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

Xiang Zhang
School of Information Science and Engineering
Southeast University
Nanjing, China
xiangzhang369@seu.edu.cn

Qiao Wang
School of Information Science and Engineering
Southeast University
Nanjing, China
qiaowang@seu.edu.cn

***Abstract*—We study the problem of learning graphs in Gaussian graphic models by assuming that the underlying precision matrix has a "low-rank and diagonal" (LRaD) structure. This assumption enjoys a latent factor representation interpretation and can reduce the dimensionality of the problem, thus leading to superior performance with fewer samples. To address the optimization challenges caused by the LRaD constraint, we design a Riemannian quotient manifold to transform the proposed model into an unconstrained problem on the manifold. Then, we devise a Riemannian conjugate gradient algorithm to solve the proposed model. Experimental results on synthetic and real data illustrate the effectiveness of the proposed method.**

***Index Terms*—Graph learning, Gaussian graphic model, Gaussian precision factor model, Riemannian manifold optimization**

I. Introduction

Graphs can flexibly represent irregular structures behind high dimensional data and have proven their power in many applications [1]. Inferring graph topology behind data, also known as graph learning (GL), has been increasingly studied and applied in numerous fields, such as social networks [2] and medical data [3]. Extensive studies are conducted on GL [4], [5], including models based on causality dependencies [6], [7] and graph signal processing [2], [8].

Here, we focus on GL in Gaussian graphical models (GGMs), where graphs encode conditional dependence between two variables in a multivariate Gaussian model. In GGMs, GL is equivalent to estimating precision (inverse covariance) matrices, which is challenging, especially for small sample sizes and high-dimensional data [9]. Thus, many works impose prior structures on the graphs to improve performance, such as sparsity [10], [11], separated clusters [12], non-negative correlations [13], [14], and tree structures [15].

In this study, we assume that the precision matrix enjoys a "low-rank and diagonal" (LRaD) structure. This structure is inspired by the Gaussian precision factor model [16], which provides a latent factor representation interpretation for data with an LRaD precision matrix. The LRaD assumption can reduce the dimensionality of the problem, thereby improving the estimation performance under small sample sizes. In the literature, the LRaD structure is widely used to estimate covariance matrices, which is derived from the Gaussian factor analysis (GFA) model [17]–[19]. Our study differs from these models in that we aim to estimate precision matrices, on which the LRaD structure is imposed. The work [16] also uses the LRaD structure to construct a precision matrix estimator from a Bayesian perspective. Our model learns graphs by solving a penalized Gaussian maximum likelihood estimation problem.

However, the LRaD structure poses optimization challenges due to the non-convex low-rank constraint. Existing methods include relaxing the low-rank constraint by penalizing nuclear norm [20] so that the problem can be solved using convex methods [21]. Albeit feasible, the solutions of these methods are approximations of original problems and require additional tuning parameters. To this end, we propose an algorithm based on manifold optimization [22], [23]. Specifically, we design a Riemannian quotient manifold [24], [25] based on the LRaD structure. We then propose a Riemannian conjugate gradient (RCG) algorithm to solve our model on this manifold. The merits of our algorithm are two-fold. First, optimizing on the manifold transforms our model into an unconstrained problem, and the solution will exactly and automatically satisfy the LRaD constraint. Second, the solution is not an approximation of the original problem and is free of extra parameter tuning.

The main contributions of this study are summarized as follows. (i) We propose a GL model in GGMs by assuming the LRaD structure on precision matrices, which reduces the dimensionality of the problem and thus improves estimation performance under small sample sizes. (ii) We design a Riemannian manifold to describe the LRaD constraint, on which an RCG algorithm is proposed to solve our problem. (iii) Extensive experiments, including synthetic and real data, are conducted to illustrate the effectiveness of our approach.

II. Problem Formulation

In this section, we first introduce the background of GL in GGMs and then present the proposed model.

A. Learning Graphs in GGMs

Let $\mathcal{G} = \{\mathcal{A}, \mathcal{E}\}$ be an undirected graph with the set of p nodes $\mathcal{A}$ and the set of edges $\mathcal{E}$. Assume that a random vector $\mathbf{x} \in \mathbb{R}^p$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, and let

This research is partially supported by Project Four of the National Natural Science Foundation of China (NSFC) Major Program, Prototype Construction of Carbon Neutrality Building Technology for High Density Urban Environment, Project Number: 52394224.

