholds $\widehat{\alpha}_f$, $\widehat{\alpha}_b$ is

$$(\widehat{\alpha}_f, \widehat{\alpha}_b) =: \underset{(\alpha_f, \alpha_b) \in \mathcal{H}}{\operatorname{argmin}} \; CV(\alpha_f, \alpha_b)$$

where $\mathcal{H}$ is a grid of ordered pairs (α_f, α_b) in $[0, 1] \times [0, 1]$ over which we perform the search. For a detailed description see Appendix A.

2.5. Example

To illustrate the algorithm we consider the GGM with 16 edges given in the first panel of Figure 1. We draw $n = 1000$ independent observations from this model (see the next section for details). The values for the threshold parameters $\alpha_f = 0.17$ and $\alpha_b = 0.09$ are determined by 5-fold cross-validation. The figure also displays the selected pairs of edges at each step in a sequence of successive updates of $\widehat{\mathcal{A}}_j^k$, for $k = 1, 4, 9, 12$ and the final step $k = 16$, showing that the estimated graph is identical to the true graph.

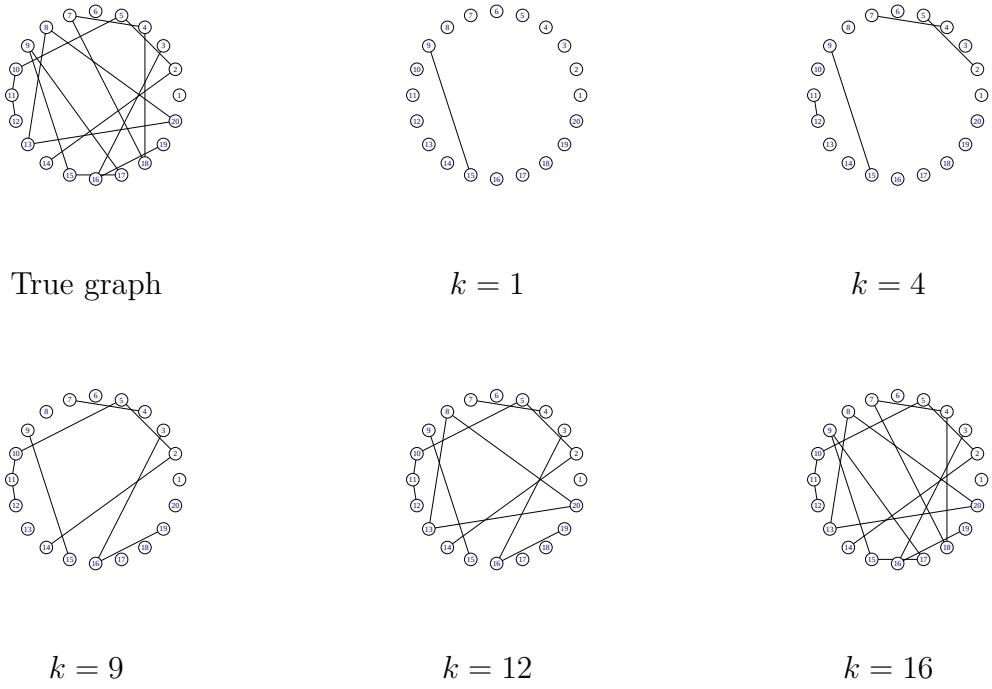


Figure 1: True graph and sequence of successive updates of $\widehat{\mathcal{A}}_j^k$, for $k = 1, 4, 9, 12, 16$ of StepGraph.

3. Numerical Results and Real Data Example

We conducted extensive Monte Carlo simulations to investigate the performance of StepGraph. In this section we report some results from this study and a numerical experiment using real data.

3.1. Monte Carlo Simulation Study

Simulated Models

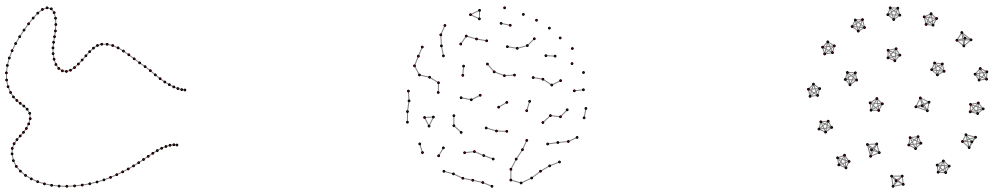
We consider three dimension values $p = 50, 100, 150$ and three different models for $\mathbf{\Omega}$:

Model 1 Autoregressive model of orden 1, denoted AR(1). In this case $\Sigma_{ij} = 0.4^{|i-j|}$ for $i, j = 1, \ldots p$.

Model 2 Nearest neighbors model of order 2, denoted NN(2). For each node we randomly select two neighbors and choose a pair of symmetric entries of $\mathbf{\Omega}$ using the NeighborOmega function of the R package Tlasso.

Model 3 Block diagonal matrix model with q blocks of size p/q, denoted BG. For $p = 50, 100$ and 150, we use $q = 10, 20$ and 30 blocks, respectively. Each block, of size $p/q = 5$, has diagonal elements equal to 1 and off-diagonal elements equal to 0.5.

