Graphical continuous Lyapunov models

Gherardo Varando Niels Richard Hansen

Abstract

We introduce a new graphical model via the linear Lyapunov equation of a covariance matrix. This model class parametrizes equilibrium covariances for stochastic processes. We show how the model class behaves under marginalization and introduce a method for structure learning via ℓ_1-penalized loss minimization. Our proposed method is demonstrated to outperform alternative structure learning algorithms in a simulation study.

1 INTRODUCTION

Path analysis as introduced by Wright (1921, 1934) illustrates how covariance computations in linear models can benefit from a graphical model representation. Today there is a vast literature on linear structural equation models and their corresponding algebraic and graphical model theory, see e.g. Drton (2018). Within this framework, the standard parametrization gives the covariance matrix Σ as a solution to the equation

$$(I - \Lambda)^T \Sigma (I - \Lambda) = \Omega \tag{1}$$

for matrix parameters Λ and Ω. The associated mixed graph has directed edges and bidirected edges determined by the non-zero entries of Λ and Ω, respectively. If we fix an acyclic graph, say, the framework provides a parametrization of the observables from a directed acyclic model potentially with latent variables, see (Richardson and Spirtes, 2002). In the cyclic case the parametrization can, moreover, be interpreted as an equilibrium distribution for a deterministic process whenever the spectrum of Λ is inside the unit circle, see e.g. (Hyttinen et al., 2012).

It is, however, well known that for certain continuous time stochastic processes the invariant covariance matrix doesn't have a simple graphical representation using the parametrization above, see e.g. (Mogensen et al., 2018). However, there is an alternative parametrization corresponding to the graphical representation of the dynamics of the process. In this parametrization, Σ is the solution to the continuous Lyapunov equation,

$$B\Sigma + \Sigma B^T + C = 0 \tag{2}$$

where B and C are matrices parametrizing Σ.

Models given by (2) are of practical interest when only cross-sectional data from the stochastic process can be obtained. This is the case for biological systems such as gene regulatory or protein signalling networks, where cells are destroyed in the measurement process. Existing methods based on correlation or mutual information, such as the ARACNe method by Basso et al. (2005), the use of directed graphical models, (Sachs et al., 2005), or the graphical lasso giving undirected graphs, (Friedman et al., 2007), cannot represent feedback processes, whereas cycles can be encoded naturally by (2).

The main objective of this paper is to introduce the novel framework of graphical models parametrized by (2) and to develop a learning algorithm of the graphical structure. In the preparation of this paper we found that similar ideas were recently considered by Young et al. (2019) and Fitch (2019). The work by Fitch (2019) is based on (2) and a learning algorithm was proposed, while Young et al. (2019) considered the vector autoregressive model, whose invariant covariance matrix solves the *discrete* Lyapunov equation. However, no substantial contribution has so far been made toward understanding the models parametrized by (2) as graphical models.

Wright's path analysis demonstrated how a mixed graph makes it possible to express entries of Σ as polynomials in the entries of Λ and Ω. Such formulas are in modern terminology known as *trek rules*, and they explain how graphical structural constraints are encoded into Σ. By introducing trek seperation, Sullivant et al. (2010) gave, for instance, a complete graph-theoretic characterization in the acyclic case of when submatrices of Σ will drop rank. Another example is the half-trek criterion for generic identifiability by Foygel et al. (2012).

In this paper we associate a mixed graph to the covariance matrix solving (2) and establish a version of trek rules when B is a stable matrix. We use this to introduce a novel graphical projection yielding a parametrization of marginalized models in terms of solutions to Lyapunov equations. To fit models parametrized by (2), but with an unknown graphical structure, we propose ℓ_1-penalized loss minimization using either the Frobenius norm or the Gaussian log-likelihood loss. They outperformed the learning algorithm proposed by Fitch (2019) in a simulation study.

2 GRAPHICAL CONTINUOUS LYAPUNOV MODELS

We will consider models of covariance matrices determined as solutions to the Lyapunov equation (2) and parametrized by the matrices B and C. We note that the solution to (2) is unique whenever the sum of any two eigenvalues of B is nonzero, in which case $\Sigma(B, C)$ will denote the solution.

To motivate (2) consider the p-dimensional Ornstein-Uhlenbeck process given as a solution to the stochastic differential equation

$$dX_t = B(X_t - a)dt + DdW_t \tag{3}$$

where B and D are $p \times p$ matrices, $a \in \mathbb{R}^p$ and W_t is a standard Brownian motion in $\mathbb{R}^p$. If B is a *stable matrix*, that is, all its eigenvalues have strictly negative real part, (3) has a Gaussian invariant distribution with covariance matrix $\Sigma(B, DD^T)$. Thus solutions of (2) arise as equilibrium covariances for continuous time stochastic processes.

Some notation and terminology is needed to study solutions of (2). Introduce $\mathrm{Mat}_0(p)$ as the $p \times p$ matrices that don't have two eigenvalues summing to zero, and let $\mathrm{Sym}(p)$ denote the symmetric $p \times p$ matrices. Let $\mathrm{Stab}(p)$ denote the set of stable $p \times p$ matrices. Obviously, $\mathrm{Stab}(p) \subseteq \mathrm{Mat}_0(p)$. The set of $p \times p$ positive definite matrices is denoted $\mathrm{PD}(p)$.

