Sparse Quadratic Approximation for Graph Learning

Dimosthenis Pasadakis [1], Matthias Bollhöfer [2], and Olaf Schenk [2]

[1]Università della Svizzera italiana — USI
[2]Affiliation not available

October 30, 2023

Abstract

Learning graphs represented by M-matrices via an l1-regularized Gaussian maximum-likelihood method is a popular approach, but also one that poses computational challenges for large scale datasets. Recently proposed methods cast this problem as a constrained optimization variant of precision matrix estimation. In this paper, we build on a state-of-the-art sparse precision matrix estimation method and introduce two algorithms that learn M-matrices, that can be subsequently used for the estimation of graph Laplacian matrices. In the first one, we propose an unconstrained method that follows a post processing approach in order to learn an M-matrix, and in the second one, we implement a constrained approach based on sequential quadratic programming. We also demonstrate the effectiveness, accuracy, and performance of both algorithms. Our numerical examples and comparative results with modern open-source packages reveal that the proposed methods can accelerate the learning of graphs by up to 3 orders of magnitude, while accurately retrieving the latent graphical structure of the data. Furthermore, we conduct large scale case studies for the clustering of COVID-19 daily cases and the classification of image datasets to highlight the applicability in real-world scenarios.

Sparse Quadratic Approximation for Graph Learning

Dimosthenis Pasadakis & Matthias Bollhöfer & Olaf Schenk *Senior Member, IEEE*

Abstract—Learning graphs represented by M-matrices via an ℓ_1-regularized Gaussian maximum-likelihood method is a popular approach, but also one that poses computational challenges for large scale datasets. Recently proposed methods cast this problem as a constrained optimization variant of precision matrix estimation. In this paper, we build on a state-of-the-art sparse precision matrix estimation method and introduce two algorithms that learn M-matrices, that can be subsequently used for the estimation of graph Laplacian matrices. In the first one, we propose an unconstrained method that follows a post processing approach in order to learn an M-matrix, and in the second one, we implement a constrained approach based on sequential quadratic programming. We also demonstrate the effectiveness, accuracy, and performance of both algorithms. Our numerical examples and comparative results with modern open-source packages reveal that the proposed methods can accelerate the learning of graphs by up to 3 orders of magnitude, while accurately retrieving the latent graphical structure of the data. Furthermore, we conduct large scale case studies for the clustering of COVID-19 daily cases and the classification of image datasets to highlight the applicability in real-world scenarios.

Index Terms—ℓ_1-regularization, Gaussian Markov Random Fields, M-matrices, partial correlations, precision matrix estimation, sign constraints

1 Introduction and related work

The representation of data in the form of graphs is a ubiquitous task in every scientific domain that deals with interacting or interconnected data. Graphs are fundamental mathematical entities with nodes (or vertices) and edges connecting them. The relationship between two connected nodes is usually captured by the scalar value of the weight of the edge that links them. In many domains data is generally available in the form of an unstructured list of samples or variables, with no available relational information among them. The construction of the latent graphical structure of such a dataset often offers an intuitive representation of the data. It can also result in a dimensionality reduction of the problem through the utilization of prior knowledge about the underlying graph (e.g. the level of sparsity or a priori information about the connectivity of the nodes). Overviews of various recent graph learning approaches can be found in [1], [2], [3].

Undirected weighted graphs, with edges representing the conditional dependence among the variables, are typically constructed with a Gaussian graphical modeling (GCM) approach [4]. In this context, each vertex corresponds to a variable, with edges being present between the vertices only if the vertices are conditionally dependent. These dependencies among the data points can be both positive and negative, and are encoded in a matrix that represents the graphical structure. The non-zero entries of this matrix correspond to the dependencies between two variables. This matrix is the inverse of the covariance matrix, also known as the precision matrix, which encodes the graphical structure of a Gaussian Markov random field (GMRF). A common prior imposed on the estimation of the precision matrix is that the conditional correlations among the random variables are sparse [5], i.e., there is a limited number of conditional correlations between the variables. This prior corresponds to imposing a degree of sparsity on the estimated precision matrix. A widely used approach for the estimation of sparse precision matrices is the ℓ_1-regularized maximum likelihood estimation (MLE), commonly referred to as the "graphical LASSO" problem [6]. A popular second-order solution method for the graphical LASSO problem with superlinear convergence is the **QU**adratic approximation of **I**nverse **C**ovariance matrices (QUIC) algorithm [7]. The Sparse QUIC (SQUIC) algorithm [8] continues the progress on large-scale, second-order methods by exploiting the underlying sparse linear algebra operations. In [9], [10] it has been shown that SQUIC is equivalently accurate and significantly faster than other state-of-the-art precision matrix estimation routines (e.g. [11], [12], [13]) in both, synthetic and real-world datasets.