$\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the precision matrix. Associating $\mathbf{x}$ with the graph $\mathcal{G}$ forms a GGM satisfying

$$\boldsymbol{\Theta}[ij] \neq 0 \Leftrightarrow (i,j) \in \mathcal{E}, \forall i \neq j \tag{1}$$

$$\boldsymbol{\Theta}[ij] = 0 \Leftrightarrow \mathbf{x}[i] \perp\!\!\!\perp \mathbf{x}[j] \,|\, \mathbf{x}[\mathcal{A}\backslash\{i,j\}], \tag{2}$$

where $\boldsymbol{\Theta}[ij]$ is the (i,j) entry of $\boldsymbol{\Theta}$, $\mathbf{x}[i] \perp\!\!\!\perp \mathbf{x}[j] \,|\, \mathbf{x}[\mathcal{A}\backslash\{i,j\}]$ means that $\mathbf{x}[i]$ is conditionally independent of $\mathbf{x}[j]$ given the other variables. Thus, the graph $\mathcal{G}$ or $\boldsymbol{\Theta}$ characterizes conditional dependence between two variables in a multivariate vector $\mathbf{x}$. Within GGMs, GL aims to infer the precision matrix $\boldsymbol{\Theta}$ from N independently observed data $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{p\times N}$, which boils down to solving the following penalized Gaussian maximum likelihood estimation problem:

$$\min_{\boldsymbol{\Theta}\in\mathcal{S}_{++}^p} f(\boldsymbol{\Theta}) = \min_{\boldsymbol{\Theta}\in\mathcal{S}_{++}^p} \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - \log\det(\boldsymbol{\Theta}) + \beta h(\boldsymbol{\Theta}), \tag{3}$$

where $\mathbf{S} = \frac{1}{N}\sum_{n=1}^N \mathbf{x}_n\mathbf{x}_n^\top$ is the sample covariance matrix, $\operatorname{tr}(\cdot)$ is the trace operator, $\det(\cdot)$ is the determinant of a matrix, and $\mathcal{S}_{++}^p$ is the set of $p\times p$ positive definite (PD) matrices. Moreover, $h(\boldsymbol{\Theta}) = \sum_{i\neq j}\psi(\boldsymbol{\Theta}[ij])$, $\psi(\cdot)$ is a sparsity penalty function applied to the off-diagonal elements of $\boldsymbol{\Theta}$, and $\beta \geq 0$ is a constant controlling sparsity.

B. Learning Graphs with LRaD Structures

The model (3) imposes sparsity via the regularizer $h(\boldsymbol{\Theta})$, which is not sufficient and may lead to poor estimation results when N is small [12]. In this study, we assume that $\boldsymbol{\Theta}$ enjoys the LRaD structure. Specifically, for any $\boldsymbol{\Theta} \in \mathcal{S}_{++}^p$, we have

$$\boldsymbol{\Theta} = \mathbf{L} + \mathbf{D},\ \mathbf{D} \in \mathcal{D}_{++}^p,\ \mathbf{L} \in \mathcal{S}_{+}^{p,r}, \tag{4}$$

where $\mathcal{D}_{++}^p$ 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 $r, 0 \leq r \leq p$. Note that the decomposition (4) always holds for a sufficiently large r. However, in practice, we empirically find that values of $r \ll p$ suffice to approximate $\boldsymbol{\Theta}$ well even if $\boldsymbol{\Theta}$ does not exactly admit such an LRaD structure.

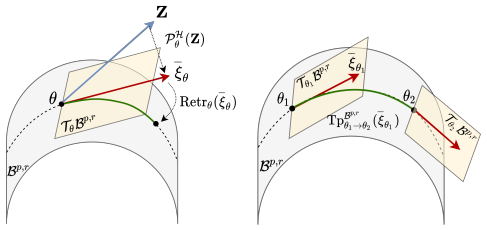
In the literature, the LRaD structure is used to estimate covariance matrices based on the GFA model [17], [18], while our model imposes the LRaD structure on precision matrices. Like the GFA model, our LRaD assumption also has a latent factor representation interpretation, known as the Gaussian precision factor model, see Proposition 1 in [16]. From a utilitarian perspective, we focus on leveraging the LRaD structures to facilitate the estimation of $\boldsymbol{\Theta}$.

With the LRaD assumption, problem (3) becomes

$$\min_{\boldsymbol{\Theta}\in\mathcal{M}^{p,r}} f(\boldsymbol{\Theta}) \tag{5}$$

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

Compared with (3), our model (5) has two merits. First, it reduces the dimensionality of the estimation problem from $p(p+1)/2$ to $p(r+1)$, where we usually have $r \ll p$. Thus, our model requires fewer samples to recover the graph and can be applied to high-dimensional regimes. Second, the LRaD decomposition guarantees that the estimated $\boldsymbol{\Theta}$ lies in $\mathcal{S}_{++}^p$ automatically, while traditional methods may require post-processing to ensure the positive definiteness of $\boldsymbol{\Theta}$.



