Precision Matrix Estimation with Noisy and Missing Data

Roger Fan[1] **Byoungwook Jang**[1] **Yuekai Sun**[1] **Shuheng Zhou**[2]

[1]University of Michigan
[2]University of California, Riverside

Abstract

Estimating conditional dependence graphs and precision matrices are some of the most common problems in modern statistics and machine learning. When data are fully observed, penalized maximum likelihood-type estimators have become standard tools for estimating graphical models under sparsity conditions. Extensions of these methods to more complex settings where data are contaminated with additive or multiplicative noise have been developed in recent years. In these settings, however, the relative performance of different methods is not well understood and algorithmic gaps still exist. In particular, in high-dimensional settings these methods require using non-positive semidefinite matrices as inputs, presenting novel optimization challenges. We develop an alternating direction method of multipliers (ADMM) algorithm for these problems, providing a feasible algorithm to estimate precision matrices with indefinite input and potentially nonconvex penalties. We compare this method with existing alternative solutions and empirically characterize the tradeoffs between them. Finally, we use this method to explore the networks among US senators estimated from voting records data.

1 Introduction

Undirected graphs are often used to describe high-dimensional distributions. Under sparsity conditions, these graphs can be estimated using penalized methods such as

$$\hat{\Theta} \in \underset{\Theta \succeq 0}{\arg\min} \left\{ \operatorname{tr}(\hat{\Gamma}_n \Theta) - \log\det(\Theta) + g_\lambda(\Theta) \right\}, \quad (1)$$

where $\hat{\Gamma}_n$ is the sample covariance or correlation matrix and g_λ is a separable (entry-wise) sparsity-inducing penalty function. Although this approach has proven successful in a variety of application areas such as neuroscience and genomics, its soundness hinges on the positive semidefiniteness (PSD) of $\hat{\Gamma}_n$. If $\hat{\Gamma}_n$ is indefinite, the objective may be unbounded from below.

In order to ensure this penalized M-estimator is well-behaving, Loh and Wainwright (2015) impose a side constraint of the form $\rho(\Theta) < R$, where ρ is a convex function. Here we focus on the estimator using the operator norm as a side constraint

$$\hat{\Theta} \in \underset{\Theta \succeq 0, \|\Theta\|_2 \leq R}{\arg\min} \left\{ \operatorname{tr}(\hat{\Gamma}_n \Theta) - \log\det(\Theta) + g_\lambda(\Theta) \right\}. \quad (2)$$

Loh and Wainwright (2017) adopt this method and show in theory the superior statistical properties of this constrained estimator. Their results suggest that the addition of a side constraint is not only sufficient but also almost necessary to effectively untangle the aforementioned complications.

Unfortunately, this additional constraint precludes using existing methods to solve the penalized objective with non-PSD input. To close this gap, we develop an alternating direction method of multipliers (ADMM) algorithm to implement (2) efficiently. We conduct empirical studies comparing this new method to several other precision matrix estimators. Our simulation study reveals several trends that are not present in the fully observed case. Finally, we illustrate the performance of our methods in analyzing the US senate voting data, uncovering both known and novel phenomena from the modern political landscape.

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of existing related work and describe in detail the optimization issues that arise from indefinite inputs and nonconvex penalties. In Section 3, we present the proposed ADMM algorithm and present some convergence results. Section 4 provides numerical examples and comparisons. Section 5 presents an exploratory analysis of US Senate voting records data using this method and details several interesting conclusions that can be drawn from the estimated graphs. Finally, we summarize the empirical results and their practical implications regarding choice of method in Section 6.

2 Problem formulation and existing work

There is a wide body of work proposing methods to perform precision matrix estimation in the fully observed case, including Meinshausen and Bühlmann (2006), Yuan and Lin (2007), Rothman et al. (2008), Friedman et al. (2008), Banerjee et al. (2008), and Zhou et al. (2010), most of which are essentially a ℓ_1-penalized likelihood approach (1) which we will refer to as the graphical Lasso.

Recent work has focused on using nonconvex regularizers such as SCAD and MCP for model selection in the regression setting (Fan and Li, 2001; Zhang, 2010; Breheny and Huang, 2011; Zhang and Zhang, 2012). Loh and Wainwright (2015, 2017) extend this analysis to general M-estimators, including variants of the graphical Lasso objective, and show their statistical convergence and support recovery properties. Estimators with these penalties have been shown to attain model selection under weaker theoretical conditions, but require more sophisticated optimization algorithms to solve, such as the local linear approximation (LLA) method of Fan et al. (2014).

In a fully observed and noiseless setting, $\hat{\Gamma}_n$ is the sample covariance and guaranteed to be at least positive semidefinite. Then, if g_λ is the ℓ_1-penalty, the objective of (1) is convex and bounded from below. In this setting, one can show that for $\lambda > 0$ a unique optimum $\hat{\Theta}$ exists with bounded eigenvalues and that the iterates for any descent algorithm will also have bounded eigenvalues (for example, see Lemma 2 in Hsieh et al., 2014).

When working with missing, corrupted, and dependent data, the likelihood is nonconvex, and the expectation-maximization (EM) algorithm has traditionally been used to perform statistical inference. However, in these noisy settings, the convergence of the EM algorithm is difficult to guarantee and is often slow in practice. For instance, Städler and Bühlmann (2012) implement a likelihood-based method for inverse covariance estimation with missing values, but their EM algorithm requires solving a full graphical Lasso optimization problem in each M-step.

An alternative approach is to develop M-estimators that account for missing and corrupted data. For graphical models, Loh and Wainwright (2015) establish that the graphical Lasso, including a version using nonconvex penalties, can be modified to accommodate noisy or missing data by adjusting the sample covariance estimate.

These modified estimators depend on the observation that statistical theory for the graphical Lasso generally requires that $\|\hat{\Gamma}_n - \Sigma\|_\infty$ converges to zero at a sufficiently fast rate (e.g. Rothman et al., 2008; Zhou et al., 2010; Loh and Wainwright, 2017). When considering missing or corrupted data, it is often possible to construct covariance estimates $\hat{\Gamma}_n$ that satisfy this convergence criteria but are not necessarily positive semidefinite. In fact, in high-dimensional settings $\hat{\Gamma}_n$ may even be guaranteed to be indefinite. Attempting to input these indefinite covariance estimates into the graphical Lasso, however, presents novel optimization issues.

Unbounded objective. When attempting to move beyond the ℓ_1 penalized case with positive semidefinite input, the problem in (1) becomes unbounded from below, so an optimum may not necessarily exist. This issue comes from two potential sources: 1) negative eigenvalues in $\hat{\Gamma}_n$, or 2) zero eigenvalues combined with the boundedness of the nonconvex penalty g_λ. For example, consider the restriction of the objective in (1) to a ray defined by an eigenvalue-vector pair σ_1, v_1 of $\hat{\Gamma}_n$:

$$\begin{aligned} &f(I + tv_1v_1^T) \\ &\quad = \operatorname{tr}(\hat{\Gamma}_n) + t\operatorname{tr}(\hat{\Gamma}_n v_1 v_1^T) - \log(1+t) + g_\lambda(tv_1v_1^T) \\ &\quad = \operatorname{tr}(\hat{\Gamma}_n) + t\sigma_1 - \log(1+t) + g_\lambda(tv_1v_1^T). \end{aligned} \tag{3}$$

If $\sigma_1 < 0$, we see that f is unbounded from below due to the $t\sigma_1$ and $-\log(1+t)$ terms. In fact, if $\sigma_1 = 0$ and g_λ is bounded from above, as is the case when using standard nonconvex penalties, the objective is also unbounded from below.

So unboundedness can occur anytime there is a negative eigenvalue in the input matrix, or whenever there are zero eigenvalues combined with a nonconvex penalty function g_λ. Unboundedness creates optimization issues, as an optimum no longer necessarily

exists.

Handling unboundedness. In order to guarantee that an optimum exists for (1), an additional constraint of the form $\rho(\Theta) \leq R$ can be imposed, where ρ is some convex function. In this paper, we consider the estimator (2), which uses a side constraint of the form $\|\Theta\|_2 \leq R$. Loh and Wainwright (2017) show the rates of convergence of this estimator (2) and show that it can attain model selection consistency and spectral norm convergence without the incoherence assumption when used with a nonconvex penalty (see Appendix E therein), but do not discuss implementation or optimization aspects of the problem.

To our knowledge, there is currently no feasible optimization algorithm for the estimator defined in (2), particularly when the input is indefinite. Loh and Wainwright (2015) present a composite gradient descent method for optimizing a subset of side-constrained versions of (1). However, their algorithm requires a side constraint of the form $\rho(\Theta) = \frac{1}{\lambda}(g_\lambda(\Theta) + \frac{\mu}{2}\|\Theta\|_F^2)$, which does not include the spectral norm constraint and therefore cannot attain the better theoretical results it achieves (Section C.7 compares the performance of different side constraints). It may be possible to develop heuristic algorithms that alternate performing a proximal gradient update ignoring the side constraint and projecting to the constraint set, but as far as we know there has not been any analysis of algorithms of this type (we discuss this in more detail in Section C.4).

An alternative approach to solving this unbounded issue is to project the input matrix $\hat{\Gamma}_n$ to the positive semidefinite cone before inputting into (1). We discuss this further in Section 4.1, but this only solves the unbounded issue when using the ℓ_1 penalty; nonconvex penalties still require a side constraint to have a bounded objective and therefore our algorithm is still useful even for the projected methods.

3 ADMM Algorithm

Our algorithm is similar to the algorithm in Guo and Zhang (2017), which applies ADMM to the closely related problem of condition number-constrained sparse precision matrix estimation using the same splitting scheme as below. We discuss their method in more detail in Section A.6. The following algorithm is specialized to the case where the spectral norm is used as the side constraint. In Section B we derive a similar ADMM algorithm that can be used for any side constraint with a computable projection operator.

Algorithm 1: ADMM for graphical Lasso with a side constraint

Input: $\hat{\Gamma}_n$, ρ, g_λ, R
Output: $\hat{\Theta}$
Initialize $V^0 = \Theta^0 \succ 0$, $\Lambda^0 = \mathbf{0}$;
while *not converged* **do**
$V^{k+1} = \text{Prox}_{g_\lambda/\rho}\left(\frac{\rho\Theta^k + \Lambda^k}{\rho}\right)$
$\Theta^{k+1} = T_\rho\left(\frac{\rho V^{k+1} - \hat{\Gamma}_n - \Lambda^k}{\rho}\right)$
$\Lambda^{k+1} = \Lambda^k + \rho(\Theta^{k+1} - V^{k+1})$
end

Rewrite the objective from (2) as

$$f(\Theta) = \text{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) + g_\lambda(\Theta) + \mathbb{1}_{\mathcal{X}_R}(\Theta) \quad (4)$$

where $\mathcal{X}_R = \{\Theta : \Theta \succeq 0, \|\Theta\|_2 \leq R\}$ and $\mathbb{1}_{\mathcal{X}}(\Theta) = 0$ if $\Theta \in \mathcal{X}$ and ∞ otherwise.

Let $\rho > 0$ be a penalty parameter and let $\text{Prox}_{g_\lambda/\rho}$ be the prox operator of g_λ/ρ. We derive these updates for SCAD and MCP in Section A.2. Let $T_\rho(A)$ be the following prox operator for $-\log\det\Theta + \mathbb{1}_{\mathcal{X}_R}(\Theta)$, which we derive in Section A.3,

$$T_\rho(A) = T_\rho(UMU^T) = U\tilde{D}U^T$$
$$\text{where } \tilde{D}_{ii} = \min\left\{\frac{M_{ii} + (M_{ii}^2 + \frac{4}{\rho})^{1/2}}{2}, R\right\},$$

where UMU^T is the eigendecomposition of A. Then the ADMM algorithm for solving (4), which we derive in Section A.2, is described in Algorithm 1. Computationally this algorithm is dominated by the eigendecomposition used to evaluate T_ρ, and therefore has a complexity of $O(m^3)$, which matches the scaling of other graphical Lasso solvers (e.g. Meinshausen and Bühlmann, 2006; Friedman et al., 2008; Hsieh et al., 2014).

3.1 Convergence

The following proposition applies standard results on the convergence of ADMM for convex problems to show convergence when the ℓ_1 penalty is used. Details are in Section A.4.

Proposition 1. *If the penalty is convex and satisfies the conditions in Section A.1, Algorithm 1 converges to a global minimum of* (4).

Remark. Regarding the nonconvex penalty, recent work has established ADMM convergence results in some nonconvex settings (see Hong et al., 2016; Wang et al., 2015), but to our knowledge there is no convergence result that encompasses this nonsmooth and

nonconvex application. We can show convergence if a fairly strong assumption is made on the iterates, but we are currently working on extending existing results to this case.

Proposition 2 shows that any limiting point of Algorithm 1 is a stationary point of the original objective (4). This is proved in Section A.5. When using the ℓ_1 penalty or a nonconvex penalty with $R \leq \sqrt{2/\mu}$, where μ is the weak convexity constant of g_λ, the objective f is convex and therefore any stationary point is unique and also the global optimum. See Section C.5 for a more detailed discussion.

Proposition 2. *Assume that the penalty g_λ satisfies the conditions in Section A.1. Then for any limit point $(\Theta^*, V^*, \Lambda^*)$ of the ADMM algorithm defined in Algorithm 1, Θ^* is also a stationary point of the objective f as defined in* (4).

