Latent Variable Selection with Convex Optimization

Tina Behrouzi, Yuan Liu

I. Abstract

In this project, we aim to investigate a convex optimization problem for latent variable selection. Its significance include automated determination and interpretation of latent factors which are used in graphical models. Typical construction of graphical models uses some fixed prior or greedy search strategy which are suboptimal and/or non-convergent in general. In our investigation, with access to only observable variables, we aim to decompose the observed relationships into a sparse and a low rank part by formulating as a convex optimization problem, the results of which can fed into graphical model construction which we omit in this paper. We pay special attention to consistency, identifiability and convergence properties of the optimization formulation. Finally, we explore convex optimization solvers for practically solving the problem using real world stock data.

II. Introduction

Graph modeling is employed to capture the complicated relationship of variables in a variety of fields, ranging from communication patterns and the stock market to bio information. In graphical model, observed variables are shown as node, and edges indicate connection between them. In practise, lots of time some of the variable are unobserved or latent. Considering these latent variable plays an important role on obtaining the relationship of the data. However, the number and structure of these variables are unknown and difficult to detect. Therefore, determining the structure of latent variables subject to observed ones has been the interest of many studies. The challenge faced when solving this optimization problem on recent data, which usually has high dimensionality and size.

It is usually considered that latent and observed variables are jointly Gaussian, and the estimation of the covariance matrix between nodes is considered as the representation of the model. Previously, the Expectation-Maximization (EM) algorithm was widely used to learn latent variables of tree-like graphs [9]. However, it has a very slow convergence and poor local optima. Furthermore, the problem can be written as Semidefinite Programming (SDP) format and solved with Interior Point (IP) method in polynomial time [2]. However, it does not consider the sparsity and structure of the problem. Therefore, it is not computationally applicable.

The method [1], defined the novel solution for finding the structure of the observed variable with respect to the latent variable. The method is based on considering the sparse and low-rank structure for the precision matrix. In section III, we precisely explain the latent variable method [1] and its' proof of convergence. In section IV, this method [1] is compared based on performance and computational time with other models. The data that we used and the result of the optimization are explained in section V. Finally, the brief conclusion is provided in section VI.

III. Latent Variable method

In this section, first, the main problem is precisely explained. Then, we describe conditions where the algorithm is expected to converge. Finally, the numerical strategy to solve the problem is mentioned.

main problem

This paper [1] considers all observed and latent variable are jointly Gaussian. This is a very common assumption for graphical models' covariance estimation. Likelihood, assuming normal distribution:

$L(\theta; x) = p_\theta(X = x|\theta)$

$$L(\theta; x) = \frac{1}{(2\pi)^{\frac{D}{2}}(det\Sigma)^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

$\theta = (\mu, \Sigma)$

Log likelihood:

$$\begin{aligned} l(\theta; x) &= logL(\theta; x) \\ &= -log((\frac{1}{2\pi})^{\frac{D}{2}}) - log(det\Sigma)^{\frac{1}{2}} \\ &\quad - \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \end{aligned}$$

Optimize over θ:

$$\max_\theta \ logL(\theta; x)$$

$$\max_\theta \ -\frac{1}{2}log(det\Sigma) - \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$$

Optimal solution:

$$\begin{aligned}&\underset{\theta}{\text{argmax}}\ -log(det\Sigma) - (x-\mu)^T\Sigma^{-1}(x-\mu)\\&let\ z = x-\mu\ \&\ \Sigma_s = zz^T\\&let\ K = \Sigma^{-1}\\&Therefore, (x-\mu)^T\Sigma^{-1}(x-\mu) =\\&tr(z^T\Sigma z) = tr(zz^T\Sigma^{-1}) = tr(\Sigma_s K)\\&\underset{\theta}{\text{argmax}}\ log(detK) - tr(\Sigma_s K)\end{aligned}$$