More recently, GCMs under the constraint that all partial correlations are non-negative have received significant attention. The problem of finding variables that are only non-negatively correlated corresponds to enforcing an M-matrix structure on the precision matrix [14], [15]. Symmetric M-matrices are positive definite with non-positive off-diagonal elements, i.e. they are part of the set

$$\mathcal{S}_M = \left\{ \boldsymbol{\Theta} \in \mathbb{R}^{p\times p} |\, \boldsymbol{\Theta}_{ij} = \boldsymbol{\Theta}_{ji} \leqslant 0 \;\forall i \neq j,\;\; \boldsymbol{\Theta} \succ 0 \right\}, \quad (1)$$

where $\succ, \succcurlyeq$ denote positive (semi-)definiteness. Slawski and Hein [16] estimate matrices from the set (1) with a sign-constrained log-determinant divergence minimization algorithm without regularization, thus limiting the applicability of their algorithm to smaller datasets. They also establish that an a-posteriori thresholding of the off-diagonal entries of the precision matrix successfully retrieves matrices that encapsulate only the positively correlated variables. In [17] an algorithm that does not require any tuning parameters is proposed that estimates only the graphical structure without the weights. In [18] the optimization problem is solved with an alternating direction method of multipliers (ADMM) algorithm with LASSO and adaptive LASSO penalties.

Dimosthenis Pasadakis, and Olaf Schenk are with the Advanced Computing Laboratory at the Institute of Computing, Università della Svizzera italiana (USI), Lugano, Switzerland. email: {dimosthenis.pasadakis, olaf.schenk}@usi.ch.
Matthias Bollhöfer is with TU Braunschweig, Germany. email: m.bollhoefer@tu-bs.de.

A tightly connected research direction is concerned with the estimation of the combinatorial graph Laplacian, a symmetric, positive semidefinite and weakly diagonally dominant matrix. If we allow $\mathbf{\Theta} \succcurlyeq 0$ in (1), then graph Laplacians were part of the subset of (1) defined as

$$\mathcal{S}_L = \{\mathbf{\Theta} \in \mathbb{R}^{p\times p} | \, \mathbf{\Theta}_{ij} = \mathbf{\Theta}_{ji} \leqslant 0 \; \forall i \neq j; \\ \mathbf{\Theta}_{ii} = -\sum_{j:i\neq j} \mathbf{\Theta}_{ij}, \;\; \mathbf{\Theta} \succcurlyeq 0\}. \tag{2}$$

The matrices in the set (2) are singular, with their off-diagonal entries capturing the weight of the edges of the graph in reversed sign. Here, the initial work of Lake and Tenenbaum [19] focused in the estimation of graph Laplacians through the optimization of an ℓ_1-regularized MLE problem by adding a positive constant values to the diagonal entries of the graph Laplacian to account for its singularity. In [20], Egilmez et al. build upon previous of their work in the field [21], and propose an optimization framework for the estimation of graph Laplacian matrices by introducing new problem formulations with sign and structural (i.e., connectivity) constraints and develop tailored algorithms for these problems using again an ℓ_1 regularization term to enforce sparsity in the graph. Similarly, in [22] the authors convert combinatorial structural constraints into spectral ones on graph matrices, and develop an optimization framework based on block majorization-minimization to solve the graph learning problem. In [23] nonconvex regularization terms are proposed in order to enforce sparsity in the retrieved matrices.

Additionally, various M-matrix learning algorithms have been proposed based on the assumption that the graph structure emerges from a set of smooth signals. The authors in [24] adopted a factor analysis model and imposed a Gaussian probabilistic prior on the latent variables that control these signals to obtain a graphical representation. In [25] the same problem is formulated as a weighted ℓ_1 minimization, and in [26] a scalable variant is proposed that utilizes approximate nearest neighbors techniques to reduce the dimensionality of the problem.

Among a plethora of applications, M-matrices in the sets (1), (2) are commonly used in regularization [27] and clustering applications [28], [29], and their spectrum is utilized in graph partitioning tasks [30], [31].

