A Riemannian Geometric Approach for Graph Learning

Xiang Zhang

School of Information Science and Engineering
Southeast University

Jan. 9th, 2025

Contents

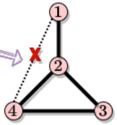
Contents

Motivations

- Gausssian Graphic Models (GGMs) [YL07]

$$\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{\dagger} \qquad \mathcal{G} = \{\mathcal{A}, \mathcal{E}\}$$

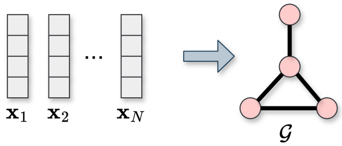
$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \qquad \begin{bmatrix} 1 & -1 & 0 & {\color{red}0} \\ -1 & 6 & -2 & -3 \\ 0 & -2 & 3 & -1 \\ {\color{red}0} & -3 & -1 & 4 \end{bmatrix}$$



$$\boldsymbol{\Theta}[14] = \boldsymbol{\Theta}[41] = 0$$
$$\Leftrightarrow \mathbf{x}[1] \perp \mathbf{x}[4] \mid \mathbf{x}[\mathcal{A} \backslash \{1,4\}]$$

- Graph Learning in GGMs [EPO17]

Graph Learning Task

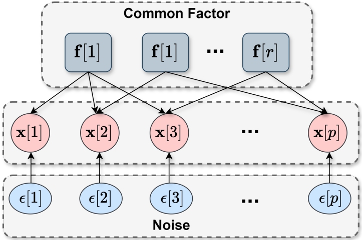


Regularized MLE Problem

$$\min_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}) = \underbrace{\operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - \log\det(\boldsymbol{\Theta})}_{g(\boldsymbol{\Theta})} + \beta h(\boldsymbol{\Theta})$$

$$\text{s. t. } \boldsymbol{\Theta} \in \mathcal{S}^p_{++}, \ \boldsymbol{\Theta}[ij] \leq 0, \forall i \neq j.$$

- Low-rank Factor Analysis Model in GGMs [ZYP22]

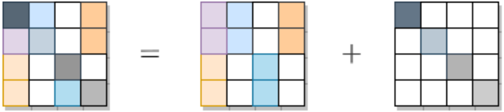


$$\left.\begin{array}{l} \mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^r \\ \\ \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \in \mathbb{R}^p \\ \\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{N}) \in \mathbb{R}^p \end{array}\right\} \quad \mathbf{x} = \mathbf{B}\mathbf{f} + \epsilon$$

The covariance matrix of $\mathbf{x}$ is

$$\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{B}\mathbb{E}[\mathbf{f}\mathbf{f}^\top]\mathbf{B}^\top + \mathbb{E}[\epsilon\epsilon^\top] = \mathbf{K} + \mathbf{N} = \mathbf{\Sigma},$$

which exhibit a low-rank and diagonal (LRaD) structure:



$$\mathbf{\Sigma} \quad = \quad \mathbf{K} \quad + \quad \mathbf{N}$$

Motivations

What are the motivations?

Motivations

- Does the LRaD structure of the covariance matrix improve the estimation of the precision matrix?
- What effect does r have on the graph (precision matrix) estimation?
- How to choose r reasonably?
- How to handle the LRaD constraint when optimizing?

Motivations

- Does the LRaD structure of the covariance matrix improve the estimation of the precision matrix?
- What effect does r have on the graph (precision matrix) estimation?
- How to choose r reasonably?
- How to handle the LRaD constraint when optimizing?

Motivations

- Does the LRaD structure of the covariance matrix improve the estimation of the precision matrix?
- What effect does r have on the graph (precision matrix) estimation?
- How to choose r reasonably?
- How to handle the LRaD constraint when optimizing?

Motivations

- Does the LRaD structure of the covariance matrix improve the estimation of the precision matrix?
- What effect does r have on the graph (precision matrix) estimation?
- How to choose r reasonably?
- How to handle the LRaD constraint when optimizing?

Contents

- Learning Precision Matrices with LRaD Structures

For covariance matrices with LRaD structures, we have [HS81]:

$$\mathbf{\Theta} = \mathbf{\Sigma}^{-1} = (\mathbf{K}+\mathbf{N})^{-1} = -\underbrace{\mathbf{N}^{-1}\left(\mathbf{I}+\mathbf{K}\mathbf{N}^{-1}\right)^{-1}\mathbf{K}\mathbf{N}^{-1}}_{\mathbf{L}} + \underbrace{\mathbf{N}^{-1}}_{\mathbf{D}},$$

where $\mathbf{L} \in \mathcal{S}_{+}^{p,\leq r}$ and $\mathbf{D} \in \mathcal{D}_{++}^{p}$. Then,

Original Problem

$$\min_{\mathbf{\Theta}} f(\mathbf{\Theta}) = \underbrace{\mathrm{tr}(\mathbf{\Theta}\mathbf{S}) - \log\det(\mathbf{\Theta})}_{g(\mathbf{\Theta})} + \beta h(\mathbf{\Theta})$$

$$\text{s.t. } \mathbf{\Theta} \in \mathcal{S}_{++}^{p},\ \mathbf{\Theta}[ij] \leq 0, \forall i \neq j.$$

Graph Learning with the LRaD structure

$$\min_{\mathbf{\Theta}\in\mathcal{Q}^{\leq r}} \mathrm{tr}(\mathbf{\Theta}\mathbf{S}) - \log\det(\mathbf{\Theta}) + \beta h(\mathbf{\Theta})$$

$$\text{with } \mathcal{Q}^{\leq r} := \{\mathbf{\Theta} : \mathbf{\Theta} \in \mathcal{S}_{++}^{p},\ \mathbf{\Theta}[ij] \leq 0, \forall i \neq j,\ \mathbf{\Theta} = -\mathbf{L}+\mathbf{D}, \mathbf{D} \in \mathcal{D}_{++}^{p}, \mathbf{L} \in \mathcal{S}_{+}^{p,\leq r}\}$$

Problem Formulation

- Theoretical Analysis

  We consider two cases, where $r \geq r^*$ and $r < r^*$ (r^* is the real rank).
  First, define

$$\varrho = \min_{\boldsymbol{\Theta} \in \mathcal{Q}^{\leq r}} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\|_{\mathrm{F}},$$

  where $\boldsymbol{\Theta}^*$ is the real graph.
  Then, we make the following assumptions

Assumption 1

There exist constants $C_L, C_U > 0$ such that $C_L \leq \lambda_{\min}(\boldsymbol{\Sigma}^*) \leq \lambda_{\max}(\boldsymbol{\Sigma}^*) \leq C_U$, where $\boldsymbol{\Sigma}^*$ is the real covariance matrix of the observed data.