Consider a mixed graph $\mathcal{G} = ([p], E)$ with vertices $[p] = \{1, \ldots, p\}$ and with E containing directed as well as bidirected edges. Self loops are allowed. We say that a pair of matrices $(B, C) \in \mathrm{Mat}_0(p) \times \mathrm{Sym}(p)$ are compatible with the mixed graph $\mathcal{G}$ if $B_{ji} \neq 0$ implies $i \to j$ and $C_{ij} \neq 0$ implies $i \leftrightarrow j$. The set of $\mathcal{G}$-compatible matrix pairs is denoted $\Xi_{\mathcal{G}} \subseteq \mathrm{Mat}_0(p) \times \mathrm{Sym}(p)$, and $\Theta_{\mathcal{G}} = \Xi_{\mathcal{G}} \cap \mathrm{Stab}(p) \times \mathrm{PD}(p)$.

Given a mixed graph $\mathcal{G}$, the map $(B, C) \mapsto \Sigma(B, C)$ is well defined on $\Xi_{\mathcal{G}}$ with image in $\mathrm{Sym}(p)$. The restriction of this map to $\Theta_{\mathcal{G}}$ has image in $\mathrm{PD}(p)$, which follows from Proposition 2.1 below. Let $\mathcal{M}_{\mathcal{G}} = \Sigma(\Theta_{\mathcal{G}}) \subseteq \mathrm{PD}(p)$ denote the image of $\Theta_{\mathcal{G}}$, which we call the *graphical continuous Lyapunov model* (GCLM) with graph $\mathcal{G}$. The *extended* GCLM is $\mathcal{M}^e_{\mathcal{G}} = \Sigma(\Xi_{\mathcal{G}})$.

2.1 Treks

To obtain a graphical representation of $\Sigma = \Sigma(B, C) \in \mathcal{M}_{\mathcal{G}}$ for a mixed graph $\mathcal{G}$ we introduce

$$\Sigma(s) = \int_0^s e^{uB} C e^{uB^T} \mathrm{d}u. \tag{4}$$

The following is a well known result, but we state it for completeness.

Proposition 2.1. *For* $(B, C) \in \Theta_{\mathcal{G}}$

$$\Sigma(B, C) = \lim_{s \to \infty} \Sigma(s) = \int_0^\infty e^{uB} C e^{uB^T} \mathrm{d}u. \tag{5}$$

Proof. First note that stability of B ensures that the solution to the Lyapunov equation is unique. It also ensures that the integral in (5) is convergent. We see that if Σ is given by the r.h.s. of (5) then

$$\begin{aligned} B\Sigma + \Sigma B^T &= \int_0^\infty B e^{uB} C e^{uB^T} + e^{uB} C e^{uB^T} B^T \mathrm{d}u \\ &= \int_0^\infty \frac{\mathrm{d}}{\mathrm{d}u} e^{uB} C e^{uB^T} \mathrm{d}u = -C, \end{aligned}$$

which shows that Σ solves (2). □

The representation (5) implies that Σ is positive definite if C is, which shows that $\mathcal{M}_{\mathcal{G}} \subseteq \mathrm{PD}(p)$ as claimed above.

A *trek* from i to j, denoted $i \rightsquigarrow j$, is a walk of the form

$$\tau : \underbrace{i \leftarrow \cdots \leftarrow i_1}_{n(\tau)} \leftarrow k \leftrightarrow l \rightarrow \underbrace{j_1 \rightarrow \cdots \rightarrow j}_{m(\tau)}.$$

Here $n(\tau)$ and $m(\tau)$ denote the lengths of the left hand side and right hand side of the trek, respectively. Note that $n(\tau) = 0$ with $i = k$ as well as $m(\tau) = 0$ with $j = l$ are allowed. Define also

$$\kappa(s, \tau) = \frac{s^{(n(\tau)+m(\tau)+1)}}{(n(\tau) + m(\tau) + 1)n(\tau)!m(\tau)!}$$

for any trek τ and $s \in \mathbb{R}$, and introduce for $(B, C) \in \Theta_{\mathcal{G}}$ and a trek τ the *trek weight*

$$\omega(B, C, \tau) = C_{k,l} \prod_{g \rightarrow h \in \tau} B_{hg}.$$

Proposition 2.2. *For $(B, C) \in \Theta_{\mathcal{G}}$*

$$\Sigma(s)_{ij} = \sum_{\tau \in \mathcal{T}(i,j)} \kappa(s, \tau)\omega(B, C, \tau)$$

where $\mathcal{T}(i, j)$ denotes the set of all treks from i to j.

Proof. Using the series expansion of the matrix exponential we find that

$$\begin{aligned}
\Sigma(s)_{ij} &= \int_0^s \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \sum_{k,l=1}^{p} \frac{t^n t^m}{n!m!} (B^n)_{ik} C_{kl} (B^m)_{jl} \mathrm{d}t \\
&= \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \sum_{k,l=1}^{p} \frac{s^{(n+m+1)}}{(n + m + 1)n!m!} (B^n)_{ik} C_{kl} (B^m)_{jl} \\
&= \sum_{\tau \in \mathcal{T}(i,j)} \kappa(s, \tau)\omega(B, C, \tau). \qquad \square
\end{aligned}$$

The following corollary is an immediate consequence of Propositions 2.1 and 2.2.

