Graph Learning with Low-rank and Diagonal Structures: A Riemannian Geometric Approach

Xiang Zhang[1] Qiao Wang[1]

[1]School of Information Science and Engineering
Southeast University

Hyderabad, India, April 2025

Contents

Contents

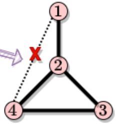
Motivations

- Gausssian Graphic Models (GGMs) [YL07]

$$\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{\dagger}$$

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

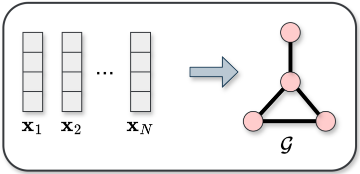
$$\mathcal{G} = \{\mathcal{A}, \mathcal{E}\}$$



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

- Graph Learning in GGMs [EPO17]

Graph Learning Task



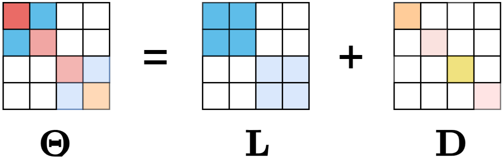
Regularized MLE Problem

$$\min_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}) = \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - \log\det(\boldsymbol{\Theta}) + \beta h(\boldsymbol{\Theta})$$
$$\text{s.t. } \boldsymbol{\Theta} \in \mathcal{S}^{p}_{++},$$

Motivations

- Gaussian Precision Factor Model [CMS22]

  The precision matrix of $\mathbf{\Theta} \in \mathcal{S}^p_{++}$ a GGM can be decomposed into a low-rank matrix $\mathbf{L} \in \mathcal{S}^{p,r}_{+}$ and a diagonal matrix $\mathbf{D} \in \mathcal{D}^p_{++}$, where $\mathcal{S}^p_{++}$ is the set of $p \times p$ positive definite (PD) matrices, $\mathcal{D}^p_{++}$ is the set of $p \times p$ diagonal PD matrices, and $\mathcal{S}^{p,r}_{+}$ is the set of all $p \times p$ positive semi-definite (PSD) matrices with rank $r, 0 \leq r \leq p$. This is termed the low-rank and diagonal (LRaD) structure.



Motivations

What are the motivations?

Motivations

- Does the LRaD structure improve the estimation of the precision matrix?
- How to handle the LRaD constraint when optimizing?

Motivations

- Does the LRaD structure improve the estimation of the precision matrix?
- How to handle the LRaD constraint when optimizing?

Contents

Problem Formulation

- Learning Precision Matrices with LRaD Structures

Original Problem

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

$$\mathrm{s.t.}\ \Theta \in \mathcal{S}_{++}^{p}$$

Graph Learning with the LRaD structure

$$\min_{\Theta \in \mathcal{M}^{p,r}} f(\Theta) = \mathrm{tr}(\Theta \mathbf{S}) - \log\det(\Theta) + \beta h(\Theta)$$

$$\mathcal{M}^{p,r} := \{\Theta : \Theta \in \mathcal{S}_{++}^{p},\ \Theta = \mathbf{L} + \mathbf{D}, \mathbf{D} \in \mathcal{D}_{++}^{p}, \mathbf{L} \in \mathcal{S}_{+}^{p,r}\}$$

- Merits of The Proposed Model
  - It reduces the dimensionality of the estimation problem from $p(p+1)/2$ to $p(r+1)$, where we usually have $r \ll p$.
  - The LRaD decomposition guarantees that the estimated Θ lies in $\mathcal{S}_{++}^{p}$ automatically.