Contributions and outline

The focus of our work is centered around the fact that the learning of M-matrices belonging to the set (1) via an ℓ_1-regularized Gaussian maximum-likelihood method is currently prohibitive for high dimensional data. Motivated by the effectiveness of SQUIC [8], [9], [10] in learning precision matrices of very large dimensions we introduce hereby two algorithms that learn graphs of non-negatively correlated random variables. The first one, SQUIC-fit, performs two consecutive unconstrained precision matrix estimations with an ℓ_1-regularized minimization. It utilizes the positively correlated variables identified in the first run as graphical bias for the retrieval of the second precision matrix, which is subsequently thresholded in order to retrieve the final M-matrix. The second one, SQUIC-sqp, is a constrained method that effectively enforces the non-positivity in the off-diagonal entries of the estimated precision matrix. The constrained minimization is achieved by means of a sequential quadratic programming (SQP) algorithm, and the corresponding projected Karush–Kuhn–Tucker (KKT) system is solved by a preconditioned conjugate gradient method (PCG).

Extensive numerical experiments are provided for both introduced algorithms. We begin with a performance and accuracy comparison with several state-of-the-art M-matrix estimation packages for synthetic datasets with up to 10^4 random variables. Then we proceed with a study on the recovery accuracy of SQUIC-fit and SQUIC-sqp when prior graphical information is available and incorporated in the optimization procedure. Following the synthetic tests, we present two didactic case studies where we highlight the applicability of the introduced algorithms in real-world datasets. For the first case study we perform the spectral clustering of $p = 3{\cdot}10^3$ US counties based on the number of daily COVID-19 cases they reported for a window of 671 days. Finally, we classify image datasets with up to $p = 7{\cdot}10^4$ dimensions based on the eigenvectors of the M-matrices estimated by the proposed algorithms.

The remainder of this paper is organized as follows. In Section 2, we briefly recap the learning of graphs in the form of precision and M-matrices when assuming that data samples are drawn from a GMRF field. In Section 3 we initially present at a high level the plain SQUIC method for large-scale sparse precision matrix estimation. We then proceed with introducing the SQUIC-fit and SQUIC-sqp algorithms for the learning of graphs in the set (1). In Section 4, we perform numerical experiments on synthetic datasets and compare with state-of-the-art methods in order to validate our proposed routines. In Section 5, we present the case studies on real-world datasets, and finally in Section 6 we draw conclusions from this work.

Notation

In what follows, we denote scalar quantities with lowercase, vectors with lowercase bold, sets by uppercase, and matrices with uppercase bold characters. The (i, j)th entry of a matrix $\mathbf{A}$ is symbolized by $\mathbf{A}_{ij}$ and all entries in row i or column j by $\mathbf{A}_{i:}$ and $\mathbf{A}_{:j}$, respectively. Sets are denoted by capital calligraphic characters, for example, $\mathcal{A}$, the identity matrix as $\mathbf{I}$ and the vector of all ones as $\mathbf{e}$.

2 Graph learning background

A common approach for various graph learning approaches consists of assuming that the data samples are drawn from a GMRF field. In subsection 2.1 we describe the problem of estimating precision matrices from GMRFs. Subsequently, in subsection 2.2 we show how this optimization procedure can be formulated in order to learn graph M-matrices in the set (1).

2.1 Sparse precision matrix estimation

The retrieval of the graphical structure of a GMRF model corresponds to the estimation of the precision matrix (inverse covariance matrix) by means of the MLE problem. The basic assumption on the given data $\mathbf{Y} \in \mathbb{R}^{p\times n}$ is that one reads its columns as a set n independently and identically distributed (i.i.d.) samples of a p-variate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^*, \mathbf{\Sigma}^*)$, where $\mathbf{\Sigma}^* \in \mathbb{R}^{p\times p}$ and $\boldsymbol{\mu}^* \in \mathbb{R}^p$ are the true covariance matrix and mean, respectively. Even if the assumption i.i.d. is not fulfilled, determining $\mathbf{\Sigma}^*$ or its inverse $\mathbf{\Theta}^* := (\mathbf{\Sigma}^*)^{-1}$ yields sufficient information that could be used for graph learning purposes. More specifically, in a setting with non-Gaussian distribution, the estimation of positive definite precision matrices can be related to the Bregman divergence regularized optimization problem [32]. The entries $\mathbf{\Theta}^*_{ij}$ of the precision matrix describe the conditional dependence of