For each p and each model we generate $R = 50$ random samples of size $n = 100$. These graph models are widely used in the genetic literature to model gene expression data. See for example Lee and Liu (2015) and Lee and Ghi (2006). Figure 2 displays graphs from Models 1-3 with $p = 100$ nodes.



AR(1) NN(2) BG

Figure 2: Graphs of AR(1), NN(2) and BG graphical models for $p = 100$ nodes.

Methods

We compare the performance of StepGraph with graphical lasso (Glasso), constrained l_1-minimization for inverse matrix estimation (CLIME) and equivalent partial correlation (EPC) proposed by Friedman et al. (2008), Cai et al. (2011) and Liang et al. (2015) respectively. More precisely, the methods compared in our simulation study are:

1. The Glasso estimate obtained by solving the ℓ_1 penalized-likelihood problem:

$$\min_{\mathbf{\Omega} \succ 0} \left(-\log\{\det[\mathbf{\Omega}]\} + \mathrm{tr}\{\mathbf{\Omega}\mathbf{X}^\top\mathbf{X}\} + \lambda \parallel \mathbf{\Omega} \parallel_1 \right). \tag{10}$$

In our simulations and examples we use the R-package CVGLASSO with the tuning parameter λ selected by 5−fold crossvalidation (the package default).

2. The CLIME estimate obtained by symmetrization of the solution of

$$\min\{\parallel \mathbf{\Omega} \parallel_1 \text{ subject to } |S\mathbf{\Omega} - \boldsymbol{I}|_\infty \leq \lambda\}, \tag{11}$$

where S is the sample covariance, $\boldsymbol{I}$ is the identity matrix, $|\cdot|_\infty$ is the elementwise l_∞ norm, and λ is a tuning parameter. For computations, we use the R-package CLIME with the tuning parameter λ selected by 5−fold crossvalidation (the package default).

3. The EPC method, which performs multiple hypothesis tests based on an equivalent measure to the partial correlation coefficient. This method starts by constructing a reduced system of neighboorhoods based on correlation screening step, which identifies correlation coefficients that are significantly different for zero. We use the R-package EQUSA with default choice of parameters.

4. The proposed method StepGraph with the forward and backward thresholds, $\alpha_f > \alpha_b$, determined by 5-fold crossvalidation, as described in Appendix A. A

slight modification of StepGraph, called StepGraph$_2$, uses a reduced sytem of neighborhoods as in EPC. This version is available as an option in our implementation.

To evaluate the graph recovery we compute the Matthews correlation coefficient (Matthews 1975)

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}, \tag{12}$$

the Specificity $= \text{TN}/(\text{TN}+\text{FP})$ and the Sensitivity $= \text{TP}/(\text{TP}+\text{FN})$. Here TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. Larger values of MCC, Sensitivity and Specificity indicate better performances (Fan et al. 2009; Baldi et al. 2000).

The performance of $\widehat{\mathbf{\Omega}}$ as an estimate for $\mathbf{\Omega}$ is measured by $m_F = ||\widehat{\mathbf{\Omega}} - \mathbf{\Omega}||_F$ (where $||\cdot||_F$ denotes the Frobenius norm) and by the normalized Kullback-Leibler divergence defined by $m_{NKL} = D_{KL}/(1+D_{KL})$ where

$$D_{KL} = \frac{1}{2}\left(\text{tr}\left\{\widehat{\mathbf{\Omega}}\mathbf{\Omega}^{-1}\right\} - \log\left\{\det\left[\widehat{\mathbf{\Omega}}\mathbf{\Omega}^{-1}\right]\right\} - p\right)$$

is the the Kullback-Leibler divergence between $\widehat{\mathbf{\Omega}}$ and $\mathbf{\Omega}$.

Results

Table 1 shows the MCC performance for the three methods under Models 1-3. For models 1 and 2, StepGraph and EPC clearly outperforms the other two methods, with CLIME being only slightly better than Glasso. EPC is slightly better than StepGraph and worse than StepGraph$_2$. Moreover, the EQUSA package often crashes in the case of model 3 (NA values reported in the table). Cai et al. (2011) pointed out that a procedure yielding a more sparse $\widehat{\mathbf{\Omega}}$ is preferable because this facilitates interpretation of the data. The sensitivity and specificity results, reported in Table 5 in Appendix B, show that in general StepGraph, StepGraph$_2$ and EPC estimate more sparse graphs than the CLIME and Glasso, yielding fewer false positives (more specificity) but a few more false negatives (less sensitivity). Table 2 shows that all the methods are roughly comparable under AR(1) and show equally poor performances under NN(2). StepGraph and StepGraph$_2$ outperform the competitors under model BG.

The axes in the panels in Figure 3 display the graph p-nodes in a given order. Each cell displays a gray level proportional to the frequency with which the corresponding pair of nodes appear in the estimated graph from the $R = 50$ simulation runs. Hence a white color in a given cell (i, j) means that nodes i and j are never adjacent in the graph. On the other hand, a pair of nodes that are always adjacent in the graph are given a black color. Notice that the sparsity patterns estimated by StepGraph and StepGraph$_2$ best match those of the true models. As noticed before, EPC results are missing for the case of BG. Figures 1 -3 in Appendix B display similar heatmaps and conclusions for 100 and 150 nodes.

