



HAL
open science

Learning Graphical Factor Models with Riemannian Optimization

Alexandre Hippert-Ferrer, Florent Bouchard, Ammar Mian, Titouan Vayer,
Arnaud Breloy

► To cite this version:

Alexandre Hippert-Ferrer, Florent Bouchard, Ammar Mian, Titouan Vayer, Arnaud Breloy. Learning Graphical Factor Models with Riemannian Optimization. Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2023), Sep 2023, Torino, Italy. <10.1007/978-3-031-43421-1_21>. <hal-04249001>

HAL Id: hal-04249001

<https://hal.science/hal-04249001v1>

Submitted on 19 Oct 2023


HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Learning Graphical Factor Models with Riemannian Optimization

Alexandre Hippert-Ferrer ¹[0000-0002-7740-5415], Florent Bouchard²[0000-0003-3003-7317], Ammar Mian³[0000-0003-1796-8707], Titouan Vayer⁴[0000-0002-8115-572X], and Arnaud Breloy⁵[0000-0002-3802-9015]

¹ Univ Gustave Eiffel, IGN, ENSG, LASTIG, F-77454 Marne-la-Vallée, France
alexandre.hippert-ferrer@univ-eiffel.fr

² Univ Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, Gif-sur-Yvette, France

³ Univ Savoie Mont Blanc, LISTIC, Annecy, France

⁴ Univ Lyon, Inria, CNRS, ENS de Lyon, UCB Lyon 1, LIP UMR 5668, F-69342, Lyon, France

⁵ Univ Paris-Nanterre, LEME, IUT Ville d'Avray, Ville d'Avray, France

Abstract. Graphical models and factor analysis are well-established tools in multivariate statistics. While these models can be both linked to structures exhibited by covariance and precision matrices, they are generally not jointly leveraged within graph learning processes. This paper therefore addresses this issue by proposing a flexible algorithmic framework for graph learning under low-rank structural constraints on the covariance matrix. The problem is expressed as penalized maximum likelihood estimation of an elliptical distribution (a generalization of Gaussian graphical models to possibly heavy-tailed distributions), where the covariance matrix is optionally constrained to be structured as low-rank plus diagonal (low-rank factor model). The resolution of this class of problems is then tackled with Riemannian optimization, where we leverage geometries of positive definite matrices and positive semi-definite matrices of fixed rank that are well suited to elliptical models. Numerical experiments on synthetic and real-world data sets illustrate the effectiveness of the proposed approach.

Keywords: Graph learning · Low-rank factor models · Riemannian optimization.

1 Introduction

Graphical models allow us to represent specific correlation structures between any two variables (entries) of multivariate observations. Inferring the topology of this structure directly from the data is referred to as *graph learning*, which has been increasingly leveraged in numerous applications, such as biology (Li and Gui, 2006; Smith et al., 2011; Stegle et al., 2015), finance (Marti et al., 2021), or signal processing (Shuman et al., 2013; Kalofolias, 2016).

Within Gaussian graphical models (GGMs), graph learning boils down to the problem of estimating the precision (inverse covariance) matrix of a Gaussian Markov random field (Friedman et al., 2008; Lake and Tenenbaum, 2010). In practice, achieving an accurate covariance matrix estimation is often a difficult task due to low sample support. Thus, it is common to introduce prior assumptions on the structure of this matrix that guarantee a correct estimation with fewer samples. A popular approach is related to low-rank factorizations, which relies on the assumption that the data is driven by an underlying low-dimensional linear model, corrupted by an independent perturbation. The resulting covariance matrix decomposition then involves a core that is a low-rank positive semi-definite matrix. Such model is ubiquitous in statistics, and for example, at the heart of probabilistic principal component analysis (Tipping and Bishop, 1999), low-rank factor analysis (Robertson and Symons, 2007; Khamaru and Mazumder, 2019; Rubin and Thayer, 1982), and their many generalizations.

Since GGMs and low-rank factorizations share a common root in structured covariance (or precision) matrix estimation, it appears desirable to leverage both approaches in a unified graph learning formulation. On one hand, linear dimension reduction approaches rely on particular spectral structures that can be beneficial for graph learning (Kumar et al., 2020). On the other hand, it also opens the way to graph-oriented view of sparse principal component analysis (Yoshida and West, 2010; Meng et al., 2014). Though theoretically appealing, such unification is challenging because it formulates optimization problems with objective functions and constraints that apply both on the covariance matrix and its inverse. Thus, deriving single-step learning algorithms for these models has only recently been addressed (Chandra et al., 2021).

In this paper, we propose a new family of methods for graph learning with low-rank constraints on the covariance matrix, hereafter referred to as graphical factor models (GFM). First, we reformulate graph learning as a problem that encompasses both elliptical distributions and low-rank factor models. The main interest of generalizing Gaussian graphical models to elliptical ones is to ensure robustness to underlying heavy-tailed distributions (Vogel and Fried, 2011; Finegold and Drton, 2014; Zhang et al., 2013; de Miranda Cardoso et al., 2021). Moreover, additionally considering low-rank factor models allows for an effective dimensionality reduction. The main novelty of our approach is to tackle the resulting class of constrained and penalized maximum likelihood estimation in a unified way with Riemannian optimization (Absil et al., 2008; Boumal, 2020). To do so, we leverage geometries of both the positive definite matrices (Bhatia, 2009), and positive semi-definite matrices of fixed rank (Bonnabel and Sepulchre, 2009; Bouchard et al., 2021) that are well suited to the considered models. The corresponding tools allows us to develop optimization methods that ensure the desired structures for the covariance matrix, thus providing a flexible and numerically efficient framework for learning graphical factor models.

Finally, experiments¹ conducted on synthetic and real-world data sets demonstrate the interest of considering both elliptical distributions and factor model

¹ The code is available at: <https://github.com/ahippert/graphfactormodel>.

structures in a graph learning process. We observe that the proposed algorithms lead to more interpretable graphs compared to unstructured models. Notably, the factor model approaches compare well with the current state-of-the-art on Laplacian-constrained graph learning methods that require to set the number of components as additional prior information (Kumar et al., 2020; de Miranda Cardoso et al., 2021). The interest of our method is twofold: *i*) it requires less supervision to unveil meaningful clusters in the conditional correlation structure of the data; *ii*) the computational bottleneck is greatly reduced, as the proposed algorithm iterations only requires the thin-SVD of a low-rank factor, rather than the whole SVD of the Laplacian (or adjacency) matrix.

2 Background and proposed framework

2.1 Gaussian graphical and related models

Gaussian graphical models assume that each observation is a centered multivariate Gaussian random vector $\mathbf{x} = [x_1, \dots, x_p]^\top$ with covariance matrix $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \boldsymbol{\Sigma}$, denoted $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. For the corresponding Gaussian Markov random field, an undirected graph is matched to the variables as follows: each variable corresponds to a vertex, and an edge is present between two vertices if the corresponding random variables are conditionally dependent given the others (Dempster, 1972; Lauritzen, 1996). The support of the precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ directly accounts for this conditional dependency, since

$$\text{corr}[x_q x_\ell | \mathbf{x}_{[1,p] \setminus \{q,\ell\}}] = -\boldsymbol{\Theta}_{q\ell} / \sqrt{\boldsymbol{\Theta}_{qq} \boldsymbol{\Theta}_{\ell\ell}}. \quad (1)$$

Hence, a non-zero entry $\boldsymbol{\Theta}_{q\ell}$ implies a conditional dependency between the variables x_q and x_ℓ , underlined by an edge between vertices q and ℓ of the graph. Within Gaussian graphical models, graph learning is therefore tied to the problem of estimating the precision matrix $\boldsymbol{\Theta}$ from a set of observations $\{\mathbf{x}_i\}_{i=1}^n \in (\mathbb{R}^p)^n$. In order to exhibit such correlation structure, a standard approach is to resort to regularized maximum likelihood estimation, *i.e.*, solving the problem:

$$\underset{\boldsymbol{\Theta} \in \mathcal{S}_p^{++}}{\text{maximize}} \quad \log \det(\boldsymbol{\Theta}) - \text{Tr}\{\mathbf{S}\boldsymbol{\Theta}\} - \lambda h(\boldsymbol{\Theta}), \quad (2)$$

where $\mathbf{S} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$ is the sample covariance matrix, h is a regularization penalty, and $\lambda \in \mathbb{R}^+$ is a regularization parameter. The ℓ_1 -norm is often used as penalty in order to promote a sparse structure in $\boldsymbol{\Theta}$, which is at the foundation of the well-known GLasso algorithm (Friedman et al., 2008; Lake and Tenenbaum, 2010; Mazumder and Hastie, 2012; Fattahi and Sojoudi, 2019). Many other convex or non-convex penalties have been considered in this setting (Lam and Fan, 2009; Shen et al., 2012; Benfenati et al., 2020). Depending on the assumptions, it can also be beneficial to consider penalties that promote certain types of structured sparsity patterns (Heinävaara et al., 2016; Tarzanagh and Michailidis, 2018). Another main example of structure within the conditional

correlations is total positivity, also known as attractive Gaussian graphical models, that assumes $\Theta_{q\ell} \leq 0$, $\forall q \neq \ell$ (Fallat et al., 2017; Lauritzen et al., 2019). In attractive Gaussian graphical models, the identifiability of the precision matrix to the graph Laplacian (Chung, 1997), has also attracted recent interest in graph learning (Egilmez et al., 2017; Ying et al., 2020; Kumar et al., 2020).

2.2 Elliptical distributions

A first shortcoming of graph learning as formulated in (2) is its lack of robustness to outliers, or heavy-tailed distributed samples. This is a consequence of the underlying Gaussian assumption, that cannot efficiently describe such data. A possible remedy is to consider a larger family of multivariate distributions. In this context, elliptical distributions (Anderson and Fang, 1990; Kai-Tai and Yao-Ting, 1990) offer a good alternative, that are well-known to provide robust estimators of the covariance matrix (Maronna, 1976; Tyler, 1987; Wald et al., 2019). In our present context, this framework has been successfully used to extend graphical models (Vogel and Fried, 2011; Finegold and Drton, 2014; Zhang et al., 2013; de Miranda Cardoso et al., 2021), as well as low-rank structured covariance matrices models (Zhao and Jiang, 2006; Zhou et al., 2019; Bouchard et al., 2021) for corrupted or heavy-tailed data.

A vector is said to follow a centered elliptically symmetric distribution of scatter matrix Σ and density generator g , denoted $\mathbf{x} \sim \mathcal{ES}(\mathbf{0}, \Sigma, g)$, if its density has the form

$$f(\mathbf{x}) \propto \det(\Sigma)^{-1/2} g(\mathbf{x}_i^\top \Sigma^{-1} \mathbf{x}_i), \quad (3)$$

which yields the negative log-likelihood

$$\mathcal{L}(\Sigma) \propto \frac{1}{n} \sum_{i=1}^n \rho(\mathbf{x}_i^\top \Sigma^{-1} \mathbf{x}_i) + \frac{1}{2} \log |\Sigma| + \text{const.} \quad (4)$$

for the sample set $\{\mathbf{x}_i\}_{i=1}^n$, where $\rho(t) = -\log g(t)$. Notice that using $g(t) = \exp(-t/2)$ allow us to recover the Gaussian case. However, the density generator g enables more flexibility, and notably to encompass many heavy-tailed multivariate distributions. Among popular choices, elliptical distributions include the multivariate t -distribution with degree of freedom $\nu > 2$, which is obtained with $g(t) = (1+t/\nu)^{-\frac{\nu+p}{2}}$. For this distribution, the parameter ν measures the rate of decay of the tails. The Gaussian case also corresponds to the limit case $\nu \rightarrow \infty$.