For K positive definite, the above problem can be cast as a convex optimization.
The $X_O \in \mathbb{R}^{p\times 1}$ is the observed variables, and $X_L \in \mathbb{R}^{r\times 1}\ r << p$ indicates the hidden variable. Then we can write $\Sigma_{(O,L)} = [\Sigma_O\ \Sigma_{OL}; \Sigma_{LO}\ \Sigma_L]$ and precision matrix as $K_{(O,L)} = [K_O\ K_{OL}; K_{LO}\ K_L]$. Based on Schur compliment:

$$K = \Sigma_O^{-1} = K_O - K_{OL}K_L^{-1}K_{LO} \tag{1}$$

$let\ K = S - L\ The\ Concentration\ matrix$
$decomposition\ into\ a\ sparse\ and\ low\ rank\ term.$
$let\ S = K_O\ which\ is\ related\ to\ X_O\ condition\ on\ X_L$
$let\ L\ = K_{OL}K_L^{-1}K_{LO}\ be\ marginalization\ over\ X_L$
$\max\limits_{S,L}\ l(\theta = S-L;\Sigma_s) = -\min\limits_{S,L}\ -l(\theta = S-L;\Sigma_s)$

Optimization Formulation:

$$\begin{aligned}&\underset{S,L}{\text{argmin}} -logdet(S-L) + tr(\Sigma_s(S-L))\\&s.t.\ S-L \succ 0, L \succeq 0\end{aligned}$$

Adding a regularization term (from main paper): The l_1 norm is applied to S to recover sparsity of the matrix, and nuclear norm is used for indicating low-rankness of L.

$$\begin{aligned}&\underset{S,L}{\text{argmin}}\ -l(\theta = S-L;\Sigma_s) + \lambda(\gamma\|S\|_1 + tr(L))\\&\underset{S,L}{\text{argmin}}\ -log\ det(S-L) + tr(\Sigma_s(S-L))\\&\qquad + \lambda(\gamma\|S\|_1 + tr(L))\\&s.t.\ S-L \succ 0, L \succeq 0\end{aligned} \tag{2}$$

Where λ is a regularisation term, and γ is a trade of between sparsity and rank terms.

A. convergence and optimizer

The fisher information, $\mathcal{I}$, is used to provide estimate of certainty, and it is defined as second derivative of the objective function Equ. 2.

$$\mathcal{I}(K) = K^{-1}\otimes K^{-1}\ \ where\ \otimes is\ tensor\ product$$

- Sparse matrix (observable variables)

This is expected not to be densely connected subgraphs, which may otherwise be mistaken for latent variable marginalization. Therefore, a minimum number of zero entries of each row (or matrix degree) should be bounded. To address this point the tangent space and the quantity to measure sparsity are defined as Equ. 3.

$$\begin{aligned}\Omega(M) &= \{N \in \mathbb{R}^{p\times p} | support(N) \subseteq support(M)\}\\ \mu(\Omega(M)) &= \max_{N\in\Omega(M), ||N||_\infty \le 1} ||N||_2\end{aligned} \tag{3}$$

Support stands for number of non-zero entries.

- Low rank matrix corresponding to effect of marginalization over latent variables

The L is expected not to be nearly aligned with the coordinate axis in order for increased chance of identifiability. In order to avoid diffusion of latent variables' effect across the observed variables, minimum singular value of matrix should not be too small. Therefore, the tangent space and the quantity to identify that is defined as Equ. 4.

$$\begin{aligned}T(M) &= \{UY_1^T + Y_2V^T | Y_1, Y_2 \in \mathbb{R}^{p\times r}\}\\ \xi(T(M)) &= \max_{N\in T(M), ||N||_2 \le 1} ||N||_\infty\end{aligned} \tag{4}$$

Where singular value decomposition of M is UDV^T. The two tangent spaces T and Ω represent local identifiability around low-rank and sparse matrix. Therefore, if these two spaces have a transverse intersection at the origin (the individual vector can be recovered from the sum), then the consistent estimate for problem 2 can be obtained.

The appropriate measurement for evaluating transversality of tangent spaces is the gain, $P_y A^T \mathcal{I} A P_Y$. Where $\mathcal{I}$ is a fisher information defined above, and P_Y is a projection into Y. $A : \mathbb{R}^{p\times p}\times\mathbb{R}^{p\times p} \to \mathbb{R}^{p\times p}$ is restricted to cartesian product $Y = \Omega \times T$. The bellow variable are defined to ensure that Ω and T are identifiable under the map of $\mathcal{I}$.

$$\begin{aligned}\alpha_\Omega &= \min_{M\in\Omega, ||M||_\infty = 1} ||P_\Omega \mathcal{I} P_\Omega(M)||_\infty\\ \alpha_T &= \min_{\rho(T,T')\le\xi(T)/2, M\in T', ||M||_2=1} \min ||P_{T'}\mathcal{I}P_{T'}(M)||_2\end{aligned}$$

Where $\rho(T, T')$ measures twisting between projection of T and T'. Moreover, α_T and α_Ω represent minimum gain of $\mathcal{I}$ restricted to Ω and T' (close to T), respectively.