components i, j provided that all other components are fixed and the associated graph leads to the GMRF. In statistics, the MLE method is employed to approximate $\mathbf{\Theta}^*$. To do so, the negative log-likelihood objective function

$$f(\mathbf{\Theta}) = -\log\det\mathbf{\Theta} + \mathrm{tr}[\mathbf{S}\mathbf{\Theta}] \tag{3}$$

is minimized, where $\mathbf{S} \in \mathbb{R}^{p\times p}$ is the sample covariance matrix. The graph of $\mathbf{\Theta}^*$ is essential to describe the GMRF, thus one reformulates the minimization as a LASSO problem by adding an additional ℓ_1 regularization term. This term enforces sparsity in the graphical representation of $\mathbf{\Theta}^*$ and explains the term graphical LASSO (GLASSO). Given a sparsity parameter matrix $\mathbf{\Lambda} \in \mathbb{R}^{p,p}$ with $\mathbf{\Lambda}_{ij} > 0$, we aim to solve the following convex ℓ_1-regularized negative log-likelihood problem

$$\widehat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}\succ 0}{\mathrm{argmin}}\ f(\mathbf{\Theta}) + \|\mathbf{\Lambda}\odot\mathbf{\Theta}\|_1, \tag{4}$$

where $\odot$ denotes the element-wise Hadamard product and $\mathbf{\Theta} \succ 0$ denotes positive-definiteness. The regularization term can be expanded as $\|\mathbf{\Lambda}\odot\mathbf{\Theta}\|_1 = \sum_{i,j=1}^{p}\mathbf{\Lambda}_{ij}|\mathbf{\Theta}_{ij}|$. Typically, small entries in $\mathbf{\Lambda}$ result in reduced sparsity in the estimated precision matrix $\widehat{\mathbf{\Theta}}$. Besides enforcing sparsity for the computed $\widehat{\mathbf{\Theta}}$, the minimization in (4) is also suitable when the number of dimensions p is significantly larger than the number of samples n, thus making (4) an appealing problem formulation for large-scale data science applications.

There is a plethora of methods available for solving (4), see, e.g. [6], [7], [8], [33], [34], [35] for a selection of them. Recently some of these methods have gained attraction because of using a quadratic approximation, briefly outlined hereby, with the purpose of accelerating convergence.

Let $f : \mathbf{\Theta} \to \mathbb{R}$ be the nonregularized negative log-likelihood function in (3). Up to a constant, the second-order Taylor expansion of f around $\mathbf{\Theta}$ is

$$\hat{f}(\mathbf{\Delta}) = \mathrm{tr}[(\mathbf{S}-\mathbf{W})\mathbf{\Delta}] + \frac{1}{2}\mathrm{tr}[\mathbf{W}\mathbf{\Delta}\mathbf{W}\mathbf{\Delta}], \tag{5}$$

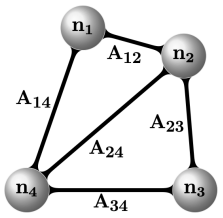
where $\mathbf{W} = \mathbf{\Theta}^{-1}$ denotes the inverse of the computed approximation $\mathbf{\Theta}$. The Newton direction $\mathbf{\Delta} \in \mathbb{R}^{p\times p}$ of the approximate objective function $\hat{f}$ can now be written as the solution of the following problem:

$$\underset{\mathbf{\Delta}}{\mathrm{argmin}}\Big\{\hat{f}(\mathbf{\Delta}) + \|\mathbf{\Lambda}\odot(\mathbf{\Theta}+\mathbf{\Delta})\|_1\Big\}. \tag{6}$$

The quadratic approximation approach solves (4) as a sequence of optimization problems of the form (6). Since (6) is an ℓ_1-regularized quadratic convex minimization problem, one can find a closed form solution if $\mathbf{\Delta}$ is restricted to the scalar case with either one single diagonal entry $\mathbf{\Delta}_{ii}$ or two single identical off-diagonal entries $\mathbf{\Delta}_{ij} = \mathbf{\Delta}_{ji}$. A natural solution strategy is a coordinate descent update where one successively minimizes over all indices $\{i, j\}$. Looking at the subgradient of $f(\mathbf{\Theta})$, it suffices to only consider $\{i, j\}$ from the free set $\mathcal{I}_{free}$, where

