Fast Projected Newton-like Method for Precision Matrix Estimation under Total Positivity

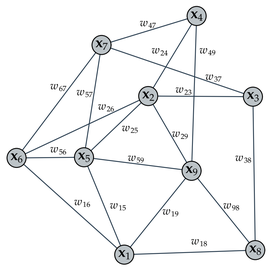
Jian-Feng Cai, José Vinícius de M. Cardoso, Daniel P. Palomar, Jiaxi Ying

Code: https://github.com/jxying/mtp2

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

1 Introduction

1.1 MTP_2 Gaussian graphical models



- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an intuitive way to represent relationships between entities.
- Nodes: $\mathcal{V} = \{1, 2, \ldots, p\}$ correspond to the random variables.
- Edges: $\mathcal{E} = \{(1,2), \ldots, (i,j), \ldots\}$ encodes conditional dependence between variables.
- Total positivity is also called multivariate totally positive of order two (MTP_2). A multivariate Gaussian distribution is MTP_2 if and only if precision matrix satisfies

$$\Theta_{ij} \leq 0, \quad \text{for all } i \neq j. \tag{1}$$

Equivalently, Θ is an *M-matrix*, or all partial correlations are nonnegative.

- Total positivity is a special form of positive dependence:

$$\mathrm{MTP}_2 \Longrightarrow \Sigma_{ij} \geq 0, \quad \text{for any } i \neq j. \tag{2}$$

- This model has broad applications including financial time series [1], taxonomic reasoning [2], and factor analysis in psychometrics [3].

1.2 Existing algorithms

- **Block coordinate descent** [4]: Updates one column/row at a time by solving a nonnegative quadratic program, and cyclically updates all columns/rows. Each cycle requires $O(p^4)$ operations.
- **Proximal point algorithm** [5]: Compute an inexact Newton direction from a $p^2 \times p^2$ linear system during each inner loop iteration, which is time-consuming in high-dimensional scenarios.
- **Projected gradient method** [6]: Offer computational efficiency, requiring only $O(p^3)$ operations per iteration, but it often suffers from a low convergence rate.

1.3 Problem formulation

- Learning Gaussian graphical models under total positivity can be formulated as

$$\begin{array}{ll} \underset{\Theta}{\text{minimize}} & -\log\det(\Theta) + \mathrm{tr}\,(\Theta S) + \sum_{i \neq j} \lambda_{ij} |\Theta_{ij}|, \\ \text{subject to} & \Theta_{ij} = 0, \ \forall (i,j) \in \mathcal{E}, \ \Theta \in \mathcal{M}^p. \end{array} \tag{3}$$

where λ_{ij} is regularization parameter, $\mathcal{M}^p := \left\{\Theta \in \mathbb{S}^p_{++} | \, \Theta_{ij} \leq 0, \ \forall i \neq j\right\}$, and $\mathcal{E}$ is disconnectivity set that forces the pairs of nodes to be disconnected.

- The disconnectivity set can be obtained in several ways: (i) Prior knowledge indicates that certain edges do not exist; (ii) it may be estimated from initial estimators; (iii) it can be obtained in tasks of learning structured graphs such as bipartite graph.
- The maximum likelihood estimator (MLE) in a Gaussian MTP_2 distribution exists and is unique with probability 1 if $n \geq 2$ [2, 3].

2 Main Results

2.1 Proposed algorithm

Step 1: Partition variables into *restricted* set and *free* set

- The *restricted* set $\mathcal{I}_k$ in k-th iteration is set as

$$\mathcal{I}_k := \mathcal{T}(\Theta_k, \epsilon_k) \cup \mathcal{E}. \tag{4}$$

where $\mathcal{T}(\Theta, \epsilon) := \left\{(i,j) \in [p]^2 \, \middle| \, -\epsilon \leq \Theta_{ij} \leq 0, \ \left[\nabla f(\Theta)\right]_{ij} < 0\right\}$.

- Variables in $\mathcal{T}(\Theta_k, \epsilon_k)$ are close to boundary and tending to move outside feasible region.

Step 2: Compute the search direction

- We use the special structure of the Hessian of the problem:

$$\left[P_k\right]_{\mathcal{I}_k^c} = \left[\Theta_k \mathcal{P}_{\mathcal{I}_k^c}(\nabla f(\Theta_k))\, \Theta_k\right]_{\mathcal{I}_k^c}, \tag{5}$$

where $\left[\mathcal{P}_{\mathcal{I}_k^c}(A)\right]_{ij} = A_{ij}$ if $(i,j) \in \mathcal{I}_k^c$, and zero otherwise.

Step 3: Compute the step size

- We try the step size $\gamma_k \in \{\beta^0, \beta^1, \beta^2, \ldots\}$, until we find the smallest $m \in \mathbb{N}$ such that the next iterate Θ_{k+1} with $\gamma_k = \beta^m$ satisfies $\Theta_{k+1} \succ 0$ and

$$f(\Theta_{k+1}) \leq f(\Theta_k) - \alpha\beta^m \left\langle \left[\nabla f(\Theta_k)\right]_{\mathcal{I}_k^c}, \left[P_k\right]_{\mathcal{I}_k^c} \right\rangle - \alpha \left\langle \left[\nabla f(\Theta_k)\right]_{\mathcal{I}_k}, \left[\Theta_k\right]_{\mathcal{I}_k} - \left[\Theta_{k+1}\right]_{\mathcal{I}_k} \right\rangle. \tag{6}$$

The proposed algorithm requires the same orders of per-iteration computation and memory costs as those of projected gradient method, while it ensures faster convergence by integrating second-order information into the search direction.