Assumption 2

Let the set $\mathcal{F} := \{(i,j) : \boldsymbol{\Theta}^*[i,j] \neq 0, i \neq j\}$ and, $\mathcal{F}_\varrho := \{(i,j) : \boldsymbol{\Theta}_\varrho[i,j] \neq 0, i \neq j\}$. Then, there exists a constant m such that $\mathrm{card}(\mathcal{F}) \leq m$ and $\mathrm{card}(\mathcal{F}_\varrho) \leq m$.

Problem Formulation

- Theoretical Analysis
  We consider two cases, where $r \geq r^*$ and $r < r^*$ (r^* is the real rank).
  First, define

$$\varrho = \min_{\boldsymbol{\Theta} \in \mathcal{Q}^{\leq r}} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\|_{\mathrm{F}},$$

where $\boldsymbol{\Theta}^*$ is the real graph.
Then, we make the following assumptions

Assumption 1

There exist constants $C_L, C_U > 0$ such that $C_L \leq \lambda_{\min}(\boldsymbol{\Sigma}^*) \leq \lambda_{\max}(\boldsymbol{\Sigma}^*) \leq C_U$, where $\boldsymbol{\Sigma}^*$ is the real covariance matrix of the observed data.

Assumption 2

Let the set $\mathcal{F} := \{(i,j) : \boldsymbol{\Theta}^*[i,j] \neq 0, i \neq j\}$ and, $\mathcal{F}_\varrho := \{(i,j) : \boldsymbol{\Theta}_\varrho[i,j] \neq 0, i \neq j\}$. Then, there exists a constant m such that $\mathrm{card}(\mathcal{F}) \leq m$ and $\mathrm{card}(\mathcal{F}_\varrho) \leq m$.

Problem Formulation

- Theoretical Analysis

  We consider two cases, where $r \geq r^*$ and $r < r^*$ (r^* is the real rank).
  First, define

$$\varrho = \min_{\boldsymbol{\Theta} \in \mathcal{Q}^{\leq r}} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\|_{\mathrm{F}},$$

  where $\boldsymbol{\Theta}^*$ is the real graph.
  Then, we make the following assumptions

Assumption 1

There exist constants $C_L, C_U > 0$ such that $C_L \leq \lambda_{\min}(\boldsymbol{\Sigma}^*) \leq \lambda_{\max}(\boldsymbol{\Sigma}^*) \leq C_U$, where $\boldsymbol{\Sigma}^*$ is the real covariance matrix of the observed data.

Assumption 2

Let the set $\mathcal{F} := \{(i,j) : \boldsymbol{\Theta}^*[i,j] \neq 0, i \neq j\}$ and, $\mathcal{F}_\varrho := \{(i,j) : \boldsymbol{\Theta}_\varrho[i,j] \neq 0, i \neq j\}$. Then, there exists a constant m such that $\mathrm{card}(\mathcal{F}) \leq m$ and $\mathrm{card}(\mathcal{F}_\varrho) \leq m$.

Problem Formulation

- Theoretical Analysis
  - $r \geq r^*$

Theorem 1

Under Assumptions 1 - 2, if $r \geq r^*$, $\beta \asymp \sqrt{\log p/N}$, *and* $\widehat{\boldsymbol{\Theta}}$ *is the graph estimated from the proposed model, we have*

$$\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|_{\mathrm{F}} = \mathcal{O}\left\{\max\left(\underbrace{\left(\frac{rp}{N}\right)^{\frac{1}{2}}}_{A_{N,p,r}}, \underbrace{\left(\frac{(m+p)\log p}{N}\right)^{\frac{1}{2}}}_{B_{N,p,m}}\right)\right\}.$$

- Theoretical Analysis
  - $r < r^*$

Theorem 2

Under Assumptions 1 - 2, if $r < r^*$, $\beta \asymp \sqrt{\log p/N}$, $\varrho = \mathcal{O}\{\max(A_{N,p,r^*}, B_{N,p,m})\}$, *and* $\widehat{\boldsymbol{\Theta}}$ *is the graph estimated from the proposed model, then we have*

$$\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|_{\mathrm{F}} = \mathcal{O}\left\{\max\left(\underbrace{\left(\frac{r^*p}{N}\right)^{\frac{1}{2}}}_{A_{N,p,r^*}}, \underbrace{\left(\frac{(m+p)\log p}{N}\right)^{\frac{1}{2}}}_{B_{N,p,m}}\right)\right\}.$$

- Reformulation as Fixed-rank Optimization Problems
  If we fix $\mathbf{L}$ to be $l-$rank, which has the low-rank decomposition, i.e., $\mathbf{L} = \mathbf{Y}\mathbf{Y}^\top, \mathbf{Y} \in \mathbb{R}^{p\times l}$, then we have

$$\min_{\boldsymbol{\Theta}\in\mathcal{U}^{p,l}} \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - \log\det(\boldsymbol{\Theta}) + \beta h(\boldsymbol{\Theta})$$
$$\text{s.t. } \mathcal{U}^{p,l} := \{\boldsymbol{\Theta} : \boldsymbol{\Theta} = \mathbf{D} - \mathbf{Y}\mathbf{Y}^\top, \mathbf{Y} \in \mathbb{R}^{p\times l}, \mathbf{D} \in \mathcal{D}^p_{++}\}, \boldsymbol{\Theta} \in \mathcal{S}^p_{++}, \boldsymbol{\Theta}[ij] \leq 0, \forall i \neq j.$$

  Ideally, if $r \geq r^*$ and the rank l coincide with rank of the optimal solution of original problem, then the fixed rank optimization problem is equivalent to the original problem. However, the rank of the optimal solution of original problem is usually unknown.