$$\mathcal{I}_{free} := \Big\{\{i,j\} \in \mathcal{I} : |\mathbf{S}_{ij} - \mathbf{W}_{ij}| > \mathbf{\Lambda}_{ij} \text{ or } \mathbf{\Theta}_{ij} \neq 0\Big\}, \tag{7}$$

with the cardinality of the free set $\mathcal{I}_{free}$ typically expected to be much less than p^2. This way an update matrix $\mathbf{\Delta}$ is obtained and in order to ensure that the next iterate $\mathbf{\Theta}$ is positive definite and meets an Armijo-type criterion, we update our current estimate of the optimizer $\mathbf{\Theta}$ with $\alpha\mathbf{\Delta}$ for an appropriate step size $\alpha \in [0, 1)$. For more details we refer the reader to [7] and references therein.



$$\mathbf{A} = \begin{bmatrix} 0 & \mathbf{A}_{12} & 0 & \mathbf{A}_{14} \\ \mathbf{A}_{12} & 0 & \mathbf{A}_{23} & \mathbf{A}_{24} \\ 0 & \mathbf{A}_{23} & 0 & \mathbf{A}_{34} \\ \mathbf{A}_{14} & \mathbf{A}_{24} & \mathbf{A}_{34} & 0 \end{bmatrix},$$

$$\mathbf{T}_{ii} = \begin{bmatrix} \sum_j \mathbf{A}_{1j} \\ \sum_j \mathbf{A}_{2j} \\ \sum_j \mathbf{A}_{3j} \\ \sum_j \mathbf{A}_{4j} \end{bmatrix}, \quad \mathbf{L} = \mathbf{T} - \mathbf{A}.$$

Fig. 1: A simple, undirected, and connected graph $\mathcal{G}(V, E, \mathbf{A})$ with 4 vertices and 5 edges, with its weighted adjacency $\mathbf{A}$, degree $\mathbf{T}$, and combinatorial graph Laplacian $\mathbf{L}$ matrices.

Alternatively to optimizing $f(\mathbf{\Theta})$ by a sequence of quadratic problems such as (6), some methods directly substitute $f(\mathbf{\Theta})$ by a single quadratic loss surrogate function, which is then required to be minimized. Among these methods are, e.g. [11], [36].

2.2 Sparse M-matrix estimation

M-matrices in the set (1) can be considered as precision matrices $\mathbf{\Theta}$ whose partial correlations $-\mathbf{\Theta}_{ij}/\sqrt{\mathbf{\Theta}_{ii}\mathbf{\Theta}_{jj}}$, $i \neq j$ are all non-negative [14]. The GMRF corresponding to a precision matrix of that form is referred to as attractive [37]. The constrained GLASSO estimator is defined as

$$\widehat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}\in\mathcal{S}_M}{\mathrm{argmin}}\ f(\mathbf{\Theta}) + \|\mathbf{\Lambda}\odot\mathbf{\Theta}\|_1. \tag{8}$$

The off-diagonal elements of $\widehat{\mathbf{\Theta}}$ are now constrained to non-positive values $\widehat{\mathbf{\Theta}}_{ij} \leq 0$ and correspond to the weights of the resulting graph of the GMRF. In subsection 3.3 we describe how (8) can be approximated by a sequence of quadratic and differentiable functions with linear inequality constraints.

Connection to graph Laplacians

The combinatorial graph Laplacian $\mathbf{L} \in \mathbb{R}^{p\times p}$ is a symmetric positive semidefinite matrix in the set (2) with off-diagonal elements non-positive, thus it is considered as (singular) M-matrix. The constant vector of ones $\mathbf{e}$ is in its nullspace, i.e. $\mathbf{L}\cdot\mathbf{e} = 0$, because the row and column sums of $\mathbf{L}$ are zero, i.e. $\mathbf{L}_{ii} + \sum_{i\neq j}\mathbf{L}_{ij} = 0$. A very important property of the spectrum of $\mathbf{L}$ is that the multiplicity of the zero eigenvalue corresponds to the number of connected components k of the graph [30]. This also implies that the Laplacian matrix is singular, with rank $p - k > 0$. Note that $\mathbf{L}$ encodes an improper GMRF (IGMRF) [22], [38] of rank $p - k$, as opposed to $\mathbf{\Theta}$ in (8) which is of full rank.

