



Change-Point Detection for Graphical Models in the Presence of Missing Values

Malte Lonschien, Solt Kovács & Peter Bühlmann

To cite this article: Malte Lonschien, Solt Kovács & Peter Bühlmann (2021) Change-Point Detection for Graphical Models in the Presence of Missing Values, Journal of Computational and Graphical Statistics, 30:3, 768-779, DOI: [10.1080/10618600.2020.1853549](https://doi.org/10.1080/10618600.2020.1853549)

To link to this article: <https://doi.org/10.1080/10618600.2020.1853549>



Published online: 27 Jan 2021.



Submit your article to this journal [↗](#)



Article views: 346



View related articles [↗](#)



View Crossmark data [↗](#)



Change-Point Detection for Graphical Models in the Presence of Missing Values

Malte Lonschien* , Solt Kovács* , and Peter Bühlmann 

Seminar for Statistics, ETH Zürich, Zürich, Switzerland

ABSTRACT

We propose estimation methods for change points in high-dimensional covariance structures with an emphasis on challenging scenarios with missing values. We advocate three imputation like methods and investigate their implications on common losses used for change-point detection. We also discuss how model selection methods have to be adapted to the setting of incomplete data. The methods are compared in a simulation study and applied to a time series from an environmental monitoring system. An implementation of our proposals within the R-package `hdcd` is available via the online supplementary materials.

ARTICLE HISTORY

Received July 2019
Revised October 2020

KEYWORDS

Covariance estimation;
Graphical Lasso;
High-dimensional models;
Incomplete data; Precision
matrix; Time-varying models

1. Introduction

The area of high-dimensional inverse covariance (or precision) matrix estimation has developed considerably over the past years. This is partly due to the off-diagonal entries of the precision matrix encoding unscaled partial correlations and in the case of a multivariate normal distribution even conditional independence between two variables given all the others. The resulting graph, with variables as nodes and edges for nonzero off-diagonal entries is called a conditional independence graph or Gaussian graphical model (GGM), respectively (Lauritzen 1996).


In a lot of settings where the graphical structure of a set of observations is of interest, the data are in the form of a time series (or has another natural ordering, e.g., by space or genome location) and the underlying distribution might change over time. A lot of attention is given to the setting where there are structural breaks, so called change points, in between which the observations are identically distributed. Disregarding this structure and estimating a GGM for all observations leads to a fit of mixtures that does not model the true GGM well for any time point. In such cases to estimate the graphical structure, prior good estimation of the change points is key.

Real world datasets often suffer from missing values. In the presence of change points and without prior knowledge of them, usual imputation methods are not expected to perform well as they require observations from a homogeneous distribution. However, some kind of imputation method is necessary to apply any of the current methods to find change points, between which the distribution can be assumed to be homogeneous, leading to a chicken and egg kind of problem. In practice, we observe that the approach of naively


imputing mean values columnwise (i.e., for each variable separately) on the whole dataset and then estimating change points does not perform well. For example, if observations are not missing completely at random, but according to some structure (e.g., in some blockwise manner due to the simultaneous failure and repair of multiple sensors), this naive approach tends to (falsely) detect change points corresponding to the missingness pattern rather than the true changes in the underlying signal (see Section 5 for more description and simulation results regarding this naive approach). We will demonstrate that integrating imputation and change-point detection in a unified framework can lead to substantial gains.

1.1. Related Work

For homogeneous observations, Meinshausen and Bühlmann (2006) proposed to use nodewise regression using the Lasso (Tibshirani 1996) to recover the conditional independence structure from data. Yuan and Lin (2007) introduced an estimator for the precision matrix (and thus the GGM) via maximization of the L_1 -penalized Gaussian log-likelihood over the set of positive definite matrices. The graphical Lasso (glasso) algorithm of Friedman, Hastie, and Tibshirani (2008) with subsequent improvements (Witten, Friedman, and Simon 2011; Mazumder and Hastie 2012) gained popularity for computing such estimates such that the term glasso is now also associated with the corresponding estimator. Computational approaches for the estimator were also presented by Banerjee, El Ghaoui, and d'Aspremont (2008), theoretical properties of it were investigated by Yuan and Lin (2007), Rothman et al. (2008), Lam and Fan (2009), and Ravikumar et al. (2011).

CONTACT Solt Kovács  solt.kovacs@stat.math.ethz.ch  Seminar for Statistics, ETH Zürich, Raemistrasse 101, 8092 Zürich, Switzerland.

*The first two authors contributed equally to this work.

 Supplementary materials for this article are available at <https://github.com/mlonschien/hdcd>.

© 2021 American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America

For the nonhomogeneous case, Zhou, Lafferty, and Wasserman (2010) considered cases where the graph structure varies smoothly and Kolar and Xing (2011) investigated under which conditions the graph structure can be recovered consistently. In a setting with abrupt structural breaks, Kolar and Xing (2012a) proposed to estimate the locations of change points with a total variation penalty for consecutive observations either using nodewise regression or a penalized likelihood approach. Gibberd and Nelson (2017) proposed the group-fused graphical Lasso (with a Frobenius norm fusion penalty) and compared it to L_1 -fused methods that they called independent-fused graphical Lasso. Some statistical analysis of the group-fused graphical Lasso was provided by Gibberd and Roy (2017). A penalized likelihood approach was proposed by Angelosante and Giannakis (2011), seeking the corresponding optimal partitioning via dynamic programming. While Kolar and Xing (2012a) used an accelerated gradient descent method, Gibberd and Nelson (2017) proposed an ADMM algorithm to calculate the estimator, similarly to Hallac et al. (2017), who considered larger classes of fusion type penalties. Other computational approaches include an approximate majorize-minimize algorithm (Bybee and Atchadé 2018) or a specific greedy search (Hallac, Nystrup, and Boyd 2018). More thorough statistical analysis of similar proposals was conducted by Roy, Atchadé, and Michailidis (2017), Avanesov and Buzun (2018), Dette, Pan, and Yang (2020), and Wang, Yu, and Rinaldo (2021).

In the presence of missing values, the maximization of the L_1 -penalized log-likelihood (see Equation (6) later on) is no longer a convex problem, even for homogeneous observations. For this scenario of homogeneous observations with missing values, Städler and Bühlmann (2012) proposed to use an expectation maximization algorithm coupled with the glasso to obtain an estimate of the precision matrix. The single example of the treatment of missing values in the context of change-point detection we are aware was done by Xie, Huang, and Willett (2013). They however considered an online (sequential) setup and assumed that the observations lie close to a time-varying low-dimensional submanifold within the observation space. This assumption is appropriate in, for example, video surveillance, but it is unrealistic in the setting of GGMs, even if the underlying precision matrix is assumed to be sparse. In this latter setting of GGMs, there is currently no available method for change-point detection incorporating missing values. We aim to fill this gap.

1.2. Possible Applications

Applications of change-point detection within graphical models include the analysis of environmental measurements, biological data and financial time series, which potentially encounter the problem of missing values. One specific motivating example for our proposals was the shallow groundwater monitoring dataset, which we will discuss later on. To mention further concrete examples, multivariate change-point detection methods (in GGMs) could also be useful for example in detecting the signal of abrupt climate changes imprinted simultaneously into multiple climate proxy records, for example, ice cores in Antarctica or speleothems, all facing missing values (see, e.g., Atsawawanunt et al. 2018).

So far, it is common practice to discard variables with too much missingness and to perform simple imputations on the rest (see, e.g., Matteson and James 2014). Also, sometimes univariate methods are used to detect change points for each variable separately (see, e.g., Shirvani 2015). These approaches are suboptimal. Discarding noncomplete observations, especially with high-dimensional data and inhomogeneity of the missingness structure with respect to time, is impractical as it results in a significant loss in information, possibly leaving only a handful of observations. Discarding all variables with missing observations can lead to keeping only a fraction of variables. Both approaches might lead to no observations for an analysis in an extreme case.

1.3. Our Contribution

Our investigated setup is composed of three problems: change-point detection, estimation of GGMs and the treatment of missing values. While there are many possible applications, there is currently no readily available method combining the three and thus capable of estimating change points in (high-dimensional) GGMs in the presence of missing values. We fill this gap and provide practitioners with practically usable, and in particular computationally tractable methods that work both in the presence and in the absence of missing values. We provide an implementation of these methods within the R-package `hdcd`, see the supplementary materials.

We investigate different scenarios of missingness (both missing completely at random and with structures resembling real world scenarios), discuss the resulting difficulties and propose viable estimation approaches. Their performance is evaluated in a simulation study and applied to data from an environmental monitoring system.

2. Change-Point Detection Without Missing Values

Consider a sequence of independent Gaussian random variables $(X_i)_{i=1}^n \in \mathbb{R}^p$ with means μ_i and covariance matrices $\Sigma_i = \Omega_i^{-1}$ such that the map $i \mapsto (\mu_i, \Sigma_i)$ is piecewise constant. Let

$$\alpha^0 := \{0, n\} \cup \{i : (\mu_i, \Sigma_i) \neq (\mu_{i+1}, \Sigma_{i+1})\}$$

be the set of segment boundaries. We label the elements of α^0 by their natural order starting with zero such that consecutive pairs of elements in α^0 define segments $(\alpha_{k-1}^0, \alpha_k^0]$, $k = 1, \dots, |\alpha^0| - 1$ within which the X_i are i.i.d. For $0 \leq u < v \leq n$ let $X_{(u,v]}$ denote the matrix of the observations X_{u+1}, \dots, X_v , denote with $\hat{\mu}^{(u,v]}$ their mean and let $S_{(u,v]}$:= $(X_{(u,v]} - \hat{\mu}^{(u,v]})^T (X_{(u,v]} - \hat{\mu}^{(u,v]}) / (v - u)$ be the corresponding covariance matrix.