Table 1: Comparison of means and standard errors (in brackets) of MCC over $R = 50$ replicates.

Model	p	StepGraph		StepGraph_2		Glasso		CLIME		EPC	
	50	0.741	(0.009)	0.863	(0.005)	0.419	(0.016)	0.492	(0.006)	0.831	(0.005)
AR(1)	100	0.751	(0.004)	0.847	(0.005)	0.433	(0.020)	0.464	(0.004)	0.803	(0.005)
	150	0.730	(0.004)	0.837	(0.004)	0.474	(0.017)	0.499	(0.003)	0.778	(0.004)
	50	0.751	(0.004)	0.857	(0.006)	0.404	(0.014)	0.401	(0.007)	0.870	(0.004)
NN(2)	100	0.802	(0.005)	0.875	(0.005)	0.382	(0.006)	0.407	(0.005)	0.862	(0.000)
	150	0.695	(0.007)	0.799	(0.004)	0.337	(0.008)	0.425	(0.003)	0.762	(0.004)
	50	0.898	(0.005)	0.832	(0.028)	0.356	(0.009)	0.482	(0.005)	NA	NA
BG	100	0.857	(0.005)	0.857	(0.005)	0.348	(0.004)	0.461	(0.002)	NA	NA
	150	0.780	(0.008)	0.780	(0.008)	0.314	(0.003)	0.408	(0.003)	NA	NA

Table 2: Comparison of means and standard errors (in brackets) of m_F and m_{NKL} over $R = 50$ replicates.

		StepGraph		StepGraph_2		Glasso		CLIME		EPC	
Model	p	m_{NKL}	m_F	m_{NKL}	m_F	m_{NKL}	m_F	m_{NKL}	m_F	m_{NKL}	m_F
	50	0.70	3.82	0.66	3.59	0.64	3.90	0.63	3.91	0.67	3.75
		(0.00)	(0.00)	(0.00)	(0.03)	(0.00)	(0.02)	(0.00)	(0.01)	(0.00)	(0.03)
AR(1)	100	0.83	5.73	0.81	5.24	0.80	5.72	0.79	5.75	0.82	5.56
		(0.00)	(0.00)	(0.00)	(0.03)	(0.00)	(0.02)	(0.00)	(0.01)	(0.00)	(0.03)
	150	0.89	7.16	0.87	6.53	0.86	7.21	0.86	7.25	0.88	7.03
		(0.00)	(0.00)	(0.00)	(0.03)	(0.02)	(0.02)	(0.01)	(0.01)	(0.00)	(0.02)
	50	0.99	6.98	0.99	6.88	0.99	6.65	0.99	6.64	1.00	6.39
		(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
NN(2)	100	1.00	10.11	1.00	10.09	1.00	9.64	1.00	9.60	1.00	9.30
		(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.01)	(0.00)	(0.01)	(0.00)	(0.00)
	150	1.00	12.37	1.00	12.34	1.00	11.90	1.00	11.79	1.00	11.51
		(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	(0.00)
	50	0.46	1.44	0.50	1.97	0.85	5.45	0.82	5.03	NA	NA
		(0.00)	(0.00)	(0.02)	(0.23)	(0.00)	(0.10)	(0.00)	(0.05)	NA	NA
BG	100	0.71	2.94	0.71	2.94	0.93	9.16	0.92	8.71	NA	NA
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.07)	(0.00)	(0.02)	NA	NA
	150	0.88	6.10	0.88	6.10	0.96	11.59	0.96	11.42	NA	NA
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.06)	(0.00)	(0.02)	NA	NA

Table 3 compares the average running time, in seconds, for each method under model AR(1) with $p = 50$ nodes and $R = 50$ replications. The times were obtained using the R-package `tictoc`.

Table 3: Comparison of average running time, in seconds, for each method under model AR(1) with $p = 50$ nodes and $R = 50$ replications.

	Mean	Standard Error
StepGraph	444.50	16.83
StepGraph_2	130.81	0.75
Glasso	0.28	0.01
CLIME	122.48	0.11
EPC	2.27	0.01

In Appendix C we compare the performance of StepGraph using cross–validation and extended Bayesian criterion (EBIC). The results show that StepGraph based on EBIC trades–off a moderate decrease in sensitivity and increase the considered distances in exchange for a considerable decrease in average computing time.