2.3 Low-rank factor models

A second limitation of (2) is that it does not account for other potential structure exhibited by the covariance or precision matrices. This can be problematic when the sample support is low ($n \simeq p$, or $n < p$) as the sample covariance matrix is not a reliable estimate in these setups (Ledoit and Wolf, 2004; Smith, 2005; Vershynin, 2012). In this context, a popular approach consists in imposing low-rank

structures on the covariance matrix. These structures arise from the assumption that the data is driven by an underlying low-dimensional linear model, *i.e.*, $\mathbf{x} = \mathbf{W}\mathbf{s} + \boldsymbol{\epsilon}$, where \mathbf{W} is a rank- k factor loading matrix, and $\mathbf{s} \in \mathbb{R}^k$ and $\boldsymbol{\epsilon} \in \mathbb{R}^p$ are independent random variables. Thus the resulting covariance matrix is of the form $\boldsymbol{\Sigma} \triangleq \mathbf{H} + \boldsymbol{\Psi}$, where $\mathbf{H} = \mathbf{W}\mathbb{E}[\mathbf{s}\mathbf{s}^\top]\mathbf{W}^\top$ belongs to the set of positive semi-definite matrices of rank k , denoted $\mathcal{S}_{p,k}^+ = \{\boldsymbol{\Sigma} \in \mathcal{S}_p, \boldsymbol{\Sigma} \succcurlyeq \mathbf{0}, \text{rank}(\boldsymbol{\Sigma}) = k\}$. This model is well known in statistics, and for example, at the core of probabilistic principal component analysis, that assumes $\boldsymbol{\Psi} \propto \mathbf{I}_p$ (Tipping and Bishop, 1999). Also notice that in Laplacian-constrained models, a rank- k precision matrix implies a $(p - k)$ -component graph (Chung, 1997). Hence it is also interesting to leverage such spectral structures from the graph learning perspective (Kumar et al., 2020).

In this paper, we will focus on the low-rank factor analysis models (Robertson and Symons, 2007; Rubin and Thayer, 1982; Khamaru and Mazumder, 2019), that consider the general case $\boldsymbol{\Psi} \in \mathcal{D}_p^{++}$, where $\mathcal{D}_p^{++} = \{\boldsymbol{\Sigma} = \text{diag}(\mathbf{d}), \mathbf{d} \in \mathbb{R}_{+*}^p\}$ denotes the space of positive definite diagonal matrices. Given this model, the covariance matrix belongs to the space of rank- k plus diagonal matrices, denoted as

$$\mathcal{M}_{p,k} = \left\{ \boldsymbol{\Sigma} = \mathbf{H} + \boldsymbol{\Psi}, \mathbf{H} \in \mathcal{S}_{p,k}^+, \boldsymbol{\Psi} \in \mathcal{D}_p^{++} \right\}. \quad (5)$$

Notice that this parameterization reduces the dimension of the estimation problem reduces from $p(p + 1)/2$ to $p(k + 1) - k(k - 3)/2$, which is why it is often used in regimes with few samples, or high dimensional settings.

2.4 Learning elliptical graphical factor models

We cast the problem of graph learning for elliptical graphical factor models as

$$\begin{aligned} & \underset{\boldsymbol{\Sigma} \in \mathcal{S}_p^{++}}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{\Sigma}) + \lambda h(\boldsymbol{\Sigma}) \\ & \text{subject to} \quad \boldsymbol{\Sigma} \in \mathcal{M}_{p,k}, \end{aligned} \quad (6)$$

where \mathcal{L} is the negative-log likelihood in (4), and $\mathcal{M}_{p,k}$ is the space of rank- k plus diagonal positive definite matrices in (5). The penalty h is a smooth function that promotes a sparse structure on the graph, and $\lambda \in \mathbb{R}_+$ is a regularization parameter. Although this formulation take into account many options, we focus on the usual element-wise penalty applied to the precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$, defined as:

$$h(\boldsymbol{\Sigma}) = \sum_{q \neq \ell} \phi([\boldsymbol{\Sigma}^{-1}]_{q\ell}) \quad (7)$$

It is important to notice that the considered optimization framework will require h to be smooth. This is why we presently use a surrogate of the ℓ_1 -norm defined as

$$\phi(t) = \varepsilon \log(\cosh(t/\varepsilon)), \quad (8)$$

with $\varepsilon > 0$ ($\lim_{\varepsilon \rightarrow 0} \phi(t)$ yields the ℓ_1 -norm). In practice, we use $\varepsilon = 1e^{-12}$. However, note that the output of the algorithm is not critically sensitive to this

Algorithm 1: Graph learning with elliptical graphical (factor) models

Input: Data $\{\mathbf{x}_i\}_{i=1}^n \in (\mathbb{R}^p)^n$ **Parameters :** GGM/GGFM/EGM/EGFM \rightarrow density generator g , rank k
Regularization parameter λ and penalty shape ϕ
Tolerance threshold tol **Output:** Learned graph adjacency \mathbf{A} (Boolean)**if** $k=p$ **then**GGM/EGM initialization $\boldsymbol{\Sigma}_0 = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$ **for** $t = 0$ **to convergence do**

Compute the Riemannian gradient with (15) and (16)

 Computes steps α_t and β_t with (Hestenes and Stiefel, 1952) Update $\boldsymbol{\Sigma}_{t+1}$ with (14) (transport in (12), retraction in (13))**else**

GGFM/EGFM initialization

 $(\mathbf{V}_0 = k$ leading eigenvectors of $\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$, $\mathbf{A}_0 = \mathbf{I}_k$, $\boldsymbol{\Psi}_0 = \mathbf{I}_p)$ **for** $t = 0$ **to convergence do**

Compute the Riemannian gradient with (27)

 Computes steps α_t and β_t with (Hestenes and Stiefel, 1952) Update $(\mathbf{V}_{t+1}, \mathbf{A}_{t+1}, \boldsymbol{\Psi}_{t+1})$ with (14) transposed to $\mathcal{M}_{p,k}$ (i.e., with transport in (27), retraction in (33)) $\boldsymbol{\Sigma}_{\text{end}} = \mathbf{V}_{\text{end}} \mathbf{A}_{\text{end}} \mathbf{V}_{\text{end}}^\top + \boldsymbol{\Psi}_{\text{end}}$ Compute $\boldsymbol{\Theta} = \boldsymbol{\Sigma}_{\text{end}}^{-1}$ and conditional correlations matrix $\tilde{\boldsymbol{\Theta}}$ from (34)Activate edges (i, j) in adjacency \mathbf{A} for each $\tilde{\boldsymbol{\Theta}}_{ij} \geq \text{tol}$

parameter: the obtained results were similar for ε ranging from $1e^{-6}$ to the numerical tolerance.

In conclusion, the problem in (6) accounts for a low-rank structure in a graph learning formulation, which is also expected to be more robust to data following an underlying heavy-tailed distributions. We then introduce four main cases, and their corresponding acronyms:

GGM/GGFM: Gaussian graphical factor models (GGFM) are obtained with the Gaussian log-likelihood, *i.e.*, setting $g(t) = \exp(-t/2)$ in (4). The standard Gaussian graphical models (GGM) as in (2) are recovered when dropping the constraint $\boldsymbol{\Sigma} \in \mathcal{M}_{p,k}$.

EGM/EGFM: elliptical graphical factor models (EGFM) are obtained for the more general case where \mathcal{L} defines a negative log-likelihood of an elliptical distribution. Dropping the constraint $\boldsymbol{\Sigma} \in \mathcal{M}_{p,k}$ also yields a relaxed problem that we refer to as elliptical graphical models (EGM).

3 Learning graphs with Riemmanian optimization

The optimization problem in (6) is non-convex and difficult to address. Indeed, it involves objective functions and constraints that apply on both the covariance matrix and its inverse. One approach to handle these multiple and complex

constraints is to resort to variable splitting and alternating direction method of multipliers (Kovnatsky et al., 2016) which has, for example, been considered for Laplacian-constrained models in (Zhao et al., 2019; de Miranda Cardoso et al., 2021). In this work, we propose a new and more direct approach by harnessing Riemannian optimization (Absil et al., 2008; Boumal, 2020). Besides being computationally efficient, this solution also has the advantage of not being an approximation of the original problem, nor requiring extra tuning parameters.

For GGM and EGM, the resolution of (6) requires to consider optimization on \mathcal{S}_p^{++} . This will be presented in Section 3.1, which will serve as both a short introduction to optimization on smooth Riemannian manifold, and as a building block for solving (6) for GGFM and EGFM. For these models, we will derive Riemannian optimization algorithms on $\Sigma \in \mathcal{M}_{p,k}$ in Section 3.2, which will be done by leveraging a suitable geometry for this space. The corresponding algorithms are summarized in algorithm 1.

3.1 Learning GGM/EGM: optimization on \mathcal{S}_p^{++}

When relaxing the constraint in (6), the problem still requires to be solved on \mathcal{S}_p^{++} . Since \mathcal{S}_p^{++} is open in \mathcal{S}_p the tangent space $T_{\Sigma}\mathcal{S}_p^{++}$ can simply be identified as \mathcal{S}_p , for any $\Sigma \in \mathcal{S}_p^{++}$. To be able to perform optimization, the first step is to define a Riemannian metric, *i.e.*, an inner product on this tangent space. In the case of \mathcal{S}_p^{++} , the most natural choice is the affine-invariant metric (Skovgaard, 1984; Bhatia, 2009), that corresponds to the Fisher information metric of the Gaussian distribution and features very interesting properties from a geometrical point of view. It is defined for all $\Sigma \in \mathcal{S}_p^{++}$, $\xi, \eta \in \mathcal{S}_p$ as

$$\langle \xi, \eta \rangle_{\Sigma}^{\mathcal{S}_p^{++}} = \text{tr}(\Sigma^{-1}\xi\Sigma^{-1}\eta). \quad (9)$$