For $\delta > 0$ define $\mathcal{A}_{n,\delta}$ to be the family of possible sets of segment boundaries such that the minimal segment length is not smaller than δn . Let $L_n((u, v])$ be some loss after fitting an adequate model to $X_{(u,v]}$ that is normalized by n and scales with the segment length $v - u$ (see, e.g., Equation (4)). For a penalty parameter $\gamma > 0$ for the number of segments, an estimator for α^0 is

$$\hat{\alpha}^0 := \operatorname{argmin}_{\alpha \in \mathcal{A}_{n,\delta}} \sum_{j=1}^{|\alpha|-1} (L_n((\alpha_{j-1}, \alpha_j]) + \gamma). \tag{1}$$

This estimator can be computed using dynamic programming with $\mathcal{O}(n^2)$ evaluations of L_n , and considerably faster when certain pruning steps are applicable (typically in scenarios with many change points, see Killick, Fearnhead, and Eckley 2012). In particular without pruning steps (e.g., in scenarios with a handful of change points only), this is computationally infeasible if n is large, especially if the cost to evaluate L_n is significant.

Binary segmentation (BS, Vostrikova 1981) is a much faster greedy algorithm to estimate α^0 . For this define the gains function of some segment $(u, v]$ at some split point s to be

$$G_n^{(u,v]}(s) := L_n((u, v]) - (L_n((u, s]) + L_n((s, v])) \quad (2)$$

and define

$$\hat{\alpha}_{(u,v]} := \operatorname{argmax}_{s \in \{u+\delta n, \dots, v-\delta n\}} G_n^{(u,v]}(s). \quad (3)$$

The search for a single change-point in $(X_i)_{i=1}^n$ by solving (1) breaks down to finding $\hat{\alpha}_{(0,n]}$ and checking if $G_n^{(0,n]}(\hat{\alpha}_{(0,n]}) > \gamma$. For the multiple change-point case BS finds an approximate solution to (1) by recursively splitting segments using (3) until the resulting segment length is smaller than $2\delta n$ such that a split is no longer allowed or the corresponding gain is not bigger than γ , the minimally required gain to split. BS typically requires asymptotically $\mathcal{O}(n \log(n))$ evaluations of L_n . Due to its greedy nature, BS is not optimal in terms of statistical estimation (as opposed to computing the optimal partitioning from (1) using dynamic programming). Wild binary segmentation (WBS, Fryzlewicz 2014) as well as the recently proposed seeded binary segmentation (SeedBS) method of Kovács et al. (2020a) improve on statistical detection by evaluating the gains function (2) for various random (for WBS) or deterministically constructed (for SeedBS) background intervals, each leading to a candidate split point via (3). Out of the list of candidates the final set of change-point estimates is subsequently derived. The improved detection is due to the fact that some of the generated background intervals only contain a single change-point and in that case the detection is easier. While WBS loses some of the computational efficiency of plain BS (e.g., in scenarios with many change points or very short spacing between some change points when one needs to draw a large number of random intervals), the more efficient deterministic interval construction of SeedBS requires only $\mathcal{O}(n \log(n))$ evaluations of L_n independent of the number of change points and thus SeedBS is similarly fast as plain BS.

Nonetheless, $\mathcal{O}(n \log(n))$ evaluations of L_n for BS can still be prohibitive if the cost of evaluating L_n is large (e.g., for high-dimensional model fits) and n is big. Instead of evaluating $G_n^{(u,v]}(s)$ at every possible split in a full grid search to find its maximum, one can find (up to very few exceptions with special signal cancellation effects) one of its local maxima with adaptively chosen $\log(n)$ evaluations using the optimistic search strategies of Kovács et al. (2020b). The expected gain curve for common losses (e.g., the squared error loss) is piecewise convex between the true underlying segment boundaries, such that in particular all local maxima correspond to change points. Hence, splitting at a local maximum instead of the global maximum, one does not induce a false discovery and the missed global maximum can still be found in a later step. Doing BS with this optimistic search is called optimistic binary segmentation (OBS,

Kovács et al. 2020b) and approximately requires $\mathcal{O}(|\alpha^0| \log(n))$ evaluations of L_n . Note that we only use the so-called naive variant of optimistic search and thus the naive variant of OBS in this article (e.g., simulation results). In practice, the observed gain curve (resulting from a single draw) is a noisy version of the expected one. As a consequence, the idealized piecewise convex structure of the expected gain curve is distorted and only conserved approximately (in terms of its rough shape). In very noisy scenarios the optimistic search can get stuck in a noise induced local maximum that is far away from one corresponding to a change-point. Optimism is thus needed in noisy scenarios. In contrast to optimistic search (used in OBS), the full grid search (used in BS) finds the global maximum of the noisy observed gain curve, which is typically closer to one of the true underlying change points (provided of course that there is a sufficiently large jump in the underlying signal). We will see later on how the noisiness of the observed gain curve influences the applicability of OBS compared to BS in scenarios with lots of missing data.

Since the writing of this article, Kovács et al. (2020b) proposed two new variants, the advanced and combined optimistic search. These have improved performance and stronger theoretical results for challenging scenarios with change points close to the segment boundaries. Kovács et al. (2020b) also describe optimistic seeded binary segmentation (OSeedBS), a combination of seeded binary segmentation and optimistic search. OSeedBS is typically somewhat more costly than OBS, but as advantages, OSeedBS would be easy to parallelize and also has stronger statistical guarantees than BS (and thus also OBS). These more recent search methods could be combined with our later described methodology for detecting change points for GGMs and to increase flexibility, we added some of them to our R-package `hdcd`.

Leonardi and Bühlmann (2016) applied BS to a high-dimensional linear regression change-point problem using the negative log-likelihood for Gaussian errors (sum of squared errors) resulting from a Lasso fit as a loss measure. They provided (under technical conditions) consistency results if the penalization parameter λ for the Lasso is adjusted by the inverse of the square root of the relative segment length, that is, by using $\sqrt{n/(v-u)}\lambda_0$ for some fixed λ_0 for the segment $(u, v]$.

To adapt this approach to the multivariate normal case, we set

$$L_n(\Omega; S_{(u,v]}) := \frac{v-u}{n} (\operatorname{Tr}(\Omega^T S_{(u,v]}) - \log(|\Omega|)). \quad (4)$$

This way $L_n(\Omega; S_{(u,v]})$ is (up to a constant) the negative log-likelihood of a Gaussian with precision matrix Ω given observations $X_{(u,v]}$ that is scaled with segment length. As L_n scales with $(v-u)/n$, we set $\lambda_{(u,v]} := \sqrt{(v-u)/n}\lambda_0$ and define the estimator

$$\begin{aligned} \hat{\Omega}_{(u,v]}^{\text{lasso}} &:= \operatorname{argmin}_{\mathbb{R}^{p \times p} \ni \Omega \succ 0} L_n(\Omega; S_{(u,v]}) + \lambda_{(u,v]} \|\Omega\|_1 \\ &= \operatorname{argmin}_{\mathbb{R}^{p \times p} \ni \Omega \succ 0} \operatorname{Tr}(\Omega^T S_{(u,v]}) - \log(|\Omega|) \\ &\quad + \sqrt{n/(v-u)}\lambda_0 \|\Omega\|_1. \end{aligned} \quad (5)$$

Hence, $\hat{\Omega}_{(u,v]}^{\text{lasso}}$ is the glasso estimator from Friedman, Hastie, and Tibshirani (2008) for the precision matrix of observations $X_{(u,v]}$ with penalization parameter $\sqrt{n/(v-u)}\lambda_0$. We

thus replicate the scaling approach by Leonardi and Bühlmann (2016), but use a different notation more suited to our GGM setup. We prefer not to penalize the diagonal in $\|\Omega\|_1$ and then use the in sample loss

$$L_n((u, v]) := L_n(\hat{\Omega}_{(u,v]}^{\text{glasso}}; S_{(u,v]})$$

after fitting the glasso in Equation (2) as the loss when greedily searching for optimal splits with BS or OBS.

We choose the graphical Lasso for the estimation of the precision matrix since some sparsity assumption seems crucial in high-dimensional scenarios. Note that Avanesov and Buzun (2018) also relied on the graphical Lasso (or similar procedures for sparse precision matrices) and Roy, Atchadé, and Michailidis (2017) also had a sparsity assumption in their change-point detection proposals. Additionally, the change-point detection is expected to be more powerful and reasonable when placing the sparsity assumption on the precision matrix rather than the covariance matrix, in particular when considering sparse changes between conditional dependencies among a few variables. For low-dimensional scenarios nonsparse (e.g., ridge type) estimators could also be reasonable, but differences in estimated change-point locations compared to our glasso based approach are expected to be small.

Overall the algorithms have three tuning parameters: δ for the minimal relative segment length, γ to control for the number of segments and λ_0 for the sparsity of the precision matrices. Note that the procedure is similar to the one proposed by Angelosante and Giannakis (2011), but as a crucial difference their penalty is not scaled according to the segment length with $\lambda_{(u,v]}$, and they include the penalty term into the gains function, which seems unnatural.