(a) Projection and retraction (b) Vector transport

Fig. 1: Illustration of operations in manifold $\mathcal{B}^{p,r}$.

III. Proposed Algorithm

The proposed model (5) is non-convex and difficult to solve due to the rank constraint. Thus, we exploit manifold optimization to solve (5). The main idea is to view the LRaD constraint as a Riemann manifold, on which optimization algorithms are performed. To do this, we first construct a Riemannian manifold based on $\mathcal{M}^{p,r}$. Then, we propose an RCG algorithm over the constructed manifold to solve (5).

A. Riemannian Manifold of $\mathcal{M}^{p,r}$

We focus on using the particular structure of $\mathbf{L}$ and $\mathbf{D}$ to define a manifold. Specifically, for any $\mathbf{L} \in \mathcal{S}_{+}^{p,r}$, it allows a full rank decomposition $\mathbf{L} = \mathbf{Y}\mathbf{Y}^\top$, where $\mathbf{Y} \in \mathcal{R}^{p,r}$, $\mathcal{R}^{p,r} := \{\mathbf{Y} \in \mathbb{R}^{p\times r} : \det(\mathbf{Y}^\top\mathbf{Y}) \neq 0\}$ is the set of full-rank $p\times r$ matrices. Let $\mathcal{B}^{p,r} = \mathcal{R}^{p,r} \times \mathcal{D}_{++}^p$, and we define

$$\phi : \mathcal{B}^{p,r} \to \mathcal{M}^{p,r} : (\mathbf{Y}, \mathbf{D}) \mapsto \phi(\mathbf{Y}, \mathbf{D}) = \mathbf{Y}\mathbf{Y}^\top + \mathbf{D}. \tag{6}$$

The optimization on $\mathcal{M}^{p,r}$ can be transformed into the optimization on $\mathcal{B}^{p,r}$ via ϕ. However, for any $\mathbf{O} \in \mathcal{O}^r$, where $\mathcal{O}^r$ is the orthogonal group in dimension r, we have a equivalence class w.r.t. ϕ, i.e., $\phi(\mathbf{Y}\mathbf{O}, \mathbf{D}) = \phi(\mathbf{Y}, \mathbf{D})$. This equivalence class of $\boldsymbol{\theta} := (\mathbf{Y}, \mathbf{D}) \in \mathcal{B}^{p,r}$ is denoted by $[\boldsymbol{\theta}] := \{\boldsymbol{\theta} * \mathbf{O} : \mathbf{O} \in \mathcal{O}^r\}$, where $\boldsymbol{\theta} * \mathbf{O} = (\mathbf{Y}\mathbf{O}, \mathbf{D})$. The invariance property renders that the solution of (5) is not isolated. To address this issue, we define a quotient set $\mathcal{B}^{p,r}/\mathcal{O}^r$ as

$$\mathcal{B}^{p,r}/\mathcal{O}^r := \{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \mathcal{B}^{p,r}\}. \tag{7}$$

The quotient map is then defined as

$$\pi : \mathcal{B}^{p,r} \to \mathcal{B}^{p,r}/\mathcal{O}^r : \boldsymbol{\theta} \mapsto \pi(\boldsymbol{\theta}) = [\boldsymbol{\theta}]. \tag{8}$$

In practice, we can manipulate objects in $\mathcal{B}^{p,r}$ and eliminate the effect of the equivalence class. As shown in Fig.1, we provide the operations required to design algorithms on $\mathcal{B}^{p,r}$.

(i) **Riemannian metric**: The premise of operations on manifolds is to define metrics. First, the tangent space at a point $\boldsymbol{\theta} \in \mathcal{B}^{p,r}$ is

$$\mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,r} = \{(\boldsymbol{\xi}_{\mathbf{Y}}, \boldsymbol{\xi}_{\mathbf{D}}) : \boldsymbol{\xi}_{\mathbf{Y}} \in \mathbb{R}^{p\times r}, \boldsymbol{\xi}_{\mathbf{D}} \in \mathcal{D}^p\}. \tag{9}$$

For $\boldsymbol{\xi} = (\boldsymbol{\xi}_{\mathbf{Y}}, \boldsymbol{\xi}_{\mathbf{D}})$ and $\boldsymbol{\zeta} = (\boldsymbol{\zeta}_{\mathbf{Y}}, \boldsymbol{\zeta}_{\mathbf{D}}) \in \mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,r}$, the metric is defined as sum of the metrics of $\mathcal{R}^{p,r}$ and $\mathcal{D}_{++}^p$, i.e.,

$$\langle\boldsymbol{\xi}, \boldsymbol{\zeta}\rangle_{\boldsymbol{\theta}}^{\mathcal{B}^{p,r}} = \operatorname{tr}(\boldsymbol{\xi}_{\mathbf{Y}}^\top\boldsymbol{\zeta}_{\mathbf{Y}}) + \operatorname{tr}(\mathbf{D}^{-1}\boldsymbol{\xi}_{\mathbf{D}}\mathbf{D}^{-1}\boldsymbol{\zeta}_{\mathbf{D}}). \tag{10}$$

The first term of the r.h.s. of (10) is from [26], and the second term is the affine-invariant one in [27].