Problem Formulation

- Learning Precision Matrices with LRaD Structures

Original Problem

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

$$\mathrm{s.t.}\ \mathbf{\Theta} \in \mathcal{S}_{++}^{p}$$

Graph Learning with the LRaD structure

$$\min_{\mathbf{\Theta}\in\mathcal{M}^{p,r}} f(\mathbf{\Theta}) = \mathrm{tr}(\mathbf{\Theta}\mathbf{S}) - \log\det(\mathbf{\Theta}) + \beta h(\mathbf{\Theta})$$

$$\mathcal{M}^{p,r} := \{\mathbf{\Theta} : \mathbf{\Theta} \in \mathcal{S}_{++}^{p},\ \mathbf{\Theta} = \mathbf{L} + \mathbf{D}, \mathbf{D} \in \mathcal{D}_{++}^{p}, \mathbf{L} \in \mathcal{S}_{+}^{p,r}\}$$

- Merits of The Proposed Model
  - It reduces the dimensionality of the estimation problem from $p(p+1)/2$ to $p(r+1)$, where we usually have $r \ll p$.
  - The LRaD decomposition guarantees that the estimated $\mathbf{\Theta}$ lies in $\mathcal{S}_{++}^{p}$ automatically.

Contents

- Riemannian Manifold Optimization

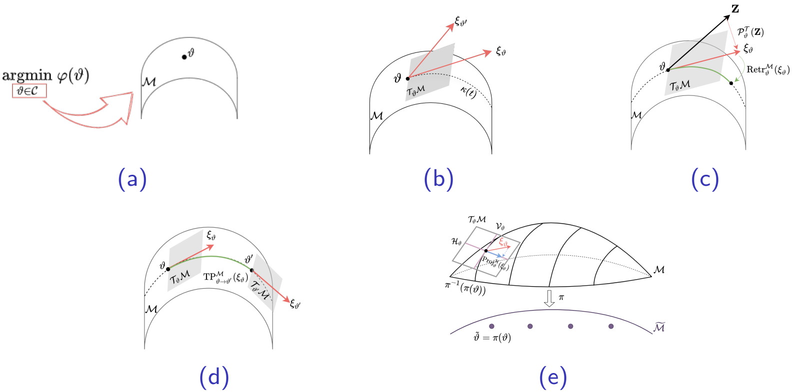


Figure: The illustration of Riemannian manifold optimization: (a) Overview of manifold optimization; (b) Tangent vectors and tangent space; (c) Projection onto $\mathcal{T}_{\boldsymbol{\vartheta}}\mathcal{M}$ and retraction back to $\mathcal{M}$; (d) Vector transport; (e) Quotient manifold $\widetilde{\mathcal{M}}$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

- Riemannian Manifold of the LRaD structure

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

$$\phi : \mathcal{B}^{p,r} \to \mathcal{M}^{p,r} : (\mathbf{Y}, \mathbf{D}) \mapsto \phi(\mathbf{Y}, \mathbf{D}) = \mathbf{D} + \mathbf{Y}\mathbf{Y}^\top.$$

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

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

Algorithm

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

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

  - **Quotient map:**

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

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

Algorithm

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

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

  - **Quotient map:**

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

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

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

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

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

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

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

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

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

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

  - **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 $\boldsymbol{\Omega} \in \mathcal{S}_{\perp}^{r}$ is the matrix satisfying Sylvester equation

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

Algorithm

- Riemannian Manifold of the LRaD structure
  - **Retraction:** Let $\boldsymbol{\theta} \in \mathcal{B}^{p,r}$ 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 $\mathcal{B}^{p,r}$ is selected as

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

  - **Vector transport:** Given $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{B}^{p,r}$ 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

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

- Proposed Riemannian ADMM Algorithm

  The problem with fixed rank r can be rewritten as

$$\min_{\boldsymbol{\Theta}} f(\boldsymbol{\Theta}) = g(\boldsymbol{\Theta}) + \beta h(\boldsymbol{\Theta})$$

$$\text{s.t. } \boldsymbol{\Theta} : \boldsymbol{\Theta} \in \mathcal{S}^p_{++} = \mathbf{D} + \mathbf{Y}\mathbf{Y}^\top, \mathbf{Y} \in \mathbb{R}^{p \times l}, \mathbf{D} \in \mathcal{D}^p_{++}$$