The assumptions on g_λ in Section A.1 are the same as those assumed in Loh and Wainwright (2015, 2017), and are satisfied by the Lasso, SCAD, and MCP functions.

Note that if a limiting point is found to exist when using a nonconvex penalty the result in Proposition 2 will still hold. Empirically we find that the algorithm performs well and converges consistently when used with nonconvex penalties, but there is no existing theoretical guarantee that a limiting point of ADMM will exist in that setting.

4 Simulations

We evaluate the proposed estimators using the relative Frobenius norm and the sum of the false positive rate and false negative rate (FPR+FNR). We present results over a range of λ values, noting that all the compared methods would use similar techniques to perform model tuning. Section C.1 presents an example of how to use BIC or cross-validation to tune these methods. We present results using covariance matrices from auto-regressive and Erdős-Rényi random graph models. See Section C for descriptions of these models as well as additional simulation results.

4.1 Alternative methods

When faced with indefinite input, there are two alternative graphical Lasso-style estimators that can be used besides (2), which involve either ℓ_∞ projection to the positive semidefinite cone or nodewise regression in the style of Meinshausen and Bühlmann (2006).

Projection. Given an indefinite input matrix $\hat{\Gamma}_n$, Park (2016) and Greenewald et al. (2017) propose performing the projection $\hat{\Gamma}_n^+ = \arg\min_{\Gamma \succeq 0} \|\Gamma - \hat{\Gamma}_n\|_\infty$. They then input $\hat{\Gamma}_n^+$ into the optimization problem (1). This is similar to the projection done in Datta and Zou (2017). In terms of the upper bound on statistical convergence rates, this method pays a constant factor cost, though in practice projection may result in a loss of information and therefore a decrease in efficiency.

After projecting the input, existing algorithms can be used to optimize (1) with the ℓ_1 penalty. However, as mentioned in Section 2, using a nonconvex penalty still leads to an unbounded objective and therefore still requires using our ADMM algorithm to solve (2).

Nodewise regression. Loh and Wainwright (2012) and Rudelson and Zhou (2017) both study the statistical and computational convergence properties of using errors-in-variables regression to handle indefinite input matrices in high-dimensional settings. Following the nodewise regression ideas of Meinshausen and Bühlmann (2006) and Yuan (2010), we can perform m Lasso-type regressions to obtain estimates $\hat{\beta}_j$ and form estimates $\hat{a}_j$, where

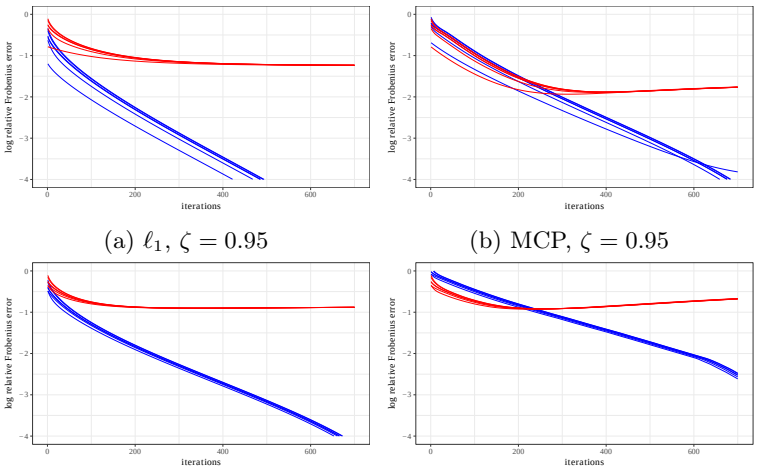
$$\begin{aligned} \hat{\beta}_j &\in \underset{\|\beta\|_1 \leq R}{\arg\min} \left\{ \frac{1}{2} \beta^T \hat{\Gamma}_{n,-j,-j} \beta - \langle \hat{\Gamma}_{n,-j,j}, \beta \rangle + \|\beta\|_1 \right\} \\ \hat{a}_j &= -(\hat{\Gamma}_{n,j,j} - \langle \hat{\Gamma}_{n,-j,j}, \hat{\beta}_j \rangle)^{-1} \end{aligned} \tag{5}$$

and combine to get $\tilde{\Theta}$ with $\tilde{\Theta}_{-j,j} = \hat{a}_j \hat{\beta}_j$ and $\tilde{\Theta}_{j,j} = -\hat{a_j}$. Finally, we symmetrize the result to obtain $\hat{\Theta} = \arg\min_{\Theta \in S^m} \|\Theta - \tilde{\Theta}\|_1$, where S^m is the set of symmetric matrices.

These types of nodewise estimators have gained popularity as they require less restrictive incoherence conditions to attain model selection consistency and often perform better in practice in the fully observed case. They have not, however, been as well studied when used with indefinite input.

4.2 Data models

We test these methods on two models that result in indefinite covariance estimators, the non-separable Kronecker sum model from Rudelson and Zhou (2017) and the missing data graphical model described in Loh and Wainwright (2015). In the main paper we focus on the missing data model, but Section C contains a detailed description of the Kronecker sum model as well as simulation results using it.



(a) ℓ_1, $\zeta = 0.95$ (b) MCP, $\zeta = 0.95$

(c) ℓ_1, $\zeta = 0.7$ (d) MCP, $\zeta = 0.7$

Figure 1: Convergence of the ADMM algorithm for several initializations. Blue lines show the relative optimization error ($\|\Theta^k - \hat{\Theta}\|_F / \|\Theta^*\|_F$, where $\hat{\Theta}$ is the result of running our algorithm to convergence) while red lines show the statistical error ($\|\Theta^k - \Theta^*\|_F / \|\Theta^*\|_F$). All panels use an AR1(0.7) covariance with $m = 300$ and $n = 125$ and set $\rho = 12$. The left panels use an ℓ_1 penalty, while the right panels use MCP with $a = 2.5$. R is set to be three times the oracle spectral norm.

Missing data (MD). As discussed above, Loh and Wainwright (2013, 2015) propose an estimator for a graphical model with missing-completely-at-random observations.

Let $W \in \mathbb{R}^{n \times m}$ be a mean-zero subgaussian random matrix. Let $U \in \{0,1\}^{n \times m}$ where $U_{ij} \sim$ Bernoulli(ζ_j) are independent of W. This corresponds to entries of the jth column of the data matrix being observed with probability ζ_j. Then we have an unobserved matrix Z and observed matrix X generated by $Z = WA^{1/2}$ and $X = U \circ X$, where $\circ$ denotes the Hadamard, or element-wise, product. Here the covariance estimate for A is

$$\hat{\Gamma}_n = \frac{1}{n} X^T X \oslash M \text{ where } M_{k\ell} = \begin{cases} \zeta_k & \text{if } k = \ell \\ \zeta_k \zeta_\ell & \text{if } k \neq \ell \end{cases} \tag{6}$$

where $\oslash$ denotes element-wise division. As we divide off-diagonal entries by smaller values, $\hat{\Gamma}_n$ will not necessarily be positive semidefinite.

4.3 Simulation results

Optimization performance. Figure 1 shows the optimization performance of Algorithm 1 using nonprojected input matrices from the missing data model with both ℓ_1 and nonconvex penalties (MCP). The top two panels present an "easy" scenario with a higher sampling rate, while the bottom two have a more challenging scenario with significant missing data. Blue lines report the optimization error while red lines are the statistical error.

All the plots in Figure 1 have their optimization error quickly converge to below the statistical error. These plots also suggest that our algorithm can attain linear convergence rates. We find that the algorithm consistently converges well over a range of tested scenarios.

Comparing the statistical error of the top two plots, we see that MCP achieves significantly lower error for the easier scenario. But in the bottom two plots, where there is more missing data, it struggles relative to the ℓ_1 penalty. This is a common trend through our simulations, as the performance of estimators using MCP degrades as missingness increases while the ℓ_1-penalized versions are more robust.

Method comparisons. Figure 2 demonstrates the statistical performance along the full regularization path. Across the panels from left to right, the sampling rate decreases and therefore the magnitude of the most negative eigenvalue increases (see Table 4).

In terms of Frobenius error, both projected methods and the nonprojected estimator with the ℓ_1 penalty get slightly worse across panels, but the nodewise regression and the nonprojected MCP estimator react much more negatively to more indefinite input. The nodewise regression in particular goes from being among the best to among the worst estimators as the sampling rate decreases.

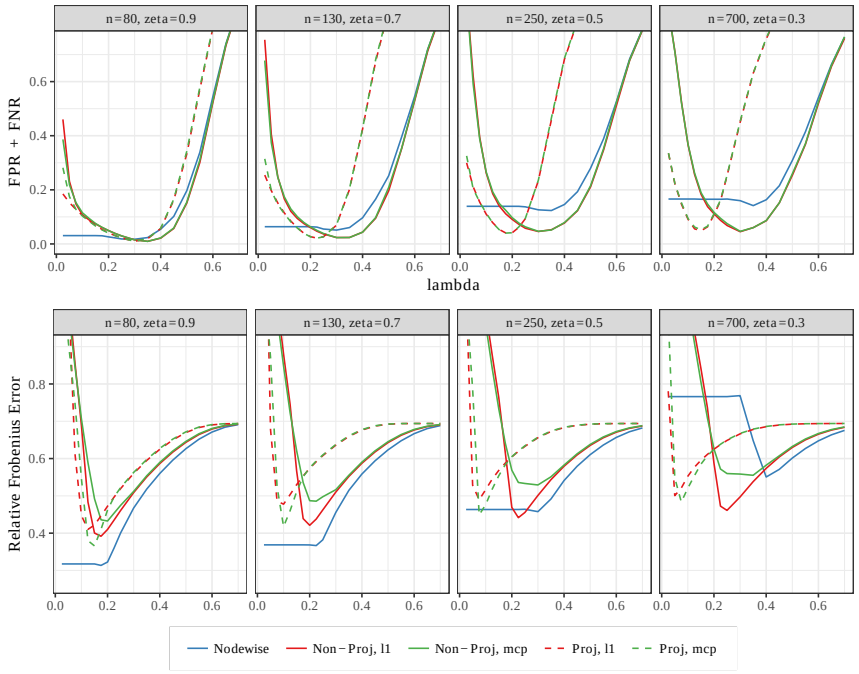


Figure 2: The performance of the various estimators for the missing data model in terms of relative Frobenius error ($\|\hat{\Theta} - \Theta^*\|_F / \|\Theta^*\|_F$) and model selection as measured by FPR + FNR. We use an AR(0.6) covariance and set $m = 1200$. Settings are chosen so that the effective sample size ($n\zeta^2$) is roughly equivalent. The MCP penalty uses $a = 2.5$. We set R to be 1.5 times the oracle value for each method and set $\rho = 24$. Our convergence criteria is $\|\Theta^{k+1} - \Theta^k\|_F / \|\Theta^k\|_F < 5\text{e}{-5}$.

Comparing the projected and nonprojected curves in Figure 2, we see that the optimal value of λ, as well as the range of optimal values, shrinks for the projected method as the sampling rate decreases. This pattern is consistently repeated across models and scenarios, likely because the ℓ_∞ projection is shrinking the off-diagonal entries of the input matrix. We find that the nonprojected graphical Lasso performs slightly better than the projected version when used with the ℓ_1 penalty, likely due to the information lost in this shrinkage.

Figure 2 also shows how these methods perform in terms of model selection. We can see that the nonconvex penalties perform essentially identically to their ℓ_1 penalized counterparts. In particular, the degradation of the nonprojected MCP estimator in terms of norm error does not seem to affect its model selection performance. The nodewise regression, however, still demonstrates this pattern, as its model selection performance degrades across the panels. For scenarios with more missing data, the nonprojected estimators seem to be easier to tune, maintaining a wider range of λ values where they perform near-optimally. In Section C of the supplement we perform similar experiments in a variety of different noise and model settings.

Sensitivity to R. Figure 3 demonstrates the sensitivity of the nonprojected estimators to the choice of R, the size of the side constraint. We can see that all these methods are sensitive to the choice of R for small values of λ in terms of norm error. None of the methods are sensitive in terms of model selection.

The nonprojected graphical Lasso with MCP is the most sensitive to R and is also sensitive for larger choices of λ, which is important since it never reaches its oracle minimum norm errors when R is chosen to be larger than the oracle. The nonprojected graphical Lasso with ℓ_1 and the projected graphical Lasso with MCP both still achieve the same best-case performance when R is misspecified, though tuning λ becomes more difficult.