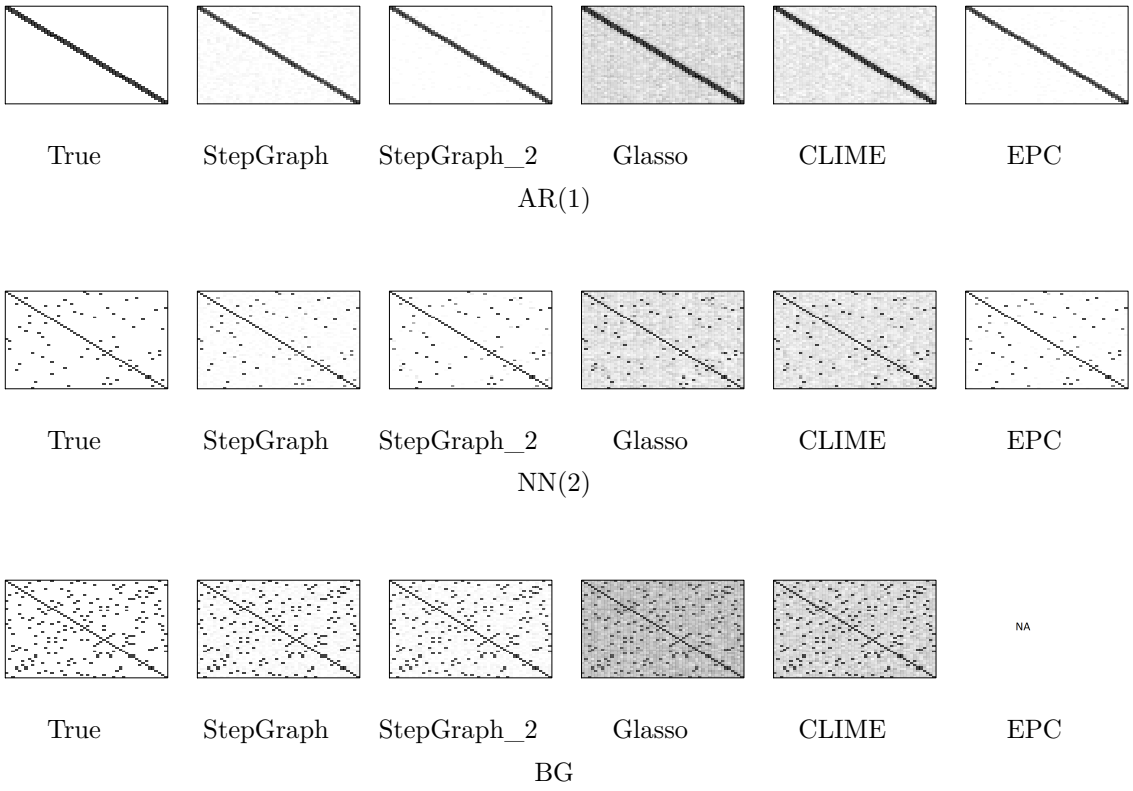


Figure 3: Models heatmaps for the frequency of adjancency for each pair of nodes, for models AR(1), NN(2) and BG, with $p = 50$ nodes. The axes display the graph p-nodes in a given order.

3.2. Analysis of Breast Cancer Data

In preoperative chemoterapy, the complete eradication of all invasive cancer cells is referred to as *pathological complete response*, abbreviated as pCR. It is known in medicine that pCR is associated with the long-term cancer-free survival of a patient. Gene expression profiling (GEP) – the measurement of the activity (expression level) of genes in a patient – could in principle be a useful predictor for the patient's pCR.

Using normalized gene expression data of patients in stages I-III of breast cancer, Hess et al. (2006) aim to identify patients that may achieve pCR under *sequential anthracycline paclitaxel* preoperative chemotherapy. When a patient does not achieve pCR state, he is classified in the group of residual disease (RD), indicating that cancer still remains. Their data consist of 22283 gene expression levels for 133 patients, with 34 pCR and 99 RD. Following Fan et al. (2009) and Cai et al. (2011) we randomly split the data into a training set and a testing set. The testing set is formed by randomly selecting 5 pCR patients and 16 RD patients (roughly 1/6 of the subjects) and the remaining patients form the training set. From the training set, a two sample t-test is performed to select the 50 most significant genes. The data is then standardized using the standard deviation estimated from the training set.

We apply a linear discriminant analysis (LDA) to predict whether a patient may achieve pathological complete response (pCR), based on the estimated inverse covariance ma-

trix of the gene expression levels. We label with $r = 1$ the pCR group and $r = 2$ the RD group and assume that data are normally distributed, with common covariance matrix $\boldsymbol{\Sigma}$ and different means $\boldsymbol{\mu}_r$. From the training set, we obtain $\widehat{\boldsymbol{\mu}}_r$, $\widehat{\boldsymbol{\Omega}}$ and for the test data compute the linear discriminant score as follows

$$\delta_r(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\Omega}} \widehat{\boldsymbol{\mu}}_r - \frac{1}{2} \boldsymbol{\mu}_r^\top \widehat{\boldsymbol{\Omega}} \boldsymbol{\mu}_r + \log \widehat{\pi}_r \quad \text{for } i = 1, \ldots, n, \tag{13}$$

where $\widehat{\pi}_r$ is the proportion of group r subjects in the training set. The classification rule is

$$\widehat{r}(\mathbf{x}) = \operatorname{argmax} \delta_r(\mathbf{x}) \quad \text{for } r = 1, 2. \tag{14}$$

For every method we use 5-fold cross validation on the training data to select the tuning constants. We repeat this scheme 100 times.