In addition, maximum effect of Ω and T' in the orthogonal direction of $\Omega^\perp$ and $T'^\perp$, respectively is defined below:

$$\begin{aligned}\delta_\Omega &= \max_{M\in\Omega, ||M||_\infty = 1} ||P_{\Omega^\perp} \mathcal{I} P_\Omega(M)||_\infty\\ \delta_T &= \max_{\rho(T,T')\le\xi(T)/2, M\in T', ||M||_2=1} ||P_{T'^\perp}\mathcal{I}P_{T'}(M)||_2\end{aligned}$$

Lastly, the Equ. 5 and 6 indicates behaviour of $\mathcal{I}$ in ℓ_2 and ℓ_∞ norm. These two quantities combined with $\mu(\Omega)$ and $\xi(T)$ control (I) restricted to direct sum of $\Omega \oplus T$.

$$\beta_\Omega = \max_{M\in\Omega, ||M||_2=1} ||\mathcal{I}(M)||_2 \tag{5}$$

$$\beta_T = \max_{\rho(T,T')\le\xi(T)/2, M\in T', ||M||_\infty=1} ||\mathcal{I}(M)||_\infty \tag{6}$$

$\delta = max(\delta_\Omega, \delta_T)$ $\beta = max(\beta_\Omega, \beta_T)$ $\alpha = min(\alpha_\Omega, \alpha_T)$ The α, β, and δ bounds the information $\mathcal{I}$ to T and Ω spaces. Main assumption of paper [1]:

$$\exists v \in (0, 0.5] \; such \; that \; \frac{\delta}{\alpha} \leq 1 - 2v \tag{7}$$

This assumption is generalization of the Lasso irrepresentability condition [7].

Let n denote to number of samples $\{{X_O}^i\}_{i=1}^{\;n}$ of p observed variable. Suppose that $\mu(\Omega)\xi(T) \leq \frac{1}{6}(\frac{va}{\beta(2-v}$ and Equ. 8 hold. Also consider $n \gtrsim \frac{p}{\xi(T)^4}$, $\lambda \asymp \frac{1}{\xi(T)}\sqrt{\frac{p}{n}}$, $\lambda_m in(L*) \gtrsim \frac{1}{\xi(T)^3}\sqrt{\frac{p}{n}}$, and $degree(S*) \gtrsim \frac{1}{\xi(T)\mu(\Omega)}\sqrt{\frac{p}{n}}$. Then the main paper [1] proposes that with probability greater than $1 - 2e^{-p}$ we can have algebraic correctness 9 and error bound defined in Equ. 10.

$$\gamma \in [\frac{3\xi(T)\beta(2-v)}{v\alpha}, \frac{v\alpha}{2\mu(\Omega}\beta(2-v)] \tag{8}$$

$$sign(\hat{S}) = sign(S^*) \; and \; rank(\hat{L}) = rank(L^*) \tag{9}$$

$$||(\hat{S} - \hat{L})^{-1} - {\Sigma_O}^*||_2 \lesssim \frac{1}{\xi(T)}\sqrt{\frac{p}{n}} \tag{10}$$

Therefore, for choose of λ and γ in range of $\sqrt{\frac{p}{n}}$ the primal-dual for optimizer can be specified to guarantee optimality. For more detail explanation of convergence proof see [1, 8].

Optimizer: The main paper [1] has used general-purpose solvers [8], Newton-CG Primal Proximal Point Algorithm, for optimization in polynomial time. This is a special-purpose solver for logdet problems.

The Primal Proximal Point algorithm considers that Slater's condition for both primal and dual problems exists. Moreover, both primal and dual are convex. Therefore, the KKT condition is necessary and sufficient for them. This optimizer computes the full eigenvalue decomposition in each iteration. Moreover, this method [1] is applicable for high-dimensional problem, where number of latent and observed variable grow simultaneously.