The nodewise regression results are also plotted here. Here R is the ℓ_1 side constraint level in (5). For smaller values of λ the nodewise estimator levels off, corresponding to when the side constraint becomes active over the penalty. Different values of R change when this occurs and, if R is chosen large enough, do not significantly affect ideal performance. Note that these use a stronger oracle that knows each column-

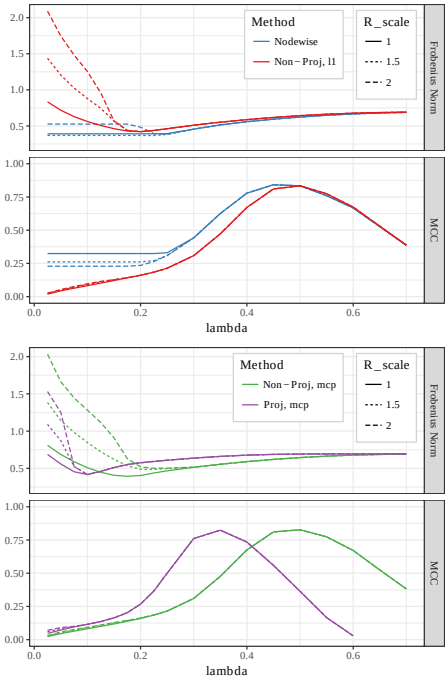


Figure 3: The performance of missing data estimators over different choices of R. The non-nodewise estimators set $R = \text{R\_scale} \times \|A\|_2$, while each node's regression in the nodewise estimator sets R to be R_scale times that node's oracle ℓ_1 value. We use an $\text{AR}(0.6)$ covariance, set $m = 1200$, $n = 130$, and choose a sampling rate of $\zeta = 0.7$. The MCP penalty is chosen with $a = 2.5$.

wise ℓ_1 norm, but do show that this method can be improved with careful tuning.

5 Senate voting analysis

Based on the missing data model from Section 4.2, we estimate the conditional dependence graph among senators using the ADMM algorithm from Section 3. The dataset includes voting records from the United States Senate during the 112th Congress (2011-2013). We drop senators who serve partial terms and unanimous votes, resulting in a dataset of voting records for 99 senators over 426 votes. Appendix D contains further details regarding data processing and the methods used as well as additional analysis.

Missing values in this data correspond to votes that are missed by senators and consist of roughly 2.6% of total votes. Note that only 109 of the votes are fully observed, so some type of correction or imputation should be used instead of omitting rows.

A major story at this time was the rise of the tea party movement in the Republican party. Across the US government tea party challengers rose to prominence. Though it was not an official party, politicians associated with the tea party movement tended to be more conservative and less likely to compromise than establishment Republicans, leading to a particularly politically polarized period of government.

Figure 4 plots the estimated graph among senators. As expected the distinction between Republicans and Democrats is stark. Both independent senators caucus with the Democrats, so as expected they are part of the Democratic component of the graph.

We identify senators who were present at the inaugural meeting of the unofficial Senate Tea Party Caucus as well as those elected in 2010 with significant tea party support.[1] These senators are colored in black, and we can see that within the Republican party they are clustered together.

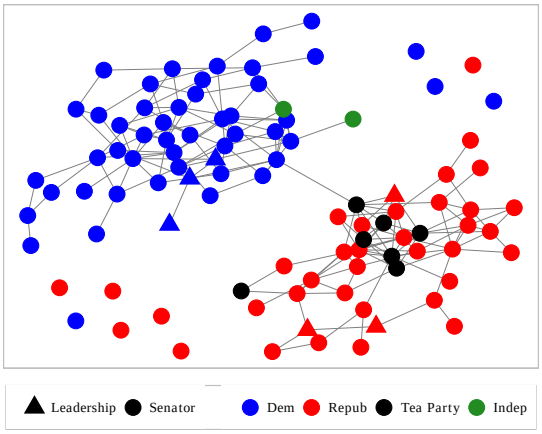
In Figure 4a we can see that the sole connection between parties runs through the tea party (Rand Paul) and Jeff Merkley, a Democratic senator. This may be surprising, as Rand Paul is one of the most conservative senators and Merkley one of the most liberal. Paul is, however, regarded as a relatively libertarian conservative. So though he is extremely conservative in some dimensions, he may share liberal views with Merkeley on others.

Figure 4b plots the same graph estimated at a lower penalization level. The Republicans who have cross-party connections include some of both the most conservative (Paul) and the most moderate (Thad Cochran, Lisa Murkowski).[2] On the Democratic side the cross-connected senators also include both the most liberal (Sanders, Merkley, Tom Udall) and relatively moderate (Claire McCaskill). As expected, moderates are among those most connected opposing party, but this shows that the most extreme members of a party can also be linked to the opposing party. Appendix D discusses these cross-party links in more detail.

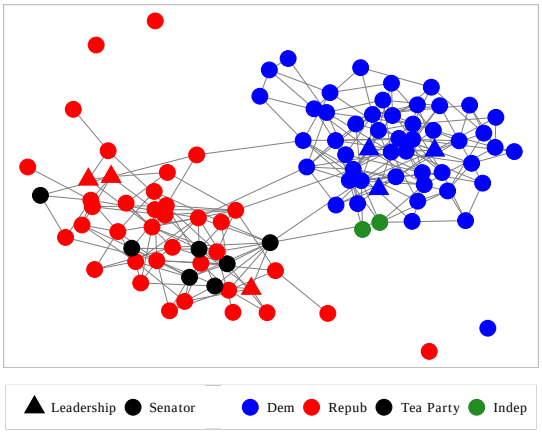
Figure 4c shows the Republican subgraph from Figure 4a. Here we can identify other senators who are closely associated with the tea party. In particular, two nodes near the tea party cluster are marked 'H' and 'C,' corresponding to Senators Orrin Hatch and Tom Coburn. Both have been linked to the tea party in the media, either as candidates supported by it or

[1] The marked tea party senators are Marco Rubio, Mike Lee, Jerry Moran, Jim DeMint, Rand Paul, Ron Johnson, and Pat Toomey.

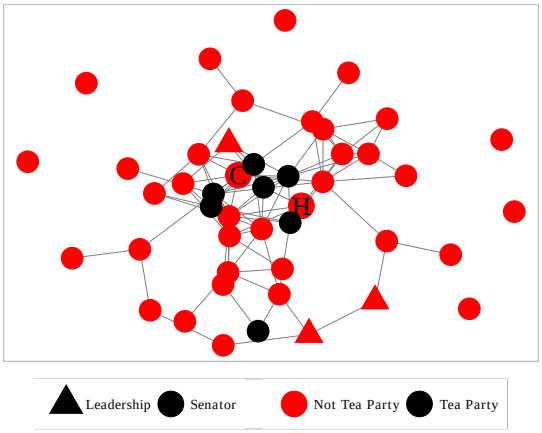
[2] Here we are measuring ideology by NOMINATE, a standard method in political science for assessing a representative's position on the political spectrum (Poole, 2005). See Appendix D for more details.



(a) $\lambda = 0.21$



(b) $\lambda = 0.15$



(c) $\lambda = 0.15$, Republican subgraph

Figure 4: Graphs among senators estimated on Senate voting records from the 112th US Congress using an ℓ_1 penalty with penalty λ as indicated. We set $R = 10$ and the ADMM algorithm was run with $\rho = 10$. After estimation, the precision matrix is thresholded at 0.04 for the top panel and 0.055 for the bottom two.

as being supportive of the movement.

It is also of interest that one marked senator is not clustered with the others, Jerry Moran. This suggests that he is not as closely connected to the tea party movement as the others we have identified.

6 Summary and discussion

In this paper, we study the estimation of sparse precision matrices from noisy and missing data. To close an existing algorithmic gap, we propose an ADMM algorithm that allows for fast optimization of the side-constrained graphical Lasso, which is needed to implement the graphical Lasso with either indefinite input and/or nonconvex penalties. We investigate its convergence properties and compare its performance with other methods that handle the indefinite sample covariance matrices that arise with dirty data.

We find that methods with nonconvex penalties are quite sensitive to the indefiniteness of the input covariance estimate, and are particularly sensitive to the magnitude of its negative eigenvalues. They may have better existing theoretical guarantees, but in practice we find that with nontrivial missingness or noise they perform worst than or, at best, recover the performance of their ℓ_1-normalized counterparts. The nonconvex methods can outperform the ℓ_1-penalized ones when there is a small amount of missingness or noise, but in these cases we often find the nodewise estimator to perform best.

In difficult settings with significant noise or missingness, the most robust and efficient method seems to be using the graphical Lasso with nonprojected input and an ℓ_1 penalty. As the application becomes easier – with more observations or less missing data – the nodewise estimator becomes more competitive, just as it is understood to be with fully observed data.

The projected graphical Lasso estimator with an ℓ_1 penalty seems to be slightly worse than its nonprojected counterpart. Projection does, however, allow for the use of nonconvex penalties in more difficult settings without the large degradation in performance we have observed. This may be desired in some scenarios when the nonzero off-diagonal precision matrix entries are expected to be large.

Finally, we also use this new algorithm to estimate conditional dependence graphs among US senators using voting records data. We identify several interesting patterns in these graphs, especially regarding the rise of the tea party movement and cross-party connections between senators.

Acknowledgements

The research is supported in part by the NSF under grants DMS-1316731 and NSF-1830247.

References

Agarwal, A., Negahban, S., and Wainwright, M. J. (2012). Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482.

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

Belloni, A., Rosenbaum, M., and Tsybakov, A. B. (2017). Linear and conic programming estimators in high dimensional errors-in-variables models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):939–956.

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and distributed computation: Numerical methods*. Prentice Hall Englewood Cliffs, NJ. Republished by Athena Scientific in 1997.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232.

Datta, A. and Zou, H. (2017). Cocolasso for high-dimensional error-in-variables regression. *The Annals of Statistics*, 45(6):2400–2426.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J., Xue, L., and Zou, H. (2014). Strong oracle optimality of folded concave penalized estimation. *Annals of Statistics*, 42(3):819.

Farrell, R. H. (1985). Multivariate calculation: Use of the continuous groups.

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

Greenewald, K., Park, S., Zhou, S., and Giessing, A. (2017). Time-dependent spatially varying graphical models, with application to brain fMRI data analysis. In *Advances in Neural Information Processing Systems*, pages 5834–5842.

Guo, X. and Zhang, C. (2017). The effect of L_1 penalization on condition number constrained estimation of precision matrix. *Statistica Sinica*, 27:1299–1317.

Hong, M., Luo, Z.-Q., and Razaviyayn, M. (2016). Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364.

Hornstein, M., Fan, R., Shedden, K., and Zhou, S. (2018). Joint mean and covariance estimation with unreplicated matrix-variate data. *Journal of the American Statistical Association*.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., and Ravikumar, P. (2014). QUIC: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):2911–2947.

Loh, P.-L. and Wainwright, M. J. (2012). High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *The Annals of Statistics*, 40(3):1637–1664.

Loh, P.-L. and Wainwright, M. J. (2013). Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *The Annals of Statistics*, 41(6):3022.

Loh, P.-L. and Wainwright, M. J. (2015). Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616.

Loh, P.-L. and Wainwright, M. J. (2017). Support recovery without incoherence: A case for nonconvex regularization. *The Annals of Statistics*, 45(6):2455–2482.

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

Mota, J. F., Xavier, J. M., Aguiar, P. M., and Püschel, M. (2011). A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions. *arXiv preprint arXiv:1112.2295*.

Park, S. (2016). *Selected Problems for High-Dimensional Data–Quantile and Errors-in-Variables Regressions*. PhD thesis, University of Michigan.

Park, S., Shedden, K., and Zhou, S. (2017). Nonseparable covariance models for spatio-temporal

data, with applications to neural encoding analysis. *arXiv preprint arXiv:1705.05265*.

Poole, K. T. (2005). *Spatial models of parliamentary voting*. Cambridge University Press.

Rosenbaum, M. and Tsybakov, A. B. (2010). Sparse recovery under matrix uncertainty. *The Annals of Statistics*, pages 2620–2651.

Rosenbaum, M. and Tsybakov, A. B. (2013). Improved matrix uncertainty selector. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, pages 276–290. Institute of Mathematical Statistics.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515.

Rudelson, M. and Zhou, S. (2017). Errors-in-variables models with dependent measurements. *Electronic Journal of Statistics*, 11(1):1699–1797.

Städler, N. and Bühlmann, P. (2012). Missing values: sparse inverse covariance estimation and an extension to sparse regression. *Statistics and Computing*, 22(1):219–235.

Wang, Y., Yin, W., and Zeng, J. (2015). Global convergence of ADMM in nonconvex nonsmooth optimization. *arXiv preprint arXiv:1511.06324*.

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

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

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942.

Zhang, C.-H. and Zhang, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, pages 576–593.

Zhou, S. (2014). GEMINI: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562.

Zhou, S. (Forthcoming, 2019). Sparse Hanson-Wright inequalities for subgaussian quadratic forms. *Bernoulli*. Available at https://arxiv.org/abs/1510.05517.

Zhou, S., Lafferty, J., and Wasserman, L. (2010). Time varying undirected graphs. *Machine Learning*, 80:295–319.

A Auxiliary Results

A.1 Nonconvex penalties

The nonconvex penalties we will focus on are the SCAD and MCP functions, introduced in Fan and Li (2001) and Zhang (2010), respectively. Following Loh and Wainwright (2015), we make the following assumptions regarding the (univariate) penalty function $g_\lambda \colon \mathbb{R} \to \mathbb{R}$.

(i) $g_\lambda(0) = 0$ and $g_\lambda(t) = g_\lambda(-t)$.

(ii) $g_\lambda(w)$ is nondecreasing for $w >= 0$.

(iii) $g_\lambda(w)/w$ is nonincreasing for $w > 0$.

(iv) $g'_\lambda(w)$ exists for all $w \neq 0$ and $\lim_{w \to 0^+} g'_\lambda(w) = \lambda$.

(v) g_λ is weakly convex, i.e. there exists $\mu > 0$ such that $g_\lambda(w) + (\mu/2)w^2$ is convex.

Note that Loh and Wainwright (2017) show stronger model selection results under the following additional assumption.