Table 4 displays the means and standard errors (in brackets) of Sensitivity, Specificity, MCC and Number of selected Edges using $\widehat{\boldsymbol{\Omega}}$ over the 100 replications. As measured by MCC, the performance of StepGraph and CLIME are similar. However notice that StepGraph is preferable because the recovered graph is much more sparse. On the other hand, the performances of Glasso and EPC are similarly poor. The results of StepGraph_2 are similar to those of StepGraph and therefore omited.

Table 4: Comparison of means and standard errors (in brackets) of Sensitivity, Specificity, MCC and Number of selected edges over 100 replications.

	StepGraph		Glasso		CLIME		EPC	
Sensitivity	0.798	(0.020)	0.612	(0.021)	0.786	(0.020)	0.682	(0.021)
Specificity	0.784	(0.010)	0.754	(0.011)	0.788	(0.010)	0.712	(0.007)
MCC	0.520	(0.020)	0.342	(0.021)	0.516	(0.020)	0.346	(0.017)
Number of Edges	54	(2)	1712	(63)	4823	(8)	13	(0)

4. Concluding Remarks

This paper introduces a stepwise procedure, called StepGraph, to perform covariance selection in high dimensional Gaussian graphical models. StepGraph uses a different parametrization of the Gaussian graphical model based on Pearson correlations between the best-linear-predictors prediction errors. The algorithm begins with a family of empty neighborhoods and using basic steps, forward and backward, adds or delete edges until appropriate thresholds are reached. These thresholds are automatically determined by cross–validation.

StepGraph is compared with Glasso, CLIME and EPC under different Gaussian graphical models (AR(1), NN(2) and BG) and using different performance measures regarding network recovery and sparse estimation of the precision matrix Ω. StepGraph is shown to have good support recovery performance and to produce more sparse models than Glasso and CLIME (i.e. StepGraph is a parsimonious estimation procedure). StepGraph and StepGraph_2 (a variant including a pre-processing correlation screening step) compare well with standard procedures including Glasso, CLIME and EPC. Particularly good simulation results are obtained under block models where the other approaches face some difficulties.

We apply StepGraph for the analysis of breast cancer data and show that our method is a useful tool for applications in medicine and other fields.

Acknowledgments

The authors thanks the generous support of NSERC, Canada, the Institute of Financial Big Data, University Carlos III of Madrid and the CSIC, Spain and the University of Rio Cuarto, Argentina.

References

Anderson, T. (2003). *An Introduction to Multivariate Statistical Analysis*. John Wiley.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics*, 16(5):412–424.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Cai, T., Liu, W., and Luo, X. (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607.

Cramér, H. (1999). *Mathematical Methods of Statistics*. Princeton University Press.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, pages 157–175.

Eaton, M. L. (2007). *Multivariate Statistics : A Vector Space Approach*. Institute of Mathematical Statistics.

Edwards, D. (2000). *Introduction to Graphical Modelling*. Springer Science & Business Media.

Fan, J., Feng, Y., and Wu, Y. (2009). Network exploration via the adaptive lasso and scad penalties. *The Annals of Applied Statistics*, 3(2):521–541.

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

Hess, K. R., Anderson, K., Symmans, W. F., Valero, V., Ibrahim, N., Mejia, J. A., Booser, D., Theriault, R. L., Buzdar, A. U., Dempsey, P. J., et al. (2006). Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology*, 24(26):4236–4244.

Huang, S., Jin, J., and Yao, Z. (2016). Partial correlation screening for estimating large precision matrices, with applications to classification. *The Annals of Statistics*, 44(5):2018–2057, DOI: 10.1214/15-AOS1392, http://dx.doi.org/10.1214/15-AOS1392.

Johnson, C. C., Jalali, A., and Ravikumar, P. (2011). High-dimensional sparse inverse covariance estimation using greedy methods. *arXiv preprint arXiv:1112.6411*.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *The Journal of Machine Learning Research*, 8:613–636.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford University Press.

Lawrance, A. J. (1976). On conditional and partial correlation. *The American Statistician*, 30(3):146–149.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411.

Lee, H. and Ghi, J. (2006). Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317.

Lee, W. and Liu, Y. (2015). Joint estimation of multiple precision matrices with common structures. *Journal of Machine Learning Research*, 16(1):10351062.

Liang, F., Song, Q., and Qiu, P. (2015). An equivalent measure of partial correlation coefficients for high-dimensional gaussian graphical models. *Journal of the American Statistical Association*, 110(511):1248–1265.

Liu, H. and Wang, L. (2012). Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models. *arXiv preprint arXiv:1209.2437*.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, 405(2):442451.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746.

Ravikumar, P., Wainwright, M. J., Raskutti, G., Yu, B., et al. (2011). High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980.

Ren, Z., Sun, T., Zhang, C.-H., Zhou, H. H., et al. (2015). Asymptotic normality and optimalities in estimation of large gaussian graphical models. *The Annals of Statistics*, 43(3):991–1026.

Rütimann, P., Bühlmann, P., et al. (2009). High dimensional sparse covariance estimation via directed acyclic graphs. *Electronic Journal of Statistics*, 3:1133–1160.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT press.

Yuan, M. (2010). High dimensional inverse covariance matrix estimation via linear programming. *The Journal of Machine Learning Research*, 11:2261–2286.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhou, S., Rütimann, P., Xu, M., and Bühlmann, P. (2011). High-dimensional covariance estimation based on gaussian graphical models. *The Journal of Machine Learning Research*, 12:2975–3026.