Algorithm 1: Fast Projected Newton-like method (FPN)

Input: Sample covariance S, λ, $\hat{w}^{(0)}$;
for $k = 1, 2, \ldots$ **do**
  Identify the *restricted* set $\mathcal{I}_k$ and *free* set $\mathcal{I}_k^c$;
  Compute the approximation Newton direction over *free* set:

$$\left[P_k\right]_{\mathcal{I}_k^c} = \left[\Theta_k \mathcal{P}_{\mathcal{I}_k^c}(\nabla f(\Theta_k))\Theta_k\right]_{\mathcal{I}_k^c};$$

  $m \leftarrow 0$;
  repeat
    Update $\left[\Theta_{k+1}\right]_{\mathcal{I}_k} = 0, \quad \left[\Theta_{k+1}\right]_{\mathcal{I}_k^c} = \mathcal{P}_\Omega\left(\left[\Theta_k\right]_{\mathcal{I}_k^c} - \beta^m \left[P_k\right]_{\mathcal{I}_k^c}\right)$;
  until $\Theta_{k+1} \succ 0$, *and equation* (6) *holds*;;
end
Output: Θ_k.

2.2 Theoretical results

Assumption 1. For the minimizer $\Theta^\star$ of Problem (3), we assume that the gradient of the objective function f at $\Theta^\star$ satisfies

$$\left[\nabla f(\Theta^\star)\right]_{ij} < 0, \quad \forall (i,j) \in \mathcal{V} \setminus \mathcal{E},$$

where $\mathcal{V} = \left\{(i,j) \in [p]^2 \, \middle| \, \left[\Theta^\star\right]_{ij} = 0\right\}$, and $\mathcal{E}$ is the disconnectivity set.

- This assumption is equivalent to the strict complementary slackness condition.
- Note that the minimizer must satisfy $\left[\nabla f(\Theta^\star)\right]_{ij} \leq 0$, for any $(i,j) \in \mathcal{V} \setminus \mathcal{E}$. This assumption requires that $\left[\nabla f(\Theta^\star)\right]_{ij}$ is strictly negative for any $(i,j) \in \mathcal{V} \setminus \mathcal{E}$.

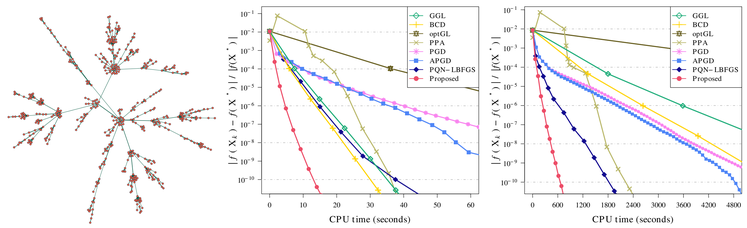
Theorem 2. *The sequence $\{\Theta_k\}$ generated by Algorithm 1 converges to the minimizer $\Theta^\star$, with $\{f(\Theta_k)\}$ monotonically decreasing. Moreover, under Assumption 1, there exists some $k_o \in \mathbb{N}_+$ such that*

$$\mathcal{I}_k^c = \mathrm{supp}(\Theta^\star), \quad \forall\, k \geq k_o.$$

In other words, the set of free variables is consistent with the support of $\Theta^\star$ when $k \geq k_o$.

3 Experiments

3.1 Synthetic data



(a) BA graph (b) $p = 1000$ (c) $p = 5000$

Figure 1: Relative errors of the objective function values versus time on BA graphs with 1000 and 5000 nodes respectively.

3.2 *Concepts* data

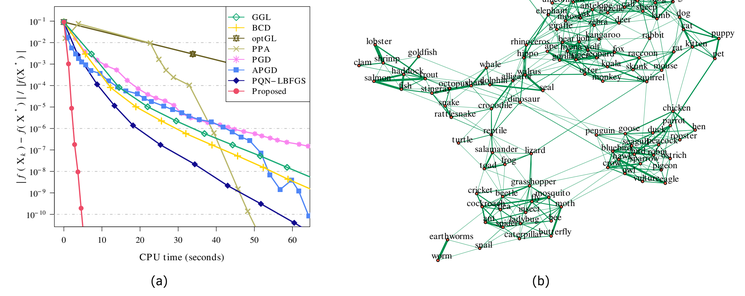


Figure 2: (a) Relative errors of the objective function values versus time for different algorithms on the *concepts* data set, consisting of 1000 nodes. (b) The learned subgraph, consisting of 132 nodes.

References

[1] R. Agrawal, U. Roy, and C. Uhler, "Covariance Matrix Estimation under Total Positivity for Portfolio Selection," *Journal of Financial Econometrics*, vol. 20, no. 2, pp. 367–389, 2020.

[2] M. Slawski and M. Hein, "Estimation of positive definite M-matrices and structure learning for attractive Gaussian Markov random fields," *Linear Algebra and its Applications*, vol. 473, pp. 145–179, 2015.

[3] S. Lauritzen, C. Uhler, and P. Zwiernik, "Maximum likelihood estimation in Gaussian models under total positivity," *The Annals of Statistics*, vol. 47, no. 4, pp. 1835–1863, 2019.

[4] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constrints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, 2017.

[5] Z. Deng and A. M.-C. So, "A fast proximal point algorithm for generalized graph laplacian learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5425–5429.

[6] J. Ying, J. V. d. M. Cardoso, and D. P. Palomar, "Adaptive estimation of graphical models under total positivity," in *International Conference on Machine Learning*, 2023, pp. 40 054–40 074.