3. Adapting to Missing Data

For noncomplete data denote with $x_{\text{obs},i}$ the observed part of the i th observation and with $\Omega_{\text{obs},i}$ and $\mu_{\text{obs},i}$ the submatrix of Ω and the subvector of μ corresponding to the observed variables of x_i . Similarly denote with $\mu_{\text{mis},i}$ the subvector of μ corresponding to the variables of x_i with missing values. Note that the observed components can differ across observations x_i . The normalized negative Gaussian log-likelihood of observations from a segment $(u, v]$ for a given mean μ and precision matrix Ω is

$$\ell(\mu, \Omega; (x_i)_{i=u+1}^v) = \frac{1}{2n} \sum_{i=u+1}^v \left((x_{\text{obs},i} - \mu_{\text{obs},i})^T \Omega_{\text{obs},i} (x_{\text{obs},i} - \mu_{\text{obs},i}) - \log(|2\pi \Omega_{\text{obs},i}|) \right), \tag{6}$$

treating observations with missing values as if they arose from a lower-dimensional Gaussian (by considering only the log-likelihood corresponding to the observed parts, that is, submatrices of Ω and subvectors of μ). Set $\hat{\mu}^{(u,v]}$ to be the empirical mean of the observed part of $X_{(u,v]}$ (discarding the missing values). Note that $\hat{\mu}^{(u,v]}$ might have missing values itself if for some variable there is no observed value in the segment $(u, v]$. To simplify notation in terms of the mean vector, by plugging $\hat{\mu}^{(u,v]}$ into (6), let

$$\ell(\Omega; (x_i)_{i=u+1}^v) := \ell(\hat{\mu}^{(u,v]}, \Omega; (x_i)_{i=u+1}^v), \tag{7}$$

still evaluating the log-likelihood according to the observed parts. The L_1 -penalized maximum likelihood estimator for observations $X_{(u,v]}$ is then

$$\hat{\Omega}_{(u,v]} = \underset{\mathbb{R}^{p \times p} \ni \Omega \succ 0}{\text{argmin}} \ell(\Omega; (x_i)_{i=u+1}^v) + \lambda_{(u,v]} \frac{1}{v-u} \sum_{i=u+1}^v \|\Omega_{\text{obs},i}\|_1.$$

Contrary to (5), this problem is no longer convex and cannot be solved efficiently with an update-based approach like the glasso. The Miss-Glasso algorithm proposed by Städler and Bühlmann (2012) combines the glasso and an expectation maximization (EM) algorithm to estimate the precision matrix in the presence of missing data. However, the algorithm needs complete observations for a good initialization, is computationally expensive due to a new glasso fit for each EM iteration and might get stuck in local optima. These features are especially critical in our setting of high-dimensional change-point detection. High computational cost is prohibitive since we do a lot of evaluations of the loss function. More importantly even, for each split, Miss-Glasso would be initialized slightly differently and in some situations it could converge to a very different local optimum. This would result in jumps in the gain curve from Equation (2) for some neighboring splits s .

While an accurate estimate $\hat{\Omega}_{(u,v]}$ (as aimed e.g., by Miss-Glasso) is necessary for a good fit, this is not needed for change-point detection. It is sufficient to roughly approximate the piecewise convex shape of the (expected) gains function (2), as this makes sure that pronounced local maxima lie in a neighborhood of the true underlying change points. This key new idea helps to avoid the heavy computations and local optima of the EM algorithm when doing BS or OBS. In the following, we propose estimators $\tilde{S}_{(u,v]}$ that approximate $S_{(u,v]}$ based on $(x_{\text{obs},i})_{i=u+1}^v$. We can then use $\tilde{S}_{(u,v]}$ in the glasso estimator (5) to obtain $\tilde{\Omega}_{(u,v]}^{\text{glasso}}$ instead of $\hat{\Omega}_{(u,v]}^{\text{glasso}}$ and use the resulting log-likelihood $\ell(\tilde{\Omega}_{(u,v]}^{\text{glasso}}; (x_i)_{i=u+1}^v)$ as in Equation (7) with minor modifications (see Section 3.2) as a loss measure.

3.1. Missing Value Imputations

We propose three different estimators $\tilde{S}_{(u,v]}$ for data with missing values. These will lead to different results, computational costs and applicabilities.

3.1.1. The Average Imputation Estimator

In a first attempt, we impute the missing values $(x_{\text{mis},i})_{i=u+1}^v$ with the average value of the corresponding variables within the interval $(u, v]$. Thus define centered variables

$$z_{\text{obs},i}^{(u,v]} := x_{\text{obs},i} - \hat{\mu}_{\text{obs},i}^{(u,v]} \text{ and } \hat{z}_i^{(u,v]} := (\hat{z}_{\text{obs},i}^{(u,v]}, 0).$$

The latter notation indicates that the centered vector $\hat{z}^{(u,v]}$ is padded with zeros at the appropriate positions of missing entries. Then set

$$\tilde{S}_{(u,v]}^{\text{av}} := \frac{1}{v-u} (\hat{z}_{u+1}^{(u,v]}, \dots, \hat{z}_v^{(u,v]})^T (\hat{z}_{u+1}^{(u,v]}, \dots, \hat{z}_v^{(u,v]})$$

as the average imputation estimator of the in-segment covariance matrix. This will underestimate the variance and covariance of variables where values are missing. Nonetheless the average method serves as a baseline.

3.1.2. The Loh–Wainwright Bias Corrected Estimator

We try to counteract the shortcomings of the average imputation method using the bias correction presented by Loh and Wainwright (2012). The authors showed that if the j th variable of an observation in $X_{(u,v]}$ is discarded with probability ρ_j , setting

$$M_{i,j} := \begin{cases} \frac{1}{(1-\rho_i)(1-\rho_j)} & i \neq j, \\ \frac{1}{1-\rho_i} & i = j, \end{cases}$$

the matrix $\tilde{S}_{(u,v]}^{\text{av}} \circ M$ is an unbiased estimator of $S_{(u,v]}$. Here, \circ denotes the Hadamard (pointwise) product for matrices. In practice, we have to estimate the ρ_i based on the proportion of missing observations, leading to \hat{M} . Note that the resulting matrix $\tilde{S}_{(u,v]}^{\text{av}} \circ \hat{M}$ is not necessary positive semidefinite which is required for the glasso algorithm (Friedman, Hastie, and Tibshirani 2008) to converge. We thus compute the closest positive semidefinite matrix to $\tilde{S}_{(u,v]}^{\text{av}} \circ \hat{M}$ with respect to the Frobenius norm using the Higham algorithm (Higham 2002) and take this as our final Loh–Wainwright (LW) bias corrected estimator $\tilde{S}_{(u,v]}^{\text{LW}}$. Note that the Higham algorithm has an asymptotic complexity of $\mathcal{O}(p^3)$ similar to the glasso and thus evaluating the gain using the LW estimator is computationally more expensive than the average imputation method.

Alternatively, one could plug in $\tilde{S}_{(u,v]}^{\text{av}} \circ \hat{M}$ as the sample covariance matrix into Equation (5), similar to what has been suggested by Kolar and Xing (2012b) as an alternative to the Miss-Glasso procedure of Städler and Bühlmann (2012) in scenarios with homogeneous distributions with missing data. While one can skip the calculation of the nearest positive definite matrix this way, the problem is that not all algorithms that have been developed to solve estimation problems from Equation (5) work with input matrices that are not positive semidefinite. In particular, as mentioned previously, the glasso algorithm itself would not work either. As we intend to rely on standard algorithms and corresponding software that are fast (as several repeated fits are necessary for change-point detection), we do not further pursue this alternative.

3.1.3. Pairwise Covariance Estimation

A third estimate can be based on pairwise covariance estimates, where the covariance between two variables is calculated from observations where both corresponding variables are available. Similar to the LW estimator, this gives a valid estimate of the variances and covariances of variables, even if the missingness structure in the data is not homogeneous. If there are less than two complete observations for a pair of variables in a segment, we set the covariance between these variables to be zero. Again, as with the LW approach, this might yield a matrix that is not positive semidefinite. We hence apply the Higham algorithm to compute the closest positive semidefinite matrix and denote the resulting estimator by $\tilde{S}_{(u,v]}^{\text{pair}}$.

3.2. Avoiding Jumps in the Gain Curve

For all of the three proposals above, the estimation of the variance of a variable on a given segment requires at least two

observations with nonmissing values. To obtain a meaningful estimate, more nonmissing observations are necessary. As a consequence, with a lot of missing values or small segments, we might only be able to estimate a submatrix of the full covariance matrix. For such segments, the log-likelihood can then only be computed for a submodel of the entire multivariate Gaussian distribution. When evaluating the gains function (2) at some split point s , it might thus happen that the log-likelihood of the segment $(u, v]$ is calculated based on a larger covariance matrix (and thus a model with more parameters) than for $(u, s]$ or $(s, v]$. As the log-likelihoods of multivariate Gaussians of different dimensions are not easily comparable, this is especially problematic for split points s such that the estimated covariance matrix for the neighboring split $s + 1$ has a different size. In such scenarios, the gains curve often has jumps between s and $s + 1$ (see Figure 2). To alleviate this, we propose to restrict the log-likelihood of $(u, v]$ to the dimensions available for $(u, s]$ and $(s, v]$. To make this more concrete, let us introduce some notation.