(ii) **Projection**: The equivalence class $[\boldsymbol{\theta}]$ is a subspace of $\mathcal{B}^{p,r}$, and its tangent space $\mathcal{T}_{\boldsymbol{\theta}}[\boldsymbol{\theta}]$ is also a subspace of $\mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,r}$, which is known as vertical space $\mathcal{V}_{\boldsymbol{\theta}}$

$$\mathcal{V}_{\boldsymbol{\theta}} = \{(\mathbf{Y}\boldsymbol{\Omega}, \mathbf{0}) : \boldsymbol{\Omega} = -\boldsymbol{\Omega}^\top, \boldsymbol{\Omega} \in \mathbb{R}^{p\times p}\}, \tag{11}$$

where $\mathbf{0}$ is the matrices of zeros. The space $\mathcal{V}_{\boldsymbol{\theta}}$ contains directions moving along the equivalence class $[\boldsymbol{\theta}]$, which

should be eliminated. Thus, the vectors of interest in $\mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,r}$ lie in the orthogonal complement of $\mathcal{V}_{\boldsymbol{\theta}}$, termed horizontal space $\mathcal{H}_{\boldsymbol{\theta}}$

$$\mathcal{H}_{\boldsymbol{\theta}} = \{(\mathbf{Z}_1, \mathbf{Z}_2) : \mathbf{Y}^\top \mathbf{Z}_1 = \mathbf{Z}_1^\top \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{p\times r}, \mathbf{Z}_2 \in \mathcal{D}^p\}. \quad (12)$$

Any tangent vector $\boldsymbol{\xi}_{[\boldsymbol{\theta}]} \in \mathcal{T}_{[\boldsymbol{\theta}]}\mathcal{B}^{p,r}/\mathcal{O}^r$ can be represented by a unique vector $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}} = (\overline{\boldsymbol{\xi}}_{\mathbf{Y}}, \overline{\boldsymbol{\xi}}_{\mathbf{D}}) \in \mathcal{H}_{\boldsymbol{\theta}}$ satisfying $\mathrm{D}\pi(\boldsymbol{\theta})[\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}] = \boldsymbol{\xi}_{[\boldsymbol{\theta}]}$, where $\mathrm{D}\pi(\boldsymbol{\theta})[\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}]$ is the directional derivative of $\pi(\boldsymbol{\theta})$ in the direction $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}$. The $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}$ is called horizontal lift of $\boldsymbol{\xi}_{[\boldsymbol{\theta}]}$ at $\boldsymbol{\theta}$ [23]. We can eliminate the effect of equivalence classes by projecting points onto $\mathcal{H}_{\boldsymbol{\theta}}$. For any point $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2)$ in the ambient space $\mathbb{R}^{p\times r} \times \mathbb{R}^{p\times p}$, the projection is defined as

$$\mathcal{P}^{\mathcal{H}}_{\boldsymbol{\theta}}(\mathbf{Z}) = (\mathbf{Z}_1 - \mathbf{Y}\boldsymbol{\Omega}, \mathrm{ddiag}(\mathbf{Z}_2)), \quad (13)$$

where $\mathrm{ddiag}(\cdot)$ means only retaining diagonal elements, and $\boldsymbol{\Omega}$ is a skew-symmetric matrix satisfying Sylvester equation

$$\boldsymbol{\Omega}\mathbf{Y}^\top\mathbf{Y} + \mathbf{Y}^\top\mathbf{Y}\boldsymbol{\Omega} = \mathbf{Y}^\top\mathbf{Z}_1 - \mathbf{Z}_1^\top\mathbf{Y}. \quad (14)$$

(iii) **Retraction**: Given a tangent vector $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}} \in \mathcal{H}_{\boldsymbol{\theta}}$, retracting it back to the manifold $\mathcal{B}^{p,r}$ is defined as

$$\mathrm{Retr}_{\boldsymbol{\theta}}(\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}}) = (\mathbf{Y} + \overline{\boldsymbol{\xi}}_{\mathbf{Y}}, \mathbf{D} + \overline{\boldsymbol{\xi}}_{\mathbf{D}} + \frac{1}{2}\overline{\boldsymbol{\xi}}_{\mathbf{D}}\mathbf{D}^{-1}\overline{\boldsymbol{\xi}}_{\mathbf{D}}). \quad (15)$$