The Riemannian gradient (Absil et al., 2008) at $\Sigma \in \mathcal{S}_p^{++}$ of an objective function $f : \mathcal{S}_p^{++} \rightarrow \mathbb{R}$ is the only tangent vector such that for all $\xi \in \mathcal{S}_p$,

$$\langle \nabla^{\mathcal{S}_p^{++}} f(\Sigma), \xi \rangle_{\Sigma}^{\mathcal{S}_p^{++}} = Df(\Sigma)[\xi], \quad (10)$$

where $Df(\Sigma)[\xi]$ denotes the directional derivative of f at Σ in the direction ξ . The Riemannian gradient $\nabla^{\mathcal{S}_p^{++}} f(\Sigma)$ can also be obtained from the Euclidean gradient $\nabla^{\mathcal{E}} f(\Sigma)$ through the formula

$$\nabla^{\mathcal{S}_p^{++}} f(\Sigma) = \Sigma \text{sym}(\nabla^{\mathcal{E}} f(\Sigma)) \Sigma, \quad (11)$$

where $\text{sym}(\cdot)$ returns the symmetrical part of its argument. The Riemannian gradient is sufficient in order to define a descent direction of f , *i.e.*, a tangent vector inducing a decrease of f , hence to define a proper Riemannian steepest descent algorithm. However, if one wants to resort to a more sophisticated optimization algorithm such as conjugate gradient or BFGS, vector transport (Absil et al., 2008) is required to transport a tangent vector from one tangent space to another. In the case of \mathcal{S}_p^{++} , we can employ the most natural one, *i.e.*, the

one corresponding to the parallel transport associated with the affine-invariant metric (Jeuris et al., 2012). The transport of the tangent vector ξ of Σ onto the tangent space of $\bar{\Sigma}$ is given by

$$\mathcal{T}_{\Sigma \rightarrow \bar{\Sigma}}^{\mathcal{S}_p^{++}}(\xi) = (\bar{\Sigma} \Sigma^{-1})^{1/2} \xi (\Sigma^{-1} \bar{\Sigma})^{1/2}. \quad (12)$$

Now that we have all the tools needed to obtain a proper descent direction of some objective function f , it remains to be able to get from the tangent space back onto the manifold \mathcal{S}_p^{++} . This is achieved by means of a retraction map. The best solution on \mathcal{S}_p^{++} in order to ensure numerical stability while taking into account the chosen geometry is

$$R_{\Sigma}^{\mathcal{S}_p^{++}}(\xi) = \Sigma + \xi + \frac{1}{2} \xi \Sigma^{-1} \xi. \quad (13)$$

This retraction corresponds to a second order approximation of the geodesics of \mathcal{S}_p^{++} (Jeuris et al., 2012), which generalize the concept of straight lines for a manifold. We now have all the necessary elements to perform the optimization of f on \mathcal{S}_p^{++} . For instance, the sequence of iterates $\{\Sigma_t\}$ and descent directions $\{\xi_t\}$ generated by a Riemannian conjugate gradient algorithm is

$$\begin{aligned} \Sigma_{t+1} &= R_{\Sigma_t}^{\mathcal{S}_p^{++}}(\xi_t) \\ \xi_t &= \alpha_t (-\nabla^{\mathcal{S}_p^{++}} f(\Sigma_t) + \beta_t \mathcal{T}_{\Sigma_{t-1} \rightarrow \Sigma_t}^{\mathcal{S}_p^{++}}(\xi_{t-1})), \end{aligned} \quad (14)$$

where α_t is a stepsize that can be computed through a linesearch (Absil et al., 2008) and β_t can be computed using the rule in (Hestenes and Stiefel, 1952).

From there, to obtain a specific algorithm that solves (6) on \mathcal{S}_p^{++} , it only remains to provide the Riemannian gradients of the negative log-likelihood \mathcal{L} and penalty h . The Riemannian gradient of \mathcal{L} at $\Sigma \in \mathcal{S}_p^{++}$ is

$$\nabla^{\mathcal{S}_p^{++}} \mathcal{L}(\Sigma) = \frac{1}{2} \Sigma - \frac{1}{2n} \sum_i u(\mathbf{x}_i^\top \Sigma^{-1} \mathbf{x}_i) \mathbf{x}_i \mathbf{x}_i^\top, \quad (15)$$

where $u(t) = -2g'(t)/g(t)$. For the Gaussian distribution, we have $u(t) = 1$. For the t -distribution with degree of freedom ν , we have $u(t) = (\nu + p)/(\nu + t)$. Concerning h , the $q\ell$ element of its Riemannian gradient at $\Sigma \in \mathcal{S}_p^{++}$ is

$$[\nabla^{\mathcal{S}_p^{++}} h(\Sigma)]_{q\ell} = \begin{cases} 0 & \text{if } q = \ell \\ \phi'([\Sigma^{-1}]_{q\ell}) & \text{if } q \neq \ell, \end{cases} \quad (16)$$

where $\phi'(t) = \tanh(t/\varepsilon)$. In the next section, the Euclidean gradients of \mathcal{L} and h in \mathcal{S}_p^{++} are used. They are easily obtained from (11) and by noticing that the Euclidean gradient is already symmetric.

3.2 Learning GGFm/EGFM: optimization on $\mathcal{M}_{p,k}$

When considering the factor model, we aim at finding a structured matrix Σ in $\mathcal{M}_{p,k}$, which is a smooth submanifold of \mathcal{S}_p^{++} . The Riemannian geometry of

$\mathcal{M}_{p,k}$ is not straightforward. In fact, defining an adequate geometry for low-rank matrices is a rather difficult task and, while there were many attempts, see *e.g.*, (Bonnabel and Sepulchre, 2009; Meyer et al., 2011; Vandereycken et al., 2012; Massart and Absil, 2018; Neuman et al., 2021; Bouchard et al., 2021), no perfect solution has been found yet. Here, we choose the parametrization considered in (Bonnabel and Sepulchre, 2009; Meyer et al., 2011; Bouchard et al., 2021). From a geometrical perspective, the novelty here is to adopt the particular structure of $\mathcal{M}_{p,k}$, *i.e.*, by combining a low-rank positive semi-definite matrix and a diagonal positive definite matrix. Each of the two manifolds of this product are well-known and the proofs of the following can easily be deduced from the proofs in (Bouchard et al., 2021). All $\Sigma \in \mathcal{M}_{p,k}$ can be written as

$$\Sigma = \mathbf{V}\mathbf{A}\mathbf{V}^\top + \Psi, \quad (17)$$

where $\mathbf{V} \in \text{St}_{p,k} = \{\mathbf{V} \in \mathbb{R}^{p \times k} : \mathbf{V}^\top \mathbf{V} = \mathbf{I}_k\}$ (Stiefel manifold of $p \times k$ orthogonal matrices), $\mathbf{A} \in \mathcal{S}_k^{++}$ and $\Psi \in \mathcal{D}_p^{++}$. Let $\mathcal{N}_{p,k} = \text{St}_{p,k} \times \mathcal{S}_k^{++} \times \mathcal{D}_p^{++}$ and

$$\begin{aligned} \varphi : \quad & \mathcal{N}_{p,k} \rightarrow \mathcal{S}_p^{++} \\ & (\mathbf{V}, \mathbf{A}, \Psi) \mapsto \mathbf{V}\mathbf{A}\mathbf{V}^\top + \Psi. \end{aligned} \quad (18)$$

It follows that $\mathcal{M}_{p,k} = \varphi(\mathcal{N}_{p,k})$. Therefore, to solve (6) on $\mathcal{M}_{p,k}$, one can exploit φ and solve it on $\mathcal{N}_{p,k}$. However, $\mathcal{N}_{p,k}$ contains invariance classes with respect to φ , that is for any $\mathbf{O} \in \mathcal{O}_k$ (orthogonal group with $k \times k$ matrices),

$$\varphi(\mathbf{V}\mathbf{O}, \mathbf{O}^\top \mathbf{A}\mathbf{O}, \Psi) = \varphi(\mathbf{V}, \mathbf{A}, \Psi). \quad (19)$$

As a consequence, the space that best corresponds to $\mathcal{M}_{p,k}$ is the quotient manifold $\mathcal{N}_{p,k}/\mathcal{O}_k$ induced by equivalence classes on $\mathcal{N}_{p,k}$

$$\pi(\mathbf{V}, \mathbf{A}, \Psi) = \{(\mathbf{V}, \mathbf{A}, \Psi) * \mathbf{O} : \mathbf{O} \in \mathcal{O}_k\}, \quad (20)$$

where $(\mathbf{V}, \mathbf{A}, \Psi) * \mathbf{O} = (\mathbf{V}\mathbf{O}, \mathbf{O}^\top \mathbf{A}\mathbf{O}, \Psi)$. To efficiently solve an optimization problem on $\mathcal{M}_{p,k}$ with the chosen parametrization, we thus need to consider the quotient $\mathcal{N}_{p,k}/\mathcal{O}_k$. In practice, rather than dealing with the quite abstract manifold $\mathcal{N}_{p,k}/\mathcal{O}_k$ directly, we will manipulate objects in $\mathcal{N}_{p,k}$, implying that the different points of an equivalence class (20) are, in fact, one and only point.

Our first task is to provide the tangent space of $\theta = (\mathbf{V}, \mathbf{A}, \Psi) \in \mathcal{N}_{p,k}$. It is obtained by aggregating tangent spaces $T_{\mathbf{V}}\text{St}_{p,k}$, $T_{\mathbf{A}}\mathcal{S}_k^{++}$ and $T_{\Psi}\mathcal{D}_p^{++}$, *i.e.*,

$$T_{\theta}\mathcal{N}_{p,k} = \{(\xi_{\mathbf{V}}, \xi_{\mathbf{A}}, \xi_{\Psi}) \in \mathbb{R}^{p \times k} \times \mathcal{S}_k \times \mathcal{D}_p : \mathbf{V}^\top \xi_{\mathbf{V}} + \xi_{\mathbf{V}}^\top \mathbf{V} = \mathbf{0}\}. \quad (21)$$

Then, we define a Riemannian metric on $\mathcal{N}_{p,k}$. It needs to be invariant along equivalence classes, *i.e.*, for all $\theta \in \mathcal{N}_{p,k}$, $\xi, \eta \in T_{\theta}\mathcal{N}_{p,k}$ and $\mathbf{O} \in \mathcal{O}_k$

$$\langle \xi, \eta \rangle_{\theta}^{\mathcal{N}_{p,k}} = \langle \xi * \mathbf{O}, \eta * \mathbf{O} \rangle_{\theta * \mathbf{O}}^{\mathcal{N}_{p,k}}. \quad (22)$$

In this work, the Riemannian metric $\mathcal{N}_{p,k}$ is chosen as the sum of metrics on $\text{St}_{p,k}$, \mathcal{S}_k^{++} and \mathcal{D}_p^{++} . The metric on $\text{St}_{p,k}$ is the so-called canonical metric (Edelman

et al., 1998), which yields the simplest geometry of $\text{St}_{p,k}$. The metrics on \mathcal{S}_k^{++} and \mathcal{D}_p^{++} are the affine-invariant ones. Hence, our chosen metric on $\mathcal{N}_{p,k}$ is

$$\langle \xi, \eta \rangle_{\theta}^{\mathcal{N}_{p,k}} = \text{tr}(\boldsymbol{\xi}_{\mathbf{V}}^{\top}(\mathbf{I}_p - \frac{1}{2}\mathbf{V}\mathbf{V}^{\top})\boldsymbol{\eta}_{\mathbf{V}}) + \text{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}_{\boldsymbol{\Lambda}}) + \text{tr}(\boldsymbol{\Psi}^{-2}\boldsymbol{\xi}_{\boldsymbol{\Psi}}\boldsymbol{\eta}_{\boldsymbol{\Psi}}). \quad (23)$$