A. Selection of the Thresholds Parameters by Cross-Validation

In this section we describe the selection of the forward and backward thresholds for StepGraph.

Let $\boldsymbol{X}$ be the $n \times p$ matrix with rows $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, n$, corresponding to n observations. For each $j = 1, \ldots, p$, let $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})^\top$ denote the jth–column of the matrix $\mathbf{X}$.

We randomly partition the dataset $\{\mathbf{x}_i\}_{1 \leq i \leq n}$ into K disjoint subsets of approximately equal size, the t^{th} subset being of size $n_t \geq 2$ and $\sum_{t=1}^{K} n_t = n$. For every t, let $\{\mathbf{x}_i^{(t)}\}_{1 \leq i \leq n_t}$ be the t^{th} *validation subset*, and its complement $\{\widetilde{\mathbf{x}}_i^{(t)}\}_{1 \leq i \leq n-n_t}$, the t^{th} *training subset*. For every $t = 1, \ldots, K$ and threshold parameters $(\alpha_f, \alpha_b) \in [0,1] \times [0,1]$ let $\widehat{\mathcal{A}}_1^{(t)}, \ldots, \widehat{\mathcal{A}}_p^{(t)}$ be the estimated neighborhoods given by StepGraph using the t^{th} training subset $\{\widetilde{\mathbf{x}}_i^{(t)}\}_{1 \leq i \leq n-n_t}$ with $\widetilde{\mathbf{x}}_i^{(t)} = (\widetilde{x}_{i1}^{(t)}, \ldots, \widetilde{x}_{ip}^{(t)})$, $1 \leq i \leq n - n_t$. Consider for every node j the estimated neighborhood $\widehat{\mathcal{A}}_j^{(t)} = \{l_1, \ldots, l_q\}$ and let $\widehat{\beta}_{\widehat{\mathcal{A}}_j^{(t)}}$ be the estimated coefficient of the regression of $\widetilde{\mathbf{X}}_j = (\widetilde{x}_{1j}^{(t)}, \ldots, \widetilde{x}_{n-n_t j}^{(t)})^\top$ on $X_{l_1}, \ldots, X_{l_q}$, represented in (16) (red colour).

Consider the t^{th} validation subset $\{\mathbf{x}_i^{(t)}\}_{1 \leq i \leq n_t}$ with $\mathbf{x}_i^{(t)} = (x_{i1}^{(t)}, \ldots, x_{ip}^{(t)})$, $1 \leq i \leq n_t$ and for every j let $\mathbf{X}_j^{(t)} = \left(x_{1j}^{(t)}, \ldots, x_{n_t j}^{(t)}\right)^\top$ and define the vector of predicted values

$$\widehat{\mathbf{X}}_j^{(t)}(\alpha_f, \alpha_b) = \mathbf{X}_{\widehat{\mathcal{A}}_j^{(t)}} \widehat{\beta}_{\mathcal{A}_j^{(t)}}^{(t)},$$

where $\mathbf{X}_{\widehat{\mathcal{A}}_j^{(t)}}$ is the matrix with rows $(x_{il_1}^{(t)}, \ldots, x_{il_q}^{(t)})$, $1 \leq i \leq n_t$ represented in (16) (in blue colour). If the neighborhood $\mathcal{A}_j^{(t)} = \emptyset$ we define

$$\widehat{\mathbf{X}}_j^{(t)}(\alpha_f, \alpha_b) = (\bar{x}_j^{(t)}, \ldots, \bar{x}_j^{(t)})^\top$$

where $\bar{x}_j^{(t)}$ is the mean of the sample of observations $x_{1j}^{(t)}, \ldots, x_{n_t j}^{(t)}$.

We define the K–fold cross–validation function as

$$CV(\alpha_f, \alpha_b) = \frac{1}{n} \sum_{t=1}^{K} \sum_{j=1}^{p} \left\| \mathbf{X}_j^{(t)} - \widehat{\mathbf{X}}_j^{(t)}(\alpha_f, \alpha_b) \right\|^2$$

where $\|\cdot\|$ the L2-norm or euclidean distance in $\mathbb{R}^p$. Hence the K–fold cross–validation forward–backward thresholds $\widehat{\alpha}_f$, $\widehat{\alpha}_b$ is

$$(\widehat{\alpha}_f, \widehat{\alpha}_b) =: \underset{(\alpha_f, \alpha_b) \in \mathcal{H}}{\operatorname{argmin}} \; CV(\alpha_f, \alpha_b) \tag{15}$$

where $\mathcal{H}$ is a grid of ordered pairs (α_f, α_b) in $[0,1] \times [0,1]$ over which we perform the search.