- Reformulation as Fixed-rank Optimization Problems

Proposition 1

A local minimizer $(\mathbf{D}, \mathbf{Y})$ *of the fixed rank problem provides a stationary point* $(\mathbf{D}, \mathbf{L})$, $\mathbf{L} = \mathbf{Y}\mathbf{Y}^\top$, *of the original problem with* $r > r^*$ *if and only if the matrix*

$$\boldsymbol{\Upsilon}_Y = (\mathbf{D} - \mathbf{Y}\mathbf{Y}^\top)^{-1} - \mathbf{S} + \beta\widetilde{\mathbf{I}} - \boldsymbol{\Gamma}_Y + \boldsymbol{\Xi}_Y \succeq \mathbf{0},$$

where $\widetilde{\mathbf{I}} = \mathbf{1} - \mathbf{I}$. *Here,* $\mathbf{1}$ *is a* $p \times p$ *matrix of ones. Furthermore,* $\boldsymbol{\Gamma}_Y, \boldsymbol{\Xi}_Y \in \mathbb{R}^{p \times p}$ *are the dual variables associated with the constraints of the fixed rank problem in the first-order KKT conditions.*

Contents

- Riemannian Manifold Optimization

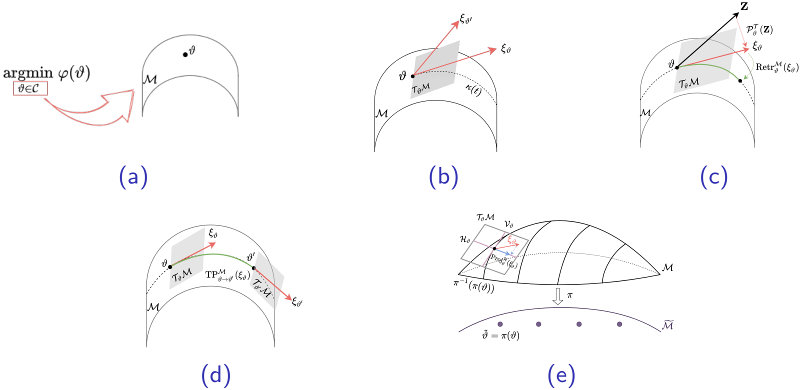
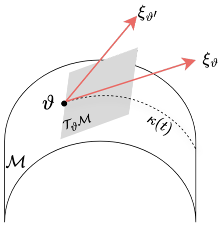


Figure: The illustration of Riemannian manifold optimization: (a) Overview of manifold optimization; (b) Tangent vectors and tangent space; (c) Projection onto $\mathcal{T}_\vartheta\mathcal{M}$ and retraction back to $\mathcal{M}$; (d) Vector transport; (e) Quotient manifold $\widetilde{\mathcal{M}}$ of manifold $\mathcal{M}$.

Algorithm

- Key Ingredients of Riemannian Manifold Optimization
  - **Curve:** For a manifold $\mathcal{M}$, a smmoth mapping $\kappa : \mathbb{R} \to \mathcal{M} : t \mapsto \kappa(t)$ is termed a curve in $\mathcal{M}$.
  - **Tangent vector:** A tangent vector $\boldsymbol{\xi}_{\boldsymbol{\vartheta}}$ of $\mathcal{M}$ at point $\boldsymbol{\vartheta} \in \mathcal{M}$ is a mapping from $\mathcal{B}_{\boldsymbol{\vartheta}}(\mathcal{M})$ to $\mathbb{R}$ such that there exists a curve κ on $\mathcal{M}$ with $\kappa(0) = \boldsymbol{\vartheta}$, satisfying $\boldsymbol{\xi}_{\boldsymbol{\vartheta}} b = \frac{\mathrm{d}(b(\kappa(t)))}{\mathrm{d}t}|_{t=0}$ for all $b \in \mathcal{B}_{\boldsymbol{\vartheta}}(\mathcal{M})$, where $\mathcal{B}_{\boldsymbol{\vartheta}}(\mathcal{M})$ denotes the set of smooth real-valued functions defined on a neighborhood of $\boldsymbol{\vartheta}$.
  - **Tangent space:** For every point $\boldsymbol{\vartheta} \in \mathcal{M}$, it is attached to a tangent space $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$, which contains all tangent vectors $\boldsymbol{\xi}_{\boldsymbol{\vartheta}}$ to $\mathcal{M}$ at $\boldsymbol{\vartheta}$.



- Key Ingredients of Riemannian Manifold Optimization
  - **Riemannian metric:**

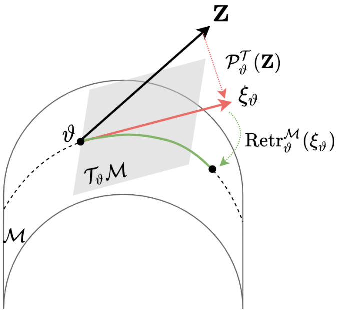
$$\langle \cdot, \cdot \rangle_{\boldsymbol{\vartheta}}^{\mathcal{M}} : \mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M} \times \mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M} \to \mathbb{R}.$$

  - **Riemannian gradient:** Given $\varphi : \mathcal{M} \to \mathbb{R}$, the Riemannian gradient of φ at $\boldsymbol{\vartheta} \in \mathcal{M}$, denoted by $\mathrm{grad}_{\mathcal{M}}\varphi(\boldsymbol{\vartheta})$, is defined via the Riemannian metric as the unique tangent vector in $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$ satisfying