(iv) **Vector transport**: Given $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{B}^{p,r}$ and $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}_1} \in \mathcal{H}_{\boldsymbol{\theta}_1}$, the vector transport of $\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}_1}$ to $\mathcal{H}_{\boldsymbol{\theta}_2}$ is

$$\mathrm{Tp}^{\mathcal{B}^{p,r}}_{\boldsymbol{\theta}_1\to\boldsymbol{\theta}_2}(\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}_1}) = \mathcal{P}^{\mathcal{H}}_{\boldsymbol{\theta}_2}(\overline{\boldsymbol{\xi}}_{\boldsymbol{\theta}_1}). \quad (16)$$

We have defined all required operations on the manifold $\mathcal{B}^{p,r}$. Next, we will solve (5) on the $\mathcal{B}^{p,r}$ using these operations.

B. Riemannian Conjugate Gradient Algorithm

The objective function $f(\boldsymbol{\Theta}) : \mathcal{S}^p_{++} \to \mathbb{R}$ can be rewritten as $\overline{f}(\boldsymbol{\theta}) : \mathcal{B}^{p,r} \to \mathbb{R}$, where $\overline{f} = f \circ \phi$. To solve (5) on $\mathcal{B}^{p,r}$, we need to derive the Riemannian gradient $\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta})$, which can be calculated from the Euclidean gradient $\nabla\overline{f}(\boldsymbol{\theta}) = (\mathbf{G}_{\mathbf{Y}}, \mathbf{G}_{\mathbf{D}})$ via metric (10). Specifically, $\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta})$ is equivalent to $\nabla\overline{f}(\boldsymbol{\theta})$ projected onto $\mathcal{T}_{\boldsymbol{\theta}}\mathcal{B}^{p,r}$, i.e.,

$$\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta}) = (\mathbf{G}_{\mathbf{Y}}, \mathbf{D}\mathrm{ddiag}(\mathbf{G}_{\mathbf{D}})\mathbf{D}), \quad (17)$$

where

$$\mathbf{G}_{\mathbf{Y}} = 2\nabla f(\phi(\boldsymbol{\theta}))\mathbf{Y}, \mathbf{G}_{\mathbf{D}} = \mathrm{ddiag}(\nabla f(\phi(\boldsymbol{\theta}))), \quad (18)$$

and

$$\nabla f(\phi(\boldsymbol{\theta})) = \nabla f(\boldsymbol{\Theta}) = \mathbf{S} - \boldsymbol{\Theta}^{-1} + \beta\nabla h(\boldsymbol{\Theta}). \quad (19)$$

Here, we use $\psi(z) = \nu\log(\cosh(z/\nu)), \nu > 0$, as an alternative to widely-used ℓ_1 norm since the algorithm requires $\psi(z)$ to be smooth [18]. When $\nu \to 0$, $\psi(z)$ yields ℓ_1 norm. Empirically, we find that the results are insensitive to ν, and we let $\nu = 10^{-12}$. Then, $\nabla h(\boldsymbol{\Theta})$ is calculated as

$$\nabla h(\boldsymbol{\Theta}) = \begin{cases} 0, & i = j \\ \tanh(\boldsymbol{\Theta}[ij]/\nu), & i \neq j. \end{cases} \quad (20)$$

Since the objective function of interest is invariant to the equivalence class, $\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta})$ naturally lies in $\mathcal{H}_{\boldsymbol{\theta}}$, which can be used directly. After obtaining $\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta})$, the update of our RCG algorithm [28] at the t-th iteration is

$$\boldsymbol{\eta}^{(k)} = -\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta}^{(k)}) + \mu^{(k)}\mathrm{Tp}^{\mathcal{B}^{p,r}}_{\boldsymbol{\theta}^{(k-1)}\to\boldsymbol{\theta}^{(k)}}\left(\boldsymbol{\eta}^{(k-1)}\right) \quad (21)$$

$$\boldsymbol{\theta}^{(k+1)} = \mathrm{Retr}_{\boldsymbol{\theta}^{(k)}}\left(\gamma^{(k)}\boldsymbol{\eta}^{(k)}\right), \quad (22)$$

where $\gamma^{(k)}$ is the stepsize determined by linesearch [22], and $\mu^{(k)}$ can be computed via the method in [29]. Eq.(21) calculates the descent direction $\boldsymbol{\eta}^{(k)}$, which is composed of $-\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta})$ plus the previous descent direction $\boldsymbol{\eta}^{(k-1)}$. Moreover, $\boldsymbol{\eta}^{(k-1)}$ is transported to $\mathcal{H}_{\boldsymbol{\theta}^{(k)}}$ so that it can be added to $-\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta}^{(k)})$. Eq.(22) performs updates using the calculated $\boldsymbol{\eta}^{(k)}$ and retracts the updates onto the manifold so that $\boldsymbol{\theta}^{(k+1)}$ still lies in $\mathcal{B}^{p,r}$. The complete procedure of the proposed algorithm is displayed in Algorithm 1.