$$\left(\begin{array}{cccccc}
t^{th} & \text{training} & \text{subset} & & & \\
\cdots & \widetilde{x}_{1j}^{(t)} & \cdots & \widetilde{x}_{1l_1}^{(t)} & \cdots & \widetilde{x}_{1l_q}^{(t)} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & \widetilde{x}_{n-n_t j}^{(t)} & \cdots & \widetilde{x}_{n-n_t l_1}^{(t)} & \cdots & \widetilde{x}_{n-n_t l_q}^{(t)} & \cdots \\
\hline
t^{th} & \text{validation} & \text{subset} & & & & \\
\cdots & x_{1j}^{(t)} & \cdots & x_{1l_1}^{(t)} & \cdots & x_{1l_q}^{(t)} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & x_{n_t j}^{(t)} & \cdots & x_{n_t l_1}^{(t)} & \cdots & x_{n_t l_q}^{(t)} & \cdots
\end{array}\right) \tag{16}$$

Remark 3 *Matrix* (16) *represents, for every node* j *the comparison between estimated and predicted values for cross-validation.* $\widehat{\beta}_{\widehat{\mathcal{A}}_j^{(t)}}$ *is computed using the observations* $\widetilde{\mathbf{X}}_j = (\widetilde{x}_{1j}^{(t)}, \ldots, \widetilde{x}_{n-n_t j}^{(t)})^\top$ *and the matrix* $\widetilde{\mathbf{X}}_{\widehat{\mathcal{A}}_j^{(t)}}$ *with rows* $(\widetilde{x}_{il_1}^{(t)}, \ldots, \widetilde{x}_{il_q}^{(t)})$, $i = 1, \ldots, n - n_t$ *in the* t^{th} *training subset (red colour). Based on the* t^{th} *validation set* $\widehat{\mathbf{X}}_j^{(t)}$ *is computed using* $\mathbf{X}_{\widehat{\mathcal{A}}_j^{(t)}}$ *and compared with* $\mathbf{X}_j$ *(in blue color).*

B. Additional Simulation Results

In this section we give aditional simulation results. Table 5 reports additional Specificity and Sensitivity results from our simulation study. Figures 3 - 6 display the heatmaps for the three considered models and p equal to 50, 100 and 150.

Table 5: Comparison of means and standard errors (in brackets) of Specificity (TN%), Sensitivity (TP%) and MCC over $R = 50$ replicates.

		StepGraph			StepGraph_2			Glasso			CLIME			EPC		
Model	p	TP%	TN%	MCC	TP%	TN%	MCC	TP%	TN%	MCC	TP%	TN%	MCC	TP%	TN%	MCC
	50	0.756	0.988	0.741	0.812	0.997	0.863	0.994	0.823	0.419	0.988	0.891	0.492	0.750	0.998	0.831
		(0.015)	(0.002)	(0.009)	(0.011)	(0.000)	(0.005)	(0.002)	(0.012)	(0.016)	(0.002)	(0.003)	(0.006)	(0.011)	(0.000)	(0.005)
AR(1)	100	0.632	0.999	0.751	0.771	0.999	0.847	0.989	0.897	0.433	0.983	0.934	0.464	0.689	0.999	0.803
		(0.007)	(0.000)	(0.004)	(0.008)	(0.000)	(0.005)	(0.002)	(0.009)	(0.020)	(0.002)	(0.001)	(0.004)	(0.009)	(0.000)	(0.005)
	150	0.607	0.999	0.730	0.749	0.999	0.837	0.981	0.943	0.474	0.972	0.964	0.499	0.636	1.000	0.778
		(0.006)	(0.000)	(0.004)	(0.007)	(0.000)	(0.004)	(0.002)	(0.007)	(0.017)	(0.002)	(0.001)	(0.003)	(0.007)	(0.000)	(0.004)
	50	0.632	0.999	0.751	0.787	0.999	0.857	0.971	0.864	0.404	0.984	0.875	0.401	0.798	0.999	0.870
		(0.007)	(0.000)	(0.004)	(0.012)	(0.000)	(0.006)	(0.004)	(0.010)	(0.014)	(0.003)	(0.004)	(0.007)	(0.008)	(0.000)	(0.004)
NN(2)	100	0.730	0.999	0.802	0.831	0.999	0.875	0.987	0.924	0.382	0.985	0.937	0.407	0.791	0.999	0.862
		(0.008)	(0.000)	(0.005)	(0.007)	(0.000)	(0.005)	(0.002)	(0.004)	(0.006)	(0.002)	(0.001)	(0.005)	(0.007)	(0.000)	(0.000)
	150	0.555	0.999	0.695	0.693	0.999	0.799	0.952	0.936	0.337	0.934	0.965	0.425	0.621	1.000	0.762
		(0.017)	(0.000)	(0.007)	(0.006)	(0.000)	(0.004)	(0.004)	(0.002)	(0.008)	(0.003)	(0.001)	(0.003)	(0.007)	(0.000)	(0.004)
	50	0.994	0.981	0.898	0.904	0.983	0.832	0.867	0.697	0.356	0.962	0.807	0.482	NA	NA	NA
		(0.002)	(0.001)	(0.005)	(0.039)	(0.001)	(0.028)	(0.032)	(0.021)	(0.009)	(0.004)	(0.005)	(0.005)	NA	NA	NA
BG	100	0.949	0.989	0.857	0.949	0.989	0.857	0.569	0.908	0.348	0.818	0.920	0.462	NA	NA	NA
		(0.007)	(0.000)	(0.005)	(0.007)	(0.000)	(0.005)	(0.039)	(0.011)	(0.004)	(0.005)	(0.005)	(0.002)	NA	NA	NA
	150	0.782	0.994	0.780	0.782	0.994	0.780	0.426	0.952	0.314	0.626	0.959	0.408	NA	NA	NA
		(0.021)	(0.000)	(0.008)	(0.021)	(0.000)	(0.008)	(0.035)	(0.006)	(0.003)	(0.006)	(0.001)	(0.003)	NA	NA	NA

Figure 4: Model AR(1). Heatmaps for the frequency of adjancency for each pair of nodes. The axes display the graph p-nodes in a given order.