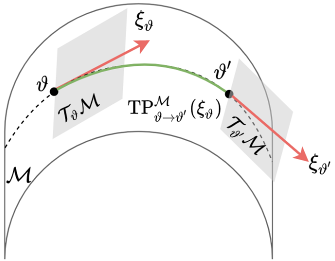
$$\langle \mathrm{grad}_{\mathcal{M}}\varphi(\boldsymbol{\vartheta}), \boldsymbol{\xi}_{\boldsymbol{\vartheta}} \rangle_{\boldsymbol{\vartheta}}^{\mathcal{M}} = \mathrm{D}\varphi(\boldsymbol{\vartheta})[\boldsymbol{\xi}_{\boldsymbol{\vartheta}}] = \lim_{c \to 0} \frac{\varphi(\boldsymbol{\vartheta} + c\boldsymbol{\xi}_{\boldsymbol{\vartheta}}) - \varphi(\boldsymbol{\vartheta})}{c}, \forall \boldsymbol{\xi}_{\boldsymbol{\vartheta}} \in \mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}.$$

Algorithm

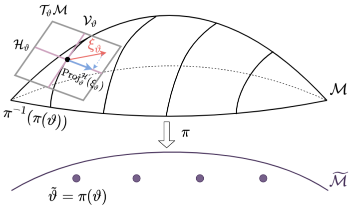
- Key Ingredients of Riemannian Manifold Optimization
  - **Retraction:** The retraction $\mathrm{Retr}_{\boldsymbol{\vartheta}}^{\mathcal{M}}(\cdot) : \mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M} \to \mathcal{M}$ builds a bridge between $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$ and $\mathcal{M}$, which is a smooth mapping such that $\mathrm{Retr}_{\boldsymbol{\vartheta}}^{\mathcal{M}}(\mathbf{0}_{\boldsymbol{\vartheta}}) = \boldsymbol{\vartheta}$ and $\mathrm{DRetr}_{\boldsymbol{\vartheta}}^{\mathcal{M}}(\mathbf{0}_{\boldsymbol{\vartheta}})[\boldsymbol{\xi}_{\boldsymbol{\vartheta}}] = \boldsymbol{\xi}_{\boldsymbol{\vartheta}}$ for all $\boldsymbol{\xi}_{\boldsymbol{\vartheta}} \in \mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$.
  - **Projection to $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$:** The projection $\mathcal{P}_{\boldsymbol{\vartheta}}^{\mathcal{T}}(\mathbf{Z})$ projects the vector $\mathbf{Z}$ in the ambient space to the tangent space $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$.



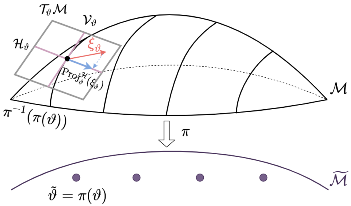
- Key Ingredients of Riemannian Manifold Optimization
  - **Vector transport:** The operation $\mathrm{TP}^{\mathcal{M}}_{\boldsymbol{\vartheta}\to\boldsymbol{\vartheta}'}(\boldsymbol{\xi}_{\boldsymbol{\vartheta}})$ denotes moving a tangent vector $\boldsymbol{\xi}_{\boldsymbol{\vartheta}}$ from the current tangent space $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$ to the tangent space $\mathcal{T}_{\boldsymbol{\vartheta}'}\mathcal{M}$ of another point $\boldsymbol{\vartheta}' \in \mathcal{M}$.



- Key Ingredients of Riemannian Manifold Optimization
  - **Quotient manifold:** A quotient manifold $\widetilde{\mathcal{M}}$ of a smooth manifold $\mathcal{M}$ is an abstract manifold, whose elements are indeed equivalence classes on $\mathcal{M}$. Specifically, define the projection $\pi : \mathcal{M} \to \widetilde{\mathcal{M}}$, which associates $\tilde{\boldsymbol{\vartheta}} = \pi(\boldsymbol{\vartheta})$. The equivalent class of $\boldsymbol{\vartheta}$, denoted by $[\boldsymbol{\vartheta}]$, is obtained on $\widetilde{\mathcal{M}}$ by $\pi^{-1}(\pi(\boldsymbol{\vartheta}))$.
  - **Vertical subspace:** Specifically, the vertical space $\mathcal{V}_{\boldsymbol{\vartheta}}$ is the subspace of $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$, which contains directions moving along the equivalence class $[\boldsymbol{\theta}]$.



- Key Ingredients of Riemannian Manifold Optimization
    - **Horizontal subspace:** The subspace that is orthogonal complement to $\mathcal{V}_{\boldsymbol{\vartheta}}$ in $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$ according to $\langle\cdot,\cdot\rangle_{\boldsymbol{\vartheta}}^{\mathcal{M}}$, termed horizontal space $\mathcal{H}_{\boldsymbol{\vartheta}}$, provides desired representatives of tangent vectors in $\mathcal{T}_{\tilde{\boldsymbol{\vartheta}}}\widetilde{\mathcal{M}}$.
    - **Orthogonal projection:** Accordingly, we can define an orthogonal projection $\mathrm{Proj}_{\boldsymbol{\vartheta}}^{\mathcal{H}}(\cdot):\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}\to\mathcal{H}_{\boldsymbol{\vartheta}}$.



Algorithm

- Riemannian Manifold of the LRaD structure

  Let $\mathcal{R}^{p,l} := \{\mathbf{Y} \in \mathbb{R}^{p\times l} : \det(\mathbf{Y}^\top \mathbf{Y}) \neq 0\}$ be the set of full rank $p \times l$ matrices and $\mathcal{B}^{p,l} = \mathcal{R}^{p,l} \times \mathcal{D}^{p}_{++}$. Define the mapping ϕ

  $$\phi : \mathcal{B}^{p,l} \to \mathcal{U}^{p,l} : (\mathbf{Y}, \mathbf{D}) \mapsto \phi(\mathbf{Y}, \mathbf{D}) = \mathbf{D} - \mathbf{Y}\mathbf{Y}^\top.$$

  However, for any $\mathbf{O} \in \mathcal{O}^l$, where $\mathcal{O}^l = \{\mathbf{O} \in \mathbb{R}^{l\times l} : \mathbf{O}\mathbf{O}^\top = \mathbf{I}\}$, we have $\phi(\mathbf{Y}\mathbf{O}, \mathbf{D}) = \phi(\mathbf{Y}, \mathbf{D})$. This equivalence class of $\boldsymbol{\theta} := (\mathbf{Y}, \mathbf{D}) \in \mathcal{B}^{p,l}$ is denoted by

  $$[\boldsymbol{\theta}] := \{\boldsymbol{\theta} * \mathbf{O} : \mathbf{O} \in \mathcal{O}^l\}, \boldsymbol{\theta} * \mathbf{O} = (\mathbf{Y}\mathbf{O}, \mathbf{D}).$$