Corollary 2.3. *If $\Sigma \in \mathcal{M}_{\mathcal{G}}$ and there is no trek from i to j in $\mathcal{G}$ then $\Sigma_{ij} = 0$.*

2.2 Marginalization

Let Σ be a $p' \times p'$ matrix that solves the Lyapunov equation for given B and C, and suppose that we only observe variables corresponding to the top left $p \times p$ block, Σ_{11}, for $p < p'$. Writing out the Lyapunov equation in block matrix form gives four coupled equations. The one corresponding to Σ_{11} is the Lyapunov equation

$$B_{11}\Sigma_{11} + \Sigma_{11}B_{11}^T + \tilde{C} = 0 \tag{6}$$

with $\tilde{C} = B_{12}\Sigma_{21} + \Sigma_{12}B_{12}^T + C_{11}$.

When C is symmetric so is $\tilde{C}$, but there is no guarantee that it is positive definite even if C is so, nor that B_{11} is stable if B is so. What we can show is that if Σ is a GCLM then Σ_{11} is an extended GCLM. To do so we will introduce a graphical projection map.

For $\mathcal{G} = ([p'], E)$ a mixed graph let $\mathcal{G}[p] = ([p], E[p])$ denote the projection onto the first $p < p'$ vertices defined as follows: for $i, j \in [p]$

- $i \to j \in E[p]$ if $i \to j \in E$.
- $i \leftrightarrow j \in E[p]$ if $i \leftrightarrow j \in E$.
- $i \leftrightarrow j \in E[p]$ if for some $k > p$ there is a trek from i to j of the forms $i \leftarrow k \rightsquigarrow j$ or $i \rightsquigarrow k \to j$.

Thus the projected graph retains all edges in $\mathcal{G}$ between vertices in $[p]$. In addition, it has bidirected edges between vertices $i, j \in [p]$ that are connected by a trek containing a vertex not in $[p]$, which is directly connected to either i or j in the trek. It should be noted that this *is not* a standard latent graph projection. For once, only bidirected arrows are added.

Proposition 2.4. *If $\Sigma \in \mathcal{M}_\mathcal{G}$ and $B_{11} \in \mathrm{Mat}_0$ then $\Sigma_{11} \in \mathcal{M}^e_{\mathcal{G}[p]}$.*

Proof. It is clear from the definitions that B_{11} fulfills the $\mathcal{G}[d]$-compatibility requirement. Observe then that

$$\tilde{C}_{ij} = C_{ij} + \sum_{k=p+1}^{p'} B_{ik}\Sigma_{kj} + \Sigma_{ik}B_{jk},$$

which is symmetric in i and j. If $C_{ij} \neq 0$ then $i \leftrightarrow j$. If $\tilde{C}_{ij} \neq 0$, but $C_{ij} = 0$, then there is a $k > d$ such that $B_{ik}\Sigma_{kj} \neq 0$ or $\Sigma_{ik}B_{jk} \neq 0$. In the first case this means that $\Sigma_{kj} \neq 0$, and by Corollary 2.3 there is a trek from k to j. Now as $B_{ik} \neq 0$ as well, we can extend the trek to the left with the edge $k \to i$, and $i \leftrightarrow j$ by the definition of $\mathcal{G}[p]$. A similar argument applies if $\Sigma_{ik}B_{jk} \neq 0$.

In conclusion, $(B_{11}, \tilde{C})$ is $\mathcal{G}[d]$-compatible, and since it is assumed that $B_{11} \in \mathrm{Mat}_0$ we have that

$$\Sigma_{11} = \Sigma(B_{11}, \tilde{C}) \in \mathcal{M}^e_{\mathcal{G}[d]}. \qquad \square$$

2.3 Example

Consider the GCLM with $\mathcal{G}$ as given by (A) in Figure 1. In this example $p = 5$ and the only bidirected edges are self loops. The specific model has

$$B = \begin{pmatrix} -1 & 1 & . & . & . \\ -1 & . & 0.2 & . & . \\ . & . & -1 & -0.5 & . \\ . & . & . & -1 & 1 \\ . & . & 1 & . & -1 \end{pmatrix}$$

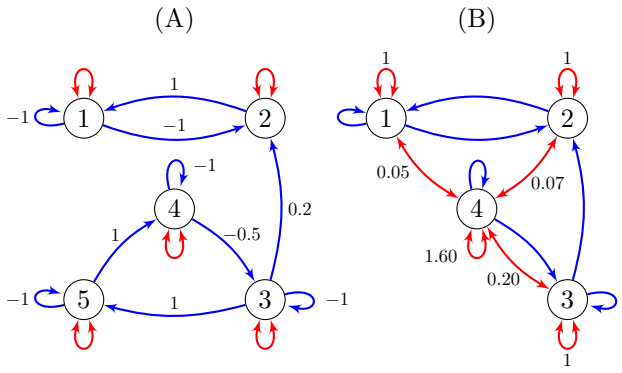

Figure 1: Mixed graphs representing a GCLM with $p = 5$ nodes (A) and the extended GCLM (B) obtained by marginalization of (A). The larger model (A) has $C = I$ and the nonzero entries of B are shown as edge weights for the directed edges. The marginalized model (B) has the same directed edge weights and the non-zero entries of $\tilde{C}$ are shown as edge weights for the bidirected edges.