Denote for a segment $(u, v]$ and some $k \geq 2$ with $j_{(u,v]}(k)$ the indices of the variables for which at least k values are observed in the segment $(u, v]$. For any imputation method $* \in \{\text{av}, \text{LW}, \text{pair}\}$ let $\tilde{S}_{(u,v]}^{*,k} := (\tilde{S}_{(u,v]}^*)_{j_{(u,v]}(k), j_{(u,v]}(k)}$ be the submatrix of $\tilde{S}_{(u,v]}^*$ where each variable has at least k observed values. Finally denote with $\tilde{\Omega}_{(u,v],k}^{\text{glasso}} = \tilde{\Omega}_{(u,v],k}^{\text{glasso}}(\tilde{S}_{(u,v]}^{*,k})$ the obtained glasso fit for the submatrix. In our simulations, we set the minimal number of required observations for keeping a variable to be $k = 5$.

A naive estimator for the gains function is

$$\begin{aligned} \tilde{G}_{n,\text{naive}}^{(u,v],k}(s) &:= \ell(\tilde{\Omega}_{(u,v],k}^{\text{glasso}}, (x_{i,j_{(u,v]}(k)}^v)_{i=u+1}^v) \\ &\quad - \ell(\tilde{\Omega}_{(u,s],k}^{\text{glasso}}, (x_{i,j_{(u,s]}(k)}^s)_{i=u+1}^s) \\ &\quad - \ell(\tilde{\Omega}_{(s,v],k}^{\text{glasso}}, (x_{i,j_{(s,v]}(k)}^v)_{i=s+1}^v). \end{aligned}$$

This naive estimator might be comparing log-likelihoods of multivariate normal distributions of different dimensions at some possible split points s if $j_{(u,s]}(k) \subsetneq j_{(u,v]}(k)$ or $j_{(s,v]}(k) \subsetneq j_{(u,v]}(k)$. See Figure 1 for an illustrative example. Here, $j_{(u,s]}(2) = \{1, 2, 3\}$, $j_{(s,v]}(2) = \{2, 3, 4\}$, and $j_{(u,v]}(2) = \{1, 2, 3, 4\}$. The log-likelihood $\ell(\tilde{\Omega}_{(u,v],2}^{\text{glasso}}, (x_{i,j_{(u,v]}(2)}^v)_{i=u+1}^v)$ of the full segment would be using the first variable of the $(s + 1)$ st observation for evaluation, whereas the log-likelihood $\ell(\tilde{\Omega}_{(s,v],2}^{\text{glasso}}, (x_{i,j_{(s,v]}(2)}^v)_{i=s+1}^v)$ of the segment on the right would not.

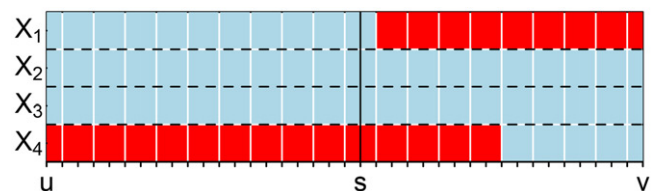


Figure 1. An illustrative example of a segment with 38 observations of dimension 4. Red blocks correspond to missing values and the black vertical line to a possible split.

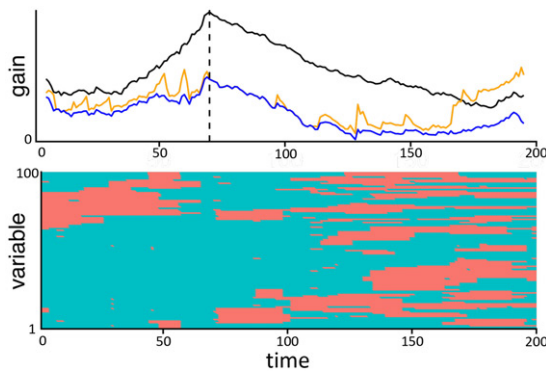


Figure 2. A scenario of simulated blockwise missing data (bottom). Red parts correspond to missing (i.e., deleted) values in the data matrix. The top shows the recovered gains if the full data without missing values is used (black), as well as in the case of missing values our proposed estimator using the LW method (blue), the naive version (orange) exhibiting jumps, and the location of the true underlying change-point (vertical dashed line).

To avoid this, we propose to use the slightly different estimator

$$\begin{aligned} \tilde{G}_n^{(u,v],k}(s) &:= \ell(\tilde{\Sigma}_{(u,v],k}^{\text{glasso}}(x_{i,j(u,s]}(k))_{i=u+1}^s) \\ &+ \ell(\tilde{\Sigma}_{(u,v],k}^{\text{glasso}}(x_{i,j(s,v]}(k))_{i=s+1}^v) \\ &- \ell(\tilde{\Sigma}_{(u,s],k}^{\text{glasso}}(x_{i,j(u,s]}(k))_{i=u+1}^s) \\ &- \ell(\tilde{\Sigma}_{(s,v],k}^{\text{glasso}}(x_{i,j(s,v]}(k))_{i=s+1}^v). \end{aligned}$$

Here, we only use the variables $j(u,s](k)$ and $j(s,v](k)$ in the calculation for the loss of the full segment $(u, v]$ when splitting at s . Hence, slightly different losses are used for the full segment depending on the split point. As we are not primarily interested in a segments' own loss, but rather a fair estimate of the gain, this is a sensible approach.

This does not increase the computational cost significantly, as for repeated evaluations of $\tilde{G}_n^{(u,v],k}$ we can keep the initial estimate $\tilde{\Sigma}_{(u,v]}^{\text{glasso}}$. Without this mechanism the gain curve exhibits jumps at the boundaries of missing blocks, as illustrated in Figure 2. This might prevent BS and OBS from correctly estimating the change points. The severity of this problem depends on the block size. When the missingness structure is homogeneous in time, the block size is small such that this is not a pronounced issue. However, in many applications one faces blockwise missing data, where such jumps would be a pronounced issue without our proposal.

4. Model Selection

A good choice of tuning parameters is essential for accurate estimation results. The parameter λ_0 controls the form of the gain curve. When chosen too small, the glasso tends to overfit and the resulting gain curve has an inverse U shape independently of the underlying change points. When chosen too big, the glasso underfits, resulting in an almost constant gain curve. Even though we adjust the penalization depending on the segment length ($\lambda_{(u,v]} = \sqrt{(v-u)/n}\lambda_0$ as discussed in Section 2), one global λ_0 might not be able to approximate the shape (piecewise convex structure) of the population version of the gain curve

simultaneously in all possible segments encountered during BS or OBS. If the sparsity pattern of the underlying graphical model in the segments differs strongly, selecting a new λ_0 in each splitting step of BS or OBS is advocated to obtain good results. We defer the details on how to locally choose the tuning parameter λ_0 (which we recommend using in practice) to the end of this section.

The parameters γ (the penalty for the number of segments) and δ (the minimum relative segment size) on the other hand control the depth of the tree structure generated by BS and thus how many change points are found. A sufficiently large value of δ is also necessary to achieve stability of fits in high-dimensional scenarios. Often, overly small segments are uninterpretable and thus uninteresting for practitioners, such that δ can be set to some predetermined value (typically around 0.1). Thus, γ is the key parameter to be chosen to avoid under- or oversegmentation. Leonardi and Bühlmann (2016) proposed to choose values for λ_0 and γ via 2-fold cross-validation, where the test data is taken from an equispaced grid across the entire sample (in this case every second observation). Since change points for γ can be regained from BS trees grown with smaller $\gamma' < \gamma$, it is only necessary to do one BS fit (with $\gamma = 0$) per fold and λ_0 .

This approach did not yield satisfactory results in our settings even when only a small amount of values were missing. Often the value chosen for γ would correspond to the correct segmentation of the test data, but would underfit on the whole data. This could be explained by the fact that while our covariance estimation methods approximately preserve the structure of the gain curve, they do not reliably preserve its magnitude and thus γ might be incomparable between folds and different segments.

To eliminate the previously mentioned issues, we propose an alternative method both as a stopping criterion for splits as well as for choosing λ_0 in practice. This approach performed empirically much better than the one that was used by Leonardi and Bühlmann (2016) and has the strong computational advantage of only requiring one single BS or OBS fit. For each investigated segment $(u, v]$ we first apply 10-fold cross-validation, taking the test data from an equispaced grid in $(u, v]$ (every 10th observation in this case) to obtain an optimal value $\hat{\lambda}_0((u, v])$ corresponding to the minimal attained cross-validated loss, which we denote by $l_{(u,v]}(\hat{\lambda}_0((u, v]))$. The loss that is minimized is the negative log-likelihood of the test data given the mean and the estimated precision matrix of the train data. We then use $\hat{\lambda}_0((u, v])$ for the evaluation of the gain curve for that segment. In the standard setting, one would then check if $G_n^{(u,v]}(\hat{\alpha}_{(u,v]}) > \gamma$ to decide whether to split further at the found point $\hat{\alpha}_{(u,v]}$ or not. Instead, we compare the cross-validated minimal loss on the full segment $(u, v]$ to the sum of cross-validated minimal losses of the subsegments $(u, \hat{\alpha}_{(u,v]})$ and $(\hat{\alpha}_{(u,v]}, v]$. We keep the split if there is a positive improvement, that is, if

$$\begin{aligned} l_{(u,v]}(\hat{\lambda}_0((u, v])) - l_{(u,\hat{\alpha}_{(u,v]})}(\hat{\lambda}_0((u, \hat{\alpha}_{(u,v]}))) \\ - l_{(\hat{\alpha}_{(u,v]},v]}(\hat{\lambda}_0((\hat{\alpha}_{(u,v]}, v]))) > 0. \end{aligned}$$

Hence, in the decision rule for keeping a candidate, γ is essentially removed and the decision to keep or discard a segment is just based on cross-validated (out of sample) losses (in particular, whether they indicate any improvement). Note that our approach is not a proper cross-validation technique

as the above described improvement at a candidate split $\hat{\alpha}_{(u,v)}$ is evaluated on the same data as was used to find the split point. Hence, this procedure might be optimistic regarding the improvements and thus slightly biased toward finding too many change points. This would not be a big problem in practice, as finding too many change points is preferred to finding too few. In Section 5, we investigate how our stopping criterion performs on simulated data both with and without change points. Contrary to expectations it does not tend to oversegment in scenarios without change points.