(vi) There exists a constant $\gamma < \infty$ such that $g'_\lambda(w) = 0$ for all $w > \gamma\lambda$.

This excludes the ℓ_1 penalty, but is satisfied by the nonconvex penalties we consider.

The SCAD penalty takes the form

$$g_\lambda(w) = \begin{cases} \lambda|w| & \text{if } |w| \leq \lambda \\ -\frac{w^2 - 2a\lambda|w| + \lambda^2}{2(a-1)} & \text{if } \lambda < |w| \leq a\lambda \\ \frac{(a+1)\lambda^2}{2} & \text{if } a\lambda < |w| \end{cases} \tag{7}$$

for some parameter $a > 2$. Note that this penalty is weakly convex with constant $\mu = 1/(a-1)$.

The MCP penalty has the form

$$g_\lambda(w) = \text{sign}(w)\lambda \int_0^{|w|} \left(1 - \frac{z}{\lambda a}\right)_+ dz \tag{8}$$

for some parameter $a > 0$. This penalty is weakly convex with $\mu = 1/a$.

A.2 Derivation of Algorithm 1

Recall that we can rewrite the objective as

$$f(\Theta) = \text{tr}(\hat{\Gamma}_n \Theta) - \log\det(\Theta) + g_\lambda(\Theta) + \mathbb{1}_{\mathcal{X}_R}(\Theta)$$

where $\mathcal{X}_R = \{\Theta : \Theta \succeq 0, \|\Theta\|_2 \leq R\}$ and $\mathbb{1}_{\mathcal{X}}(\Theta) = 0$ if $\Theta \in \mathcal{X}$ and ∞ otherwise.

We then introduce an auxiliary optimization variable $V \in \mathbb{R}^{m \times m}$ and reformulate the problem as

$$\hat{\Theta} = \underset{\Theta, V \in \mathbb{R}^{m\times m}}{\arg\max} \Big\{ \text{tr}(\hat{\Gamma}_n \Theta) - \log\det(\Theta) + \mathbb{1}_{\mathcal{X}_R}(\Theta) + g_\lambda(V) \Big\} \text{ s.t. } \Theta = V$$

For a penalty parameter $\rho > 0$ and Lagrange multiplier $\Lambda \in \mathbb{R}^{m \times m}$, we consider the augmented Lagrangian

$$\mathcal{L}_\rho(\Theta, V, \Lambda) = \text{tr}(\hat{\Gamma}_n \Theta) - \log\det(\Theta) + \mathbb{1}_{\mathcal{X}_R}(\Theta) + g_\lambda(V) + \frac{\rho}{2}\|\Theta - V\|_F^2 + \langle \Lambda, \Theta - V \rangle \tag{9}$$

The ADMM algorithm is then, given current iterates Θ^k, V^k, and Λ^k,

$$V^{k+1} = \underset{V \in \mathbb{R}^{m\times m}}{\arg\min} \Big\{ g_\lambda(V) + \frac{\rho}{2}\|\Theta^k - V\|_F^2 + \langle \Lambda^k, \Theta^k - V \rangle \Big\} \tag{10}$$

$$\Theta^{k+1} = \underset{\Theta \in \mathbb{R}^{m\times m}}{\arg\min} \Big\{ -\log\det\Theta + \text{tr}(\hat{\Gamma}_n \Theta) + \mathbb{1}_{\mathcal{X}_R}(\Theta) + \frac{\rho}{2}\|\Theta - V^{k+1}\|_F^2 + \langle \Lambda^k, \Theta - V^{k+1} \rangle \Big\} \tag{11}$$

$$\Lambda^{k+1} = \Lambda^k + \rho(\Theta^{k+1} - V^{k+1}) \tag{12}$$

Considering the V-subproblem, we can show that the minimization problem in (10) is equivalent to

$$V^{k+1} = \underset{V \in \mathbb{R}^{m\times m}}{\arg\min} \left\{ \frac{1}{\rho} g_\lambda(V) + \frac{1}{2} \left\| V - \frac{\rho\Theta^k + \Lambda^k}{\rho} \right\|_F^2 \right\}.$$

Which is a prox operator of g_λ/ρ. Let $W = \frac{\rho\Theta^k + \Lambda^k}{\rho}$ and $\nu = 1/\rho$. If g_λ is the ℓ_1 penalty then these updates simply soft-threshold the elements of W at level λ/ρ. For SCAD, these updates have the element-wise form

$$\text{Prox}_{g_\lambda/\rho}(w) = \begin{cases} 0 & \text{if } |w| \leq \nu\lambda \\ w - \text{sign}(w)\nu\lambda & \text{if } \nu\lambda \leq |w| \leq (\nu+1)\lambda \\ \frac{w - \text{sign}(w)\frac{a\nu\lambda}{a-1}}{1 - \frac{\nu}{a-1}} & \text{if } (\nu+1)\lambda \leq |w| \leq a\lambda \\ w & \text{if } a\lambda \leq |w| \end{cases} \tag{13}$$

While for MCP the updates are

$$\text{Prox}_{g_\lambda/\rho}(w) = \begin{cases} 0 & \text{if } |w| \leq \nu\lambda \\ \frac{w - \text{sign}(w)\nu\lambda}{1 - \nu/a} & \text{if } \nu\lambda \leq |w| \leq a\lambda \\ w & \text{if } a\lambda \leq |w| \end{cases} \tag{14}$$

See Loh and Wainwright (2015) for the derivations of these updates.

For the Θ-subproblem, we can similarly show that (11) is equivalent to

$$\Theta^{k+1} = \underset{\Theta \in \mathbb{R}^{m\times m}}{\arg\min} \Big\{ -\log\det\Theta + \mathbb{1}_{\mathcal{X}_R}(\Theta) + \frac{\rho}{2}\left\| \Theta - \frac{\rho V^{k+1} - \hat{\Gamma}_n - \Lambda^k}{\rho} \right\|_F^2 \Big\} \tag{15}$$

For any matrix A with corresponding eigendecomposition $A = RMR^T$ let us define the operator

$$\begin{aligned} T_\rho(A) &= T_\rho(UMU^T) \\ &= \underset{\Theta}{\arg\min} \Big\{ -\log\det\Theta + \mathbb{1}_{\mathcal{X}_R}(\Theta) + \frac{\rho}{2}\|\Theta - A\|_F^2 \Big\} \\ &= U\tilde{D}U^T \\ &\quad \text{where } \tilde{D}_{ii} = \min\left\{ \frac{M_{ii} + (M_{ii}^2 + \frac{4}{\rho})^{1/2}}{2}, R \right\} \end{aligned} \tag{16}$$

whose solution is derived in Section A.3. Then the solution to (11) is $T_\rho((\rho V^{k+1} - \hat{\Gamma}_n - \Lambda^k)/\rho)$.

Using these results, the algorithm in (10)-(12) becomes

$$\begin{aligned} V^{k+1} &= \text{Prox}_{g_\lambda/\rho}\left(\frac{\rho\Theta^k + \Lambda^k}{\rho} \right) \\ \Theta^{k+1} &= T_\rho\left(\frac{\rho V^{k+1} - \hat{\Gamma}_n - \Lambda^k}{\rho} \right) \\ \Lambda^{k+1} &= \Lambda^k + \rho(\Theta^{k+1} - V^{k+1}) \end{aligned} \tag{17}$$

A.3 Solution of T_ρ

Recall that in (16) we define

$$T_\rho(A) = \underset{\Theta}{\arg\min} \Big\{ -\log\det\Theta + \mathbb{1}_{\mathcal{X}_R}(\Theta) + \frac{\rho}{2}\|\Theta - A\|_F^2 \Big\}$$

Let $\Theta = WDW^T$ and $A = UMU^T$ be the eigendecompositions of the optimization variable and A. Then, similar to the derivation in Guo and Zhang (2017), we can rewrite this problem as

$$\begin{aligned} T_\rho(A) &= \underset{\Theta\in\mathbb{R}^{m\times m}}{\arg\min} -\log\det\Theta + \frac{\rho}{2}\operatorname{tr}(\Theta\Theta) - \rho\operatorname{tr}(\Theta A) + \mathbb{1}_{\mathcal{X}_R}(\Theta) \\ &= \underset{\Theta = WDW^T}{\arg\min} -\log\det D + \frac{\rho}{2}\operatorname{tr}(DD) - \rho\operatorname{tr}(WDW^TUMU^T) + \mathbb{1}_{\mathcal{X}_R}(D) \\ &= \underset{\Theta = WDW^T, W = U}{\arg\min} -\log\det D + \frac{\rho}{2}\operatorname{tr}(DD) - \rho\operatorname{tr}(DM) + \mathbb{1}_{\mathcal{X}_R}(D) \end{aligned}$$

The final line is since, if we denote $O(m)$ to be the set of $m \times m$ orthonormal matrices,

$$\begin{aligned} \operatorname{tr}(WDW^TUMU^T) &= \operatorname{tr}((U^TW)D(U^TW)^TM) \\ &\le \sup_{Q\in O(m)} \operatorname{tr}(QDQ^TM) = \operatorname{tr}(DM) \end{aligned}$$

Which holds with equality when $W = U$. Note that the last equality here is from Theorem 14.3.2 of Farrell (1985).

We therefore get that $T_\rho(A) = U\tilde{D}U^T$ where

$$\begin{aligned} \tilde{D} &= \underset{D \text{ diagonal}}{\arg\min} -\log\det D + \frac{\rho}{2}\operatorname{tr}(D^2) - \rho\operatorname{tr}(DM) + \mathbb{1}_{\mathcal{X}_R}(D) \\ &= \underset{D \text{ diagonal}}{\arg\min} \sum_{i=1}^m \Big(-\log D_{ii} + \frac{\rho}{2}D_{ii}^2 - \rho D_{ii}M_{ii} + \mathbb{1}(0 \le D_{ii} \le R) \Big) \end{aligned}$$

We can see that this is separable by element. Let

$$q(d; M_{ii}) = -\log d + \frac{\rho}{2}d^2 - \rho d M_{ii}$$

So $\tilde{D}_{ii} = \arg\min_d q(d; M_{ii}) + \mathbb{1}(0 \le d \le R)$. Ignoring the constraints in the indicator function for now, we can set the derivative of q equal to zero to get that

$$0 = -\frac{1}{d} + \rho d - \rho M_{ii} \implies 0 = d^2 - M_{ii}d - \frac{1}{\rho}$$

Which we can solve with the quadratic formula to show that $q(d; M_{ii})$ has a unique minimizer over $d > 0$ at

$$\underset{d}{\arg\min}\, q(d; M_{ii}) = \frac{M_{ii} + (M_{ii}^2 + \frac{4}{\rho})^{1/2}}{2}$$

Adding $\mathbb{1}(0 \le d \le R)$ back and noting that $q(d; M_{ii})$ is strictly convex over $d > 0$, we get that we simply need to truncate this value at R. Therefore we get that

$$\begin{aligned} T_\rho(UMU^T) &= U\tilde{D}U^T \\ &\quad \text{where } \tilde{D}_{ii} = \min\left\{ \frac{M_{ii} + (M_{ii}^2 + \frac{4}{\rho})^{1/2}}{2}, R \right\} \end{aligned}$$

A.4 Proof of Proposition 1

Proof. The optimization problem (4) is equivalently

$$\begin{aligned} \min_{\Theta,V} \ \phi(\Theta,V) = \min_{\Theta,V} \{f_1(\Theta) + f_2(V)\} \\ \text{s.t.} \quad A\text{vec}(V) + B\text{vec}(\Theta) = 0 \end{aligned} \tag{18}$$

where $f_1(\Theta) = \text{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) + \mathbb{1}_{\mathcal{X}_R}(V)$, $f_2(V) = g_\lambda(V)$, $A = -I_{m^2}$, and $B = I_{m^2}$.

Boyd et al. (2010) show that if f_1 and f_2 are proper convex functions and if (18) is solveable then ADMM converges in terms of the objective value $\phi(\Theta^t, V^t) \to \phi^*$ and dual variable $\Lambda^t \to \Lambda^*$. Bertsekas and Tsitsiklis (1989, Proposition 4.2) and Mota et al. (2011) show that if in addition A and B have full column rank then we get convergence of the primal iterates $\Theta^t \to \Theta^*$ and $V^t \to V^*$, where (Θ^*, V^*) is the solution to (18).

□

A.5 Proof of Proposition 2

Before we prove Proposition 2, we first define directional derivatives and stationary points.

Definition. The *directional derivative* of a lower semi-continuous function h at Θ in the direction Δ is

$$h'(\Theta;\Delta) = \lim_{t\searrow 0} \frac{h(\Theta + t\Delta) - h(\Theta)}{t}.$$

Note that we allow $h'(\Theta;\Delta) = +\infty$. We say that Θ is a *stationary point* of h if it satisfies the first-order necessary conditions to be a local extrema, i.e.

$$h'(\Theta;\Delta) \geq 0 \text{ for all directions } \Delta \in \mathbb{R}^{m\times m}$$

Note that this coincides with the definition of stationary point used in Loh and Wainwright (2017), though they use slightly different notation. Also note that $h'(\Theta;\Delta) = \langle \nabla h(\Theta), \Delta\rangle$ when h is continuously differentiable.