Algorithm 1 RCG algorithm to solve (5)

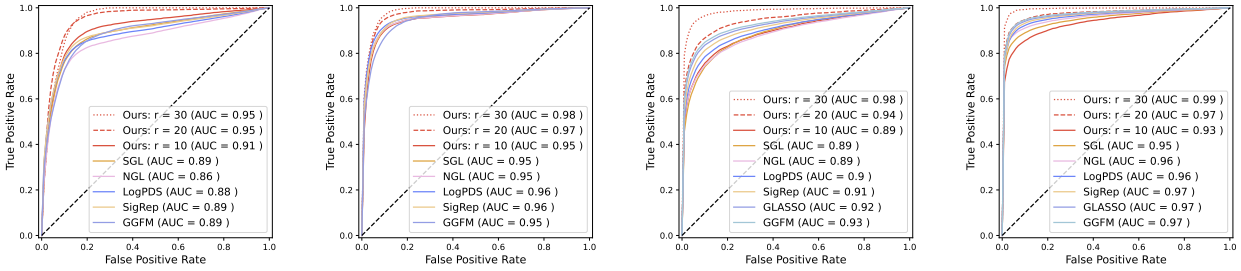
Input: Data $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$, β, rank r
Output: The learned graph (precision matrix) $\boldsymbol{\Theta}$
1: Initialize $\boldsymbol{\theta}^{(0)} \in \mathcal{B}^{p,r}$ randomly, and let $\boldsymbol{\eta}^{(0)} = -\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta}^{(0)})$
2: **for** $k = 0, 1, 2, ...,$ until convergence **do**
3: Compute the stepsize $\gamma^{(k)}$ using the linesearch in [22]
4: Update $\boldsymbol{\theta}^{(k+1)}$ using (22)
5: Compute the stepsize $\mu^{(k+1)}$ using the rule in [29]
6: Let $\boldsymbol{\Theta}^{(k+1)} = \phi(\boldsymbol{\theta}^{(k+1)})$
7: Compute $\nabla^{\mathcal{B}^{p,r}}\overline{f}(\boldsymbol{\theta}^{(k+1)})$ using (17) - (20)
8: Update $\boldsymbol{\eta}^{(k+1)}$ using (21)
9: **end for**

Our algorithm calculates the Euclidean gradient $\nabla f(\phi(\boldsymbol{\theta}))$ to obtain the Riemannian gradient, which costs $\mathcal{O}(p^3)$ per iteration due to the inversion of $\boldsymbol{\Theta}$. All other operations are quite efficient thanks to the LRaD structure. Specifically, the vector transport on the manifold needs $O(pr^2 + pr + r^3)$, and the retraction requires $O(pr)$. Usually, we have $r \ll p$. Thus, the computational burden lies in the operations of Euclidean space, and the manifold-related operations are efficient.