and $C = I$ the identity matrix. The eigenvalues of B are

$$-1.79,\ -0.60 \pm 0.69i,\ \text{and}\ -0.50 \pm 0.87i,$$

with all real parts strictly negative, whence B is stable. The graphical projection when projecting away node 5 is shown in Figure 1 (B). Since the only edge out of 5 has a head at 4, it follows from the projection map that all added bidirected edges have a head at 4. By solving the Lyapunov equation in terms of B and C the $\tilde{C}$ matrix was computed to be

$$\tilde{C} = \begin{pmatrix} 1 & . & . & 0.05 \\ . & 1 & . & 0.07 \\ . & . & 1 & 0.20 \\ 0.05 & 0.07 & 0.20 & 1.60 \end{pmatrix}.$$

3 STRUCTURE RECOVERY

We propose minimizing an ℓ_1-penalized loss to estimate the directed part of a GCLM as given by the B matrix in 2. The C matrix will be held fixed.

Specifically, we suggest estimating B by solving the following optimization problem for a generic differentiable loss function $L : \mathrm{PD}(p) \to \mathbb{R}$:

$$\begin{array}{ll} \text{minimize} & L\left(\Sigma(B, C)\right) + \lambda \rho_1(B) \\ \text{subject to} & B \text{ stable.} \end{array} \tag{7}$$

Here, $\rho_1(B) = \sum_{i \neq j} |B_{ij}|$ is the 1-norm of the off-diagonal entries of B. Exam-

ples of loss functions are the negative Gaussian log-likelihood

$$\log\det\Sigma - \operatorname{tr}\left(\hat{\Sigma}\Sigma^{-1}\right),$$

and the squared Frobenius loss

$$\|\Sigma - \hat{\Sigma}\|_F^2 = \sum_{i,j}\left(\Sigma_{ij} - \hat{\Sigma}_{ij}\right)^2.$$

We use a variation of the proximal gradient algorithm for solving (7), see (Parikh and Boyd, 2014), even though the optimization problem is in general non-convex. The proximal operator for ℓ_1-penalization is soft-thresholding ($\mathcal{S}_t(x) = \operatorname{sign}(x)\left(|x| - t\right)$), and each iteration of the algorithm amounts to

$$B^{(k)} = \mathcal{S}_{s\lambda}\left(B^{(k-1)} - s\nabla_B L(\Sigma(B^{(k-1)}, C))\right),$$

where soft-thresholding of a matrix is defined elementwise, and the step size s is chosen using line search as in Beck and Tabulle (2010).

The gradient with respect to B can be computed as follow,

$$\nabla_B(L(\Sigma(B,C))) = 2\Sigma(B,C)\Sigma(B^t, \nabla L),$$

at the cost of computing one additional solution of a Lyapunov equation, we refer to the supplementary material for details.

Our proposed proximal gradient based algorithm is given as Algorithm 1. To reduce the computational costs, the algorithm iteratively constrains the parameter space by fixing zero entries in B (line 21).

The Lyapunov equations are solved by the Bartels-Stewart algorithm (Bartels and Stewart, 1972) as implemented in LAPACK (Anderson et al., 1999). The Bartels-Stewart algorithm consists of computing the Schur decomposition of the matrix B and then solving a simplified equation by back-substitution. Observe that to solve the additional Lyapunov equation in the gradient equation the Schur decomposition of B can be used and thus it is only computed once in each iteration (in line 11). Moreover, it is immediate to check the stability of B from the diagonal elements of its Schur canonical form. The run time complexity of one step of the Algorithm 1 is thus $\mathcal{O}(p^3)$.

3.1 Estimation for a Fixed Graph

If the initial value, B_0, in Algorithm 1 is such that $(B_0, C) \in \Theta_{\mathcal{G}}$, the algorithm will remain in $\Theta_{\mathcal{G}}$ and thus optimize over matrices B compatible with $\mathcal{G}$. If we, moreover, choose $\lambda = 0$ we obtain a gradient descent algorithm for minimizing L for a fixed graph.

3.2 Regularization Paths

As for lasso, (Friedman et al., 2010), and graphical lasso, (Friedman et al., 2007), problem (7) is to be solved for a sequence of regularization parameters

Algorithm 1 Proximal gradient algorithm for minimization of ℓ_1-penalized loss

input: $\hat{\Sigma}, C \in \mathrm{PD}(p)$, $B_0 \in \mathrm{Stab}(p)$, $M \in \mathbb{N}$,
$\varepsilon, \lambda > 0, \alpha \in (0,1)$

$B = B_0$
$s = 1$
$I = \{(i,j) \in [p] \times [p] : B_{0,ij} \neq 0\}$
$\Sigma = \Sigma(B, C)$
repeat
  $f = L(\Sigma) + \lambda\rho_1(B)$
  $D = 2\Sigma(B, C)\Sigma(B^t, \nabla L)$
  $D_{ij} = 0$ for $(i,j) \notin I$
  loop
    $B' = \mathcal{S}_{s\lambda}(B - sD)$
    $\Sigma' = \Sigma(B', C)$
    $f' = L(\Sigma') + \lambda\rho_1(B')$
    if $B' \in \mathrm{Stab}(p)$ **and**
      $f' \leq f + \frac{1}{2s}||B - B'||_F^2 - \mathrm{tr}((B - B')D)$ **then**
      break
    else
      $s = \alpha s$
    end if
  end loop
  $\delta = \frac{f - f'}{|f|}$
  $\Sigma = \Sigma', B = B', f = f'$
  $I = \{(i,j) \in [p] \times [p] : B_{ij} \neq 0\}$
until $k > M$ **or** $\delta < \varepsilon$