Algorithm

- Riemannian Manifold of the LRaD structure
  - **Quotient manifold:**

$$\widetilde{\mathcal{B}}^{p,l} = \mathcal{B}^{p,l}/\mathcal{O}^l := \{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \mathcal{B}^{p,l}\}.$$

  - **Quotient map:**

$$\pi : \mathcal{B}^{p,l} \to \widetilde{\mathcal{B}}^{p,l} : \boldsymbol{\theta} \mapsto \pi(\boldsymbol{\theta}) = [\boldsymbol{\theta}].$$

We need algorithms that work conceptually on $\widetilde{\mathcal{B}}^{p,l}$ but numerically on $\mathcal{B}^{p,l}$.

Algorithm

- Riemannian Manifold of the LRaD structure
  - **Quotient manifold:**

$$\widetilde{\mathcal{B}}^{p,l} = \mathcal{B}^{p,l}/\mathcal{O}^{l} := \{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \mathcal{B}^{p,l}\}.$$

  - **Quotient map:**

$$\pi : \mathcal{B}^{p,l} \to \widetilde{\mathcal{B}}^{p,l} : \boldsymbol{\theta} \mapsto \pi(\boldsymbol{\theta}) = [\boldsymbol{\theta}].$$

We need algorithms that work conceptually on $\widetilde{\mathcal{B}}^{p,l}$ but numerically on $\mathcal{B}^{p,l}$.

- Riemannian Manifold of the LRaD structure
  - **Tangent space:**

$$\mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,l} = \{\boldsymbol{\xi}_{\boldsymbol{\theta}} = (\boldsymbol{\xi}_{\boldsymbol{\theta},Y}, \boldsymbol{\xi}_{\boldsymbol{\theta},D}) : \boldsymbol{\xi}_{\boldsymbol{\theta},Y} \in \mathbb{R}^{p\times l}, \boldsymbol{\xi}_{\boldsymbol{\theta},D} \in \mathcal{D}^p\}.$$

  - **Riemannian metric:** For $\boldsymbol{\xi}_{\boldsymbol{\theta}} = (\boldsymbol{\xi}_{\boldsymbol{\theta},Y}, \boldsymbol{\xi}_{\boldsymbol{\theta},D})$ and $\boldsymbol{\zeta}_{\boldsymbol{\theta}} = (\boldsymbol{\zeta}_{\boldsymbol{\theta},Y}, \boldsymbol{\zeta}_{\boldsymbol{\theta},D}) \in \mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,l}$,

$$\langle \boldsymbol{\xi}_{\boldsymbol{\theta}}, \boldsymbol{\zeta}_{\boldsymbol{\theta}} \rangle_{\boldsymbol{\theta}}^{\mathcal{B}^{p,l}} = \mathrm{tr}(\boldsymbol{\xi}_{\boldsymbol{\theta},Y}^{\top} \boldsymbol{\zeta}_{\boldsymbol{\theta},Y}) + \mathrm{tr}(\mathbf{D}^{-1}\boldsymbol{\xi}_{\boldsymbol{\theta},D}\mathbf{D}^{-1}\boldsymbol{\zeta}_{\boldsymbol{\theta},D}).$$

Algorithm

- Riemannian Manifold of the LRaD structure
  - **Vertical space:**

$$\mathcal{V}_{\boldsymbol{\theta}} = \{(\mathbf{Y}\boldsymbol{\Omega}, \mathbf{0}) : \boldsymbol{\Omega} \in \mathcal{S}_{\perp}^{l}\},$$

    where $\mathcal{S}_{\perp}^{l} := \{\boldsymbol{\Omega} \in \mathbb{R}^{l \times l}, \boldsymbol{\Omega} = -\boldsymbol{\Omega}^{\top}\}$.
  - **Horizontal space:**

$$\mathcal{H}_{\boldsymbol{\theta}} = \{(\boldsymbol{\xi}_{\boldsymbol{\theta},Y}, \boldsymbol{\xi}_{\boldsymbol{\theta},D}) \in \mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,l} : \mathbf{Y}^{\top}\boldsymbol{\xi}_{\boldsymbol{\theta},Y} = \boldsymbol{\xi}_{\boldsymbol{\theta},Y}^{\top}\mathbf{Y}\}.$$

  - **Projection to horizontal space:**

$$\mathrm{Proj}_{\boldsymbol{\theta}}^{\mathcal{H}}(\boldsymbol{\xi}_{\boldsymbol{\theta}}) = (\boldsymbol{\xi}_{\boldsymbol{\theta},Y} - \mathbf{Y}\boldsymbol{\Omega}, \boldsymbol{\xi}_{\boldsymbol{\theta},D}),$$

    where $\boldsymbol{\Omega} \in \mathcal{S}_{\perp}^{l}$ is the matrix satisfying Sylvester equation

$$\boldsymbol{\Omega}\mathbf{Y}^{\top}\mathbf{Y} + \mathbf{Y}^{\top}\mathbf{Y}\boldsymbol{\Omega} = \mathbf{Y}^{\top}\boldsymbol{\xi}_{\boldsymbol{\theta},Y} - \boldsymbol{\xi}_{\boldsymbol{\theta},Y}^{\top}\mathbf{Y}.$$

- Riemannian Manifold of the LRaD structure
  - **Retraction:** Let $\boldsymbol{\theta} \in \mathcal{B}^{p,l}$ and $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}} = (\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta},Y}, \overline{\boldsymbol{\xi}}_{\boldsymbol{\theta},D}) \in \mathcal{H}_{\boldsymbol{\theta}}$. The retraction of $\mathcal{B}^{p,l}$ is selected as

$$\mathrm{Retr}_{\boldsymbol{\theta}}^{\mathcal{B}^{p,l}}(\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}) = (\mathbf{Y} + \overline{\boldsymbol{\xi}}_{\boldsymbol{\theta},Y}, \mathbf{D} + \overline{\boldsymbol{\xi}}_{\boldsymbol{\theta},D} + \frac{1}{2}\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta},D}\mathbf{D}^{-1}\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta},D}).$$