IV. Comparison

In this section, the main paper [1] is compared with Alternating Thresholded Gradient Descent (AltGD) [9] and Proximal Gradient-based Alternating-Direction Method (PGADM) [2]. The main drawback of the Latent Variable Model (LVM) [1] is that R is updated without taking it's low dimensional structure into account. Therefore, the eigenvalues are estimated in each iteration, which causes the high time complexity of the method.

The AltGD [9] improves the performance of the LVM for the high-dimensional latent variable setting. For a large value of the dimension r, due to the complexity of eigenvalue decomposition computation of the LVM, the algorithm [1] becomes intractable. In AltGD the $L \in \mathbb{R}^{r \times r}$ is breakdown into $-ZZ^T$, where $Z \in \mathbb{R}^{r \times z}$ $z \leq r$. Therefore, choosing a small value for the z force the $rank(L) = rank(ZZ^T) \leq rank(Z) \leq z$ to be small.

As a result, the AlrGD [9] optimization problem is defined as Equ. 11. The constraint of Equ. 11 which is norm zero is not convex. Moreover, if we consider first part of objective ($trace(\Sigma_s(S + ZZ^T))$, the second derivative of that which is $2\Sigma_s$ is convex if it is Positive Semi-Definite (PSD). One of the method's assumption is that the eigenvalue of the true covariance matrix is finite and bounded bellow with a strictly positive number; therefore, the first term is convex. However, the second term is not convex because although $S + ZZ^T$ is convex(sum of convex and affine function), $-logdet(X)$ is convex and non-increasing; based on composition rule the second term become nonconvex.

Therefore, in Equ. 11 both objective and constraint are nonconvex, making the problem nonconvex optimization. Therefore, the AlrGD method applies the gradient descent method with an exact line search on the objective for both Z and S separately considering other variables to be fixed. For employing the constraint on the optimization, the hard-thresholding is applied to S in each iteration, forcing the number of non-zero entries of S to be bounded by s.

$$\min_{S,Z} < S + ZZ^T, \Sigma_s > -logdet(S + ZZ^T) \tag{11}$$
$$subject \; to \; ||S||_0 \leq s$$

Where $s > 0$ controls sparsity of S. AltGD doesn't consider any restriction for density of matrix S in the problem's objective, because it assumes that for low rank L^* there exists the α^* where $||L^*||_\infty \leq \frac{\alpha^*}{p}$ and $s^* = ||S^*||_0 \lesssim \frac{p^2}{z{\alpha^*}^2}$.

The statistical error of this technique is $max(O(\sqrt{\frac{zp}{n}}), O(\sqrt{\frac{s^* log(d)}{n}}))$, which is close to the LVM method. However, AltGD processing time is incredibly faster than LVM (Especially for high dimensional d) due to not calculating Singular Value Decomposition (SVD) in each iteration. The main time complexity of this algorithm comes from calculating the partial gradient concerning Z.

We notice that AltGD [9] is defining the L as $-ZZ^T$, which means it regards L to be Negative Semi-Definite (NSD). The main paper [1] considers L to be PSD. Moreover, it [1] shows that by replacing constraint $0 \leq L$ with $L \in space(T)$ and changing the objective 2 part of the $trace(L)$ to nuclear norm $||L||_*$, the algorithm will also converge. However, considering L to be completely NSD is in contrast with the structure of the latent variable. We evaluate the performance of algorithm with change of $L = ZZ^T$, and we name it AltGD_neg. The comparison with the optimum objective of AltGD_neg,

AltGD, and LVM are reported in the next section.

The PGADM [2] uses classic and gradient-based Altering Direction Method of Multipliers (ADMM) instead of Newton-CG Primal Proximal Point Algorithm (PPA). The PGADM first rewrite the objective 2 as Equ. 12. This problem is convex, and the objective is separable into three function $f(R)$, $g(S)$, and $h(L)$. However, the constraint is dependent on all variables. In each iteration, variables are updated separately considering proximal mapping of the function considering the other two variables are constant. With good choice of α and β this method obtains global convergence.

$$min < R, \Sigma_s > -logdet(R) + \alpha||S||_1 + \beta tr(L) + I(L) \quad subject\ to\ R - S + L = 0 \tag{12}$$

Where $I(R)$ is zero for $0 \leq L$ and else infinity. This method does not compute SVD in each iteration, therefore the algorithm has less time complexity. The ADMM is 5 to 35 times faster than PPA, and the AltGD method is between 30 to 50 times quicker than ADMM.