output: B, Σ such that $\Sigma = \Sigma(B, C)$

$\lambda_1 < \lambda_2 < \ldots < \lambda_k$. We have implemented the natural continuation algorithm where the solution B_{i-1} for $\lambda = \lambda_{i-1}$ is used as initial value of Algorithm 1 for $\lambda = \lambda_i$. Note, however, that contrary to e.g. `glmnet`, (Friedman et al., 2010), our continuation algorithm starts from a dense estimate and moves along the regularization parameters in increasing order toward sparser and sparser solutions. As Algorithm 1 greedily fixes zero entries of B, the resulting regularization path of solutions, $B_1, B_2, \ldots, B_k$, corresponds to a sequence of nested GCLMs.

3.3 The Direct Lasso Path

Fitch (2019) suggests estimating B as a sparse, approximate solution to the Lyapunov equation for Σ fixed and equal to the empirical covariance matrix, $\hat{\Sigma}$. For fixed λ the estimate is the solution to the lasso problem

$$\text{minimize} \quad \|B\hat{\Sigma} + \hat{\Sigma}B^t - C\|_F^2 + \lambda\rho_1(B). \tag{8}$$

for a fixed C. In Fitch (2019) all the entries of the B matrix are actually penalized, and not only the off-diagonal entries as in Equation (8).

The resulting *direct lasso path* for a sequence of regularization parameters can be computed easily by either coordinate descent, (Friedman et al., 2010), or least angle regression, (Efron et al., 2004).

4 SIMULATIONS

We carried out a simulation study to evaluate the performance of our proposed estimator and algorithm. The metrics used focus on recovery of the underlying oriented part of the graph. Performance was evaluated for Algorithm 1 using the Gaussian log-likelihood (`loglik`) as well as the Frobenius loss (`frobenius`). The results obtained were compared to the results for the direct lasso path (`lasso`), the graphical lasso (`glasso`) for undirected structure recovery (Friedman et al., 2007), and the simpler covariance thresholding method (`covthr`) (Sojoudi, 2016).

All metrics were computed as averages across 100 randomly generated GCLMs. Each GCLM was generated by simulating a stable matrix B with entries $B_{ij} = \omega_{ij}\varepsilon_{ij}$ for $i \neq j$ and $B_{ii} = -\sum_{j \neq i} |B_{ij}| - |\varepsilon_{ii}|$ where $\omega_{ij} \sim \text{Bernoulli}(d)$ and $\varepsilon_{ij} \sim N(0,1)$. Moreover, we generated diagonal C matrices with $C_{ii} \sim \text{Uniform}([0,1])$. Note that each such (B, C) pair has a corresponding mixed-graph $\mathcal{G}$ whose only bidirected edges are $i \leftrightarrow i$ and whose directed edges are generated independently and with uniform probability d.

We generated models of dimensions $p' = 10, \ldots, 40$ and considered the problem of recovering the directed part of the graph $\mathcal{G}[p]$ for the first $p = 10 \leq p'$ coordinates. The edge probabilities were $d = \frac{k}{p}$ with $k \in \{1, 2, 3, 4\}$. For each pair (p', k) we generated the 100 GCLMs as described above and applied the different structure recovery methods using $N = 50, \ldots, 5000$ observations from a

multivariate Gaussian distribution with covariance matrix solving the Lyapunov equation.

4.1 Details of the Compared Methods

For each method but `covthr` we obtained a solution path along a sequence of regularization parameters

$$0 = \lambda_0 < \lambda_1 < \ldots < \lambda_k,$$

with $\lambda_i = e^{\frac{10(1-k+i)}{k}}\lambda_{\max}$ for $i = 1, \ldots, k$ ($k = 100$ was used). For `loglik` and `frobenius` we used $\lambda_{\max} = 3$. For `lasso`, $\lambda_{\max}$ was the smallest penalization parameter such that the matrix B was diagonal. For `glasso`, $\lambda_{\max} = \max\{\hat{\Sigma}_{ij}\}$, resulting in a path similar to the default in the glasso R package, (Friedman et al., 2018). For covariance thresholding (`covthr`) we obtained instead a solution path by thresholding the absolute values in the sample covariance matrix at its off-diagonal entries.

In Algorithm 1 the relative convergence tolerance was $\varepsilon = 10^{-4}$, the maximum number of iterations was $M = 5000$ and $\alpha = 0.5$.

Data was standardized, which means that all methods used the empirical correlation matrix, $\hat{R}$, of the sample, and for `loglik`, `frobenius` and `lasso` we fixed C to the identity matrix. See the discussion for a perspective on this choice. Finally, Algorithm 1 was initialized with the stable and symmetric matrix $B_0 = -\frac{1}{2}\hat{R}^{-1}$ fulfilling $\hat{R} = \Sigma(B_0, I)$.

4.2 Results

Each method gives a solution path of graphs for a sequence of regularization parameters. We computed the following metrics to evaluate the methods:

- The path-wise maximum accuracy of edge recovery (`maxacc`).
- The path-wise maximum F1 score (`maxf1`).
- The area under the ROC curves (`auroc`), obtained as the true positive rate vs the false positive rate for each value of the regularization parameter.
- The area under the precision-recall curves (`aupr`), obtained as the precision vs the recall for each value of the regularization parameter.