An undirected weighted graph $\mathcal{G}(V, E, \mathbf{A})$, as illustrated in Figure 1, is defined by its node set $V = \{1, 2, \cdots, p\}$ representing the data points, and the similarity between the edges E which is encoded in the elements of the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{p\times p}$. Its combinatorial graph Laplacian $\mathbf{L}$ can be understood in terms of the weighted adjacency matrix $\mathbf{A}$, that encodes the weights $\mathbf{A}_{ij} \geq 0$ of the edges, and the diagonal degree matrix $\mathbf{T} \in \mathbb{R}^{p\times p}$, which captures the degree of each node $\mathbf{T}_{ii} = \sum_{j=1}^{p}\mathbf{A}_{ij}$, as $\mathbf{L} = \mathbf{T} - \mathbf{A}$. The positive entries of $\mathbf{A}$, or equivalently the negative off-diagonal entries of $\mathbf{L}$, represent the edge weights of a graph, while zero entries $\mathbf{A}_{ij} = 0, i \neq j$, imply that there is no connection between nodes i and j.

Different variants of graph Laplacian matrices have also been extensively studied. The normalized symmetric $\mathbf{L}_{\text{sym}} = \mathbf{T}^{-1/2}\mathbf{L}\mathbf{T}^{-1/2}$ and random walk $\mathbf{L}_{\text{rw}} = \mathbf{T}^{-1}\mathbf{L}$ Laplacians [39] are both scaled by the degree of the edges and have been successfully used for clustering tasks [28], [40], [29]. Additionally, nonlinear reformulations of the graph Laplacian from the traditional 2-norm to the p-norm for $p \in (1, 2]$ have proven to lead to a sharp approximation of balanced cut metrics and improved clustering assignments [41], [42], [43].

All abovementioned graph Laplacian variants can be constructed after obtaining the weights of the graph's edges, encoded in the adjacency matrix $\mathbf{A}$. In what follows we estimate non-negatively correlated variables in the form of an M-matrix $\widehat{\mathbf{\Theta}}$ from an optimization problem of the form (8), and then set $\mathbf{A} = -\widehat{\mathbf{\Theta}}$. The appropriate type of graph Laplacian is subsequently built according to the application at hand.

3 Estimating M-matrices with SQUIC

In this section we present two algorithms developed for the MLE of M-matrices emerging from high dimensional datasets. Our contributions build on top of the existing SQUIC library for large scale precision matrix estimation, thus we begin in 3.1 with a short overview of the method [8], [9] and its latest development as demonstrated in [10]. In 3.2 we present the SQUIC-fit algorithm, an unconstrained approach to M-matrix estimation based on two consecutive ℓ_1-regularized optimization problems. Then, in 3.3 we introduce SQUIC-sqp, a constrained sequential quadratic programming approach for the solution of problems of the form (8).

3.1 MLE for large dimensions

The SQUIC algorithm extends the original QUIC algorithm [7] for large-scale applications and is effective for problems that exhibit a high degree of sparsity in both $\mathbf{\Theta}$ and the intermediary computations. The critical components of the MLE method based on quadratic approximations can be summarized in five tasks, namely

1) efficient data structures for the matrices $\mathbf{S}, \mathbf{\Theta}, \mathbf{W}$ and $\mathbf{\Delta}$,
2) computation of the sparse sample covariance matrix $\mathbf{S}$,
3) Cholesky decomposition of $\mathbf{\Theta}$ to check its positive definiteness and to compute $\log\det\mathbf{\Theta}$,
4) computation of the inverse of the computed approximation $\mathbf{W} \approx \mathbf{\Theta}^{-1}$,
5) efficiently solving the quadratic approximation problem (6).

The SQUIC method addresses these challenges by using compressed sparse column storage, which is common when working with sparse matrices, and by replacing several dense matrix operations by state-of-the-art sparse matrix computations. Though the sample covariance matrix $\mathbf{S}$ is approximated as being sparse, the computation is dense due to the undetermined sparsity pattern. Initially, the off-diagonal values $|\mathbf{S}_{ij}| < \mathbf{\Lambda}_{ij}$ are discarded. During the overall iteration, any values of $\mathbf{S}$ which have not been computed yet and which have a corresponding nonzero entry in $\mathbf{W}$ are computed on the fly. The kernel operation in computing the matrix $\mathbf{S}$ is matrix-matrix multiplication, which is highly parallelizable.