We should also mention that some new methods consider differential network to be combination of sparse and low-rank matrix, e.g. [3]. The differential network is characterized by the contrast of two precision matrices, from two groups of samples distributed according to the latent variable Gaussian graphical model. These methods provide a more robust precision matrix regarding changes in sparsity.

V. Learning

For demonstration, we investigate stock price returns of selected interval over time range (see Table II). CVXPY[4] and SDPT3[6] are used to solve the optimization problem. Due to solvers differences(interior point in SDPT3 and ADMM in CVXPY's SCS backend), they yield results of different accuracy and we find SCS has trouble in making progress and returns warning of inaccurate answers even with increased iteration limits. Hence SDPT3 is chosen as our solver. As noted in [5] and seen in previous section, priors of constants related to γ and λ of the optimization problem Equ. 2 are usually unknown, thus authors perform cross validation on samples to select suitable parameters. We follow this approach by sweeping parameters ($\lambda \in \sqrt{\frac{p}{n}}$, $\gamma \in [0.1, 10]$, n is the number of samples, p is the number of observables). We divide total samples Z into training Z_{train} and test Z_{test} respectively in portions of $\frac{4}{5}$ and $\frac{1}{5}$ of the total. We then input them into optimization Equ 2 ($\Sigma_s = \frac{1}{m} Z_{train} Z_{train}^T, Z \in \mathbb{R}^{p \times m}$). Z_{test} is then used along with the output matrices L and S of the optimization problem to give objective value.

Tickers
GOOGL, AMZN, REGN, AMGN, TM, HMC, EA, ATVI

TABLE I: Tickers

Return Interval	Time Range	Total Samples
weekly	2005-01-01 to 2018-01-01	503
biweekly	2005-01-01 to 2018-01-01	251
monthly	2005-01-01 to 2018-01-01	117

TABLE II: Stock Experiments

Historical stock data are retrieved and returns are calculated as $return_i = (price_i - price_{i-1})/price_{i-1}$ for each interval and time range as indicated in Table II. We had considered 2000 to 2018 time range, however it was later found certain companies having missing data and thus shorten the time range to 2005 to 2018. Daily return was considered but not included in our analysis may not reveal as much information due to day trading noise.

Experiment	n	$\sqrt{\frac{p}{n}}$	$(\lambda, \gamma)_{sel}$	Objective
weekly	402	0.141	(0.001, 4.75)	-37.9749
biweekly	200	0.200	(0.0075, 4)	-28.9994
monthly	93	0.293	(0.01, 5)	-24.0565

TABLE III: Solving Parameters: Best objective may not result in most suitable parameters. Here we select suitable parameters based on solved variable characteristics.

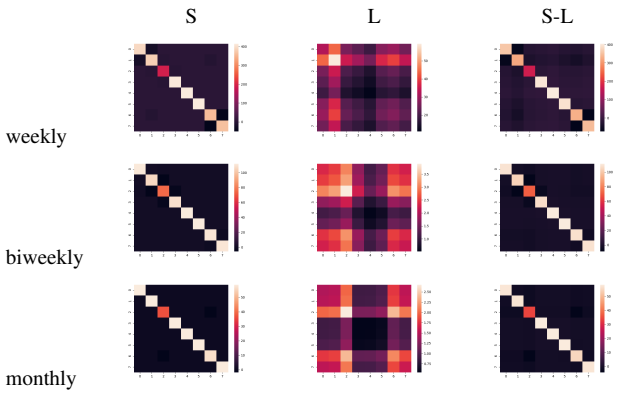

TABLE IV: Solved Variables

Parameters are swept so that combined effects of λ and γ of the sparsity penalization term is near $\sqrt{\frac{p}{n}}$. From examining a typical sweep (see Fig 1) and obtained matrices L, S, we found that SDPT3 solver returns favourable decrease in objective value at the expense of becoming further away from the ideal regime of S being sparse and L being low rank.