All the above metrics were computed considering the graph recovery as a classification problem over the $p(p-1)$ off-diagonal elements of the adjacency matrix. In particular, undirected graphs obtained with the methods `glasso` and `covthr` are evaluated as directed graphs where each undirected edge is translated into the two possible directed edges.

We note that the expected number of directed edges that are not self loops in the randomly generated graphs is $dp(p-1) = k(p-1)$. Thus a graph containing only self loops leads to a baseline accuracy of $1 - \frac{k}{p}$.

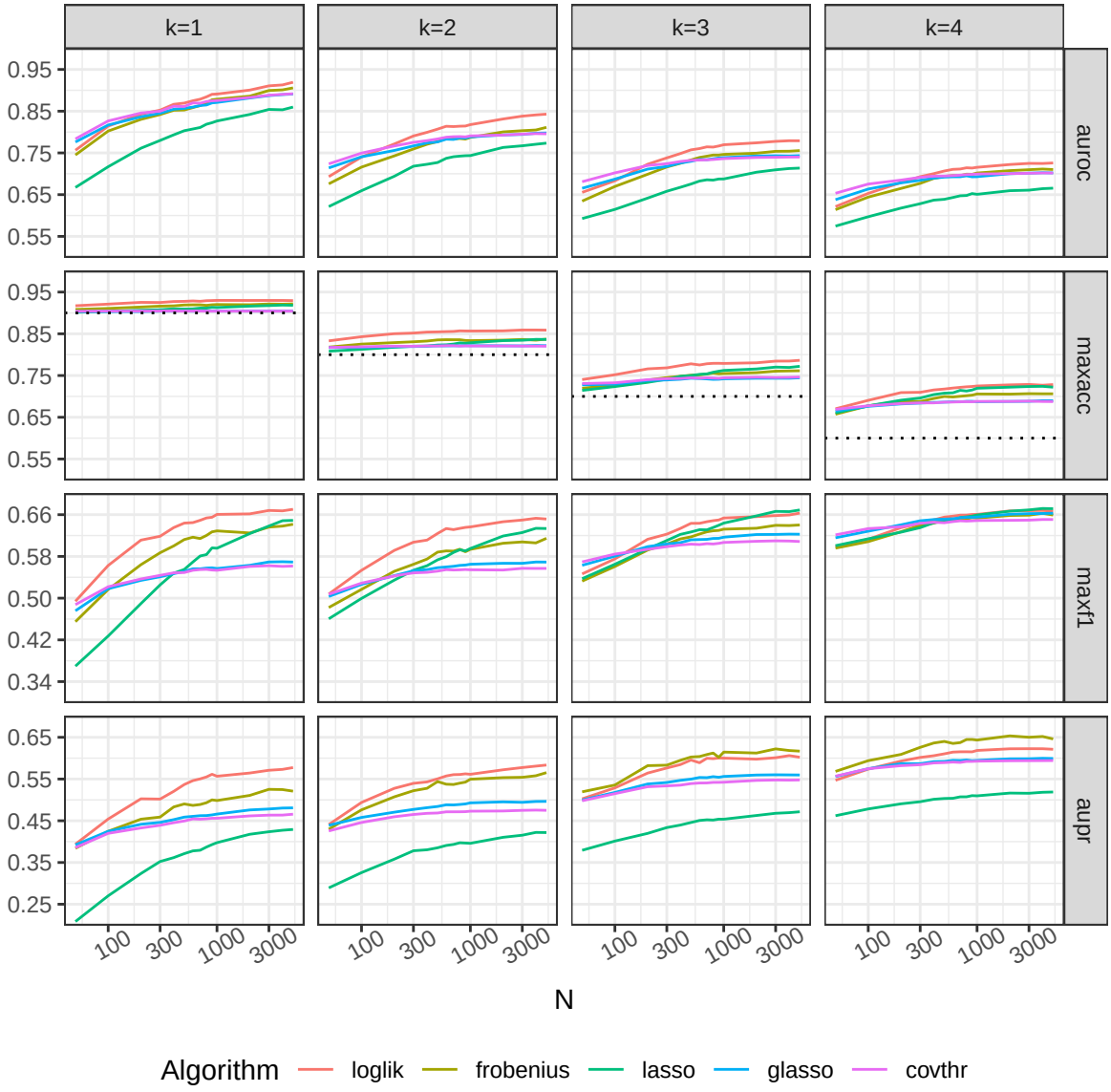

Figure 2: Average evaluation metrics (rows) as a function of the sample size (log-scale), for different edge densities (columns) and algorithms (colours). For `maxacc` the baseline accuracy $1 - \frac{k}{p}$ is indicated by a dotted line.