Proof. From the first-order necessary conditions of the subproblems (10)-(11), we get that, for all $\Delta \in \mathbb{R}^{m\times m}$,

$$\begin{aligned} 0 &\leq g'_\lambda(V^{k+1};\Delta) - \langle \rho(\Theta^k - V^{k+1}) + \Lambda^k, \Delta\rangle \\ 0 &\leq \langle \hat{\Gamma}_n - (\Theta^{k+1})^{-1} + \rho(\Theta^{k+1} - V^{k+1}) \\ &\qquad + \Lambda^k, \Delta\rangle + \mathbb{1}'_{\mathcal{X}_R}(\Theta^{k+1};\Delta) \end{aligned} \tag{19}$$

And recall that

$$\Lambda^{k+1} = \Lambda^k + \rho(\Theta^{k+1} - V^{k+1}). \tag{20}$$

We can rewrite (19)-(20) as

$$g'_\lambda(V^{k+1};\Delta) \geq \langle \rho(\Theta^k - \Theta^{k+1}) + \Lambda^{k+1}, \Delta\rangle \tag{21}$$

$$0 \leq \langle \hat{\Gamma}_n - (\Theta^{k+1})^{-1} + \Lambda^{k+1}, \Delta\rangle + \mathbb{1}'_{\mathcal{X}_R}(\Theta^{k+1};\Delta) \tag{22}$$

$$\frac{1}{\rho}(\Lambda^{k+1} - \Lambda^k) = \Theta^{k+1} - V^{k+1}. \tag{23}$$

Now consider a fixed point $(\Theta^*, V^*, \Lambda^*)$ and consider (21)-(23) evaluated at this limit point. From (23) we get that $\Theta^* = V^*$. This combined with (21) gives us that, for all $\Delta \in \mathbb{R}^{m\times m}$,

$$g'_\lambda(\Theta^*;\Delta) \geq \langle \Lambda^*, \Delta\rangle$$

Finally, (22) gives us that

$$0 \leq \langle \hat{\Gamma}_n - (\Theta^*)^{-1} + \Lambda^*, \Delta\rangle + \mathbb{1}'_{\mathcal{X}_R}(\Theta^*;\Delta)$$

Using the above and recalling the objective f as defined in (4), we get that, for all $\Delta \in \mathbb{R}^{m\times m}$,

$$\begin{aligned} 0 &\leq \langle \hat{\Gamma}_n - (\Theta^*)^{-1}, \Delta\rangle + \langle \Lambda^*, \Delta\rangle + \mathbb{1}'_{\mathcal{X}_R}(\Theta^*;\Delta) \\ &\leq \langle \hat{\Gamma}_n - (\Theta^*)^{-1}, \Delta\rangle + g'_\lambda(\Theta^*;\Delta) + \mathbb{1}'_{\mathcal{X}_R}(\Theta^*;\Delta) \\ &= f'(\Theta^*;\Delta) \end{aligned}$$

So Θ^* is a stationary point of f by definition. □

A.6 Comparison to Guo and Zhang (2017)

Guo and Zhang (2017) study the problem of condition number-constrained precision matrix estimation, where they consider the estimator

$$\hat{\Theta} = \underset{\Theta\succ 0, \text{cond}(\Theta)\leq\kappa}{\arg\min} -\log\det\Theta + \text{tr}(\hat{\Gamma}_n\Theta) + \lambda\|\Theta\|_{1,\text{off}} \tag{24}$$

Note that this is quite similar to the estimators we consider in (2), as they simply replace the maximum eigenvalue constraint with a constraint on the ratio of the maximum to minimum eigenvalues.

However, they do not study the application of their estimator to cases with indefinite input or its performance in noisy and missing data situations. In particular, constraining the condition number does not necessarily guarantee that the graphical Lasso objective (1) will be lower bounded, especially when using nonconvex penalties.

As a simple example, consider the case with an input matrix and iterates

$$\hat{\Gamma}_n = \begin{pmatrix} 1 & 0 \\ 0 & -0.2 \end{pmatrix} \qquad \Theta^t = t\begin{pmatrix} 0.1 & 0 \\ 0 & 1 \end{pmatrix}$$

In this case the objective is

$$f(\Theta^t) = \operatorname{tr}(\hat{\Gamma}_n \Theta^t) - \log\det\Theta^t = -0.1\times t - \log(0.1\times t)$$

which is unbounded below as t grows even though the condition numbers of the iterates are constant.

More generally, whenever $\hat{\Gamma}_n \in \mathbb{R}^{m\times m}$ has eigenvalues $\sigma_1, \ldots, \sigma_m$, where $\sigma_1 \geq \cdots \geq \sigma_{m_1} \geq 0$ and $0 > \sigma_{m_1+1} \geq \cdots \geq \sigma_m$. Denote $S_1 = \sum_{i=1}^{m_1} \sigma_i$ and $S_2 = \sum_{i=m_1+1}^{m} -\sigma_i$. Let $VDV^T = \hat{\Gamma}_n$ be the eigendecomposition of the covariance estimate. Then for some condition number bound κ, we can consider iterates of the form $\Theta^t = tVMV^T$, where M is a diagonal matrix with entries

$$M_{ii} = \begin{cases} 1 & \text{if } i \leq m_1 \\ \kappa & \text{if } i > m_1 \end{cases}$$

Which we note has a condition number of κ. Then we can see that the objective becomes

$$\begin{aligned} f(\Theta^t) &= t\operatorname{tr}(VDV^TVMV^T) \\ &\qquad - (m-m_1)\log(\kappa) + g_\lambda(tVMV^T) \\ &= t(S_1 - \kappa S_2) \\ &\qquad - (m-m_1)\log(\kappa) + g_\lambda(tVMV^T) \end{aligned}$$

So if $\kappa > S_1/S_2$ then this objective is still unbounded below.

Using a spectral norm bound $\|\Theta\|_2 \leq R$ as the side constraint with a indefinite input guarantees a lower bound on the graphical Lasso objective regardless of the choice of R and is therefore a more natural side constraint to use.

B ADMM for general side constraints

In this section we develop an ADMM algorithm for general side constraints, i.e. the following variant of (2).[3]

$$\hat{\Theta} \in \operatorname*{arg\,min}_{\Theta\succeq 0, h(\Theta)\leq R} \operatorname{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) + g_\lambda(\Theta).$$

This algorithm has the same convergence guarantees as Algorithm 1, but in practice we find that Algorithm 1 converges faster and more consistently when the spectral norm side constraint is used.

[3]Note that we switch the notation of the side constraint function from ρ to h to avoid confusion with the ADMM penalty parameter ρ.

B.1 Derivation

We first rewrite the objective as

$$f(\Theta) = \operatorname{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) + g_\lambda(\Theta) + \mathbb{1}_{\mathcal{X}_{h,R}}(\Theta) \tag{25}$$

where $\mathcal{X}_{h,R} = \{\Theta : \Theta \succeq 0, h(\Theta) \leq R\}$ and

$$\mathbb{1}_{\mathcal{X}}(\Theta) = \begin{cases} 0 & \text{if } \Theta \in \mathcal{X} \\ \infty & \text{otherwise.} \end{cases}$$

We can then introduce auxiliary optimization variables $V_1, V_2 \in \mathbb{R}^{m\times m}$ and reformulate the optimization problem as

$$\begin{aligned} \hat{\Theta} = \operatorname*{arg\,max}_{\Theta, V_1, V_2} \Big\{ &\operatorname{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) \\ &+ g_\lambda(V_1) + \mathbb{1}_{\mathcal{X}_{h,R}}(V_2)\Big\} \\ \text{s.t. } &\Theta = V_1 = V_2 \end{aligned}$$

For a penalty parameter $\rho > 0$ and Lagrange multiplier matrices $\Lambda_1, \Lambda_2 \in \mathbb{R}^{m\times m}$, we consider the augmented Lagrangian of this problem

$$\begin{aligned} &\mathcal{L}_\rho(\Theta, V_1, V_2, \Lambda_1, \Lambda_2) \\ &\quad = \operatorname{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) + g_\lambda(V_1) + \mathbb{1}_{\mathcal{X}_{h,R}}(V_2) \\ &\qquad + \frac{\rho}{2}\|\Theta - V_1\|_F^2 + \frac{\rho}{2}\|\Theta - V_2\|_F^2 + \langle\Lambda_1, \Theta - V_1\rangle \\ &\qquad + \langle\Lambda_2, \Theta - V_2\rangle \end{aligned} \tag{26}$$

The ADMM algorithm is then, given current iterates Θ^k, V_1^k, V_2^k, Λ_1^k, and Λ_2^k,

$$\begin{aligned} V_1^{k+1} = \operatorname*{arg\,min}_{V_1\in\mathbb{R}^{m\times m}} \Big\{ &g_\lambda(V_1) + \frac{\rho}{2}\|\Theta^k - V_1\|_F^2 \\ &+ \langle\Lambda_1^k, \Theta^k - V_1\rangle\Big\} \end{aligned} \tag{27}$$

$$\begin{aligned} V_2^{k+1} = \operatorname*{arg\,min}_{V_2\in\mathbb{R}^{m\times m}} \Big\{ &\mathbb{1}_{\mathcal{X}_{h,R}}(V_2) + \frac{\rho}{2}\|\Theta^k - V_2\|_F^2 \\ &+ \langle\Lambda_2^k, \Theta^k - V_2\rangle\Big\} \end{aligned} \tag{28}$$

$$\begin{aligned} \Theta^{k+1} = \operatorname*{arg\,min}_{\Theta\in\mathbb{R}^{m\times m}} \Big\{ &-\log\det\Theta + \operatorname{tr}(\hat{\Gamma}_n\Theta) \\ &+ \frac{\rho}{2}\|\Theta - V_1^{k+1}\|_F^2 + \frac{\rho}{2}\|\Theta - V_2^{k+1}\|_F^2 \\ &+ \langle\Lambda_1^k, \Theta - V_1^{k+1}\rangle + \langle\Lambda_2^k, \Theta - V_2^{k+1}\rangle\Big\} \end{aligned} \tag{29}$$

$$\Lambda_1^{k+1} = \Lambda_1^k + \rho(\Theta^{k+1} - V_1^{k+1}) \tag{30}$$

$$\Lambda_2^{k+1} = \Lambda_2^k + \rho(\Theta^{k+1} - V_2^{k+1}) \tag{31}$$

Considering the V_1-subproblem, we can show that the minimization problem in (27) is equivalent to

$$V_1^{k+1} = \underset{V_1 \in \mathbb{R}^{m\times m}}{\arg\min} \left\{ \frac{1}{\rho} g_\lambda(V_1) + \frac{1}{2} \left\| V_1 - \frac{\rho\Theta^k + \Lambda_1^k}{\rho} \right\|_F^2 \right\}.$$

Which is a prox operator of g_λ/ρ. These have the same form as described in Section A.2.

For the V_2-subproblem, we similarly see that (28) is equivalent to

$$V_2^{k+1} = \underset{V_2 \in \mathbb{R}^{m\times m}}{\arg\min} \left\{ \mathbb{1}_{\mathcal{X}_{h,R}}(V_2) + \frac{1}{2} \left\| V_2 - \frac{\rho\Theta^k + \Lambda_2^k}{\rho} \right\|_F^2 \right\}.$$

which is equivalent to the projection operator

$$\text{Proj}_{\mathcal{X}_{h,R}} \left(\frac{\rho\Theta^k + \Lambda_2^k}{\rho} \right) = \min_{V_2 \in \mathcal{X}_{h,R}} \left\| V_2 - \frac{\rho\Theta^k + \Lambda_2^k}{\rho} \right\|_F^2 \tag{32}$$

Note that if directly projecting onto $\mathcal{X}_{h,R}$ does not have an closed-form solution, we can perform this step using Dykstra's alternating projection algorithm.

Finally, for the Θ-subproblem, we can again show that (29) is equivalent to

$$\begin{aligned} \Theta = \underset{\Theta\in\mathbb{R}^{m\times m}}{\arg\min} \Big\{ &- \log\det\Theta \\ &+ \rho \left\| \Theta - \frac{\rho V_1^{k+1} + \rho V_2^{k+1} - \hat{\Gamma}_n - \Lambda_1^k - \Lambda_2^k}{2\rho} \right\|_F^2 \Big\} \end{aligned} \tag{33}$$

Let us define the operator

$$\begin{aligned} \tilde{T}_\rho(A) &= \underset{\Theta}{\arg\min} \left\{ -\log\det\Theta + \rho\|\Theta - A\|_F^2 \right\} \\ &= \frac{1}{2}(A + (A^2 + (2/\rho)I)^{1/2}) \end{aligned} \tag{34}$$

whose solution is derived in Section A.3 if we set $R = \infty$. Then the solution to (29) is $\tilde{T}_\rho((\rho V_1^{k+1} + \rho V_2^{k+1} - \hat{\Gamma}_n - \Lambda_1^k - \Lambda_2^k)/(2\rho))$.