To efficiently compute the Cholesky decomposition, SQUIC uses the algorithm CHOLMOD [44] which is part of the SuiteSparse Matrix Collection.[1] CHOLMOD is based on the supernodal approach, which successively detects dense block structures during the factorization and produces a matrix in a hybrid format. In this format, several consecutive columns with the same nonzero pattern are treated as one dense block. These dense blocks can be efficiently handled with high-performance libraries such as the Intel(R) Math Kernel Library (MKL). For details we refer to [45]. Once the Cholesky decomposition of $\mathbf{\Theta}$ is successfully computed, $\log\det\mathbf{\Theta}$ can be easily obtained as a by-product.

The Cholesky decomposition returns a block-structured triangular factorization $\mathbf{\Theta} = \mathbf{PBDB}^\top\mathbf{P}^\top$, where $\mathbf{P}$ is a suitably chosen permutation matrix in order to reduce the fill-in for the factorization, $\mathbf{B}$ is block lower triangular with unit diagonal and $\mathbf{D}$ is block diagonal. This factorization can be employed to approximately compute $\mathbf{W} \approx \mathbf{\Theta}^{-1}$ via $\mathbf{W} \approx \mathbf{P}(\mathbf{B}^{inv})^\top\mathbf{D}^{-1}\mathbf{B}^{inv}\mathbf{P}^\top$. Here $\mathbf{B}^{inv}$ is approximated by a Neumann series applied to $\mathbf{B}^{-1} = (\mathbf{I} - \mathbf{E})^{-1}$, with $-\mathbf{E}$ being the strictly lower triangular part of $\mathbf{B}$ and with entries of small magnitude being dropped. Similarly, the final product $(\mathbf{B}^{inv})^\top\mathbf{D}^{-1}\mathbf{B}^{inv}$ also drops entries of small magnitude. We note that the computation of $\mathbf{B}^{inv}$ as well as the final product $\mathbf{P}(\mathbf{B}^{inv})^\top\mathbf{D}^{-1}\mathbf{B}^{inv}\mathbf{P}^\top$ are also efficiently parallelized in SQUIC. The convergence of the algorithm is determined by measuring that the relative difference between the objective function at the updated $\mathbf{\Theta}$ and the previous $\mathbf{\Theta}_{\text{prev}}$ is below a threshold τ, i.e. $\frac{\|f(\mathbf{\Theta}_{\text{prev}}) - f(\mathbf{\Theta})\|}{f(\mathbf{\Theta}_{\text{prev}})} < \tau$.

The quadratic optimization problem (6) also uses block structures which leads to block coordinate descent updates by efficiently recycling as many information as possible from previous descent steps. We are not going into the details of this approach and kindly refer the reader to [10].

3.2 A post-processing approach for M-matrix estimation

The first learning algorithm of M-matrices in the set (1) that we present hereby can be considered as an unconstrained ℓ_1-regularized technique. In SQUIC-fit we do not enforce additional sign constraints in the estimation of the precision matrix, but instead follow a post processing approach coupled with the utilization of a matrix sparsity parameter in order to obtain the graphical structure of non-negatively correlated variables. Our approach consists of two consecutive estimations of precision matrices $\widehat{\mathbf{\Theta}}^{(1)}, \widehat{\mathbf{\Theta}}^{(2)}$ that are solutions of a problem of the form (4). The first precision matrix $\widehat{\mathbf{\Theta}}^{(1)}$ is computed with the aid of a scalar regularization parameter λ, and is utilized in order to estimate the binary graphical structure of the non-negatively correlated variables in the data $\mathbf{Y} \in \mathbb{R}^{p\times n}$ under question. The second precision matrix $\widehat{\mathbf{\Theta}}^{(2)}$ is estimated with a matrix sparsity parameter $\mathbf{\Lambda}$ that encodes this graphical structure. Finally, the M-matrix in the set (1) is extracted by post-processing the entries of $\widehat{\mathbf{\Theta}}^{(2)}$.