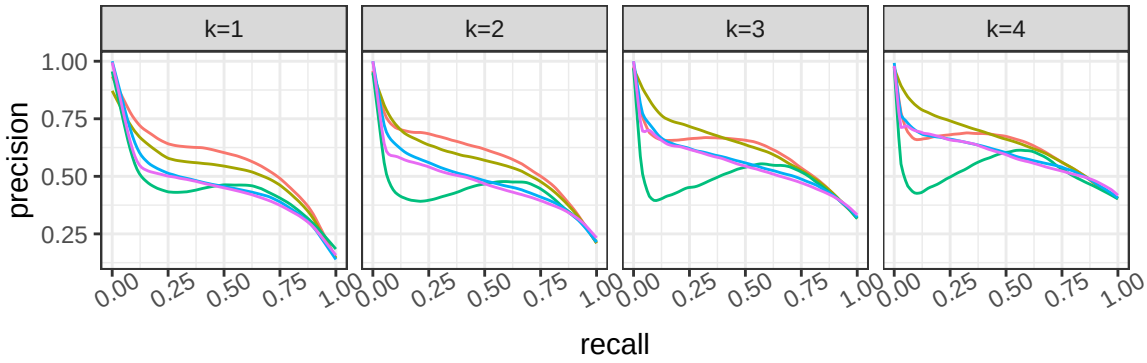

Figure 3: Average precision-recall curves for different edge densities (columns) and algorithms (colours) for $N = 3000$.

Figure 2 shows the results from the simulation experiments for $p' = p = 10$ averaged over the 100 repetitions. Figure 3 shows, in addition, the average precision-recall curves for $N = 3000$.

From Figure 2 we first observe that all methods had a path-wise maximum accuracy above the baseline. Among our proposed methods, `loglik` was slightly better than `frobenius` except in terms of precision-recall for large k. Moreover, we observe that `loglik` and `frobenius` were superior to `lasso` in the recovery of the true graph with respect to all the metrics. The direct lasso path was only competitive in terms of path-wise consistency and only for large sample sizes. Finally, `loglik` and `frobenius` also outperformed the *undirected* methods, `glasso` and `covthr` – most clearly with respect to `maxacc` and `aupr`.

Figure 4 shows `aupr` as a function of $p' \geq p$ for $N = 3000$. Additional results when data arise from a marginalization of a larger GCLM are in the supplementary material.

Again, `loglik` and `frobenius` were superior to the other methods under marginalization – especially when compared to `lasso`. Interestingly, `frobenius` clearly outperformed `loglik` for large k and p'. The performance deteriorated for all methods as a function of p'.

5 PROTEIN-SIGNALING NETWORKS

We apply the proposed method with log-likelihood loss to the flow-cytometry data in Sachs et al. (2005) containing observations of 11 phosphorylated proteins and phospholipids from 7466 cells. Data were recorded under nine different conditions consisting of different stimulatory and inhibitory interventions.

For each one of the different nine datasets we apply the following procedure, slightly inspired by stability selection methods (Meinshausen and Bhlmann,

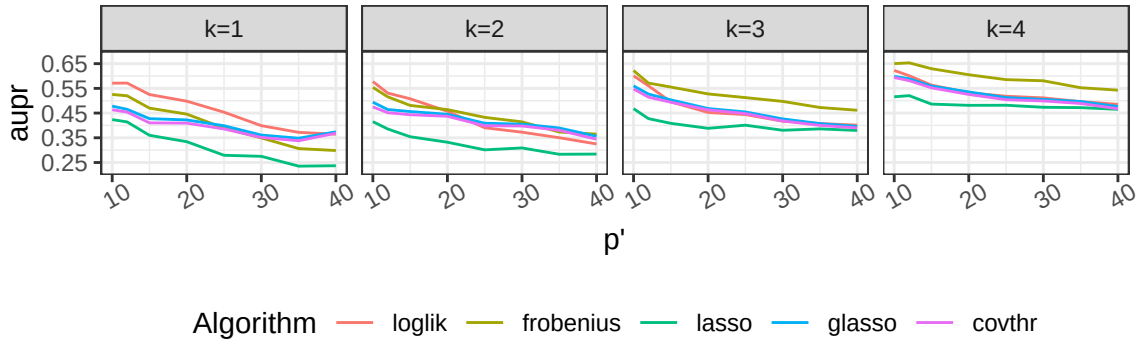

Figure 4: Average area under the precision-recall curve for different edge densities (columns) and algorithms (colours) for $N = 3000$.

2010).

1. Randomly split the observations in two subsets with the same cardinality: *Train* and *Test*.
2. Apply Algorithm 1 using the estimated correlation matrix from *Test*, to obtain the estimated B matrices along a regularization path.
3. Fit the minimum-loss estimators (using Algorithm 1 with $\lambda = 0$) for all the structures obtained in the previous point.
4. Select the structure that obtains the minimum loss with respect to the correlation matrix estimated from *Test*.
5. Repeat from point 1. and finally output the matrix with the proportion of reported edges.

In Figure 5 directed edges appearing in more of the 50% of the 200 repetitions in at least one interventional condition are reported.

We can appreciate that the method retrieves edges that are consistent with the ground truth of conventionally accepted interactions (Sachs et al., 2005; Meinshausen et al., 2016). In particular the interaction between PLC_γ, PIP2 and PIP3 molecules is appropriately obtained, apart from additional edges between PLC_γ and PIP2 or PIP3. The interactions path PKC $\rightarrow$ Raf $\rightarrow$ Mek1/2 $\rightarrow$ Erk1/2 is also obtained by our method.