At $\theta \in \mathcal{N}_{p,k}$, $T_{\theta}\mathcal{N}_{p,k}$ contains tangent vectors inducing a move along the equivalence class $\pi(\theta)$. In our setting, these directions are to be eliminated. They are contained in the so-called vertical space \mathcal{V}_{θ} , which is the tangent space $T_{\theta}\pi(\theta)$ to the equivalence class. In our case, it is

$$\mathcal{V}_{\theta} = \{(\mathbf{V}\boldsymbol{\Omega}, \boldsymbol{\Lambda}\boldsymbol{\Omega} - \boldsymbol{\Omega}\boldsymbol{\Lambda}, \mathbf{0}) : \boldsymbol{\Omega} \in \mathcal{S}_k^{\perp}\}, \quad (24)$$

where \mathcal{S}_k^{\perp} denotes the vector space of skew-symmetric matrices. Now that the unwanted vectors have been identified, we can deduce the ones of interest: they are contained in the orthogonal complement of \mathcal{V}_{θ} according to metric (23). This space is called the horizontal space \mathcal{H}_{θ} and is equal to

$$\mathcal{H}_{\theta} = \{\xi \in T_{\theta}\mathcal{N}_{p,k} : \mathbf{V}^{\top}\boldsymbol{\xi}_{\mathbf{V}} = 2(\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi}_{\boldsymbol{\Lambda}} - \boldsymbol{\xi}_{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}^{-1})\}. \quad (25)$$

The Riemannian gradient of an objective function \bar{f} on $\mathcal{N}_{p,k}$ is defined through the chosen metric as for \mathcal{S}_p^{++} in (10). Again, it is possible to obtain the Riemannian gradient $\nabla^{\mathcal{N}_{p,k}}\bar{f}(\theta)$ on $\mathcal{N}_{p,k}$ from the Euclidean one $\nabla^{\mathcal{E}}\bar{f}(\theta)$. This is achieved by

$$\nabla^{\mathcal{N}_{p,k}}\bar{f}(\theta) = (\mathbf{G}_{\mathbf{V}} - \mathbf{V}\mathbf{G}_{\mathbf{V}}^{\top}\mathbf{V}, \boldsymbol{\Lambda}\mathbf{G}_{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}, \boldsymbol{\Psi}^2 \text{ddiag}(\mathbf{G}_{\boldsymbol{\Psi}})), \quad (26)$$

where $\nabla^{\mathcal{E}}\bar{f}(\theta) = (\mathbf{G}_{\mathbf{V}}, \mathbf{G}_{\boldsymbol{\Lambda}}, \mathbf{G}_{\boldsymbol{\Psi}})$ and $\text{ddiag}(\cdot)$ cancels the off-diagonal elements of its argument. Furthermore, since the objective functions we are interested in are invariant along equivalence classes (20), their Riemannian gradient is naturally on the horizontal space \mathcal{H}_{θ} and we can use it directly. In this work, our focus is objective functions $\bar{f} : \mathcal{N}_{p,k} \rightarrow \mathbb{R}$ such that $\bar{f} = f \circ \varphi$, where $f : \mathcal{S}_p^{++} \rightarrow \mathbb{R}$; see (6). Hence, it is of great interest to be able to get the gradient of \bar{f} directly from the one of f . The Euclidean gradient $\nabla^{\mathcal{E}}\bar{f}(\theta)$ as a function of the Euclidean gradient $\nabla^{\mathcal{E}}f(\varphi(\theta))$ is

$$\nabla^{\mathcal{E}}\bar{f}(\theta) = (2\nabla^{\mathcal{E}}f(\varphi(\theta))\mathbf{V}\boldsymbol{\Lambda}, \mathbf{V}^{\top}\nabla^{\mathcal{E}}f(\varphi(\theta))\mathbf{V}, \text{ddiag}(\nabla^{\mathcal{E}}f(\varphi(\theta)))). \quad (27)$$

While it is not necessary for the gradient, the vector transport requires to be able to project elements from the ambient space $\mathbb{R}^{p \times k} \times \mathbb{R}^{k \times k} \times \mathbb{R}^{p \times p}$ onto the horizontal space. The first step is to provide the orthogonal projection from the ambient space onto $T_{\theta}\mathcal{N}_{p,k}$. For all $\theta \in \mathcal{N}_{p,k}$ and $\xi \in \mathbb{R}^{p \times k} \times \mathbb{R}^{k \times k} \times \mathbb{R}^{p \times p}$

$$P_{\theta}^{\mathcal{N}_{p,k}}(\xi) = (\boldsymbol{\xi}_{\mathbf{V}} - \mathbf{V} \text{sym}(\mathbf{V}^{\top}\boldsymbol{\xi}_{\mathbf{V}}), \text{sym}(\boldsymbol{\xi}_{\boldsymbol{\Lambda}}), \text{ddiag}(\boldsymbol{\xi}_{\boldsymbol{\Psi}})). \quad (28)$$

From there we can obtain the orthogonal projection from $T_{\theta}\mathcal{N}_{p,k}$ onto \mathcal{H}_{θ} . Given $\xi \in T_{\theta}\mathcal{N}_{p,k}$, it is

$$\mathcal{P}_{\theta}^{\mathcal{N}_{p,k}}(\xi) = (\boldsymbol{\xi}_{\mathbf{V}} - \mathbf{V}\boldsymbol{\Omega}, \boldsymbol{\xi}_{\boldsymbol{\Lambda}} + \boldsymbol{\Omega}\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{\Omega}, \boldsymbol{\xi}_{\boldsymbol{\Psi}}), \quad (29)$$

where $\boldsymbol{\Omega} \in \mathcal{S}_k^\perp$ is the unique solution to

$$2(\mathbf{A}^{-1}\boldsymbol{\Omega}\mathbf{A} + \mathbf{A}\boldsymbol{\Omega}\mathbf{A}^{-1}) - 3\boldsymbol{\Omega} = \mathbf{V}^\top \boldsymbol{\xi}_\mathbf{V} + 2(\boldsymbol{\xi}_\mathbf{A}\mathbf{A}^{-1} + \mathbf{A}^{-1}\boldsymbol{\xi}_\mathbf{A}). \quad (30)$$

We can now define an adequate vector transport operator. Given $\theta, \bar{\theta} \in \mathcal{N}_{p,k}$ and $\xi \in \mathcal{H}_\theta$, it is simply

$$\mathcal{T}_{\theta \rightarrow \bar{\theta}}^{\mathcal{N}_{p,k}}(\xi) = \mathcal{P}_{\bar{\theta}}^{\mathcal{N}_{p,k}}(\mathcal{P}_{\bar{\theta}}^{\mathcal{N}_{p,k}}(\xi)). \quad (31)$$

The last object that we need is a retraction on $\mathcal{N}_{p,k}$. To address the invariance requirement, the retraction needs to be invariant along equivalence classes, *i.e.*, for all $\theta \in \mathcal{N}_{p,k}$, $\xi \in \mathcal{H}_\theta$ and $\mathbf{O} \in \mathcal{O}_k$

$$R_{\theta * \mathbf{O}}^{\mathcal{N}_{p,k}}(\xi * \mathbf{O}) = R_\theta^{\mathcal{N}_{p,k}}(\xi). \quad (32)$$

In this paper, we choose a second order approximation of geodesics on $\mathcal{N}_{p,k}$ given by

$$R_\theta^{\mathcal{N}_{p,k}}(\xi) = (\text{uf}(\mathbf{V} + \boldsymbol{\xi}_\mathbf{V}), \mathbf{A} + \boldsymbol{\xi}_\mathbf{A} + \frac{1}{2}\boldsymbol{\xi}_\mathbf{A}\mathbf{A}^{-1}\boldsymbol{\xi}_\mathbf{A}, \boldsymbol{\Psi} + \boldsymbol{\xi}_\boldsymbol{\Psi} + \frac{1}{2}\boldsymbol{\xi}_\boldsymbol{\Psi}^2\boldsymbol{\Psi}^{-1}), \quad (33)$$

where $\text{uf}(\cdot)$ returns the orthogonal factor of the polar decomposition.

We finally have all the tools needed to solve problem (6) on $\mathcal{N}_{p,k}$ and thus to ensure the structure of $\mathcal{M}_{p,k}$. With all the objects given in this section, many Riemannian optimization algorithms can be employed to achieve the minimization (6), *e.g.*, steepest descent, conjugate gradient or BFGS. In this work, we use in practice a conjugate gradient algorithm.

3.3 Algorithms properties

In terms of convergence, the proposed algorithms inherit the standard properties from the Riemannian optimization framework (Absil et al., 2008; Boumal, 2020): each iterate satisfies the constraints and ensures a decrease of the objective function until a critical point is reached. Due to the inversion of $\boldsymbol{\Sigma}$ at each step, the computational complexity of the Riemannian conjugate gradient for GGM/EGM is $\mathcal{O}(Tp^3)$, where T is the number of iterations. This complexity has the same dependence in p than GLasso (Friedman et al., 2008) and more recent structured graph learning algorithms (Kumar et al., 2020). On the other hand, by relying on the structure of the factor models, the Riemannian conjugate gradient for GGFm/EGFM has a $\mathcal{O}(T(pk^2 + k^3))$ computational complexity². As we usually set $k \ll p$, EGM/EGFM are therefore more suited to large dimensions.

While the question of statistical properties will not be explored in this paper, it is also worth mentioning that (6) appears as a regularized maximum likelihood

² Note that GGFm/EGFM require the computation of the sample covariance matrix (SCM), which is in $\mathcal{O}(np^2)$. EGM/EGFM require to re-compute a weighted SCM at each step (*i.e.*, $\mathcal{O}(Tnp^2)$). However, these operations consist only in matrix multiplications that can be parallelized, thus are not actual computational bottlenecks.

estimation problem for elliptical models. Consequently, the algorithms presented in [algorithm 1](#) aim at solving a class of maximum a posteriori where a prior related to ϕ is set independently on the elements Θ_{ij} . Thus, one can expect good statistical performance if the data falls within the class of elliptical distribution, and especially the algorithms to be robust to a mismatch related to the choice of g ([Maronna, 1976](#); [Drašković and Pascal, 2018](#)).

4 Experiments

4.1 Validations on synthetic data

Due to space limitations, extensive experiments on synthetic data (ROC curves for edge detection, sensitivity to tuning parameters) are described in the supplementary material available in [section C](#) at the end of this paper. The short conclusions of these experiments are the following: *(i)* the proposed methods compare favorably to existing methods in terms of ROC curves for many underlying random graphs models (Erdős-Rényi, Barabási-Albert, Watts-Strogatz, and random geometric graph), especially when the sample support is limited; *(ii)* For the aforementioned graphs models, the covariance matrix does not always exhibit a low-rank structure, so factor models do not necessarily improve the ROC performance. However it is experienced to be extremely useful when applied to real-world data, as illustrated below (improved clustering capabilities); *(iii)* As all sparsity-promoting methods, the proposed algorithms require a proper set of the regularization parameter λ . However, the results are not critically sensitive to a change of this value when its order of magnitude is correctly set. The code for these experiments is available at: <https://github.com/ahippert/graphfactormodel>.

4.2 Real-world data sets

In this section, the performances of the proposed algorithms (GGM/GGFM and EGM/EGFM) are evaluated on three real-world data sets consisting of *(i)* animal species; *(ii)* GNSS time-series; *(iii)* concepts data. The purpose of the following experiments is to verify whether the factor model, which accounts for potential structures in the covariance matrix, provides an improved version of the learned graph topology. The latter is evaluated using both graph modularity m , for which high values measure high separability of a graph into sub-components ([Newman, 2006](#)), and visual inspection³.