Using these results, the algorithm in (27)-(31) becomes

$$\begin{aligned} V_1^{k+1} &= \text{Prox}_{g_\lambda/\rho} \left(\frac{\rho\Theta^k + \Lambda_1^k}{\rho} \right) \\ V_2^{k+1} &= \text{Proj}_{\mathcal{X}_{h,R}} \left(\frac{\rho\Theta^k + \Lambda_2^k}{\rho} \right) \\ \Theta^{k+1} &= \tilde{T}_\rho \left(\frac{\rho V_1^{k+1} + \rho V_2^{k+1} - \hat{\Gamma}_n - \Lambda_1^k - \Lambda_2^k}{2\rho} \right) \\ \Lambda_1^{k+1} &= \Lambda_1^k + \rho(\Theta^{k+1} - V_1^{k+1}) \\ \Lambda_2^{k+1} &= \Lambda_2^k + \rho(\Theta^{k+1} - V_2^{k+1}) \end{aligned} \tag{35}$$

B.2 Convergence

Analogues to Propositions 1 and 2 can also be shown for this algorithm using similar methods. To do this, we first note that we can rewrite the optimization problem (25) as

$$\begin{aligned} &\min_{\Theta,V} \phi(\Theta, V) = \min_{\Theta,V} \{f_1(\Theta) + f_2(V)\} \\ &\text{s.t.} \quad A\text{vec}(V) + B\text{vec}(\Theta) = 0 \end{aligned} \tag{36}$$

where

$$\begin{aligned} f_1(\Theta) &= \text{tr}(\hat{\Gamma}_n\Theta) - \log\det(\Theta) \\ f_2(V) &= g_\lambda(A_1 V) + \mathbb{1}_{\mathcal{X}_{h,R}}(A_2 V) \end{aligned}$$

and

$$A = -I_{2m^2} \qquad B = \begin{pmatrix} I_{m^2} \\ I_{m^2} \end{pmatrix}$$

$$V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \qquad A_1 = \begin{pmatrix} I_m & 0 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 0 & I_m \end{pmatrix}$$

This results in the following augmented Lagrangian that is equivalent to (26).

$$\begin{aligned} \mathcal{L}_\rho(\Theta, V, \Lambda) = f_1(\Theta) + f_2(V) \\ + \frac{\rho}{2}\|B\Theta + AV\|_F^2 + \langle \Lambda, B\Theta + AV \rangle \end{aligned}$$

Even though we present our algorithm as a three-block ADMM in Section B.1, this formulation makes it clear that we are using a two-block splitting scheme where (27) and (28) are the separable subproblems of the V-step.

Showing similar convergence results to Propositions 1 and 2 can then be done using the same techniques as in Sections A.4 and A.5.

C Additional simulation results

C.1 Tuning parameter selection

Note that in practice tuning parameters must be selected for all these methods. In particular, we must

tune λ and possibly the side-constraint R. Note that one often has a reasonable prior for the magnitude of the spectral norm of the true precision matrix, so if that is the case a multiple of that can often be used to choose R. Also, as noted in Section 4.3, when using the ℓ_1 penalty the choice of R primarily affects how difficult tuning λ will be. Though it is important to tune correctly when using nonconvex penalties, we do not recommend those methods when there is significant missing data. Therefore we will focus on tuning λ here, though the same methods can be used to choose R as well.

Two possible methods are to use cross-validation or a modified BIC criterion. Though the particular implementation of both of these will depend on the data model that is being used, as these methods can be applied to any method that generates an indefinite initial estimate of the covariance.

For the missing data case we can follow Städler and Bühlmann (2012), which uses the same data model. Recall the notation in Section 4.2, where X_{ij} denotes the ith value of variable j and U_{ij} tracks if that value is observed. Here, we define the observed log-likelihood of an observation X_i given a precision matrix estimate $\hat{\Theta}$ as

$$\ell(X_i, U_i; \hat{\Sigma}) = \log \phi(X_{i,U_i}; \hat{\Sigma}_{U_i,U_i})$$

where X_{i,U_i} is the vector of values that are observed for observation i, $\hat{\Sigma} = \hat{\Theta}^{-1}$, and ϕ is the multivariate normal density. The BIC criterion, which we minimize, is therefore

$$\text{BIC}(\lambda) = -2 \sum_i \ell(X_i, U_i; \hat{\Sigma}) + \log(n) \sum_{j \le j'} \mathbb{1}\{\hat{\Theta}_{jj'} \neq 0$$

To cross-validate, we can divide the data into V folds, where the vth fold contains indices N_v. The cross-validation score, which we maximize, is therefore

$$\text{CV}(\lambda) = \sum_v \sum_{i \in N_v} \ell(X_i, U_i; \hat{\Sigma}_{-v})$$

where $\hat{\Sigma}_{-v} = \hat{\Theta}_{-v}^{-1}$ and $\hat{\Theta}_{-v}$ is the estimate based on the sample omitting the observations in N_v.

Figure 5 presents an example of parameter tuning on a simulated scenario. We see that both BIC and CV select slightly higher-than-optimal levels of penalization in terms of model selection, but that selected model still achieves fairly good model selection.

C.2 Kronecker sum (KS) model

Park et al. (2017) present a graphical model with additive noise that is dependent across observations.

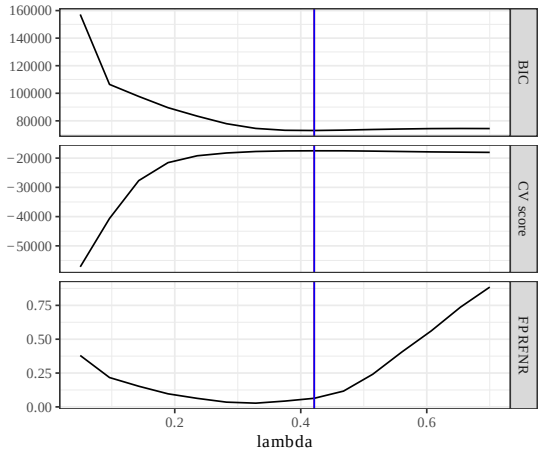


Figure 5: Example parameter tuning using BIC and CV. We additionally present the FPR+FNR rate of the estimate. The vertical lines show the optimal λ values for BIC and CV, which here happen to be identical. We set $m = 400$ and $n = 80$, the sampling rate to $\zeta = 0.8$, and let A be from an AR(0.6) model.

This noise structure was first studied in the regression setting in Rudelson and Zhou (2017) with a Kronecker sum covariance.

Let $W_1, W_2 \in \mathbb{R}^{n \times m}$ be independent mean-zero subgaussian random matrices. The data matrix is generated as $X = W_1 A^{1/2} + B^{1/2} W_2 \sim \mathcal{M}_{n,m}(0, A \oplus B)$, where $\mathcal{M}_{n,m}$ is the matrix variate normal distribution and for covariance matrices $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n \times n}$. Note that $A \oplus B = A \otimes I_n + I_m \otimes B$, where $\otimes$ denotes the Kronecker product. Here $X_0 = W_1 A^{1/2}$ contains the signal and has independent rows, while $W = B^{1/2} W_2$ is the noise matrix with independent columns but dependent rows. We are interested in estimating the signal precision matrix $\Theta = A^{-1}$, which has sparse off-diagonal entries. For our simulations, we normalize B so that $\text{tr}(B) = n\tau_B$, where τ_B is a measure of the noise level. Then the initial covariance estimate for A is given by

$$\hat{\Gamma}_n = \frac{1}{n} X^T X - \frac{\hat{\text{tr}}(B)}{n} I_m \tag{37}$$

as shown in Rudelson and Zhou (2017). Note that, in this model, $\hat{\Gamma}_n$ is guaranteed to not be positive semidefinite when $m > n$, as $X^T X$ will have zero eigenvalues.

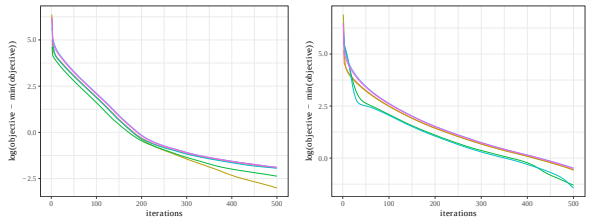
C.3 Covariance models

We look into three different models from which A Let $\Omega = A^{-1} = (\omega_{ij})$. We consider simulation settings using the following covariance models, which are also used in Zhou (2014).

- **AR1(r)**: The covariance matrix is of the form $A = (r^{|i-j|})_{ij}$.
- **Star-Block (SB)**: Here the covariance matrix is block-diagonal, where each block's precision matrix corresponds to a star-structured graph with $A_{ii} = 1$. For the corresponding edge set E, then $A_{ij} = r$ if $(i,j) \in E$ and $A_{ij} = r^2$ otherwise.
- **Erdos-Renyi random graph (ER)**: We initialize $\Omega = 0.25I$ then randomly select d edges. For each selected edge (i,j), we randomly choose $w \in [0.6, 0.8]$ and update $\omega_{ij} = \omega_{ji} \to \omega_{ij} - w$ and $\omega_{ii} \to \omega_{ii} + w$, $\omega_{jj} \to \omega_{jj} + w$.

C.4 Optimization performance

Figure 6 shows the convergence behavior for several initializations in terms of objective value. Our algorithm seems to attain a linear convergence rate in terms of the objective values even with a nonconvex penalty regardless of the initialization. We find that the algorithm consistently converges well over a range of tested scenarios.



(a) KS, AR (b) MD, ER

Figure 6: Convergence behavior of the ADMM algorithm for two objectives. Panel a shows the optimization convergence under the Kronecker sum model with $A = \text{AR1}(0.6)$, $B = \text{ER}$, $m = 300$, $n = 140$, $\tau_B = 0.3$, and $\lambda = 0.2$, while Panel b is for the missing data model with $A = ER$, $m = 400$, $n = 140$, $\zeta = 0.7$, and $\lambda = 0.2$. We choose $\rho = 12$ and the SCAD penalty is used with $a = 2.1$.

Comparison to gradient descent. Figure 7 compares the optimization performance of our ADMM algorithm to gradient descent. Note that since proximal gradient descent is difficult to do in this setting, requiring an interior optimization step, we use a heuristic version similar to that suggested by Agarwal et al. (2012) that does the proximal gradient step ignoring the side-constraint then projects back to the side-constraint space. Note that since ρ in ADMM is roughly equivalent to the inverse step size in gradient descent, we compare for difference values of ρ. These methods also take roughly the same computational time per iteration, as they are both dominated by either an eigenvalue decomposition or matrix inversion.

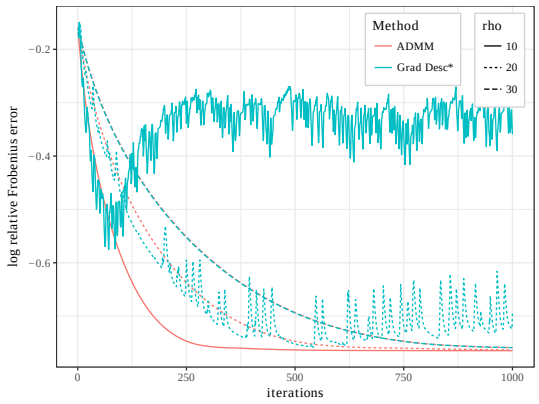


Figure 7: Comparing the convergence behavior of ADMM to gradient descent. Here we use an AR1(0.8) model with $m = 200$, $n = 150$, $\zeta = 0.6$, and use an ℓ_1 penalty with $\lambda = 0.11$. For gradient descent, ρ is the inverse of the step size. Note that since proximal gradient descent is difficult to do in this problem, this version performs the proximal gradient step without the side-constraint then projects back to the space.

We can see that for large enough values of ρ, these methods are nearly identical. Although there is no known theoretical guarantee of convergence, it seems that this heuristic gradient descent still convergence well for small enough step sizes.

But for smaller values, i.e. larger step sizes, ADMM still performs well and obtains faster convergence rates while gradient descent is unstable and inconsistent. This combined with the convergence guarantee of ADMM leads us to recommend this algorithm.

C.5 Penalty nonconvexity and R

Suppose g_λ is μ-weakly convex and $R \leq \sqrt{\frac{2}{\mu}}$. Then, as shown in Lemma 6 of Loh and Wainwright (2017), the overall objective function is strictly convex over the feasible set, and Proposition 2 therefore shows that any limiting point of ADMM algorithm corresponds to the unique global optimum of the objective. However, this choice of R radius on the $\|\cdot\|_2$ side constraint is quite restrictive. In particular, since we also require $R \geq \|\Theta^*\|_2$ we therefore need to choose large values of a in the SCAD or MCP penalties to make μ small enough, which means in practice we simply recover the performance of the ℓ_1 penalized methods. Though Loh and Wainwright (2017) show statistical properties for when the parameters are chosen satisfying this condition, in practice we

can often do better by allowing the objective to be nonconvex even though no global optimum will exist.

Once we relax this condition ($R > \sqrt{2/\mu}$), the objective becomes nonconvex, and Proposition 2 simply shows that any limiting point of our ADMM algorithm will be a stationary point of the objective. In our simulations, we generally set μ and R such that this condition is violated, and yet we show that our algorithm still results in good estimators. In fact, Figure 8 demonstrates how, in practice, choosing μ such that this condition is met tends to eliminate the advantages that nonconvex penalties provide. Here the choice of $a = 8$ is the only one that satisfies the condition, and this choice has identical performance as the ℓ_1 penalty. Using a smaller value of a violates this condition but allows the estimator to take advantage of the unbiasedness of the penalty, resulting in better performance in this setting.

Note that for both of these cases, our ADMM algorithm provides a new feasible method of implementing estimation of this type of side-constrained graphical Lasso objective. This consideration is related to tuning, where satisfying the (R, μ) condition allows the support recovery without incoherence statistical results of Loh and Wainwright (2017) but in practice results in suboptimal performance, as the nonconvex penalties have to be chosen such that they lose their unbiased advantage over the ℓ_1 penalty.