5. Simulations

It seems to be hard to provide theoretical guarantees in the setting of change-point detection in GGMs with missing values. We think that the only semirealistic case to provide theory is the one with values missing completely at random, which seems to be far from realistic for most applications (see examples in Section 6). Moreover, keeping our tuning parameter λ_0 fixed across all possible segments $(u, v]$ would be one of our technical assumptions, which however is clearly suboptimal for real and simulated data (as discussed in Section 4). We thus focus on the practical performance of our methods, which we demonstrate with the following simulations.

5.1. Setup

We first discuss how we generated data, how we deleted values and we introduce the performance measures used to evaluate the results.

5.1.1. Generating Covariance Matrices

In our simulations, we consider two methods to draw precision matrices for the segments. The first one is the random graph model (Erdős and Rényi 1960), which was used to simulate high-dimensional graphs (e.g., by Kolar et al. 2010). Here, the graph is generated by connecting nodes randomly with some probability $q > 0$, which we set to $\frac{5}{p}$ in the following to ensure sufficient sparsity. We create the corresponding precision matrix by assigning a constant value (taken here as 0.3) to the entries corresponding to the chosen edges and then adding the absolute value of the smallest eigenvalue of the resulting matrix plus some increment (here 0.1) to the diagonal. This is necessary to ensure positive definiteness of the constructed precision matrix.

We used chain networks as a second model (see, e.g., Fan, Feng, and Wu 2009, Example 4.1). Here, we set $\Sigma_{ij} := \exp(-a|s_i - s_j|)$, where $a > 0$, $s_1 < \dots < s_p$ and $s_i - s_{i-1} \sim \text{Unif}(0.5, 1)$ for $i = 2, \dots, p$. In the simulations, we set $a = 1/2$ and additionally draw $s_1 \sim \text{Unif}(0.5, 1)$. The inverse of the resulting matrix is tridiagonal. To generate precision matrices with differing sparsity patterns for different segments, we draw some permutation π of $1, \dots, p$ and set $\Sigma_{ij} = \exp(-a|s_{\pi(i)} - s_{\pi(j)}|)$. This breaks the tridiagonal structure but keeps the sparsity.

5.1.2. Types of Missingness

We consider two ways to delete values for the simulations. The first one is missing completely at random (MCAR), where a

given percentage of values is discarded uniformly at random. The second one is inspired by the missingness structure of environmental monitoring data (see bottom of Figure 6). Here, a failure of a sensor leads to missing values over several consecutive observations, while replacement of them at multiple sampling locations might occur at the same time. Moreover, it is common that simultaneously several sites are newly installed or abandoned based on the available budget. This leads to blocks of observations missing, ranging over multiple variables as well as observations. To generate a similar blockwise missingness structure, we repeatedly select $k \sim \text{Poi}(\frac{p}{20})$ variables uniformly at random and delete for all of them a segment of length $l \sim \text{Exp}(\frac{n}{8})$ with the midpoint chosen uniformly between 1 and n . We repeat this procedure until the preset percentage of missing values is reached. An example with 30% missing values for $n = 200$ and $p = 100$ is shown at the bottom of Figure 2.

5.1.3. Performance Measures

We use the adjusted Rand index (Hubert and Arabie 1985), a common measure to compare clusterings, to measure performance regarding change-point detection in our simulation study (see Table 1). Given two partitions of n observations, the Rand index (Rand 1971) is the number of agreements (pairs of observations that are either in the same subset for both partitions or are in different subsets for both partitions) divided by the total number of pairs $\binom{n}{2}$. The adjusted Rand index is the difference between the Rand index and its expectation when choosing partitions randomly, normalized by the difference between the maximum possible Rand index and its expectation. The adjusted Rand index is thus bounded by one and is expected to be zero when partitions are chosen randomly. We illustrate the estimation uncertainty of found change points corresponding to some adjusted Rand measures (taking true and estimated segments as the two partitions) via histograms in Figure 4.

An interesting point includes also the accuracy in recovering the precision matrices (or the underlying graphical models). The reasons why we only focus on the recovery of change points and do not report performance measures related to graph recovery are the following. We tuned our algorithm specifically for good detection of the change points and not necessarily for good estimation of the precision matrices (see the discussion in Section 3), because good recovery of the graph structure is impossible without precisely knowing the change-point locations, partly due to the precision matrix of the mixture distribution not being sparse in many cases. Of course, once change points are accurately localized, one can use the most favored imputation method for homogeneous data (e.g., Miss-Glasso by Städler and Bühlmann (2012), the proposals of Loh and Wainwright (2012) or Kolar and Xing (2012b), or even procedures that are not focused specifically on precision matrices such as the missForest of Stekhoven and Bühlmann (2011)) to subsequently estimate the precision matrices or the underlying graphical models. The quality of graph recovery then mainly depends on the performance of the chosen imputation method.

Table 1. Adjusted Rand indices for scenarios with $n = 500, p = 100$, and three change points.

Network	Missingness	Method	Percentage of missing values					
			10%	20%	30%	40%	50%	
CN	MCAR	LW	BS	0.999 (0.003)	0.996 (0.006)	0.993 (0.010)	0.960 (0.060)	0.430 (0.314)
CN	MCAR	LW	OBS	0.998 (0.004)	0.996 (0.006)	0.987 (0.025)	0.949 (0.068)	0.466 (0.301)
CN	MCAR	av	BS	1.000 (0.001)	0.998 (0.004)	0.952 (0.072)	0.244 (0.256)	0.000 (0.000)
CN	MCAR	av	OBS	0.999 (0.002)	0.998 (0.005)	0.957 (0.069)	0.226 (0.252)	0.000 (0.000)
CN	MCAR	pair	BS	0.999 (0.003)	0.997 (0.006)	0.993 (0.010)	0.939 (0.118)	0.372 (0.297)
CN	MCAR	pair	OBS	0.998 (0.004)	0.996 (0.006)	0.986 (0.026)	0.938 (0.074)	0.384 (0.290)
CN	block	LW	BS	0.999 (0.004)	0.994 (0.012)	0.985 (0.020)	0.953 (0.051)	0.813 (0.208)
CN	block	LW	OBS	0.998 (0.005)	0.991 (0.019)	0.977 (0.037)	0.923 (0.085)	0.727 (0.257)
CN	block	av	BS	0.957 (0.086)	0.811 (0.214)	0.367 (0.340)	0.242 (0.267)	0.280 (0.245)
CN	block	av	OBS	0.953 (0.123)	0.765 (0.243)	0.294 (0.326)	0.197 (0.268)	0.248 (0.253)
CN	block	pair	BS	0.998 (0.006)	0.990 (0.017)	0.981 (0.025)	0.941 (0.069)	0.813 (0.170)
CN	block	pair	OBS	0.997 (0.007)	0.990 (0.013)	0.958 (0.039)	0.923 (0.076)	0.732 (0.240)
RN	MCAR	LW	BS	0.991 (0.012)	0.973 (0.051)	0.943 (0.083)	0.828 (0.116)	0.510 (0.228)
RN	MCAR	LW	OBS	0.986 (0.025)	0.971 (0.046)	0.932 (0.090)	0.791 (0.146)	0.452 (0.270)
RN	MCAR	av	BS	0.981 (0.041)	0.878 (0.102)	0.578 (0.221)	0.076 (0.184)	0.005 (0.050)
RN	MCAR	av	OBS	0.983 (0.028)	0.858 (0.120)	0.499 (0.275)	0.063 (0.168)	0.011 (0.075)
RN	MCAR	pair	BS	0.988 (0.014)	0.974 (0.050)	0.931 (0.086)	0.779 (0.141)	0.438 (0.253)
RN	MCAR	pair	OBS	0.986 (0.025)	0.970 (0.050)	0.917 (0.094)	0.766 (0.156)	0.373 (0.266)
RN	block	LW	BS	0.990 (0.021)	0.974 (0.038)	0.927 (0.103)	0.791 (0.208)	0.464 (0.328)
RN	block	LW	OBS	0.979 (0.041)	0.949 (0.073)	0.904 (0.137)	0.741 (0.225)	0.386 (0.342)
RN	block	av	BS	0.752 (0.210)	0.336 (0.311)	0.173 (0.250)	0.198 (0.238)	0.276 (0.226)
RN	block	av	OBS	0.711 (0.243)	0.234 (0.287)	0.119 (0.207)	0.136 (0.224)	0.227 (0.244)
RN	block	pair	BS	0.986 (0.027)	0.967 (0.049)	0.923 (0.100)	0.759 (0.221)	0.573 (0.313)
RN	block	pair	OBS	0.980 (0.037)	0.949 (0.079)	0.863 (0.181)	0.700 (0.234)	0.483 (0.339)

5.1.4. Setup of the Main Simulation Study

We will illustrate the behavior of our methods on settings with $n = 500$ observations of dimension $p = 100$ with three change points with segments of sizes 70, 120, 120, and 190. We randomly permute the order of the segments to avoid systematic effects. Note that in the smallest segment the number of observations is smaller than the number of variables, resulting in a truly high-dimensional setting when splitting. In each simulation, we generate a precision matrix (either random or chain network) for each segment and then draw observations independently from the corresponding centered multivariate normal distribution. The parameter δ is held fixed at 0.1 and we vary the proportion of missing data in steps of 10% between 10% and 50%. Note that in this setup in expectation there is less than one complete observation available per segment when deleting only 10% of the values completely at random. Therefore, discarding incomplete observations is clearly not a viable option and some kind of imputation method is necessary.