Benchmarks and parameter setting: The proposed algorithms are compared with state-of-the-art approaches for *(i)* *connected* graphs: GLasso ([Friedman et al., 2008](#)), which uses ℓ_1 -norm as a sparse-promoting penalty, and NGL ([Ying et al.,](#)

³ Note that there is no available ground truth for these data sets. Hence, to facilitate visualization, each graph node is then clustered into colors using a community detection algorithm based on label propagation ([Cordasco and Gargano, 2010](#)).

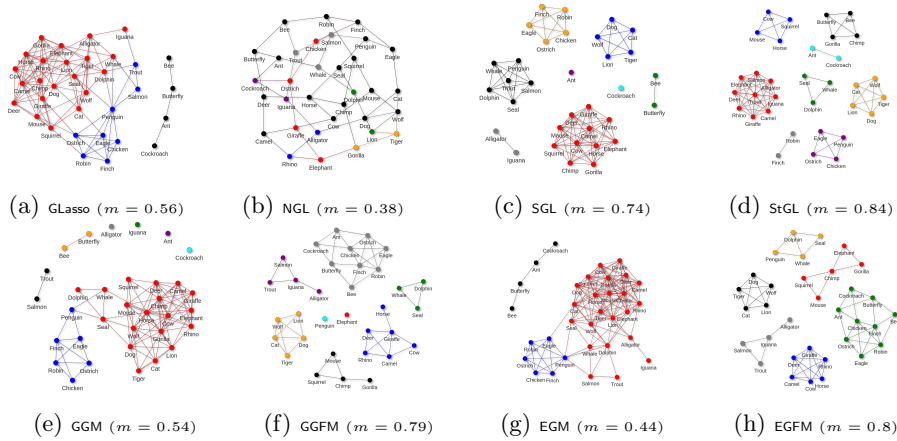


Fig. 1: Learned graphs from the *animals* data set with various algorithms and their associated modularity m . For GGFM/EGFM, the rank is 10. For SGL and StGL, the number of components is fixed to 8.

2020), which use a Gaussian Laplacian-constrained model with a concave penalty regularization; (ii) *multi-component* graphs: SGL (Kumar et al., 2020), which constraints the number of graph components in an NGL-type formulation, and StGL (de Miranda Cardoso et al., 2021), which generalizes the above to t -distributed data. For a fair comparison, the parameters of all tested algorithms (including competing methods) are tuned to display the best results. Notably, the proposed algorithms are set using algorithm 1 with a tolerance threshold of $\text{tol} = 10^{-2}$. For EGM/EGFM, the chosen density generator relates to a t -distribution with degrees of freedom $\nu = 5$. The rank of factor models is chosen manually. Still, the results were similar for other ranks around the set value.

Animals data: In the *animals* data set (Osherson et al., 1991; Lake and Tenenbaum, 2010) each node represents an animal, and is associated with binary features (categorical non-Gaussian data) consisting of answers to questions such as “has teeth?”, “is poisonous?”, *etc.* In total, there are $p = 33$ unique animals and $n = 102$ questions. Figure 1 displays the learned graphs of the *animals* data set. Following the recommendations of (Kumar et al., 2020; Egilmez et al., 2017), the SGL algorithm is applied with $\beta = 0.5$ and $\alpha = 0$, GLasso with $\alpha = 0.05$ and the input for SGL and NGL is set to $\frac{1}{n} \sum_i \mathbf{x}_i \mathbf{x}_i^\top + \frac{1}{3} \mathbf{I}_p$. While EGM yields results similar to GLasso, GGM provides a clearer structure with (*trout, salmon*) and (*bee, butterfly*) clustered together. Interestingly, with no assumptions regarding the number of components, GGFM and EGFM reach similar structure and modularity compared to SGL and StGL, that require to set *a priori* the number of components in the graph.

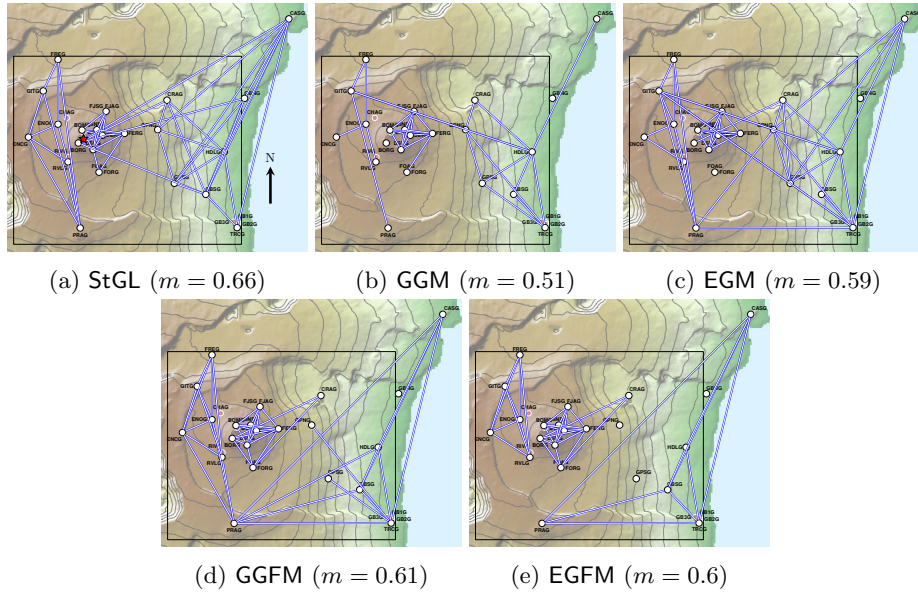


Fig. 2: Learned graphs from the GNSS data with the StGL algorithm, our proposed approaches and their associated modularity m . For StGL, the number of components is fixed to 3. For GGFM and EGFM, the rank is fixed to 4. For the t -distribution, the degrees of freedom are set to $\nu = 5$. The red star in (a) points to the summit crater. ©WEBOBS/IPGP.

GNSS data: We present an application on Earth surface displacement data collected by a network of Global Navigation Satellite System (GNSS) receivers from the volcanological observatory of Piton de la Fournaise (OVPF-IPGP), located at La Réunion Island. The presented network is composed of $p = 22$ receivers recording daily vertical displacements from January 2014 to March 2017 (Smittarello et al., 2019), with a total of $n = 1106$ observations. During this time, vertical displacements induced by volcano eruptions have been recorded by the receivers network, sometimes leading to abrupt motion changes over time. Depending on their spatial position, some receivers might move in a particular direction (upwards or downwards), thus indicating (thin) spatial correlations. Results of the learned graphs are presented in Figure 2. A general observation is that all graphs are mainly clustered into three groups: two located West (receivers ‘GITG’, ‘FREG’, etc.) and East (‘BORG’, ‘FJAG’, etc.) of the summit crater, and one extending from lower altitudes to the seashore (‘CASG’, ‘TRCG’, etc.). We call these groups *west*, *east* and *low* components, respectively. As described in (Peltier et al., 2017), the four 2015 eruptions (February, May, July and August) are characterized by asymmetric displacement patterns with respect to the North-South eruptive fissures which extend slightly west-

ward from the summit crater. Interestingly, this corresponds to the separation between *west* and *east* graph components, which is best evidenced by factor model-based algorithms, especially EGFM. This result can be explained by the fact that GGFM/EGFM are more robust to abrupt changes in the data as it occurs in heavy-tailed data distributions. The *low* graph component corresponds to receivers with small or no displacement. Note that the ‘PRAG’ receiver is also included in this group, probably because it did not record a significant motion during this period. Finally, GGM, EGM and StGL lead to similar results but with spurious connections from the west side of the crater to seashore receivers (*e.g.*, from ‘BORG’ to ‘CASG’ for StGL, and ‘PRAG’ connected to some *west* and/or *east* receivers).

Concepts data: The *concepts* data set (Lake and Tenenbaum, 2010) collected by Intel Labs, includes $p = 1000$ nodes⁴ and $n = 218$ semantic features. As stated in (Cai et al., 2022), “each node denotes a concept such as ‘house’, ‘coat’, and ‘whale’, and each semantic feature is a question such as ‘Can it fly?’, ‘Is it alive?’, and ‘Can you use it?’ The answers are on a five-point scale from ‘definitely no’ to ‘definitely yes’, conducted on Amazon Mechanical Turk”. This dataset will only be used to assess the suitability of our factor model approaches, which are expected to be more efficient for such high-dimensional setting. Since GGFM and EGFM lead to very similar results, only the former is shown. In comparison, GGM and EGM did not provide as clean interpretable results, which motivates the use of factor models in practice. Because the “true” number of components is unclear in this data, methods that require prior setting of this variable were also found hard to exploit. The graph learned by the GGFM algorithm is presented in Figure 3. This graph is mostly composed of connected sub-components, each falling into interpretable categories of similar concepts (“tools” in grey, “body parts” in blue, etc.). A closer look on the graph in Figure 4 shows interesting nodes that can be interpreted as links between the clusters of concepts (*e.g.*, the node ‘body’ linking the cluster of “body parts” to the one of “whole bodies”). Another interpretation is that concepts with ambivalent meaning also act as nodes between these clusters (‘gas’ can be associated to “drinks” as well as to “natural elements”).

5 Conclusion

In summary, we have proposed a family of graph learning algorithms in a unified formulation involving elliptical distributions, low-rank factor models, and Riemannian optimization. Our experiments demonstrated that the proposed approaches can evidence interpretable correlation structures in exploratory data analysis for various applications.

⁴ For a clearer visualization, only 500 nodes are processed, and isolated nodes are manually removed from the displayed graph.