  - **Vector transport:** Given $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{B}^{p,l}$ and $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}} \in \mathcal{H}_{\boldsymbol{\theta}}$, transporting $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}$ to another point $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}'} \in \mathcal{H}_{\boldsymbol{\theta}'}$ is given by

$$\mathrm{TP}_{\boldsymbol{\theta}\to\boldsymbol{\theta}'}^{\mathcal{B}^{p,l}}(\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}) = \mathrm{Proj}_{\boldsymbol{\theta}'}^{\mathcal{H}}(\mathrm{Proj}_{\boldsymbol{\theta}'}^{\mathcal{T}}(\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}})).$$

- **Proposed Riemannian ADMM Algorithm**

The problem with fixed rank l can be rewritten as

$$\min_{\boldsymbol{\Theta}\in\mathcal{U}^{p,l}} f(\boldsymbol{\Theta}) = g(\boldsymbol{\Theta}) + \beta h(\boldsymbol{\Theta})$$
$$\text{s.t. } \mathcal{U}^{p,l} := \{\boldsymbol{\Theta} : \boldsymbol{\Theta} = \mathbf{D} - \mathbf{Y}\mathbf{Y}^\top, \mathbf{Y} \in \mathbb{R}^{p\times l}, \mathbf{D} \in \mathcal{D}^p_{++}\},$$
$$\boldsymbol{\Theta} \in \mathcal{S}^p_{++}, \boldsymbol{\Theta}[ij] \leq 0, \forall i \neq j.$$

$$\xrightarrow{\bar{f} = f \circ \phi}$$

$$\min_{\theta\in\mathcal{B}^{p,l}} \bar{f}(\theta) = g(\phi(\theta)) + \beta h(\phi(\theta))$$
$$\text{s.t. } \phi(\theta) \in \mathcal{S}^p_{++},\ \phi(\theta)[ij] \leq 0, \forall i \neq j.$$

The above problem can be rewritten as

$$\min_{\boldsymbol{\theta}\in\mathcal{B}^{p,l}} g(\phi(\boldsymbol{\theta})) + \beta h(\mathbf{V}) + \mathrm{Ind}(\mathbf{V})$$
$$\text{s.t. } \phi(\boldsymbol{\theta}) \in \mathcal{S}^p_{++}, \phi(\boldsymbol{\theta}) = \mathbf{V},$$

where

$$\mathrm{Ind}(\mathbf{V}) = \begin{cases} 0, & \mathbf{V}[ij] \leq 0, \\ +\infty, & \mathbf{V}[ij] > 0. \end{cases}$$

- Proposed Riemannian ADMM Algorithm

The scaled augmented Lagrangian function is

$$\mathcal{L}_{\varpi}(\boldsymbol{\theta}, \mathbf{V}, \boldsymbol{\Phi}) = g(\phi(\boldsymbol{\theta})) + \beta h(\mathbf{V}) + \mathrm{Ind}(\mathbf{V}) - \frac{\varpi}{2}\|\boldsymbol{\Phi}\|_{\mathrm{F}}^2 + \frac{\varpi}{2}\|\phi(\boldsymbol{\theta}) - \mathbf{V} + \boldsymbol{\Phi}\|_{\mathrm{F}}^2,$$

where $\varpi > 0$ is an ADMM penalty parameter, and $\boldsymbol{\Phi}$ is the dual variable of $\mathbf{V}$.
The update of the ADMM framework is

$$\begin{aligned}
\boldsymbol{\theta}^{(k+1)} &= \underset{\phi(\boldsymbol{\theta}) \in \mathcal{S}_{++}^{p}, \boldsymbol{\theta} \in \mathcal{B}^{p,l}}{\operatorname{argmin}} \mathcal{L}_{\varpi}(\boldsymbol{\theta}, \mathbf{V}^{(k)}, \boldsymbol{\Phi}^{(k)}), \\
\mathbf{V}^{(k+1)} &= \underset{\mathbf{V}}{\operatorname{argmin}} \mathcal{L}_{\varpi}(\boldsymbol{\theta}^{(k+1)}, \mathbf{V}, \boldsymbol{\Phi}^{(k)}), \\
\boldsymbol{\Phi}^{(k+1)} &= \boldsymbol{\Phi}^{(k)} + \phi(\boldsymbol{\theta}^{(k+1)}) - \mathbf{V}^{(k+1)}.
\end{aligned}$$

- Proposed Riemannian ADMM Algorithm
  - **Update of $\boldsymbol{\theta}$:** The subproblem of updating $\boldsymbol{\theta}$ is

$$\begin{aligned}
\boldsymbol{\theta}^{(k+1)} &= \underset{\phi(\boldsymbol{\theta})\in\mathcal{S}_{++}^{p},\boldsymbol{\theta}\in\mathcal{B}^{p,l}}{\operatorname{argmin}} \mathcal{L}_{\varpi}(\boldsymbol{\theta}, \mathbf{V}^{(k)}, \boldsymbol{\Phi}^{(k)}) \\
&= \underset{\phi(\boldsymbol{\theta})\in\mathcal{S}_{++}^{p},\boldsymbol{\theta}\in\mathcal{B}^{p,l}}{\operatorname{argmin}} g(\phi(\boldsymbol{\theta})) + \frac{\varpi}{2}\|\phi(\boldsymbol{\theta}) - \mathbf{V}^{(k)} + \boldsymbol{\Phi}^{(k)}\|_{\mathrm{F}}^{2} \\
&:= \underset{\phi(\boldsymbol{\theta})\in\mathcal{S}_{++}^{p},\boldsymbol{\theta}\in\mathcal{B}^{p,l}}{\operatorname{argmin}} \overline{q}(\boldsymbol{\theta}) = q(\phi(\boldsymbol{\theta})).
\end{aligned}$$

We solve it by using the Riemannian conjugate gradient (RCG) algorithm [Sat22].