5.2. Results

We analyze the estimation performance (using the model selection approach of Section 4) for our three methods (average, pairwise, and LW, see Section 3.1) both using BS and OBS. We ran 100 simulations for each setting. The corresponding mean values of adjusted Rand indices are displayed in Table 1 along with their standard deviations in parenthesis. To aid the interpretability, we present the adjusted Rand indices of a selection of estimation results together with the true change points in Table 2. Note that finding the correct number of change points with an accuracy of around two observations each leads to an adjusted Rand index of around 0.95. Finding all true change points plus a false positive one leads to an adjusted Rand index of around 0.8, similar to finding only two of the three change points. Additionally, we show histograms of all the

Table 2. Examples for different adjusted Rand indices.

True change points	Found change points	Adjusted Rand
120, 190, 310	120, 188, 310	0.992
70, 190, 310	66, 191, 310	0.980
120, 240, 430	118, 239, 423	0.949
190, 260, 380	93, 190, 260, 380	0.804
120, 240, 310	119, 243	0.757
120, 240, 310	297	0.506
70, 190, 380	380	0.363

change points found over 500 simulations in Figure 4, for which we needed to consider a fixed true change points scenario to be meaningful. Note that the y -axis is on a log-scale.

Before analyzing the obtained results, we first report the performance of a naive approach (for some of the setups from Table 1). For this, we impute mean values columnwise (i.e., for each variable separately) on the whole dataset and then estimate change points using the imputed data (with our proposed methodology and implementation for the case when there are no missing values). Note that in our simulations (i.e., the setups of Table 1), there are no changes in the underlying mean values. Hence, one could expect (at least in the scenario of values missing completely at random) this naive approach to perform similar to the average imputation method. For chain networks with values missing completely at random, the naive approach performs similar to our three proposed imputation methods for up to 30% missing values. For 40–50% missing values, the naive approach performs better than the average imputation method, but clearly worse than the best performing pairwise and LW methods (by roughly 0.15 in terms of adjusted Rand index). For chain networks with blockwise missing values (and OBS), the naive approach obtained 0.61, 0.49, 0.40, 0.38, and 0.34 for the adjusted Rand indices for the scenarios with 10%–50% missing values, respectively. For low percentage of missing values, this is clearly worse than all three of our imputation

proposals (by roughly 0.4 in terms of adjusted Rand index) and for high percentage of missing values it is roughly as good as the average imputation method, and clearly worse than the best performing pairwise and LW methods (by roughly 0.4 in terms of adjusted Rand index). The bad performance of the naive approaches can be explained as follows. When values are not missing completely at random, but in some kind of blockwise structure, the naive approach tends to (falsely) find change points corresponding to the missingness structure rather than true changes in the underlying signal. Similar issues with the naive approach could happen when there are additionally also shifts in the mean of the signal. Hence, integrating imputation and change-point detection into a unified framework is essential and leads to substantial gains, especially in very challenging scenarios (with values not missing completely at random and/or high percentage of missing values).

With regards to the main simulation results comparing our three proposals, all three methods perform similarly if at most 30% of values for the chain network setup or 20% of values for the random network setup are deleted completely at random. For more challenging MCAR setups the average imputation method fails to estimate the change points reliably, whereas the LW and pairwise methods still perform reasonably well if up to 30%–40% of values are missing. The performance of all methods tends to be worse if the values are deleted blockwise, as expected. Here, the average imputation method clearly underperforms compared to the other two methods even when only 10% of the values are deleted. The pairwise and LW methods perform very well for up to 30%–40% of missing values. The LW method seems to perform somewhat better than the pairwise method. The scenarios with 50% missing values are all very challenging and the ranking of the methods may be slightly different compared to the easier scenarios.

Figure 3 provides representative examples comparing the gain curves recovered using our three methods. If 30% of the values are deleted completely at random, the expected piecewise convex shape is well conserved for all three methods. For 30% blockwise deleted values the shape is somewhat distorted when using the LW and pairwise methods but local optima still occur

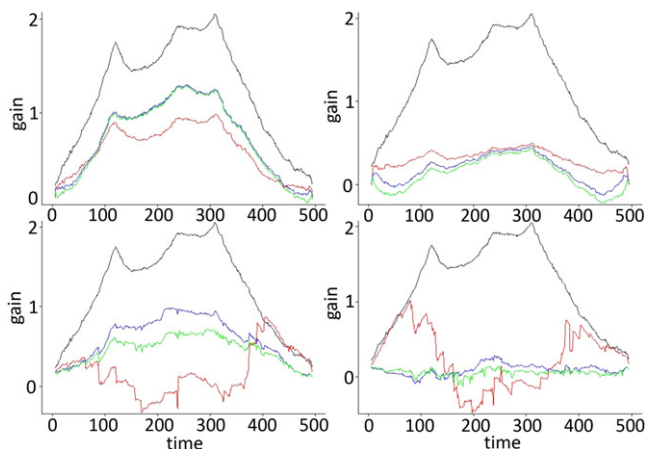


Figure 3. Recovered gain curves in random network settings with 30% MCAR (top left), 30% blockwise (bottom left), 50% MCAR (top right), and 50% blockwise (bottom right) missing values using the average (red), pairwise (green), and LW (blue) methods, compared to the case without missing values (black).

at two of the true change points. The structure (i.e., shape) is not recoverable at all due to big jumps with the average imputation method, explaining the low mean adjusted Rand index in this scenario. A key insight is that with the LW and pairwise methods the piecewise convex structure is approximately conserved, allowing for good estimation of at least one of the underlying change points, as necessary for BS and OBS. Some very challenging scenarios with 50% missing values are shown in the right panel of Figure 3. Here even in the MCAR scenario (top right), the gain curves flatten out, making it difficult to accurately locate the change points, while in the blockwise missing scenario (bottom right) all three methods would fail. Note that the imputation method (and in general also the fraction of missing values) has an effect on the magnitude of the gain curve.

The simulation scenarios with random networks seem to be harder for our methods to estimate change points compared to chain networks. As expected, at the price of higher computational cost, BS slightly outperforms OBS, especially in hard scenarios with a high fraction of missing values where we expect the piecewise convex shape of the gain curve to be strongly distorted. We also tested standard BS and OBS (as described in Section 2) on scenarios where no values are missing, but the total number of observations is scaled down by 10%–50% (Table 3). Not surprisingly, we see that complete data are generally more informative than the analogous data with missing values. Also here, the random network scenarios are more challenging compared to chain networks.

In the middle scenario of Figure 4 with 30% missing values, we estimate a total of 1343 change points with 1079 within five and 1190 within ten observations away from a true underlying change-point. The analogous numbers for the settings with 40% missingness are 1118, 754, and 882 and for 50% missingness are 724, 353, and 447. The decreasing performance is also reflected in the average adjusted Rand indices. In total, there would be $3 \times 500 = 1500$ true underlying change points. This indicates a general trend, where our algorithm seems to underfit, that is, selecting too few change points, rather than selecting incorrect ones. To investigate our model selection procedure presented in Section 4, we did a complementary study in a setting of no true

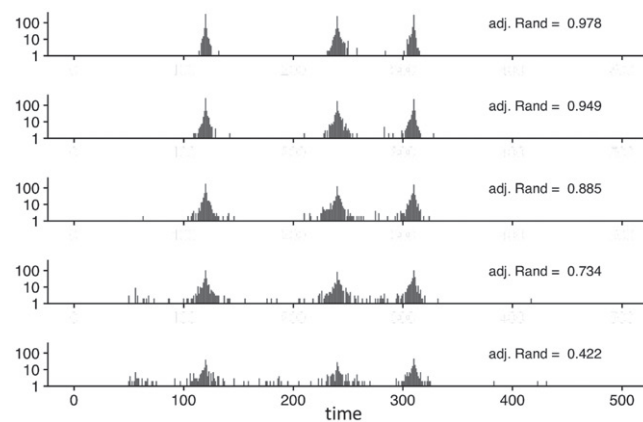


Figure 4. Cumulative estimated change points from each 500 simulations with true underlying change points at 120, 240, and 310 and random networks as the in segment precision matrices. We deleted 10% (top) to 50% (bottom) of the values blockwise and estimated the change points using OBS and the LW method. Also displayed are the average adjusted Rand indices for each set of simulations.

Table 3. Adjusted Rand indices for scenarios with $p = 100$, three change points and no missing values.