C.6 Method comparisons

Tables 1-3 present more detailed comparison based on the models from the Kronecker sum (KS) and the missing data (MD) models. We compared performance in terms of relative Frobenius and nuclear norm to the true precision matrix, as well as false positive rate plus false negative rate (FPRFNR). The Kronecker sum results are reported for two sample sizes and two values of the noise parameter τ_B, while the missing data results are reported for two covariance models and three settings of the sample size and sampling rate ζ.[4]

Comparing the projected and nonprojected methods, we see that these two methods are fairly competitive. In terms of model selection, the nonprojected methods tend to perform similarly or better than the projected methods. This improvement is particularly evident in the $n = 80$ settings in Table 1. If we focus on the methods using the ℓ_1 penalty, the nonprojected method performs at least similarly and sometimes significantly better than the projected method in terms of norm error. The lower sampling rate regime in Tables 2 and 3 shows this trend as well. Overall these results suggest a small but sometimes significant advantage for the nonprojected methods, supporting the idea that the projected methods pay a cost in terms of efficiency due to the loss of information in the projection.

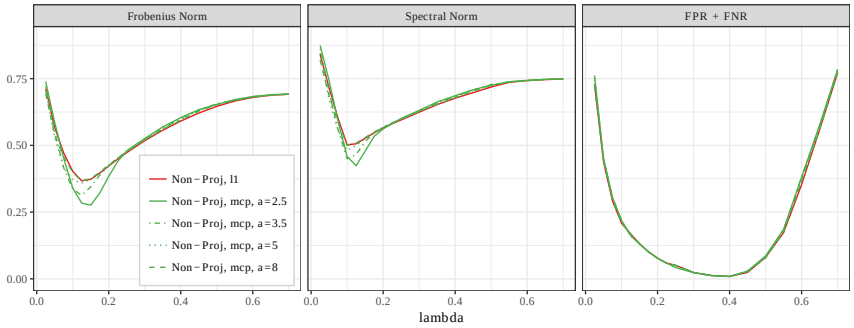
There is no significant difference in model selection between MCP and the ℓ_1 penalty. In fact, the different penalties perform almost identically across scenarios regardless of the ℓ_∞-projection step. Intuitively, the primary benefit of nonconvex penalties is their ability to more accurately estimate large entries, which are easy for the estimators to select.

In terms of norm error, however, there are significant differences depending on the indefiniteness of the optimization problem. Table 4 reports some statistics on the eigenspectrum of the input matrix. Nonprojected methods with MCP tends to perform relatively better than its ℓ_1 counterpart if the input matrix is close to the positive semidefinite space. Simulation results from the missing data model Tables 2 and 3 further support this relationship between the most negative eigenvalue and the relative performance. Here we see how the MCP nonprojected estimator goes from being significantly better than its ℓ_1 counterpart in terms of Frobenius error in the $\zeta = 0.9$ case to significantly worse when $\zeta = 0.5$. In the projected case, which projects away this indefinite issue, the MCP estimator consistently outperforms its ℓ_1 counterpart in terms of Frobenius error.

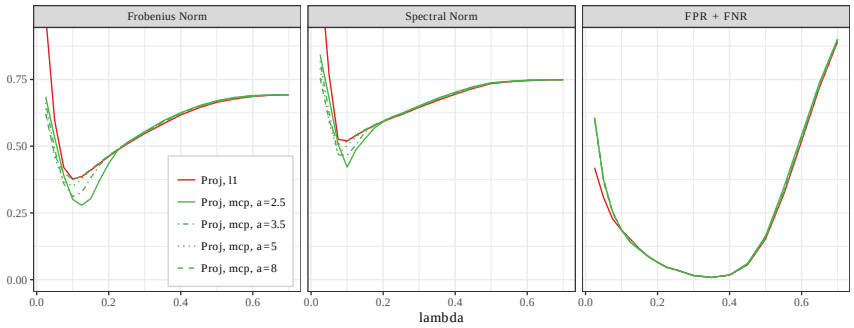
The nonconvexity of the penalty interacts poorly with indefiniteness of the input matrix. When the ℓ_1 penalty is used, it is better able to "control" the indefiniteness of the input due to its linear scaling, resulting in better norm error performance. The nonconvex penalty's inability to resolve the indefiniteness issue results in a degradation of its relative performance as the input matrix becomes more indefinite.

Turning to the nodewise estimator, we see similar patterns. Again referring to Table 4, it seems that the relative performance of the nodewise estimator varies significantly with the indefiniteness of the input matrix. When the input matrix is closer to positive semidefinite, such as the $n = 160$ situations in Table 1 or the $\zeta = 0.9$ cases in Tables 2 and 3,

[4] Note that in the initial covariance estimator for the missing data model the effective sample size for estimating an off-diagonal element of the covariance is $n\zeta^2$; four settings are designed to keep this effective sample size roughly constant while changing the sampling rate ζ. The effective sample sizes for the $n = 80$, $n = 130$, and $n = 250$ settings are 64.8, 63.7, and 62.5, respectively.



(a) Nonprojected estimators



(b) Projected estimators

Figure 8: Comparing the performance of the graphical Lasso estimators as a (and therefore the weak convexity constant μ) is changed. Here we present the results using the MCP penalty, so $\mu = 1/a$. We set R to be the oracle Note that $a = 8$ is the only value of a that satisfies the $R \leq \sqrt{2/\mu}$ condition from Loh and Wainwright (2017). Data is from a missing data model with $A = \text{AR1}(0.6)$, $m = 400$, $n = 80$, and $\zeta = 0.9$.

it performs comparably in terms of model selection and significantly better in terms of norm error. But when the input matrix is very indefinite, such as the $\zeta = 0.5$ cases in Tables 2 and 3 its relative performance quickly degrades.

Figure 9 demonstrates the patterns that we observed in Figures 2. Again, we vary the sampling rate ζ and fix the effective sample size for estimating off-diagonal entries of the covariance matrix ($n\zeta^2$), so the ℓ_∞ rate of $\hat{\Gamma}_n$ is kept constant. As the sampling rate decreases, the magnitude of the most negative eigenvalue in the covariance estimate increases, which we can see negatively affects the relative performance of the nodewise and nonprojected MCP methods. These methods are the best for high sampling rates but the worst when there is more missing data. The other methods are not as sensitive.

We can also see how the size of the off-diagonal entries in the precision matrix affect the potential benefits of the nonconvex penalties. In the top panel, which has small off-diagonal entries, the MCP estimators are consistently worse. But in the bottom panel, which has larger off-diagonal entries, the nonconvex penalties have better Frobenius norm performance when the sampling rate is high, though this advantage goes away as the sampling rate drops.

C.7 Comparison of side constraints

Here we compare using the operator norm side-constraint in (2) to the ℓ_1-side constrained version considered in Loh and Wainwright (2015). Note that theoretically Loh and Wainwright (2017) show that under certain conditions the former can attain model selection without incoherence and spectral norm convergence under the scaling $n > d^2 \log p$ (where d is the maximal node degree), which has not been shown with the latter.

Figure 10 shows the performance in terms of relative norm error for various missing data model scenarios. For large values of λ we can see that the two estimators are identical, as the side-constraints are not active.

As the penalty λ shrinks, when the ℓ_1 side constraint is used, selecting R is akin to performing

Table 1: The relative norm error and FPR + FNR performance of the Kronecker sum estimator using different methods. Here we set A to be from an AR(0.5) model and choose B from an Erdos-Renyi random graph. We set $m = 400$ and let $\tau_B = 0.5$. Metrics are reported as the minimum value over a range of penalty parameters λ. The MCP penalty is chosen with $a = 2.5$, and we set $R = 1.5\|A\|_2$.

n	τ_B	method	penalty	Frobenius	Spectral	Nuclear	FPRFNR
80	0.3	Nonproj	ℓ_1	0.422	0.598	0.406	0.107
			MCP	0.450	0.613	0.422	0.106
		Proj	ℓ_1	0.424	0.610	0.411	0.113
			MCP	0.444	0.616	0.429	0.111
		Nodewise	ℓ_1	0.391	0.517	0.383	0.130
160	0.3	Nonproj	ℓ_1	0.342	0.509	0.327	0.013
			MCP	0.363	0.518	0.345	0.013
		Proj	ℓ_1	0.356	0.525	0.343	0.016
			MCP	0.341	0.493	0.321	0.015
		Nodewise	ℓ_1	0.288	0.429	0.280	0.017
80	0.5	Nonproj	ℓ_1	0.469	0.642	0.452	0.174
			MCP	0.481	0.659	0.458	0.177
		Proj	ℓ_1	0.464	0.651	0.450	0.194
			MCP	0.483	0.658	0.467	0.197
		Nodewise	ℓ_1	0.466	0.600	0.455	0.250
160	0.5	Nonproj	ℓ_1	0.389	0.573	0.369	0.052
			MCP	0.422	0.596	0.393	0.054
		Proj	ℓ_1	0.407	0.593	0.384	0.056
			MCP	0.399	0.587	0.377	0.055
		Nodewise	ℓ_1	0.358	0.538	0.349	0.083

selection on the minimum value of λ to use, since only one of the penalty and side-constraint can be active at a time. Following the green lines, we can see that the regularization from the operator norm side-constraint can improve results. This means that ℓ_1 side-constrained estimator misses this additional improvement. At worst, for larger values of R the operator norm-constrained simply has identical performance to the ℓ_1-constained version as long as λ is appropriately selected.

Table 2: The relative norm error and FPR + FNR performance of the missing data estimator using different methods. Here we set A to be from an AR(0.6) model and set $m = 400$. Recall that ζ is the sampling rate. Metrics are reported as the minimum value over a range of penalty parameters λ. The MCP penalty is chosen with $a = 2.5$, and we set R to be 1.5 times the oracle value for each method.

A Model	n	ζ	method	penalty	Frobenius	Spectral	Nuclear	FPRFNR
AR(0.6)	80	0.9	Nonproj	ℓ_1	0.367	0.506	0.363	0.0089
				MCP	0.308	0.533	0.296	0.0088
			Proj	ℓ_1	0.377	0.520	0.375	0.0085
				MCP	0.308	0.527	0.284	0.0083
			Nodewise	ℓ_1	0.292	0.487	0.280	0.0097
	130	0.7	Nonproj	ℓ_1	0.397	0.597	0.388	0.017
				MCP	0.384	0.632	0.363	0.016
			Proj	ℓ_1	0.417	0.599	0.407	0.019
				MCP	0.348	0.626	0.326	0.018
			Nodewise	ℓ_1	0.356	0.592	0.347	0.029
	250	0.5	Nonproj	ℓ_1	0.420	0.619	0.403	0.028
				MCP	0.457	0.680	0.436	0.026
			Proj	ℓ_1	0.437	0.626	0.429	0.031
				MCP	0.391	0.600	0.369	0.032
			Nodewise	ℓ_1	0.412	0.632	0.400	0.078
	700	0.3	Nonproj	ℓ_1	0.431	0.633	0.411	0.043
				MCP	0.505	0.718	0.470	0.040
			Proj	ℓ_1	0.450	0.644	0.431	0.034
				MCP	0.422	0.664	0.391	0.031
			Nodewise	ℓ_1	0.555	0.704	0.517	0.131

Table 3: The relative norm error and FPR + FNR performance of the missing data estimator using different methods. Here we set A to be from an Erdos-Renyi random graph and set $m = 400$. Recall that ζ is the sampling rate. Metrics are reported as the minimum value over a range of penalty parameters λ. The MCP penalty is chosen with $a = 2.5$, and we set R to be 1.5 times the oracle value for each method.

A Model	n	ζ	method	penalty	Frobenius	Spectral	Nuclear	FPRFNR
ER	80	0.9	Nonproj	ℓ_1	0.398	0.426	0.369	0.133
				MCP	0.379	0.444	0.355	0.132
			Proj	ℓ_1	0.405	0.420	0.375	0.129
				MCP	0.367	0.383	0.346	0.126
			Nodewise	ℓ_1	0.349	0.357	0.334	0.160
	130	0.7	Nonproj	ℓ_1	0.409	0.495	0.372	0.137
				MCP	0.410	0.562	0.372	0.137
			Proj	ℓ_1	0.423	0.497	0.385	0.135
				MCP	0.388	0.465	0.354	0.131
			Nodewise	ℓ_1	0.372	0.463	0.346	0.194
	250	0.5	Nonproj	ℓ_1	0.421	0.556	0.379	0.163
				MCP	0.463	0.680	0.401	0.170
			Proj	ℓ_1	0.437	0.556	0.394	0.163
				MCP	0.406	0.535	0.364	0.171
			Nodewise	ℓ_1	0.431	0.654	0.376	0.241
	700	0.3	Nonproj	ℓ_1	0.427	0.604	0.383	0.193
				MCP	0.485	0.701	0.415	0.189
			Proj	ℓ_1	0.445	0.575	0.401	0.184
				MCP	0.423	0.638	0.380	0.191
			Nodewise	ℓ_1	0.500	0.719	0.413	0.276

Table 4: Measures of the indefiniteness of the input matrix $\hat{\Gamma}_n$. σ_i denote the eigenvalues of $\hat{\Gamma}_n$, while σ_i^+ denote the eigenvalues of $\hat{\Gamma}_n^+$ as defined in Section 4.1. We set $m = 400$. For data generated from each model, we report the most negative eigenvalue, the maimum eigenvalues of both the nonprojected and projected sample covariances, the sum of the negative eigenvalues, and the number of negative eigenvalues.