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

$$\min_{\theta \in \mathcal{B}^{p,r}} \bar{f}(\theta) = g(\phi(\theta)) + \beta h(\phi(\theta))$$

- Proposed Riemannian Conjugate Gradient Algorithm
    - Calculate the Riemannian gradient of $\bar{f}(\boldsymbol{\Theta})$

Proposition 1

The Riemannian gradient of $\bar{f}$ *at* $\boldsymbol{\theta} \in \mathcal{B}^{p,r}$ *is*

$$\mathrm{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta}) = (\mathbf{G}_Y, \mathbf{D}\mathrm{ddiag}(\mathbf{G}_D)\mathbf{D}),$$

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

$$\mathbf{G}_Y = 2\nabla_{\boldsymbol{\Theta}} f(\phi(\boldsymbol{\theta}))\mathbf{Y}, \mathbf{G}_D = \mathrm{ddiag}(\nabla_{\boldsymbol{\Theta}} f(\phi(\boldsymbol{\theta}))),$$
$$\nabla_{\boldsymbol{\Theta}} f(\phi(\boldsymbol{\theta})) = \mathbf{S} - \boldsymbol{\Theta}^{-1} + \beta \nabla_{\boldsymbol{\Theta}} h(\boldsymbol{\Theta}).$$

Moreover, $\mathrm{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta})$ *is invariant to the equivalence classes.*

Algorithm

- Proposed Riemannian Conjugate Gradient Algorithm
  - Calculate update direction

$$\boldsymbol{\eta}^{(k)} = -\operatorname{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta}^{(k)}) + \mu^{(k)} \mathrm{TP}^{\mathcal{B}^{p,r}}_{\boldsymbol{\theta}^{(k-1)} \to \boldsymbol{\theta}^{(k)}} \left(\boldsymbol{\eta}^{(k-1)} \right)$$

  - Retraction to the manifold

$$\boldsymbol{\theta}^{(k+1)} = \mathrm{Retr}^{\mathcal{B}^{p,r}}_{\boldsymbol{\theta}^{(k)}} \left(\gamma^{(k)} \boldsymbol{\eta}^{(k)} \right)$$

Note that $\mu^{(k)}$ can be computed via the method in [HS$^+$52], and $\gamma^{(k)}$ can be selected using the linesearch procedure in [AMS08].

- Proposed Riemannian Conjugate Gradient Algorithm

Algorithm RCG algorithm to learn graphs with the LRaD structures

Require: Data $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$, β, r

Ensure: The learned graph (precision matrix) $\boldsymbol{\Theta}$

Initialize $\boldsymbol{\theta}^{(0)} \in \mathcal{B}^{p,r}$ randomly, and let $\boldsymbol{\eta}^{(0)} = -\mathrm{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta}^{(0)})$

for $k = 0, 1, 2, ...,$ until convergence **do**

  Compute the stepsize $\gamma^{(k)}$ using the linesearch in [AMS08]

  Update $\boldsymbol{\theta}^{(k+1)}$ via $\boldsymbol{\theta}^{(k+1)} = \mathrm{Retr}_{\boldsymbol{\theta}^{(k)}}^{\mathcal{B}^{p,r}} \left(\gamma^{(k)} \boldsymbol{\eta}^{(k)}\right)$

  Compute the stepsize $\mu^{(k+1)}$ using the rule in [HS+52]

  Let $\boldsymbol{\Theta}^{(k+1)} = \phi(\boldsymbol{\theta}^{(k+1)})$

  Compute $\mathrm{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta}^{(k+1)})$ via $\mathrm{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta}) = (\mathbf{G}_Y, \mathbf{D}\mathrm{ddiag}(\mathbf{G}_D)\mathbf{D})$