IV. Experiments

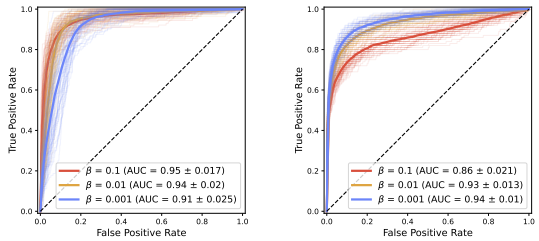
A. Synthetic Data

In this study, we generate two synthetic graphs with $p = 50$, i.e., the graph with LRaD structures and the Watts-Strogatz (WS) random graph. For the LRaD graph, we first generate a sparse matrix $\mathbf{Y}^* \in \mathbb{R}^{50\times 10}$ and let $\boldsymbol{\Theta}^* = \mathbf{Y}^*\mathbf{Y}^{*\top} + \mathbf{D}^*$, where $\mathbf{D}^* \in \mathcal{D}^p_{++}$ is a random diagonal matrix such that $\boldsymbol{\Theta}^* \in \mathcal{S}^{50\times 50}_{++}$. We generate the adjacency matrix $\mathbf{W}^*$ of WS random graphs, where each node has five neighbours, and the edges are randomly reconnected with probability 0.1. The edge weights are sampled from the uniform distribution $\mathcal{U}(2, 5)$ by following [13]. Then, we let $\boldsymbol{\Theta}^* = \boldsymbol{\Xi}^* + 0.1\cdot\mathbf{I}$, where $\boldsymbol{\Xi}^*$ is the Laplacian matrix corresponding to $\mathbf{W}^*$, and $\mathbf{I}$ is the identity matrix. Based on the generated $\boldsymbol{\Theta}^*$, N samples are drawn from $\mathcal{N}(\mathbf{0}, (\boldsymbol{\Theta}^*)^{-1})$. We compare our method with six baselines, i.e., GGFM [18], GLASSO [9], SGL [12], NGL [13], SigRep [2], and LogPDS [8]. The first four methods are based on GGMs, and the last two are based on graph signal processing. In GL, detecting the existence of edges in a graph can be regarded as a binary classification problem. Thus, we evaluate results using the receiver operating characteristic (ROC) curves obtained from the estimated precision matrices. The ROC curves depict the relationship between the true positive rate (TPR) and the false positive rate (FPR), where TPR denotes the capacity to recover real edges, and FPR denotes the false recovery of non-existent edges. We calculate the area under curve (AUC) of each ROC curve, whose value is in $[0, 1]$, where 1 represents a perfect recovery of the true edges. We do not compare edge weights learned by different baselines since they use different normalization methods, which may not be comparable. The parameters of all baselines are selected



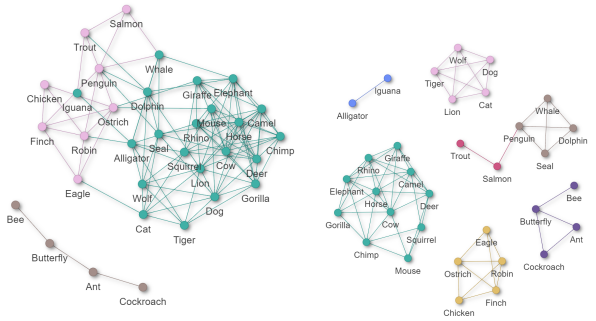
(a) LRaD graph, $N/p = 2, \beta = 0.05$ (b) LRaD graph, $N/p = 4, \beta = 0.05$ (c) WS graph, $N/p = 2, \beta = 0.05$ (d) WS graph, $N/p = 4, \beta = 0.05$

Fig. 2: ROC curves of the learned graphs. The GLASSO is ignored in the LRaD graph due to numerical divergence.



(a) LRaD graph, $N/p = 2, r = 20$ (b) WS graph, $N/p = 2, r = 20$

Fig. 3: Effect of sparsity parameter β of our method.



(a) GLASSO (b) Ours: $r = 10$, $\beta = 2.5$

Fig. 4: The graphs of the *animal* dataset. Colors of the nodes represent communities detected from the learned graphs.

as those achieving the best AUC values. All results are the average of 50 independent experiments.

Figure 2 depicts ROC curves of the graphs estimated by different methods. Overall, our method outperforms all baselines, especially when N is small. As N increases, the performance improvement of our method becomes less significant. In the LRaD graph, our method achieves competitive AUC results when $r = 10$ due to the low-rank structure of the underlying precision matrix. The performance of our method improves as r increases. In the WS graph, our method is inferior to the baselines when $r = 10$ since it may not suffice to describe the real WS graph. However, our method obtains better AUC results when r increases to 15 and 20 despite WS graphs do not exactly have LRaD structures. The results show that we can reasonably reduce the dimensionality by assuming the LRaD structure without significantly sacrificing performance.

TABLE I: Modularity of the learned graphs.

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

Figure 3 depicts the effect of the parameter β. It is observed that our method can achieve satisfactory performance in a wide range of β. In the WS graph, our method is more sensitive to β, meaning that we should carefully choose this parameter.

B. Real Data

We test the proposed method using the *animal* dataset [11], which contains the descriptions of different animals. For example, is the animal poisonous or can the animal fly? In total, there are $N = 102$ questions for 33 animals, and hence $\mathbf{X} \in \mathbb{R}^{33\times102}$. Since the ground-truth graph is unavailable in the real dataset, we cannot use ROC curves to evaluate the learned graphs. Note that animals of different species (such as mammals and birds) are distinct, while animals of the same species share similar descriptions. The learned graphs should have several communities. Thus, we employ the algorithm in [30] to detect communities in the learned graphs and evaluate the detection results using $modularity$ (Mod) [31], a metric to measure the separability of a graph into sub-components. High Mod values mean better separability.

Table I lists the modularity of the graphs learned by different methods. Our method obtains the highest Mod values, implying that the corresponding graph better preserves the underlying communities. We visualize the learned graphs in Fig.4. Due to space limitations, only the results of our method and GLASSO are provided. In our graph, the nodes of the animals in the same species are closely connected, while those corresponding to different species (such as mammals, birds, and insects) are disconnected. Compared with GLASSO, our method can better recover the underlying communities.