Model	A	n	$\min \sigma_i$	$\max \sigma_i$	$\max \sigma_i^+$	$\sum_{\sigma_i<0} \sigma_i$	$\#\{\sigma_i < 0\}$
KS	AR(0.5)	$n = 80, \tau_B = 0.3$	-0.51	17.0	15.3	-100.5	320
		$n = 160, \tau_B = 0.3$	-0.42	10.3	9.6	-74.1	240
		$n = 80, \tau_B = 0.5$	-0.93	21.3	18.1	-170.1	320
		$n = 160, \tau_B = 0.5$	-0.78	12.0	10.7	-124.6	243
MD	AR(0.6)	$n = 80, \zeta = 0.9$	-0.26	14.2	13.6	-36.2	320
		$n = 130, \zeta = 0.7$	-0.63	12.3	11.0	-116.6	270
		$n = 250, \zeta = 0.5$	-1.19	11.4	9.7	-183.6	218
		$n = 700, \zeta = 0.3$	-2.17	9.2	7.5	-228.9	188
	ER	$n = 80, \zeta = 0.9$	-0.26	13.4	12.7	-36.6	320
		$n = 130, \zeta = 0.7$	-0.62	11.7	10.4	-116.7	270
		$n = 250, \zeta = 0.5$	-1.20	10.3	8.7	-180.7	214
		$n = 700, \zeta = 0.3$	-2.17	8.5	6.9	-223.0	184

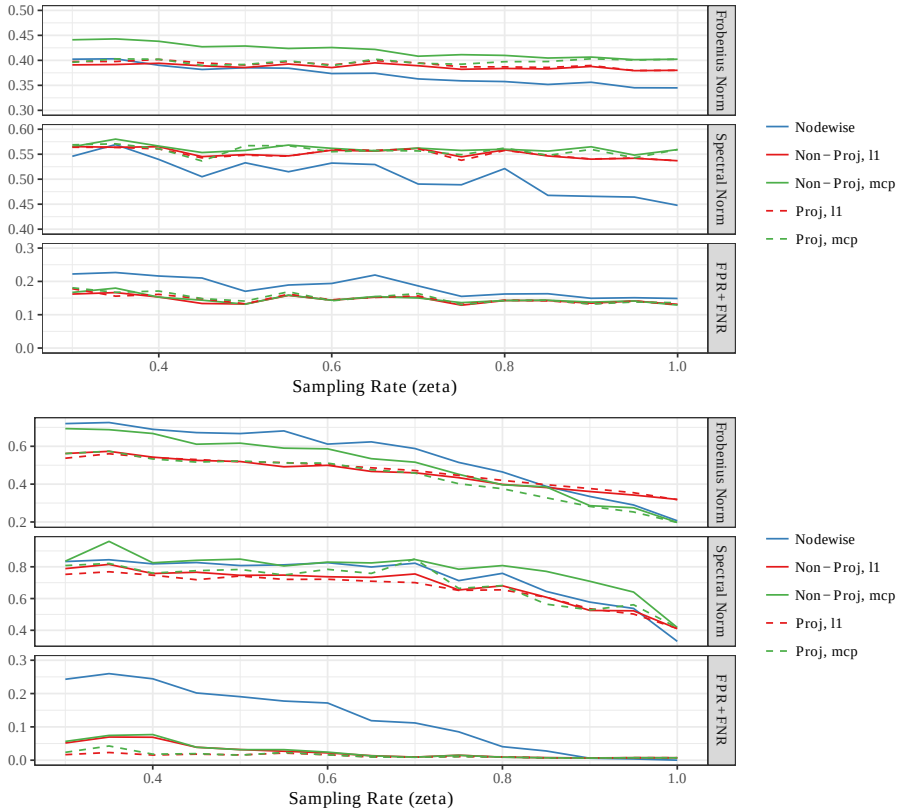
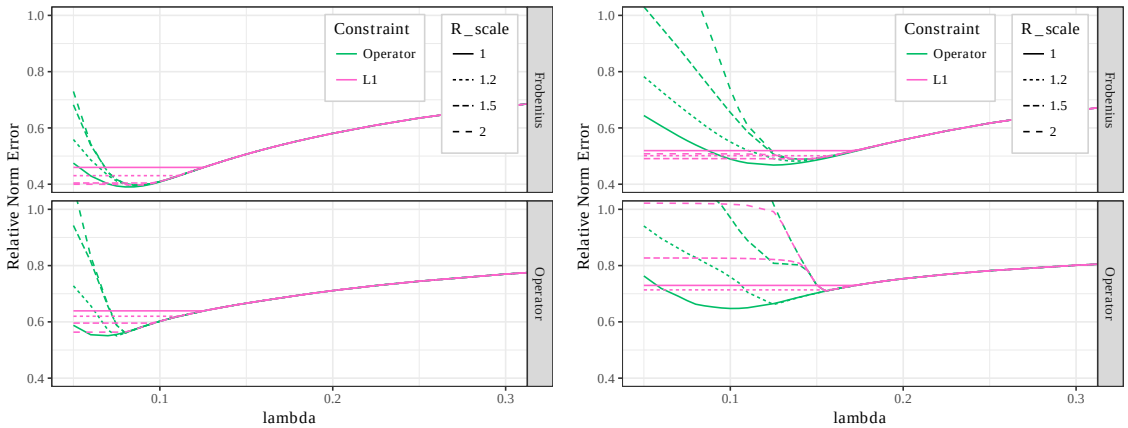


Figure 9: The performance of the various estimators for the missing data model as we vary the sampling rate. Note that these are minimums over a range of λ values. For each ζ, n is chosen so that the effective sample size for estimating off-diagonal entries of the covariance is constant, so $n\zeta^2 = 80$. On the top panel, we set A to be from an AR(0.4) with $m = 400$, while the bottom panel uses A as AR(0.8). In both cases, the effective sample size ($n\zeta^2$) is set at 80. The MCP penalty is chosen with $a = 1.5$, and we set R to be the 2 times the oracle value for each method.



(a) AR1(0.8), $n = 100$, $\zeta = 0.8$ (b) AR1(0.8), $n = 150$, $\zeta = 0.6$

Figure 10: Comparing the performance of the operator- and ℓ_1-norm side constrained estimators. For each model, errors in terms of relative Frobenius norm (top panel) and relative operator norm (bottom panel) are shown. All simulations were done with $m = 200$ and use an ℓ_1 penalty. For each method, R was set to R_scale times the oracle value.

D Additional data analysis

As discussed in Section 5, we collected voting records data from the Senate during the 112th US Congress, which was from January 3, 2011 to January 3, 2013. This data is part of the public record and open-source code to download and process the data can be found at `https://github.com/unitedstates/congress`.

Due to changes in membership, there are data on 102 senators, which we drop three of due to serving incomplete terms. The data contains 486 votes in total. We drop votes that are unanimous or unanimous within both parties, resulting in 426 votes. Roughly 2.6% of values are missing in this data.

We use the ADMM algorithm from Section 3 to estimate the nonprojected version of (2) with an ℓ_1 penalty to estimate conditional dependence graphs among senators. Since this is an exploratory analysis, we ran estimators using various levels of the penalization parameter λ and have chosen plots to display based on the number of estimated edges and maintaining visual clarity.

For our preliminary analysis, we use a modified version of the missing data estimator as described in Zhou (2019), where bills have varying missing probabilities while we estimate the edges among senators. We also demean each vote by political party, similar to the demeaning done in Hornstein et al. (2018). See our future work for a more detailed study of this estimator and its properties.

In addition to the analysis in Section 5, note that senators from the same state are often linked. Of the 33 states where both senators are in the same party, 19 of the pairs are linked.

Figure 11 plots the subgraph of senators with cross-party connections or links to those with cross-party connections from Figure 4b. We look at the NOMINATE scores of these senators (Figure 12) to determine their positions on the political spectrum. NOMINATE is a probabilistic geometric model that places each senator in a two-dimensional space representing their ideological beliefs (Poole, 2005).

Although most of the linked senators are are either on the extremes or are moderates (see Section 5), the main exception to this is Dean Heller, who is linked to Mark Warner. This outlier connection is perhaps worth further investigation as to why they are linked.

The McCaskill-Cochran connection is unsurprising, as both are among the most moderate senators from

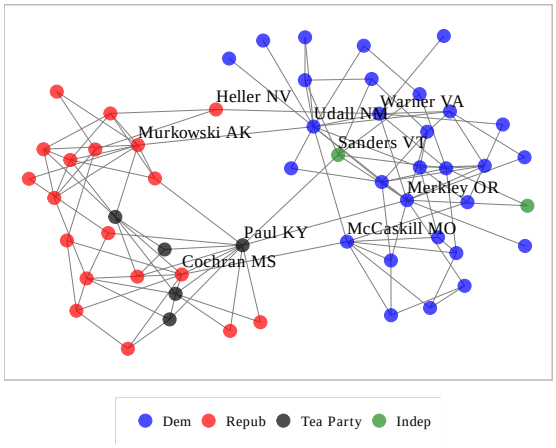


Figure 11: The subgraph of nodes that are 1- or 2-steps removed from the opposing party from 4b.

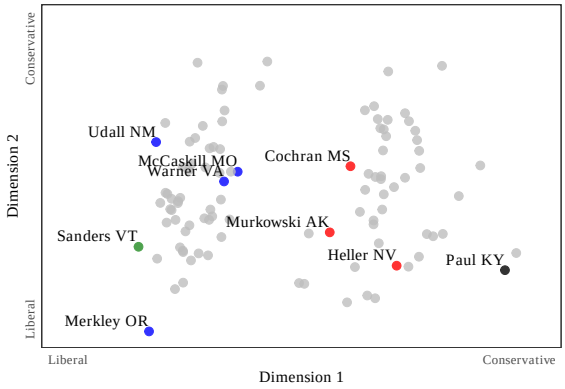


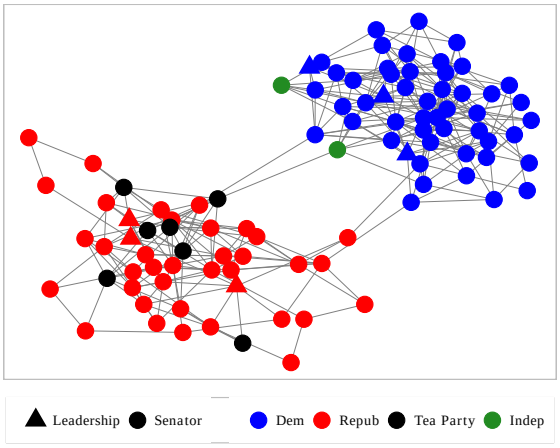
Figure 12: NOMINATE scores for the senators of the 112th Congress. Data are from `https://voteview.com/about`.

their respective parties. The Udall-Murkowski connection is more interesting since Tom Udall is viewed as a relatively liberal Democrat while Murkowski is a moderate Republican. Murkowski connecting across the aisle is expected, but why she would be linked to Udall is unknown.

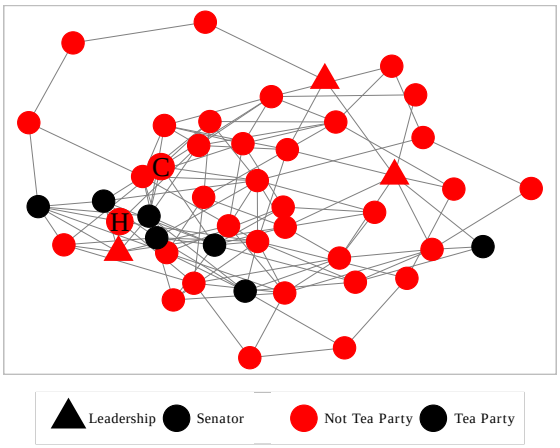
Paul-Sanders and Paul-Merkley are both interesting connections since they are between left-wing and right-wing senators. Especially since the Paul-Sanders edge has a negative estimated partial correlation, while the Paul-Merkley correlation is positive. Sanders and Merkley are similar ideologically and close allies in the Senate, so investigating the differences in their voting records would explain this discrepancy.

This could be explained by the fact that all three senators are liberal on NOMINATE's second dimension combined with Paul's particular voting patterns.

We also demonstrate the usage of nodewise regression in Figure 13. Since the sampling rate for this dataset is fairly high, we expect nodewise regression to perform well, and the results are overall similar to those in Figure 4b. Of the four cross-party links, three (Cochran-McCaskill, Sanders-Paul, and Warner-Heller) are also present in the previously estimated graph. The link between Democrat Mark Begich and Lisa Murkowski is new, a natural one since they both represent Alaska in the Senate.



(a) All Senators



(b) Republican subgraph

Figure 13: Graphs among 112th Congress senators estimated with nodewise regression. We set $\lambda = 0.12$ and $R = 10$. After estimation, the precision matrix is thresholded at 0.03.

Figure 13b exhibits similar patterns to those identified in Figure 4c. Hatch and Coburn, marked as 'H' and 'C,' still appear to be closely connected to the tea party cluster. Moran is also still disconnected from the rest of the tea party despite attending the inaugural meeting of the Senate Tea Party Caucus.