- Proposed Riemannian ADMM Algorithm
  - **Update of $\boldsymbol{\theta}$:** The subproblem of updating $\boldsymbol{\theta}$ is

$$
\begin{aligned}
\boldsymbol{\theta}^{(k+1)} &= \underset{\phi(\boldsymbol{\theta})\in\mathcal{S}_{++}^{p},\boldsymbol{\theta}\in\mathcal{B}^{p,l}}{\operatorname{argmin}} \mathcal{L}_{\varpi}(\boldsymbol{\theta}, \mathbf{V}^{(k)}, \boldsymbol{\Phi}^{(k)}) \\
&= \underset{\phi(\boldsymbol{\theta})\in\mathcal{S}_{++}^{p},\boldsymbol{\theta}\in\mathcal{B}^{p,l}}{\operatorname{argmin}} g(\phi(\boldsymbol{\theta})) + \frac{\varpi}{2}\|\phi(\boldsymbol{\theta}) - \mathbf{V}^{(k)} + \boldsymbol{\Phi}^{(k)}\|_{\mathrm{F}}^{2} \\
&:= \underset{\phi(\boldsymbol{\theta})\in\mathcal{S}_{++}^{p},\boldsymbol{\theta}\in\mathcal{B}^{p,l}}{\operatorname{argmin}} \overline{q}(\boldsymbol{\theta}) = q(\phi(\boldsymbol{\theta})).
\end{aligned}
$$

We solve it by using the Riemannian conjugate gradient (RCG) algorithm [Sat22].

Algorithm

- Proposed Riemannian ADMM Algorithm
  - **Update of θ:**

Proposition 2

The Riemannian gradient of $\overline{q}$ *at* $\boldsymbol{\theta} \in \mathcal{B}^{p,l}$ *is*

$$\operatorname{grad}_{\mathcal{B}^{p,l}} \overline{q}(\boldsymbol{\theta}) = (\mathbf{G}_Y, \mathbf{D}\operatorname{ddiag}(\mathbf{G}_D)\mathbf{D}),$$

where $\nabla_{\boldsymbol{\theta}} \overline{q}(\boldsymbol{\theta}) = (\mathbf{G}_Y, \mathbf{G}_D)$ *is the Euclidean gradient of* $\overline{q}$ *in* $\mathbb{R}^{p \times l} \times \mathbb{R}^{p \times p}$ *and is given by*

$$\mathbf{G}_Y = 2\nabla_{\boldsymbol{\Theta}} q(\phi(\boldsymbol{\theta}))\mathbf{Y}, \mathbf{G}_D = \operatorname{ddiag}(\nabla_{\boldsymbol{\Theta}} q(\phi(\boldsymbol{\theta}))),$$
$$\nabla_{\boldsymbol{\Theta}} q(\phi(\boldsymbol{\theta})) = \mathbf{S} - \boldsymbol{\Theta}^{-1} + \varpi(\boldsymbol{\Theta} - \mathbf{V}^{(k)} + \boldsymbol{\Phi}^{(k)}).$$

Moreover, $\operatorname{grad}_{\mathcal{B}^{p,l}} \overline{q}(\boldsymbol{\theta})$ *is invariant to the equivalence classes.*

Algorithm

- Proposed Riemannian ADMM Algorithm
  - **Update of θ:** Let $\boldsymbol{\theta}^{(k,0)} = \boldsymbol{\theta}^{(k)}$, and at the t-th iteration,

$$
\begin{aligned}
\boldsymbol{\eta}^{(k,t)} &= -\operatorname{grad}_{\mathcal{B}^{p,l}} \overline{q}(\boldsymbol{\theta}^{(k,t)}) + \mu^{(k,t)} \mathrm{TP}^{\mathcal{B}^{p,l}}_{\boldsymbol{\theta}^{(k,t-1)} \to \boldsymbol{\theta}^{(k,t)}} \left(\boldsymbol{\eta}^{(k,t-1)} \right) \\
\boldsymbol{\theta}^{(k,t+1)} &= \mathrm{Retr}^{\mathcal{B}^{p,l}}_{\boldsymbol{\theta}^{(k,t)}} \left(\gamma^{(k,t)} \boldsymbol{\eta}^{(k,t)} \right),
\end{aligned}
$$

    where $\mu^{(k,t)}$ can be computed via the method in [HS+52], and $\gamma^{(k,t)}$ needs to be carefully selected since we should ensure $\phi(\boldsymbol{\theta}^{(k,t+1)}) \in \mathcal{S}^p_{++}$. We use the linesearch procedure in [AMS08].

Moreover, [KGB16] suggests that there is no need to solve the $\boldsymbol{\theta}$-step problem exactly. Thus, we update $\boldsymbol{\theta}$ for T iterations and let $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k,T)}$.

- Proposed Riemannian ADMM Algorithm
  - **Update of $\mathbf{V}$:** The subproblem of updating $\mathbf{V}$ is

$$
\begin{aligned}
&\mathbf{V}^{(k+1)} \\
=&\operatorname*{argmin}_{\mathbf{V}} \mathcal{L}_{\varpi}(\boldsymbol{\theta}^{(k+1)}, \mathbf{V}, \boldsymbol{\Phi}^{(k)}) \\
=&\operatorname*{argmin}_{\mathbf{V}} \beta h(\mathbf{V}) + \mathrm{Ind}(\mathbf{V}) + \frac{\varpi}{2}\|\phi(\boldsymbol{\theta}^{(k+1)}) - \mathbf{V} + \boldsymbol{\Phi}^{(k)}\|_{\mathrm{F}}^2 \\
=&\mathrm{Prox}_{h,\mathrm{Ind}}^{\beta/\varpi}(\phi(\boldsymbol{\theta}^{(k+1)}) + \boldsymbol{\Phi}^{(k)}).
\end{aligned}
$$