V. Conclusion

We proposed a graph estimator in GGMs by assuming an LRaD structure, which can reduce the dimensionality of the problem. A Riemannian manifold describing the LRaD constraint was constructed, and an RCG algorithm was designed to solve the proposed model. Finally, experiments on synthetic and real data showed the superiority of our method.

REFERENCES

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process.Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[3] X. Pu, T. Cao, X. Zhang, X. Dong, and S. Chen, "Learning to learn graph topologies," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 4249–4262, 2021.

[4] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16– 43, 2019.

[5] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[6] Y. Shen, G. B. Giannakis, and B. Baingana, "Nonlinear structural vector autoregressive models with application to directed brain networks," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5325–5339, 2019.

[7] A. Bolstad, B. D. Van Veen, and R. Nowak, "Causal network inference via group sparse regularization," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2628–2641, 2011.

[8] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Stat.* PMLR, 2016, pp. 920–929.

[9] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[10] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[11] B. Lake and J. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proc. 33rd Annu. Cogn. Sci. Conf*, vol. 32, no. 32, 2010.

[12] S. Kumar, J. Ying, J. V. d. M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *J. Mach. Learn. Res.*, vol. 21, no. 22, pp. 1–60, 2020.

[13] J. Ying, J. V. de Miranda Cardoso, and D. Palomar, "Noncon-vex sparse graph learning under laplacian constrained graphical model," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 7101–7113, 2020.

[14] J. Ying, J. V. De Miranda Cardoso, and D. P. Palomar, "Adaptive estimation of graphical models under total positivity," in *Proc. Int. Conf. Mach. Learn.*, vol. 202. PMLR, 23–29 Jul 2023, pp. 40 054–40 074.

[15] A. Anandkumar, V. Y. Tan, F. Huang, and A. S. Willsky, "High-dimensional gaussian graphical model selection: Walk summability and local separation criterion," *J. Mach. Learn. Res.*, vol. 13, no. 76, pp. 2293–2337, 2012.

[16] N. K. Chandra, P. Mueller, and A. Sarkar, "Bayesian scalable precision factor analysis for massive sparse gaussian graphical models," *arXiv preprint arXiv:2107.11316*, 2021.

[17] R. Zhou, J. Ying, and D. P. Palomar, "Covariance matrix estimation under low-rank factor model with nonnegative correlations," *IEEE Trans. Signal Process.*, vol. 70, pp. 4020–4030, 2022.

[18] A. Hippert-Ferrer, F. Bouchard, A. Mian, T. Vayer, and A. Breloy, "Learning graphical factor models with Riemannian optimization," in *Proc. Eur. Conf. Mach. Learn.* (ECML). Springer, 2023, pp. 349–366.

[19] P. Stoica and P. Babu, "Low-rank covariance matrix estimation for factor analysis in anisotropic noise: Application to array processing and portfolio selection," *IEEE Trans. Signal Process.*, vol. 71, pp. 1699–1711, 2023.

[20] Y. Wu, Y. Qin, and M. Zhu, "High-dimensional covariance matrix estimation using a low-rank and diagonal decomposition," *Can. J. Stat.*, vol. 48, no. 2, pp. 308–337, 2020.

[21] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

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

[23] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[24] E. Massart and P.-A. Absil, "Quotient geometry with simple geodesics for the manifold of fixed-rank positive-semidefinite matrices," *Siam J. Matrix Anal.* A., vol. 41, no. 1, pp. 171–198, 2020.

[25] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, "Low-rank optimization on the cone of positive semidefinite matrices," *SIAM J. Optim.*, vol. 20, no. 5, pp. 2327–2351, 2010.

[26] S. Zheng, W. Huang, B. Vandereycken, and X. Zhang, "Riemannian optimization using three different metrics for hermitian psd fixed-rank constraints: an extended version," *arXiv preprint arXiv:2204.07830*, 2023.

[27] R. Bhatia, *Positive definite matrices*. Princeton University Press, 2009.

[28] H. Sato, "Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses," *SIAM J. Optimiz.*, vol. 32, no. 4, pp. 2690–2717, 2022.

[29] M. R. Hestenes, E. Stiefel et al., *Methods of conjugate gradients for solving linear systems*. NBS Washington, DC, 1952, vol. 49, no. 1.

[30] G. Cordasco and L. Gargano, "Community detection via semi-synchronous label propagation algorithms," *arXiv preprint arXiv:1103.4550*, 2011.

[31] M. E. Newman, "Modularity and community structure in networks," in *Proc. Natl. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.