Network	Method	Number of observations				
		450	400	350	300	250
CN	BS	1.000 (0.001)	1.000 (0.001)	0.999 (0.002)	1.000 (0.002)	0.997 (0.009)
CN	OBS	1.000 (0.002)	0.999 (0.003)	0.998 (0.006)	0.999 (0.004)	0.996 (0.016)
RN	BS	0.990 (0.019)	0.990 (0.018)	0.980 (0.039)	0.947 (0.077)	0.875 (0.095)
RN	OBS	0.987 (0.021)	0.987 (0.019)	0.969 (0.051)	0.931 (0.083)	0.833 (0.084)

underlying change points. Testing all scenarios of Table 1 with 100 simulations each resulted in a total of $24 \times 5 \times 100 = 12,000$ simulations. In the setting where $n = 500$ and $p = 100$ and without true underlying change points our estimators found a total of 734 change points, 728 of which were found with the average imputation method for data where values were deleted blockwise, four with the pairwise method and two with the LW method, both with blockwise deleted values. Our model selection did not select any change points in any of the 6000 simulation with values deleted completely at random or any of the 4800 simulations with the LW or pairwise estimator where up to 30% of values were deleted blockwise. In an analogous simulation study with $n = 100$ and $p = 100$ (setting $\delta = 0.2$ to obtain reasonable fits) again no change points were found in the simulations with values missing completely at random. In total 615 change points were found in the 12,000 simulations. 109 were found with the average method, 285 were found with the LW method, and 221 with the pairwise method. Only 3, 1, and 3, respectively, were found in scenarios with up to 30% missing values.

Finally, we indicate the computational cost of the different methods. Running on a single Intel Xenon 3.0 GHz processor core, with our current R implementation each BS search including model selection took around 130, 150, and 170 sec for the average, LW and pairwise method for an average scenario of Table 1. Change-point estimation with OBS took around 40 sec for each of the three methods, thus enabling massive speed-ups.

6. Applications

In every real application care needs to be taken as the assumption of underlying piecewise constant GGMs might not be fully valid. In particular, the underlying model might also change smoothly over time. Such deviations from the model assumptions are usually visible in the shape of the gain curves. Other deviations from the model assumptions could be dependence across observations for example. In this case, various preprocessing steps (e.g., smoothing, differencing or various transformations) might be helpful to provide more reliable results.

We applied our methods on a real data example from a shallow groundwater monitoring system with naturally occurring missing values. The dataset contains $n = 753$ monthly shallow groundwater level measurements at $p = 136$ sampling locations between the Rivers Danube and Tisza within Hungary (Figure 5) from January 1951 to September 2013. The original measurements were seasonally adjusted for each sampling location individually. This dataset has approximately 35% naturally occurring missing values.

Due to the nature of the monitoring system there are a lot of missing values with a blockwise missingness structure that is

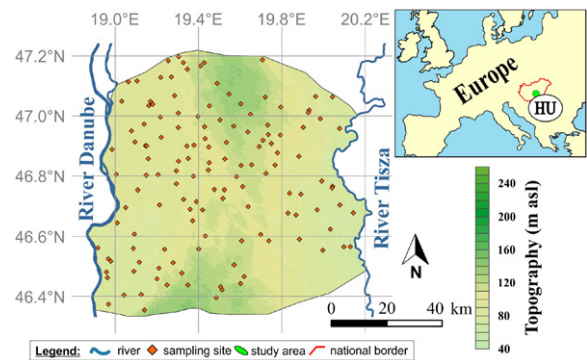


Figure 5. Location map of the shallow groundwater monitoring area within Hungary.

strongly inhomogeneous in time, see bottom right of Figure 6. The reasons for missing observations in similar monitoring systems could be the following. The failure of a sensor can lead to missing values over several consecutive observations, while replacement of them at multiple sampling locations might occur at the same time. More importantly, it is common that several sites are simultaneously newly installed or abandoned based on the available budget or even due to changing standards toward monitoring systems. These typically lead to blocks of missing observations, ranging over multiple sites (variables) as well as observations. In addition, individual randomly missing entries can occur as well.

Out of the 136 sampling sites only 93 had nonmissing values for more than $n/2 \approx 377$ months, with some having values missing for three quarters of the months. The reduced set of 93 sites only has approximately 16% missing values. We first applied BS with our three proposed methods to this subset (left of Figure 6). We used cross-validated λ_0 for each segment as described in Section 4 and a small $\delta = 0.025$ for better visualization of the gain curves at the boundaries. Note that plotting the whole gain curves requires a full search, for which BS (including the proposed model selection) took roughly 30 min. In many other scenarios (e.g., financial data) both the number of observations as well as the number of variables can be larger, quickly bringing the full grid search beyond what is computationally feasible. In such cases the faster OBS still remains a viable option.

While the order of the splits is different across the three methods, the finally found change points lie very close to each other. The three change points which all methods agree on for both the full and reduced dataset with the highest cross-validated improvements lie around November 1964, April 1983 and January 1996 (with up to only two observations difference). Besides these three, several smaller ones are found. The major change points have a clear hydrogeological interpretation. For

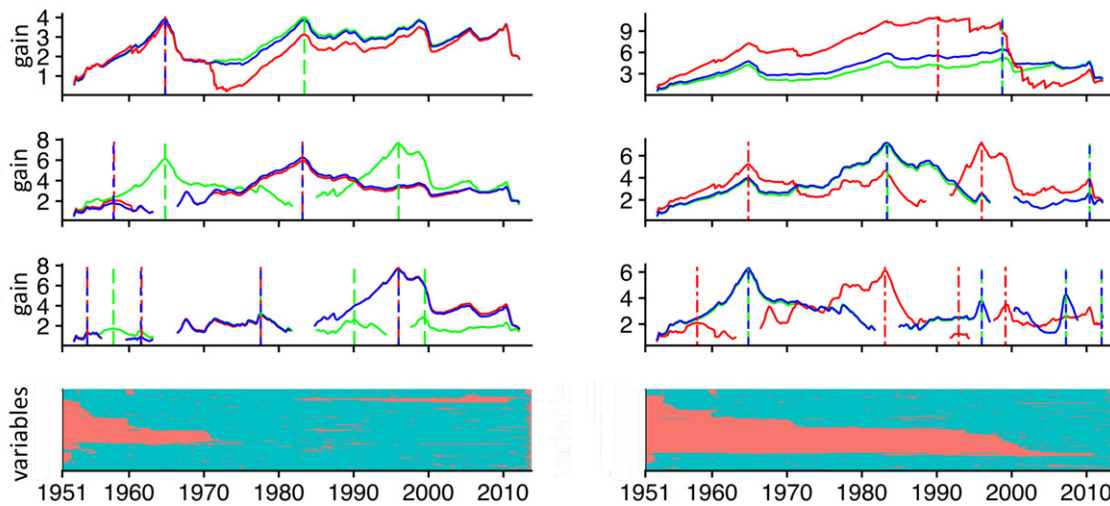


Figure 6. Gain curves of the first three BS iterations with the average imputation (red), LW (blue), and pairwise (green) methods applied to the full groundwater data (right) and to the subset only including sampling locations with less than 50% missing values (left). Splits found in each step are marked with a vertical line of the corresponding color and the gain curves of the subsegments are shown in the plot below. The missingness structure of the datasets is displayed in the bottom row with red parts corresponding to missing values.

example, around 1983, there was a drop in groundwater levels unequally impacting areas within the monitoring system. This presumably shows up jointly as a mean and covariance shift in our analysis. Note that our log-likelihood based approach will also detect shifts in mean.

Interestingly, applying BS to the very challenging full dataset yields very similar results (right of Figure 6). This would not have been possible without the techniques preventing jumps in the gain curves discussed in Section 3.2 due to the big blocks of missing values. Even with our technique, the gain curve for the first split flattens out, showing the challenges in such complicated scenarios. Also note the jumps in the gain curve of the average imputation method around the boundaries of big missing blocks, similar to Figure 3.

7. Conclusions

Our estimation methods enable practitioners to search for change points in settings where this was impossible before. The LW based imputation method presented in Section 3.1 has a stable performance even in challenging high-dimensional scenarios with lots of missing values. Since its computational cost is not significantly higher than the other methods considered, we generally recommend its usage. The choice of BS or OBS should be based on computational resources. BS results in slightly better results and the possibility of better visualization by drawing the full gain curves, however at a considerably higher computational cost than OBS. We emphasize that technical adjustments regarding the evaluation of the gains introduced in Section 3.2 as well as the model selection procedure of Section 4 lie at the core of our methodology, enabling good performances both on simulated and real data with missing values.

Supplementary Materials

R-package hcd: The R-package hcd for high-dimensional change point detection available at <https://github.com/mlondschi/hcd> implements our methods to find change points in GGMs with possibly

missing values. README files explain the usage via some examples and describe how to reproduce the above mentioned simulation results and figures. Additionally, the hcd R-package also implements some ongoing work on multivariate nonparametric change-point detection (see Kovács, Londschi, and Bühlmann 2021a). We plan to update the package to use a faster Graphical Lasso implementation and more flexible Graphical Elastic Net (see Kovács et al. 2021b) and eventually upload also to CRAN.

Acknowledgments

We thank an associate editor and two anonymous reviewers for constructive comments. Furthermore, we thank Tamás Garamhegyi, József Kovács (Department of Geology, Eötvös Loránd University, Budapest, Hungary), and József Szalai (General Directorate of Water Management, Budapest, Hungary) for the groundwater dataset and for providing hydrogeological interpretation of the found change points.

Funding

Solt Kovács and Peter Bühlmann have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement no. 786461 CausalStats—ERC-2017-ADG).