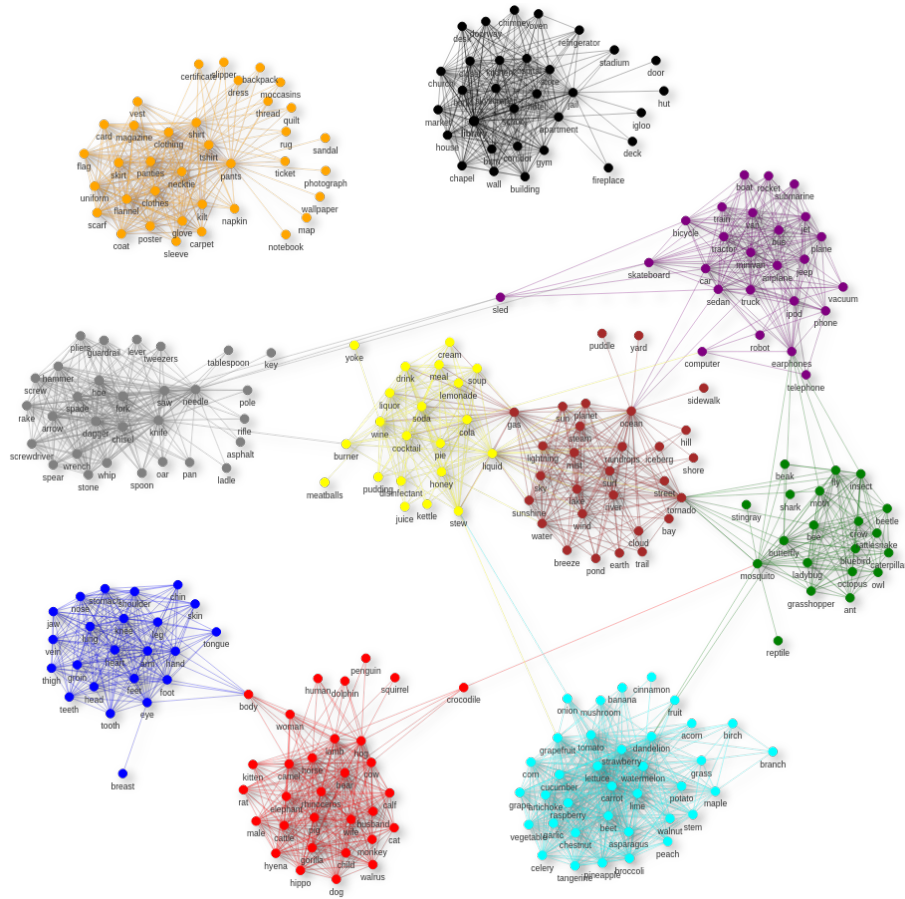


Fig. 3: Learned graph with the GGFM algorithm ($k = 10$ and $\lambda = 3.5$). Note that black and orange graph components are disconnected from the rest. Isolated nodes have been removed by hand for better visualization;

Acknowledgements This work was supported by the MASSILIA project (ANR-21-CE23-0038-01) of the French National Research Agency (ANR). We thank V. Pinel for providing us the GNSS data set and D. Smittarello for insightful comments on the GNSS results. We also thank J. Ying for sharing the *concepts* data set.

Supplementary materials to: “Learning Graphical Factor Models with Riemannian Optimization”

A Outline

Section B explains potential ethical issues regarding the collection of the ground truth of the *concepts* data set and underlines the limits of our graph predictions.

Section C presents validation experiments on synthetic data:

- Subsection C.1 describes the experimental setup, as well as the graph and data generation parameters.
- Subsection C.2 displays performance comparisons between the proposed algorithms and other state of the art graph learning methods.
- Subsection C.3 presents a study of robustness regarding the selection of the parameters of the proposed methods (namely, the rank k in factor models, and the regularization parameter λ for the sparsity promoting penalty).

The code for reproducing these experiments is made available in the following repository:

<https://github.com/ahippert/graphfactormodel>

B Ethical statement

B.1 Data set collection

A part of our numerical experiments are conducted on data sets representing related “entities” of the real world (*animals* and *concepts* data sets). We (scientists of the data/machine learning/signal processing communities) are conscious that both data sets are limited, non-exhaustive and partial. For example, the *concepts* data set has been labelled by humans using a survey conducted on the Amazon Mechanical Turk (AMT) platform. As recently documented in [Crawford \(2021\)](#), these data sets might incorporate sociological and cultural biases, embedding a particular view of the world they intend to fit. Furthermore, AMT, amongst other micro-working platforms, employs precarious *crowdworkers* around the world at low pay rates ([Gray and Suri, 2019](#)) for annotating, labelling and correcting data that help to train and test AI models. This, somehow, maintains the illusion that AI systems are autonomous and intelligent by hiding real workers ([Aytes, 2012](#)) and the labor-intensive process they require ([Tubaro and Casilli, 2019](#)).

B.2 Limits of the graph prediction

Our proposed algorithms have been designed to learn graph connections between related entities. As discussed above, the *concepts* data set might incorporate various forms of biases because each semantic feature are questions which require a subjective answer. As no ground truth is available for the considered data sets, we advise users to carefully interpret the predicted graph structures. Indeed, spurious graph links might be learned, which would highlight the aforementioned biases. Any predicted graph is no immutable truth and should be questioned in a critical manner. We strongly encourage the development of strategies to mitigate spurious correlations, as reliance on expert advice (*i.e.*, a volcanologist for the GNSS data set) or the design of more open, transparent and representative training data.

C Numerical experiments on synthetic graphs

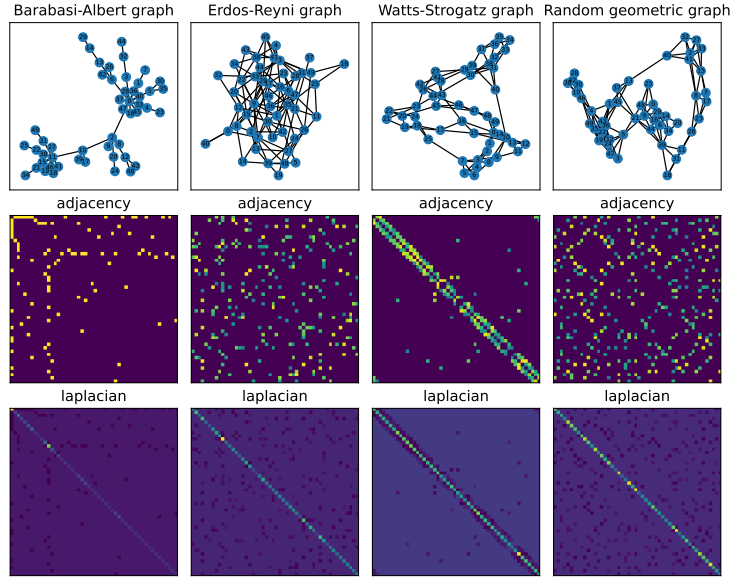
In the following, we present extensive validation experiments on synthetic graphs to evaluate the performance of the proposed algorithms, *i.e.*, GGM, EGM, GGFM and EGFM.

C.1 Experimental settings

Graph and sample generation: For a fixed dimension p , we consider four standard random graphs models that are described in [Table 1](#). Given a sampled support of the graph, weights of the edges then are sampled from $U(2, 5)$ (as in ([Ying et al., 2020](#))), which yields a symmetric weighted adjacency matrix \mathbf{A} for each random graph. [Figure 5](#) shows examples of sampled graph structures and

Graph model	Edge weights	Probability	Neighbors	Radius
Barabási-Albert	$U(2, 5)$	–	–	–
Erdős-Rényi	$U(2, 5)$	0.1	–	–
Watts-Strogatz	$U(2, 5)$	0.1	5	–
Random geometric	$U(2, 5)$	–	–	0.2

Table 1: Considered graph models in numerical experiments.

Fig. 5: Considered graph structures (here with $p = 50$ nodes) and associated adjacency and laplacian (precision) matrices.

their adjacency matrices for each of the four models. The Laplacian matrix of the graph is then used to construct the precision matrix Θ , using the relation $\Theta = L + \kappa I$, where L is the Laplacian matrix $L = D - A$ (where D is the degree matrix of A), and where $\kappa = 1e^{-1}$ is used so that Θ is non-singular. For the data generation, a total of n samples $\{\mathbf{x}_i \in \mathbb{R}^p\}_{i=1}^n$ are drawn from an elliptical graphical model, parameterized by its covariance matrix $\Sigma = \Theta^{-1}$ and density generator g , *i.e.*, $\mathbf{x}_i \sim \mathcal{ES}(\mathbf{0}, \Theta^{-1}, g)$. For the density generator g , we use a Student's t -distribution with degrees of freedom ν .

Performance evaluation: For a given parameter setup, we sample the graph-data pair (\mathbf{A}, \mathbf{X}) as described above. Graph learning algorithms are then applied to the input data \mathbf{X} . These algorithms can provide different types of estimate structures with various inherent normalization, which are not always comparable. For a meaningful comparison, we consider evaluating the receiver operating characteristic (ROC) curves obtained from the estimated adjacency matrix $\hat{\mathbf{A}}$.

The ROC curves displays the true positive rate (tpr) as function of the false positive rate (fpr): in our case, tpr denotes the capacity of the algorithm to recover actual edges of the algorithm, whereas fpr accounts for the false discovery of non-existing edges. For each curve, the area under curve (AUC) is computed. The AUC takes values in $[0, 1]$, with 1 indicating a perfect recovery of the true edges. These ROC curves are obtained as follows: First, we compute the conditional correlation coefficients

$$\tilde{\Theta}_{ij} = \text{corr} [x_q x_\ell | \mathbf{x}_{[1,p] \setminus \{q,\ell\}}] = -\Theta_{q\ell} / \sqrt{\Theta_{qq} \Theta_{\ell\ell}}. \quad (34)$$

obtained from any output estimate of the precision matrix Θ (respectively, Σ^+ if the algorithm provides an estimate of the covariance matrix). Given these coefficients estimates, a ROC curve is obtained by varying a threshold tol , i.e., the edge (i, j) is considered active if $\tilde{\Theta}_{ij} > \text{tol}$. The displayed ROC curves are finally obtained from the average of 50 Monte-Carlo experiments.

Compared methods: The proposed algorithms (GGM/GGFM/EGM/EGFM) are compared with state-of-the-art approaches to learn unstructured graphs: GLasso (Friedman et al., 2008), which uses a Gaussian model and ℓ_1 -norm as a sparse-promoting penalty; NGL and SGL (Ying et al., 2020; Kumar et al., 2020), which use a Gaussian Laplacian-constrained model with a concave penalty regularization (without prior knowledge on the number of graph components NGL and SGL solve a similar problem with a slightly different implementation); StGL (de Miranda Cardoso et al., 2021), which generalizes the above to t -distributed data. For a fair comparison, the regularization parameters of all tested algorithms (including competing methods) are tuned to the best of our ability in order to display the best results. Notably, we notice that all these methods are not critically sensitive to a slight change of their regularization parameter λ (given a reasonable order of magnitude). Furthermore, a robustness analysis concerning this point (performance versus change in parameters) is also performed subsequently for the proposed methods.

C.2 Results: ROC curves comparisons in different setups

Figure 6, 7, 8, 9 show the results with $p = 50$ and $\nu = 3.5$ (non-Gaussian distributed data) for Barabási-Albert, Erdos-Rényi, Watts-Strogatz and random geometric models, respectively. Two sample support settings are considered, i.e., $n = 2p$ and $n = 5p$. Overall, the proposed approaches outperform all compared methods (i.e., larger AUC), except for Barabási-Albert where no significant differences are observed. Notable gains are obtained when n is low (i.e., when less samples are available). When n increases, the gap in performances between proposed and compared methods gets thinner. We also notice that EGM outperforms GGM, which was to be expected since the underlying data distribution is not Gaussian. As the generated graphs do not necessarily yield a low-rank structured covariance matrix, EGFM and GGFM do not outperform their full-rank counterparts in the considered setting. Still, these algorithms do not perform

significantly worse, which means that one can reasonably reduce the dimension of the model (and benefit from a lower computational complexity).