6 DISCUSSION

We have presented a novel graphical model yielding a parametrization of covariance matrices via solutions of the continuous Lyapunov equation with parameter

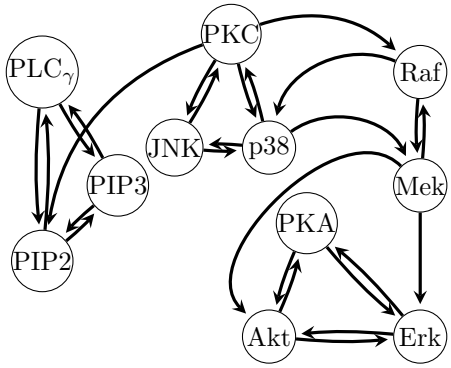

Figure 5: Estimated graph from data in Sachs et al. (2005). Self loops and bidirected edges are not plotted.

matrices (B, C) compatible with a given mixed graph. Using a trek representation and a graphical projection we showed that also marginalized models can be parametrized by the continuous Lyapunov equation.

We investigated the performance of learning the directed part of the graph via penalized loss minimization where we fixed $C = I$. A similar approach was considered by Fitch (2019). In our simulation experiment this amounts to a model misspecification even without marginalization, but in our experience attempts to estimate even a diagonal C in addition to B did not improve performance. Thus we chose the algorithmically simpler solution with fixed $C = I$. As shown in Section 2.2, marginalization may result in the C matrix being increasingly misspecified, thus the general deterioration of the performances for `loglik`, `frobenius` and `lasso` in our simulation experiment is as expected.

It was pivotal for our implementation of the proximal gradient algorithm that gradients for the loss functions could be computed as efficiently as possible. This was achieved via the representation of the Jacobian of $\Sigma(B, I)$ via Lyapunov equations as well allowing the computation of the gradient with the additional cost of solving one more Lyapunov equation. When compared to the direct lasso path as proposed by Fitch (2019), our methods are computationally more efficient, especially for large systems. Moreover, our simulation experiment showed that minimizing the ℓ_1-penalized negative log-likelihood resulted in a more efficient estimator of the directed part of the graph.

6.1 Future Directions

A direction of research is to include estimation of the C matrix corresponding to the bidirected edges of the graph. This is particularly of interest when we consider data from a marginalized model. In the supplementary material, the gradient of the loss function w.r.t. C is computed as well, and a gradient-descent algorithms could be easily implemented to estimate C – fixing B, say. However, alternating algorithms for joint estimation of B and C typically got stuck in

local minima in our experience. The optimization problem becomes highly non-convex, and there are, moreover, fundamental open problems regarding joint identifiability of the B and C parameters from Σ. It is ongoing work to provide answers to such identifiability questions and to devise algorithms that are able to jointly estimate B and C.

6.2 Reproducibility

The simulation experiment can be replicated following the instructions and running the code available at the repository *removed for anonymity*.

An R package is available from *removed for anonymity*, implementing Algorithm 1.

Acknowledgements

removed for anonymity

References

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

R. H. Bartels and G. W. Stewart. Solution of the matrix equation AX + XB = C. *Commun. ACM*, 15(9):820–826, September 1972.

Katia Basso, Adam A. Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, 37:382 – 390, 2005.

A. Beck and M. Tabulle. Gradient-based algorithms with applications to signal recovery problems. In D. Palomar and Y. Eldar, editors, *Convex Optimization in Signal Processing and Communications*, pages 42–88. Cambridge Univeristy Press, 2010.

Mathias Drton. Algebraic problems in structural equation modeling. In *The 50th Anniversary of Gröbner Bases*, pages 35–86, Tokyo, Japan, 2018. Mathematical Society of Japan.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 04 2004.

Katherine Fitch. Learning directed graphical models from Gaussian data. *arXiv:1906.08050*, 2019.

Rina Foygel, Jan Draisma, and Mathias Drton. Half-trek criterion for generic identifiability of linear structural equation models. *Ann. Statist.*, 40(3):1682–1713, 06 2012.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 12 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. *glasso: Graphical Lasso: Estimation of Gaussian Graphical Models*, 2018. R package version 1.10.

Antti Hyttinen, Frederick Eberhardt, and Patrik O. Hoyer. Learning linear cyclic causal models with latent variables. *Journal of Machine Learning Research*, 13:3387–3439, 2012.

Nicolai Meinshausen and Peter Bhlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Nicolai Meinshausen, Alain Hauser, Joris M. Mooij, Jonas Peters, Philip Versteeg, and Peter Bühlmann. Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences*, 113(27):7361–7368, 2016.

S. W. Mogensen, D. Malinsky, and N. R. Hansen. Causal learning for partially observed stochastic dynamical systems. In *Proceedings of the UAI*, 2018.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Found. Trends Optim.*, 1 (3):127–239, January 2014.

Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *Ann. Statist.*, 30(4):962–1030, 08 2002.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Somayeh Sojoudi. Equivalence of graphical lasso and thresholding for sparse graphs. *Journal of Machine Learning Research*, 17(115):1–21, 2016.

Seth Sullivant, Kelli Talaska, and Jan Draisma. Trek separation for Gaussian graphical models. *Ann. Statist.*, 38(3):1665–1685, 06 2010.

Sewall Wright. Correlation and causation. *Journal of Agricultural Research*, 20 (7):557–585, 1921.

Sewall Wright. The method of path coefficients. *Ann. Math. Statist.*, 5(3): 161–215, 09 1934.

William Chad Young, Ka Yee Yeung, and Adrian E Raftery. Identifying dynamical time series model parameters from equilibrium samples, with application to gene regulatory networks. *Statistical Modelling*, 19(4):444–465, 2019.