Let $\mathbf{Q} = \phi(\boldsymbol{\theta}^{(k+1)}) + \boldsymbol{\Phi}^{(k)}$, and for $i \neq j$

$$
\mathrm{Prox}_{h,\mathrm{Ind}}^{\beta/\varpi}(\mathbf{Q}[ij]) = \begin{cases} 0, & \mathbf{Q}[ij] > -\frac{\beta}{\varpi} \\ \mathbf{Q}[ij] + \frac{\beta}{\varpi}, & \mathbf{Q}[ij] \leq -\frac{\beta}{\varpi}. \end{cases}
$$

- The Complete Procedure

Algorithm General algorithm to learn graphs with unknown r^*

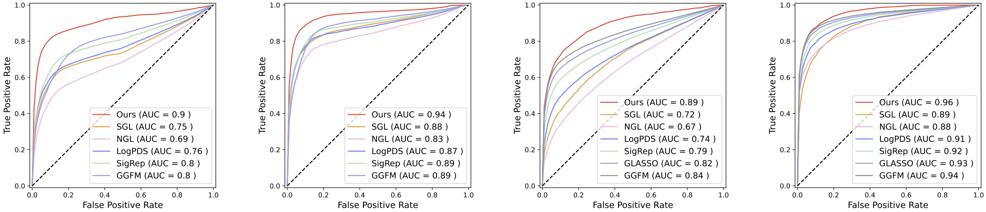
Require: Data $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$, β, ϖ, l_0
Ensure: The learned graph (precision matrix) $\boldsymbol{\Theta}$

Initialize $l = l_0, \boldsymbol{\Upsilon}_Y = -\mathbf{I}$
while $\boldsymbol{\Upsilon}_Y \prec \mathbf{0}$ and $l \leq p$ **do**
    Solve the problem with fixed rank l using Riemannian ADMM algorithm
    Calculate $\boldsymbol{\Upsilon}_Y$ using Proposition 1
    $l = l + 1$
end while
return The estimated $\widehat{\boldsymbol{\theta}}$ with rank l

Contents

Results

- Results of the LRaD graph and Watts-Strogatz (WS) graph



(a) LRaD graph, $N/p = 1, \beta = 0.05$

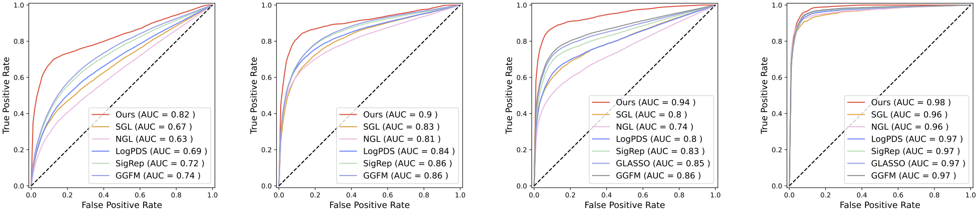
(b) LRaD graph, $N/p = 2, \beta = 0.05$

(c) WS graph, $N/p = 1, \beta = 0.05$

(d) WS graph, $N/p = 2, \beta = 0.05$

Results

- Results of the Erdos-Réyni (ER) graph and Barabási–Albert (BA) random graph



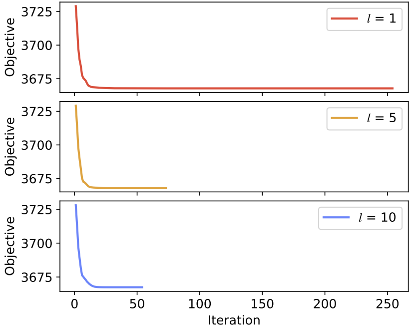
(e) ER graph, $N/p = 1, \beta = 0.05$

(f) ER graph, $N/p = 2, \beta = 0.05$

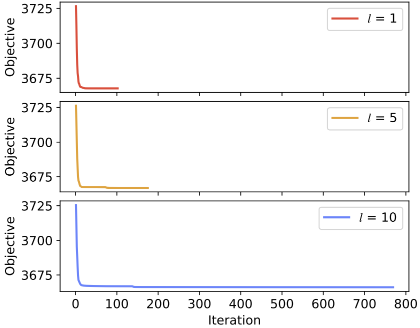
(g) BA graph, $N/p = 1, \beta = 0.05$

(h) BA graph, $N/p = 2, \beta = 0.05$

- Convergence results of the proposed Riemannian ADMM algorithm



(i) LRaD graph, $N/p = 1, \beta = 0.05$



(j) LRaD graph, $N/p = 2, \beta = 0.05$

Figure: Convergence plots of the proposed algorithm.

Contents

Conclusions

- We proposed to learn graphs (precision matrices) with the LRaD structures in GGMs.
- We discussed the estimation performance and rank selection of the proposed model.
- We proposed a Riemannian manifold to describe the LRaD structure, based on which we devised a Riemannian ADMM algorithm to solve the model.

The End

References I

[AMS08] P-A Absil, Robert Mahony, and Rodolphe Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2008.

[EPO17] Hilmi E Egilmez, Eduardo Pavez, and Antonio Ortega, *Graph learning from data under laplacian and structural constraints*, IEEE Journal of Selected Topics in Signal Processing **11** (2017), no. 6, 825–841.

[HS+52] Magnus Rudolph Hestenes, Eduard Stiefel, et al., *Methods of conjugate gradients for solving linear systems*, vol. 49, NBS Washington, DC, 1952.

[HS81] Harold V Henderson and Shayle R Searle, *On deriving the inverse of a sum of matrices*, SIAM review **23** (1981), no. 1, 53–60.

[KGB16] Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein, *Madmm: a generic algorithm for non-smooth optimization on manifolds*, Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14, Springer, 2016, pp. 680–696.

[Sat22] Hiroyuki Sato, *Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses*, SIAM Journal on Optimization **32** (2022), no. 4, 2690–2717.

[YL07] Ming Yuan and Yi Lin, *Model selection and estimation in the gaussian graphical model*, Biometrika **94** (2007), no. 1, 19–35.

References II

[ZYP22] Rui Zhou, Jiaxi Ying, and Daniel P Palomar, *Covariance matrix estimation under low-rank factor model with nonnegative correlations*, IEEE Transactions on Signal Processing **70** (2022), 4020–4030.