Figure 5: Model NN(2). Heatmaps for the frequency of adjancency for each pair of nodes. The axes display the graph p-nodes in a given order.

Figure 6: Model BG. Heatmaps for the frequency of adjancency for each pair of nodes. The axes display the graph p-nodes in a given order.

C. Comparison of Cross-Validation and EBIC

Let $(X_1, \ldots, X_p)^\top \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Omega} = (\omega_{ij})_{i,j=1\ldots,p} = \mathbf{\Sigma}^{-1}$ the *precision matrix*. Let $\mathcal{G} = (V, E)$ denote the associated *Gaussian graphical model* (GGM) where V is the set of nodes and E is the set of edges.

For every pair of forward-backward thresholds (α_f, α_b) and based in a $n \times p$ data matrix $\boldsymbol{X}$, let $\widehat{\mathbf{\Omega}} = \widehat{\mathbf{\Omega}}_{(\alpha_f,\alpha_b)}$ and $\widehat{E} = \widehat{E}_{(\alpha_f,\alpha_b)}$ be the estimated precision matrix and the set of edges, respectively, computed by the StepGraph algorithm.

The EBIC criterion, see (1) of Foygel and Drton (2010), is defined as

$$\mathrm{EBIC}_\gamma(\alpha_f, \alpha_b) = -2\mathrm{l}_n(\widehat{\mathbf{\Omega}}) + \ |\widehat{E}|\log(n) + 4|\widehat{E}|\gamma\log(p) \tag{17}$$

where $\mathrm{l}_n(\widehat{\mathbf{\Omega}})$ denotes the log-likehood function based on the estimated model $\widehat{\mathbf{\Omega}}$, $|\widehat{E}|$ is the cardinal of $\widehat{E}$ and γ is a parameter that controls the consistency when p and n increase. If $\gamma = 0$, EBIC is the classical Bayesian information criterion. Usual values of γ are $1/2$ and 1.

So, given a value of γ, the optimal forward–backward thresholds $(\widehat{\alpha}_f, \widehat{\alpha}_b)$ based on EBIC_γ is defined as

$$(\widehat{\alpha}_f, \widehat{\alpha}_b) = \underset{(\alpha_f,\alpha_b)\in\mathcal{H}}{\operatorname{argmin}} \ \mathrm{EBIC}_\gamma(\alpha_f, \alpha_b)$$

where $\mathcal{H} \subseteq [0, 1] \times [0, 1]$ is a grid over which we perform the search.

We performed a new simulation experiment comparing the performance of StepGraph using the cross-validation function and EBIC, denoted by StepGraph (CV) and StepGraph (EBIC) with $\gamma = 1/2$ as recommended by Foygel and Drton (2010), respectively. For the GGM AR(1) we generate $R = 50$ random samples of size $n = 100$.

Tables 6, 7 and 8 show that StepGraph (EBIC) trades–off a moderate decrease in sensitivity and increase the considered distances in exchange for a considerable decrease in average computing time.

Table 6: Comparison of means and standard errors (in brackets) of Specificity (TN%), Sensitivity (TP%) and MCC over $R = 50$ replicates.

	Sensitivity		Specificity		MCC	
StepGraph (CV)	0.77	(0.02)	0.99	(0.00)	0.75	(0.01)
StepGraph (EBIC)	0.63	(0.01)	1.00	(0.00)	0.77	(0.01)

Table 7: Comparison of means and standard errors (in brackets) of m_F and m_{NKL} over $R = 50$ replicates.

	m_F		m_{NKL}	
StepGraph (CV)	0.685	(0.004)	3.771	(0.033)
StepGraph (EBIC)	0.708	(0.003)	4.011	(0.024)

Table 8: Comparison ot (CPU) times, in seconds, to estimate the precision matrix using StepGraph (CV) and StepGraph (EBIC) over $R = 50$ resplicates.

	Mean	Standard Error
StepGraph (CV)	444.50	16.83
StepGraph (EBIC)	89.59	4.26

Affiliation:

Ruben Zamar
Department of Statistics
University of British Columbia
3182 Earth Sciences Building
2207 Main Mall Vancouver, BC V6T 1Z4, Canada E-mail: ruben@stat.ubc.ca
URL: https://www.stat.ubc.ca/users/ruben-h-zamar

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/

ISSN 2773-0689
July 2021, Volume I, Issue II *Submitted:* 2020-01-13
doi:10.52933/jdssv.v1i2.11 *Accepted:* 2020-06-26