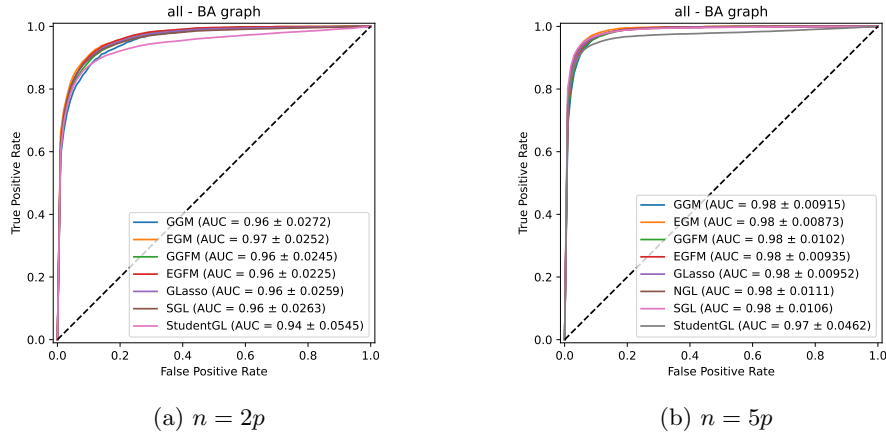


Fig. 6: Mean ROC curves obtained from estimated adjacency matrices $\hat{\mathbf{A}}$ of a **Barabási-Albert** model with GGM/EGM ($\lambda = 0.05$), GGFM/EGFM ($\lambda = 0.01; k = 20$), GLasso ($\alpha = 0.1$), NGL ($\lambda = 0.1$), SGL ($\alpha = 0.1$) and StudentGL (1 component) algorithms. Note that NGL is not displayed for $n = 2p$ because of numerical divergence.

C.3 GGM/GGFM/EGM/EGFM: robustness to parameters λ and k

Figure 10 shows the sensitivity of the proposed algorithms to the sparsity parameter λ for each considered graph model, with $p = 50$ and $\nu = 3$. We observe that the performances are slightly different as λ varies. For certain graph models (especially random geometric) GGM/EGM appear to be sensitive to this parameter, so its order of magnitude should be chosen carefully. We also observe that factor-model based approaches GGFM/EGFM have the interesting property of being less sensitive to a changing λ for each graph model.

Figure 11 shows the sensitivity to the rank k , also with $p = 50$ and $\nu = 3$. We observe that acceptable performances are obtained when k is decreased. As the data is not necessarily low-rank, similar results are obtained compared to GGM/EGM, which was already observed in subsection C.2. However, unlike GGM/EGM, we show that factor-model based approaches are particularly useful for providing interpretable and computable graphs from real-world data, as illustrated in subsection 4.2.

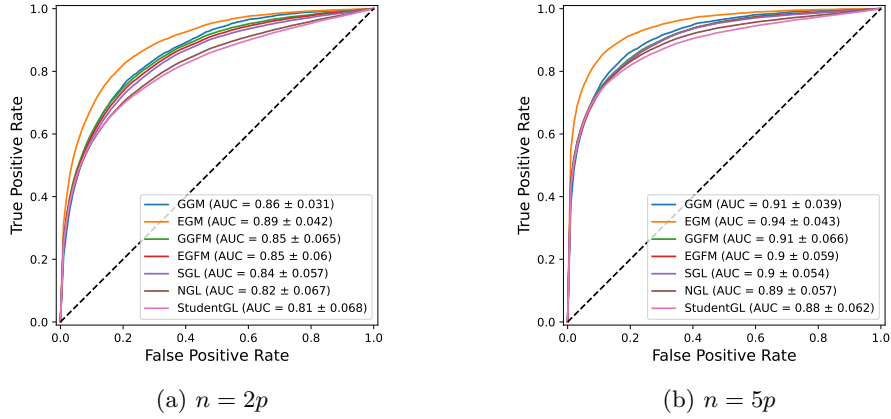


Fig. 7: Mean ROC curves obtained from estimated adjacency matrices $\hat{\mathbf{A}}$ of a **Erdős-Rényi** model with GGM/EGM ($\lambda = 0.05$), GGFM/EGFM ($\lambda = 0.01$; $k = 20$), GLasso ($\alpha = 0.1$), NGL ($\lambda = 0.1$), SGL ($\alpha = 0.1$) and StudentGL (1 component) algorithms. Note that GLasso is not displayed because of numerical divergence.

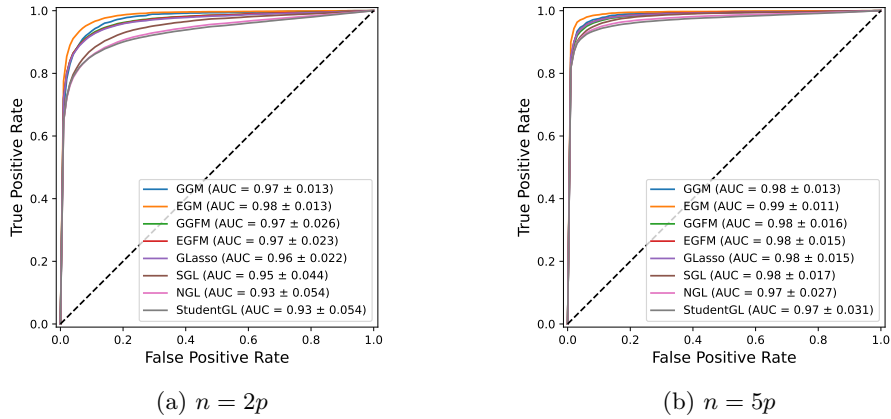


Fig. 8: Mean ROC curves obtained from estimated adjacency matrices $\hat{\mathbf{A}}$ of a **Watts Strogatz** model with GGM/EGM ($\lambda = 0.1$), GGFM/EGFM ($\lambda = 0.01$; $k = 10$), GLasso ($\alpha = 0.05$), NGL ($\lambda = 0.1$), SGL ($\alpha = 0.1$) and StudentGL (1 component) algorithms.

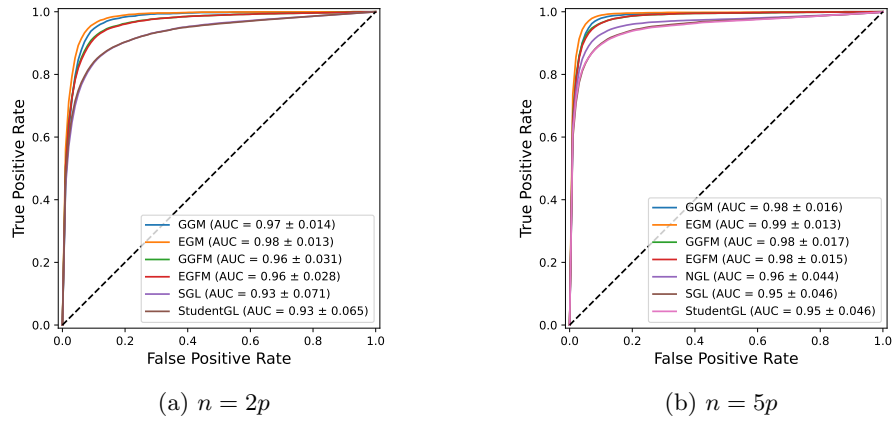


Fig. 9: Mean ROC curves obtained from estimated adjacency matrices $\hat{\mathbf{A}}$ of a **Random geometric** model with GGM ($\lambda = 0.1$), EGM ($\lambda = 0.1$), GGFM ($\lambda = 0.01$; $k = 20$), EGFM ($\lambda = 0.05$; $k = 20$), NGL ($\lambda = 0.02$), SGL ($\alpha = 0.1$) and StudentGL (1 component) algorithms. Note that GLasso is not displayed because of numerical divergence, neither is NGL for $n = 2p$.

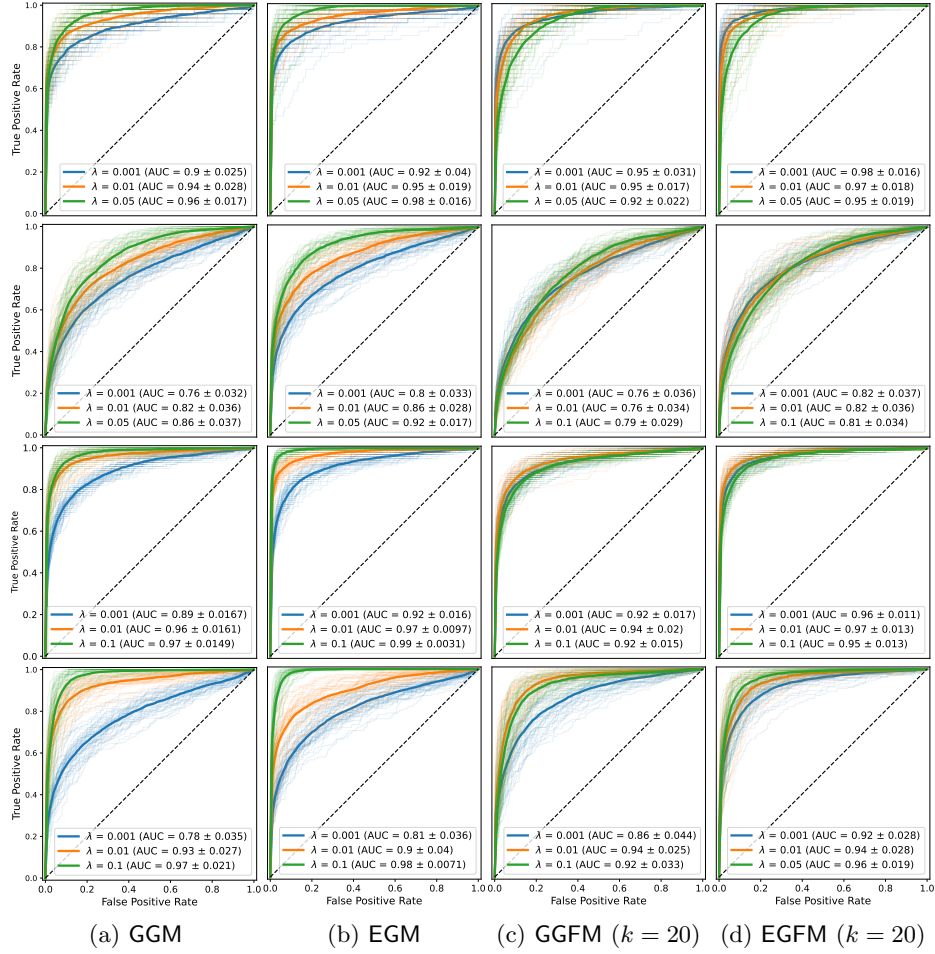


Fig. 10: **Sensitivity to λ** : mean ROC curves obtained from the estimated adjacency matrix $\hat{\mathbf{A}}$ of Barabási-Albert (first row), Erdős-Rényi (second row), Watts-Strogatz (third row) and Random geometric (fourth row) graphs with GGM/EGM/GGFM/EGFM algorithms with different values of the sparsity parameter λ .