ORCID

Malte Londschi <http://orcid.org/0000-0002-6806-492X>
 Solt Kovács <http://orcid.org/0000-0002-4691-5092>
 Peter Bühlmann <http://orcid.org/0000-0002-1782-6015>

References

- Angelosante, D., and Giannakis, G. B. (2011), “Sparse Graphical Modeling of Piecewise-Stationary Time Series,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1960–1963. [769,771]
- Atsawaranunt, K., Comas-Bru, L., Amirnezhad Mozhdhehi, S., Deininger, M., Harrison, S. P., Baker, A., Boyd, M., Kaushal, N., Ahmad, S. M., Ait Brahim, Y., Arienzo, M., Bajo, P., Braun, K., Burstyn, Y., Chawchai, S., Duan, W., Hatvani, I. G., Hu, J., Kern, Z., Labuhn, I., Lachniet, M., Lechleitner, F. A., Lorrey, A., Pérez-Mejías, C., Pickering, R., Scroton, N., and SISAL Working Group Members (2018), “The SISAL Database: A

- Global Resource to Document Oxygen and Carbon Isotope Records From Speleothems,” *Earth System Science Data*, 10, 1687–1713. [769]
- Avanesov, V., and Buzun, N. (2018), “Change-Point Detection in High-Dimensional Covariance Structure,” *Electronic Journal of Statistics*, 12, 3254–3294. [769,771]
- Banerjee, O., El Ghaoui, L., and d’Aspremont, A. (2008), “Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data,” *Journal of Machine Learning Research*, 9, 485–516. [768]
- Bybee, L., and Atchadé, Y. (2018), “Change-Point Computation for Large Graphical Models: A Scalable Algorithm for Gaussian Graphical Models With Change-Points,” *Journal of Machine Learning Research*, 19, 440–477. [769]
- Dette, H., Pan, G., and Yang, Q. (2020), “Estimating a Change Point in a Sequence of Very High-Dimensional Covariance Matrices,” *Journal of the American Statistical Association* (to appear), DOI: 10.1080/01621459.2020.1785477. [769]
- Erdős, P., and Rényi, A. (1960), “On the Evolution of Random Graphs,” *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5, 17–61. [774]
- Fan, J., Feng, Y., and Wu, Y. (2009), “Network Exploration via the Adaptive Lasso and SCAD Penalties,” *The Annals of Applied Statistics*, 3, 521–541. [774]
- Friedman, J., Hastie, T., and Tibshirani, R. (2008), “Sparse Inverse Covariance Estimation With the Graphical Lasso,” *Biostatistics*, 9, 432–441. [768,770,772]
- Fryzlewicz, P. (2014), “Wild Binary Segmentation for Multiple Change-Point Detection,” *The Annals of Statistics*, 42, 2243–2281. [770]
- Gibberd, A. J., and Nelson, J. D. (2017), “Regularized Estimation of Piecewise Constant Gaussian Graphical Models: The Group-Fused Graphical Lasso,” *Journal of Computational and Graphical Statistics*, 26, 623–634. [769]
- Gibberd, A. J., and Roy, S. (2017), “Multiple Change-point Estimation in High-Dimensional Gaussian Graphical Models,” arXiv no. 1712.05786. [769]
- Hallac, D., Nystrup, P., and Boyd, S. (2018), “Greedy Gaussian Segmentation of Multivariate Time Series,” *Advances in Data Analysis and Classification*, 13, 727–751. [769]
- Hallac, D., Park, Y., Boyd, S., and Leskovec, J. (2017), “Network Inference via the Time-Varying Graphical Lasso,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 205–213. [769]
- Higham, N. J. (2002), “Computing the Nearest Correlation Matrix—A Problem From Finance,” *IMA Journal of Numerical Analysis*, 22, 329–343. [772]
- Hubert, L., and Arabie, P. (1985), “Comparing Partitions,” *Journal of Classification*, 2, 193–218. [774]
- Killick, R., Fearnhead, P., and Eckley, I. A. (2012), “Optimal Detection of Change-points With a Linear Computational Cost,” *Journal of the American Statistical Association*, 107, 1590–1598. [770]
- Kolar, M., Song, L., Ahmed, A., and Xing, E. P. (2010), “Estimating Time-Varying Networks,” *The Annals of Applied Statistics*, 4, 94–123. [774]
- Kolar, M., and Xing, E. P. (2011), “On Time Varying Undirected Graphs,” in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 407–415. [769]
- (2012a), “Estimating Networks With Jumps,” *Electronic Journal of Statistics*, 6, 2069–2106. [769]
- (2012b), “Estimating Sparse Precision Matrices From Data With Missing Values,” in *Proceedings of the 29th International Conference on Machine Learning*, pp. 635–642. [772,774]
- Kovács, S., Li, H., Bühlmann, P., and Munk, A. (2020a), “Seeded Binary Segmentation: A General Methodology for Fast and Optimal Change Point Detection,” arXiv no. 2002.06633. [770]
- Kovács, S., Li, H., Haubner, L., Munk, A., and Bühlmann, P. (2020b), “Optimistic Search Strategy: Change Point Detection for Large-Scale Data via Adaptive Logarithmic Queries,” arXiv no. 2010.10194. [770]
- Kovács, S., Londschi, M., and Bühlmann, P. (2021a), “Random Forests and Other Nonparametric Classifiers for Multivariate Change Point Detection,” Working Paper. [778]
- Kovács, S., Ruckstuhl, T., Obrist, H., and Bühlmann, P. (2021b), “Graphical Elastic Net and Target Matrices: Fast Algorithms and Software for Sparse Precision Matrix Estimation,” Working Paper. [778]
- Lam, C., and Fan, J. (2009), “Sparsistency and Rates of Convergence in Large Covariance Matrix Estimation,” *The Annals of Statistics*, 37, 4254–4278. [768]
- Lauritzen, S. L. (1996), *Graphical Models*, Oxford: Oxford University Press. [768]
- Leonardi, F., and Bühlmann, P. (2016), “Computationally Efficient Change Point Detection for High-Dimensional Regression,” arXiv no. 1601.03704. [770,771,773]
- Loh, P.-L., and Wainwright, M. J. (2012), “Structure Estimation for Discrete Graphical Models: Generalized Covariance Matrices and Their Inverses,” in *Advances in Neural Information Processing Systems*, pp. 2087–2095. [772,774]
- Matteson, D. S., and James, N. A. (2014), “A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data,” *Journal of the American Statistical Association*, 109, 334–345. [769]
- Mazumder, R., and Hastie, T. (2012), “The Graphical Lasso: New Insights and Alternatives,” *Electronic Journal of Statistics*, 6, 2125–2149. [768]
- Meinshausen, N., and Bühlmann, P. (2006), “High-Dimensional Graphs and Variable Selection With the Lasso,” *The Annals of Statistics*, 34, 1436–1462. [768]
- Rand, W. M. (1971), “Objective Criteria for the Evaluation of Clustering Methods,” *Journal of the American Statistical Association*, 66, 846–850. [774]
- Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. (2011), “High-Dimensional Covariance Estimation by Minimizing L1-Penalized Log-Determinant Divergence,” *Electronic Journal of Statistics*, 5, 935–980. [768]
- Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008), “Sparse Permutation Invariant Covariance Estimation,” *Electronic Journal of Statistics*, 2, 494–515. [768]
- Roy, S., Atchadé, Y., and Michailidis, G. (2017), “Change Point Estimation in High Dimensional Markov Random-Field Models,” *Journal of the Royal Statistical Society, Series B*, 79, 1187–1206. [769,771]
- Shirvani, A. (2015), “Change Point Analysis of Mean Annual Air Temperature in Iran,” *Atmospheric Research*, 160, 91–98. [769]
- Städler, N., and Bühlmann, P. (2012), “Missing Values: Sparse Inverse Covariance Estimation and an Extension to Sparse Regression,” *Statistics and Computing*, 22, 219–235. [769,771,772,774]
- Stekhoven, D. J., and Bühlmann, P. (2011), “MissForest—Non-Parametric Missing Value Imputation for Mixed-Type Data,” *Bioinformatics*, 28, 112–118. [774]
- Tibshirani, R. (1996), “Regression Shrinkage and Selection via the Lasso,” *Journal of the Royal Statistical Society, Series B*, 58, 267–288. [768]
- Vostrikova, L. Y. (1981), “Detecting ‘Disorder’ in Multidimensional Random Processes,” *Soviet Mathematics Doklady*, 24, 270–274. [770]
- Wang, D., Yu, Y., and Rinaldo, A. (2021), “Optimal Covariance Change Point Localization in High Dimension,” *Bernoulli*, 27, 554–575. [769]
- Witten, D. M., Friedman, J. H., and Simon, N. (2011), “New Insights and Faster Computations for the Graphical Lasso,” *Journal of Computational and Graphical Statistics*, 20, 892–900. [768]
- Xie, Y., Huang, J., and Willett, R. (2013), “Change-Point Detection for High-Dimensional Time Series With Missing Data,” *IEEE Journal of Selected Topics in Signal Processing*, 7, 12–27. [769]
- Yuan, M., and Lin, Y. (2007), “Model Selection and Estimation in the Gaussian Graphical Model,” *Biometrika*, 94, 19–35. [768]
- Zhou, S., Lafferty, J., and Wasserman, L. (2010), “Time Varying Undirected Graphs,” *Machine Learning*, 80, 295–319. [769]