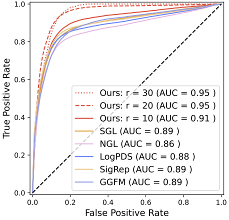
  Update $\boldsymbol{\eta}^{(k+1)}$ via $\boldsymbol{\eta}^{(k)} = -\mathrm{grad}_{\mathcal{B}^{p,r}} \bar{f}(\boldsymbol{\theta}^{(k)}) + \mu^{(k)} \mathrm{TP}_{\boldsymbol{\theta}^{(k-1)} \to \boldsymbol{\theta}^{(k)}}^{\mathcal{B}^{p,r}} \left(\boldsymbol{\eta}^{(k-1)}\right)$

end for

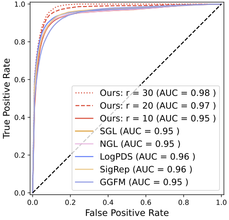
return The estimated $\widehat{\boldsymbol{\Theta}} = \phi(\widehat{\boldsymbol{\theta}})$ with rank r

Contents

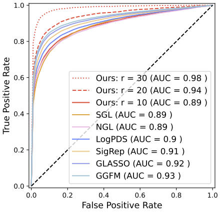
- Results of The LRaD Graph and Watts-Strogatz (WS) Graph



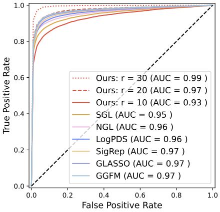
(a) LRaD, $N/p = 2$



(b) LRaD, $N/p = 4$

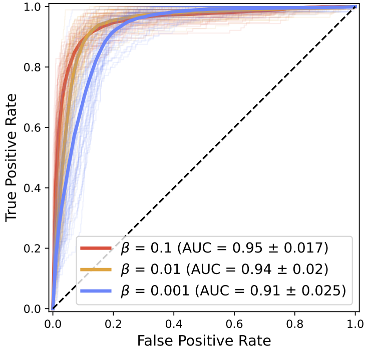


(c) WS, $N/p = 2$

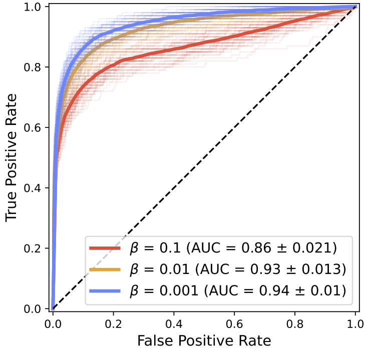


(d) WS, $N/p = 4$

- Results of Parameter Sensitivity



(e) LRaD graph, $N/p = 2, r = 20$

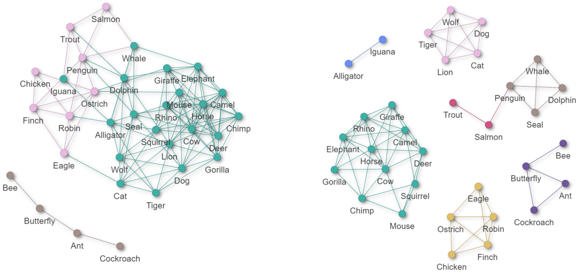


(f) WS graph, $N/p = 2, r = 20$

- Results of *ANIMAL* Dataset

Table: Modularity of the learned graphs.

	GLASSO	SGL	NGL	LogPDS	SigRep	Ours
Mod	0.57	0.74	0.42	0.70	0.74	**0.85**



(g) GLASSO (h) Ours: $r = 10,\ \beta = 2.5$

Contents

- We proposed to learn graphs (precision matrices) with the LRaD structures in GGMs.
- We proposed a Riemannian manifold to describe the LRaD structure, based on which we devised a Riemannian conjugate gradient 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.

[CMS22] Noirrit Kiran Chandra, Peter Mueller, and Abhra Sarkar, *Bayesian scalable precision factor analysis for massive sparse gaussian graphical models*.

[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