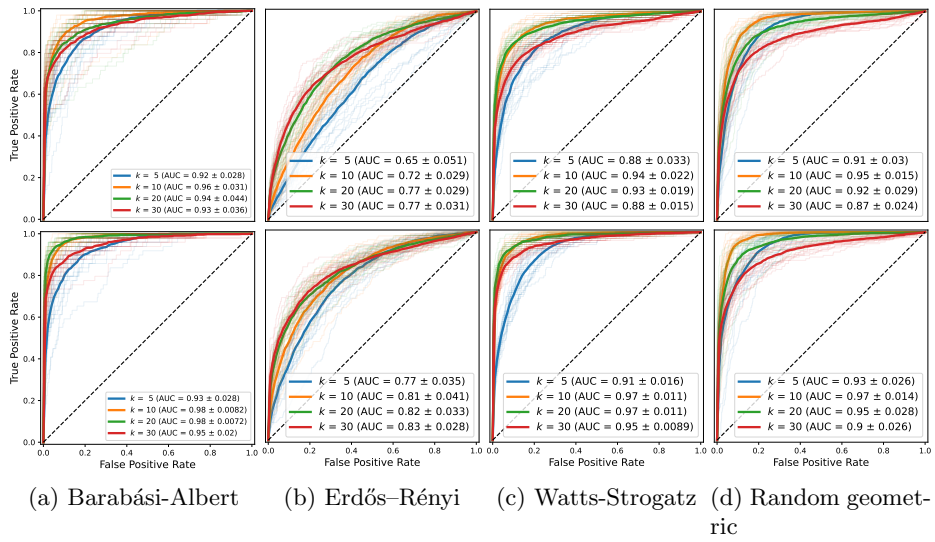


Fig. 11: **Sensitivity to k** : mean ROC curves obtained from the estimated adjacency matrix \hat{A} with GGFM/EGFM algorithms (first and second row, respectively) with different values of the rank k . For each case, parameters λ are fixed according to Figure 10.

Bibliography

- Absil, P.-A., Mahony, R., and Sepulchre, R. (2008). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA.
- Anderson, T. W. and Fang, K.-T. (1990). Theory and applications of elliptically contoured and related distributions.
- Aytes, A. (2012). Return of the crowds: Mechanical turk and neoliberal states of exception. In *Digital labor*, pages 87–105. Routledge.
- Benfenati, A., Chouzenoux, E., and Pesquet, J.-C. (2020). Proximal approaches for matrix optimization problems: Application to robust precision matrix estimation. *Signal Processing*, 169:107417.
- Bhatia, R. (2009). *Positive definite matrices*. Princeton University Press.
- Bonnabel, S. and Sepulchre, R. (2009). Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070.
- Bouchard, F., Breloy, A., Ginolhac, G., Renaux, A., and Pascal, F. (2021). A riemannian framework for low-rank structured elliptical models. *IEEE Transactions on Signal Processing*, 69:1185–1199.
- Boumal, N. (2020). An introduction to optimization on smooth manifolds. *Available online*, May, 3.
- Cai, J.-F., Cardoso, J. V. d. M., Palomar, D. P., and Ying, J. (2022). Fast projected newton-like method for precision matrix estimation under total positivity.
- Chandra, N. K., Mueller, P., and Sarkar, A. (2021). Bayesian scalable precision factor analysis for massive sparse gaussian graphical models. *arXiv preprint arXiv:2107.11316*.
- Chung, F. R. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.
- Cordasco, G. and Gargano, L. (2010). Community detection via semi-synchronous label propagation algorithms. *2010 IEEE International Workshop on: Business Applications of Social Network Analysis (BASNA)*, pages 1–8.
- Crawford, K. (2021). *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press.
- de Miranda Cardoso, J. V., Ying, J., and Palomar, D. (2021). Graphical models in heavy-tailed markets. *Advances in Neural Information Processing Systems*, 34:19989–20001.
- Dempster, A. P. (1972). Covariance selection. *Biometrics*, pages 157–175.
- Drašković, G. and Pascal, F. (2018). New insights into the statistical properties of m -estimators. *IEEE Transactions on Signal Processing*, 66(16):4253–4263.
- Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353.

- Egilmez, H. E., Pavez, E., and Ortega, A. (2017). Graph learning from data under laplacian and structural constraints. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):825–841.
- Fallat, S., Lauritzen, S., Sadeghi, K., Uhler, C., Wermuth, N., and Zwiernik, P. (2017). Total positivity in markov structures. *The Annals of Statistics*, pages 1152–1184.
- Fattahi, S. and Sojoudi, S. (2019). Graphical lasso and thresholding: Equivalence and closed-form solutions. *Journal of machine learning research*.
- Finegold, M. A. and Drton, M. (2014). Robust graphical modeling with t-distributions. *arXiv preprint arXiv:1408.2033*.
- Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.
- Gray, M. L. and Suri, S. (2019). *Ghost work: How to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books.
- Heinävaara, O., Leppä-Aho, J., Corander, J., and Honkela, A. (2016). On the inconsistency of ℓ_1 -penalised sparse precision matrix estimation. *BMC bioinformatics*, 17(16):99–107.
- Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear equation. *Journal of research of the National Bureau of Standards*, 49(6):409.
- Jeuris, B., Vandebril, R., and Vandereycken, B. (2012). A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis*, 39:379–402.
- Kai-Tai, F. and Yao-Ting, Z. (1990). *Generalized multivariate analysis*, volume 19. Science Press Beijing and Springer-Verlag, Berlin.
- Kalofolias, V. (2016). How to learn a graph from smooth signals. In *Artificial Intelligence and Statistics*, pages 920–929. PMLR.
- Khamaru, K. and Mazumder, R. (2019). Computation of the maximum likelihood estimator in low-rank factor analysis. *Mathematical Programming*, 176(1):279–310.
- Kovnatsky, A., Glashoff, K., and Bronstein, M. M. (2016). Madmm: a generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision*, pages 680–696. Springer.
- Kumar, S., Ying, J., de Miranda Cardoso, J. V., and Palomar, D. P. (2020). A unified framework for structured graph learning via spectral constraints. *J. Mach. Learn. Res.*, 21(22):1–60.
- Lake, B. and Tenenbaum, J. (2010). Discovering structure by learning sparse graphs.
- Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of statistics*, 37(6B):4254–4278.
- Lauritzen, S., Uhler, C., and Zwiernik, P. (2019). Maximum likelihood estimation in gaussian models under total positivity. *The Annals of Statistics*, 47(4):1835–1863.
- Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.
- Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.

- Li, H. and Gui, J. (2006). Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317.
- Maronna, R. A. (1976). Robust m-estimators of multivariate location and scatter. *The annals of statistics*, pages 51–67.
- Marti, G., Nielsen, F., Bińkowski, M., and Donnat, P. (2021). A review of two decades of correlations, hierarchies, networks and clustering in financial markets. *Progress in Information Geometry*, pages 245–274.
- Massart, E. and Absil, P.-A. (2018). Quotient geometry with simple geodesics for the manifold of fixed-rank positive-semidefinite matrices. *Technical Report UCL-INMA-2018.06*.
- Mazumder, R. and Hastie, T. (2012). The graphical lasso: New insights and alternatives. *Electronic journal of statistics*, 6:2125.
- Meng, Z., Eriksson, B., and Hero, A. (2014). Learning latent variable gaussian graphical models. In *International Conference on Machine Learning*, pages 1269–1277. PMLR.
- Meyer, G., Bonnabel, S., and Sepulchre, R. (2011). Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *Journal of Machine Learning Research*, 12:593–625.
- Neuman, A. M., Xie, Y., and Sun, Q. (2021). Restricted Riemannian geometry for positive semidefinite matrices. *arXiv preprint arXiv:2105.14691*.
- Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.
- Osherson, D. N., Stern, J., Wilkie, O., Stob, M., and Smith, E. E. (1991). Default probability. *Cognitive Science*, 15(2):251–269.
- Peltier, A., Froger, J.-L., Villeneuve, N., and Catry, T. (2017). Assessing the reliability and consistency of InSAR and GNSS data for retrieving 3D-displacement rapid changes, the example of the 2015 Piton de la Fournaise eruptions. *Journal of Volcanology and Geothermal Research*, 344:106–120.
- Robertson, D. and Symons, J. (2007). Maximum likelihood factor analysis with rank-deficient sample covariance matrices. *Journal of Multivariate Analysis*, 98(4):813–828.
- Rubin, D. B. and Thayer, D. T. (1982). Em algorithms for ml factor analysis. *Psychometrika*, 47(1):69–76.
- Shen, X., Pan, W., and Zhu, Y. (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232.
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98.
- Skovgaard, L. T. (1984). A riemannian geometry of the multivariate normal model. *Scandinavian journal of statistics*, pages 211–223.
- Smith, S. M., Miller, K. L., Salimi-Khorshidi, G., Webster, M., Beckmann, C. F., Nichols, T. E., Ramsey, J. D., and Woolrich, M. W. (2011). Network modelling methods for fmri. *Neuroimage*, 54(2):875–891.

- Smith, S. T. (2005). Covariance, subspace, and intrinsic crame/spl acute/r-rao bounds. *IEEE Transactions on Signal Processing*, 53(5):1610–1630.
- Smittarello, D., Cayol, V., Pinel, V., Peltier, A., Froger, J.-L., and Ferrazzini, V. (2019). Magma propagation at Piton de la Fournaise from joint inversion of InSAR and GNSS. *Journal of Geophysical Research: Solid Earth*, 124(2):1361–1387.
- Stegle, O., Teichmann, S. A., and Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145.
- Tarzanagh, D. A. and Michailidis, G. (2018). Estimation of graphical models through structured norm minimization. *Journal of machine learning research*, 18(1).
- Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.
- Tubaro, P. and Casilli, A. A. (2019). Micro-work, artificial intelligence and the automotive industry. *Journal of Industrial and Business Economics*, 46:333–345.
- Tyler, D. E. (1987). A distribution-free m-estimator of multivariate scatter. *The annals of Statistics*, pages 234–251.
- Vandereycken, B., Absil, P.-A., and Vandewalle, S. (2012). A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. *IMA Journal of Numerical Analysis*, 33(2):481–514.
- Vershynin, R. (2012). How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686.
- Vogel, D. and Fried, R. (2011). Elliptical graphical modelling. *Biometrika*, 98(4):935–951.
- Wald, Y., Noy, N., Elidan, G., and Wiesel, A. (2019). Globally optimal learning for structured elliptical losses. *Advances in Neural Information Processing Systems*, 32.
- Ying, J., de Miranda Cardoso, J. V., and Palomar, D. (2020). Nonconvex sparse graph learning under laplacian constrained graphical model. *Advances in Neural Information Processing Systems*, 33:7101–7113.
- Yoshida, R. and West, M. (2010). Bayesian learning in sparse graphical factor models via variational mean-field annealing. *J Mach Learn Res*, 11:1771–1798.
- Zhang, T., Wiesel, A., and Greco, M. S. (2013). Multivariate generalized gaussian distribution: Convexity and graphical models. *IEEE Transactions on Signal Processing*, 61(16):4141–4148.
- Zhao, J. and Jiang, Q. (2006). Probabilistic pca for t distributions. *Neurocomputing*, 69(16-18):2217–2226.
- Zhao, L., Wang, Y., Kumar, S., and Palomar, D. P. (2019). Optimization algorithms for graph laplacian estimation via ADMM and MM. *IEEE Transactions on Signal Processing*, 67(16):4231–4244.
- Zhou, R., Liu, J., Kumar, S., and Palomar, D. P. (2019). Robust factor analysis parameter estimation. In *International Conference on Computer Aided Systems Theory*, pages 3–11. Springer.