Using SPDT3, the optimization problem is sensitive to parameters $(\lambda, gamma)$. Specifically, parameters were swept at fairly large step sizes at first in order to cover more range and speed up computation, however this tends to miss parameter space in which optimization

output has a high dynamic range. We further swept in these regions in order to obtain suitable S and L. Parameters resulting in more reasonable L and S are selected in Table IV. The resulting matrices then describe covariance model of the LVM. Resulting solved L for each experiment is calculated to have 1 non-zero eigenvector and $p-1$ eigenvectors associated with 0 eigenvalue where $L \in \mathbb{R}^{p\times p}$ Since SPDT3 requires hermetian matrix variables for the SDP solver, then rank L is the equal to p minus total number of eigenvectors associated with the zero eigenvalue.

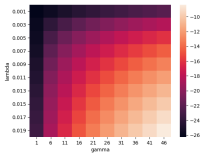

Fig. 1: Typical Parameter Sweep $(\lambda, \gamma) \rightarrow$ objective of Problem 2

Experiment/method	LVM	AltGD	AltGD_neg
weekly	-34.6845	-34.6997	-33.4612
biweekly	-40.1401	-40.4255	-39.8500
monthly	-28.2852	-24.8420	-22.0643

TABLE V: Comparison of minimum objective derived from AltGD (for z=2) [9], LVM [1], and AltGD_neg (Our change to AltGD method)

Based on table V, the minimum objective function of all three methods are very close together. Moreover, considering the L to be PSD has not improved the result of AltGD. The most difference between these methods is simulation time. However, due to small matrices, computation time is around one second for all solvers.

Extensions

The method can be used for assigning latent interpretation. In [5], latent variables are identified and matched with potential covariates data that are also provided (such as currency exchange rate, GPD index, inflation rate) alongside with observable variables. A set of candidate factor models are tested using specialized matching algorithm.

VI. Conclusion

In conclusion latent model inference and selection is tackled by assuming a graphical model from the exponentially family conditioned on latent variables and formulating it as a logdet convex optimization problem. We investigated the main paper [1] and expanded in detail its related and surrounding literature. Conditions for identifiability, convergence were studied in which authors further gave validation using known synthetic data generator of desired properties. Due to dependency on sensitive parameters, practical implementation typically involve rough estimations of parameter range, as indicated by the main authors and this difficulty is also encountered in our experiments using real world data. Additionally, different optimization methods were explored in experiments where we found varying degrees of satisfaction of the generated solution from CVXPY's SCS, SPDT3, and AltGD. From this observation, it is likely that real world stock data used in our experiment is itself challenging due to lack of externally provided signals which might aid in the inference.

References

[1] V Chandrasekaran, P.A. Parrilo, and A.S. Willsky. "Latent Variable Graphical Model Selection via Convex Optimization". In: *The Annals of Statistics* 40.4 (2012), pp. 1935–1967.

[2] Shiqian Ma, Lingzhou Xue, and Hui Zou. "Alternating direction methods for latent variable Gaussian graphical model selection". In: *Neural computation* 25.8 (2013), pp. 2172–2198.

[3] Sen Na, Mladen Kolar, and Oluwasanmi Koyejo. "Estimating differential latent variable graphical models with applications to brain connectivity". In: *arXiv preprint arXiv:1909.05892* (2019).

[4] Diamond S. and Boyd S. "CVXPY: A Python-Embedded Modeling Language for Convex Optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5. URL: http://jmlr.org/papers/v17/15-408.html.

[5] A Taeb and A Chandrasekaran. "Interpreting latent variables in factor models via convex optimization". In: *Mathematical Programming, Vol. 167, 129–154* (2018).

[6] Toh K.C. Tutuncu R.H. and M.J. Todd. "Solving semidefinite-quadratic-linear programs using SDPT3". In: *Mathematical Programming* Ser. B.95 (2003), pp. 189–217.

[7] Martin J Wainwright. "Sharp thresholds for High-Dimensional and noisy sparsity recovery using $\mathcal{L}_1$-Constrained Quadratic Programming (Lasso)". In: *IEEE transactions on information theory* 55.5 (2009), pp. 2183–2202.

[8] Chengjing Wang, Defeng Sun, and Kim-Chuan Toh. "Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm". In: *SIAM Journal on Optimization* 20.6 (2010), pp. 2994–3013.

[9] Pan Xu, Jian Ma, and Quanquan Gu. "Speeding up latent variable gaussian graphical model estimation via nonconvex optimization". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1933–1944.