An outline of the algorithmic scheme for SQUIC-fit is presented in Algorithm 1. In step 1 we aim to solve the ℓ_1-regularized negative log-likelihood problem, that is,

$$\widehat{\mathbf{\Theta}}^{(1)} = \underset{\mathbf{\Theta}\succ 0}{\operatorname{argmin}}\Big\{ -\log\det\mathbf{\Theta} + \operatorname{tr}[\mathbf{S\Theta}] + \lambda\|\mathbf{\Theta}\|_1\Big\}, \quad (9)$$

The scalar tuning parameter λ is set such that the resulting graph is sparse, and its values usually adjust the regularization according to the number of variables p and the number of features n. Then in

1. https://sparse.tamu.edu/

Algorithm 1 SQUIC-fit

input data $\mathbf{Y}$, tuning parameters λ, η, thresholds κ, τ

1: $\mathbf{estimate:}\ \widehat{\mathbf{\Theta}}^{(1)}$ // acc. (9)
2: Build graphical bias $\mathbf{G}$ // acc. (10)
3: Build matrix regularization parameter $\mathbf{\Lambda}$ // acc. (12)
4: $\mathbf{estimate:}\ \widehat{\mathbf{\Theta}}^{(2)}$ // acc. (11)
5: Build M-matrix $\widehat{\mathbf{\Theta}}$ // acc. (13)

output $\widehat{\mathbf{\Theta}}$

step 2 we estimate the structure of the negative off-diagonal entries of $\widehat{\mathbf{\Theta}}^{(1)}$ as

$$\mathbf{G}_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \mathbf{I}\left(-\widehat{\mathbf{\Theta}}^{(1)}_{ij} > \kappa\right), & \text{if } i \neq j. \end{cases} \tag{10}$$

The thresholding parameter $\kappa \geqslant 0$ is chosen sufficiently small so that all negative off-diagonal elements in $\widehat{\mathbf{\Theta}}^{(1)}_{ij}$ are detected. These values correspond to an attractive GMRF, and capture the notion of positive correlation between two nodes (variables) i, j of the graph.

In steps 3–4 we subsequently utilize the graphical structure of $\mathbf{G} \in \mathbb{R}^{p\times p}$ in the composition of the matrix tuning parameter $\mathbf{\Lambda}$ for solving

$$\widehat{\mathbf{\Theta}}^{(2)} = \underset{\mathbf{\Theta}\succ 0}{\operatorname{argmin}} \Big\{ -\log\det\mathbf{\Theta} + \operatorname{tr}[\mathbf{S}\mathbf{\Theta}] + \|\mathbf{\Lambda}\odot\mathbf{\Theta}\|_1 \Big\}. \tag{11}$$

The matrix sparsity parameter is composed as

$$\mathbf{\Lambda}_{ij} = \begin{cases} \eta & \text{for} \quad \mathbf{G}_{ij} \neq 0, \\ \lambda & \text{for} \quad \mathbf{G}_{ij} = 0. \end{cases} \tag{12}$$

where $\eta < \lambda \in \mathbb{R}$, thus the regularization matrix $\mathbf{\Lambda}$ effectively uses the sparsity pattern of $\mathbf{G}$ as a graphical bias in the estimation of the structure of $\widehat{\mathbf{\Theta}}^{(2)}$. The final step 5 of SQUIC-fit involves a post-processing procedure to construct the M-matrix from the entries of $\widehat{\mathbf{\Theta}}^{(2)}$. The final matrix $\widehat{\mathbf{\Theta}}$ is formed by selecting the structure and the weights of the non-positive off-diagonal entries of the estimated precision matrix $\widehat{\mathbf{\Theta}}^{(2)}$ as

$$\widehat{\mathbf{\Theta}}_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \mathbf{I}\left(-\widehat{\mathbf{\Theta}}^{(2)}_{ij} > \kappa\right)\widehat{\mathbf{\Theta}}^{(2)}_{ij}, & \text{if } i \neq j. \end{cases} \tag{13}$$

In both steps 1 and 4 the SQUIC algorithm is executed up to a convergence tolerance τ.

Incorporating available connectivity information for the graphical structure of non-negatively correlated variables in the data $\mathbf{Y}$ is also possible in the Algorithm 1. In this case the structure of $\mathbf{G}$ is part of the input, and the algorithm is reduced to steps 3–5.

3.3 An SQP approach for M-matrix estimation

The second learning algorithm that we introduce, SQUIC-sqp, is a constrained approach for the estimation of M-matrices based on sequential quadratic programming. The minimization of the ℓ_1-regularized log-likelihood problem in (4) restricted to the set $\mathcal{S}_M$ in (1) can be reformulated as the constrained minimization problem

$$\underset{\mathbf{\Theta}\succ 0}{\text{minimize}} \left\{ f(\mathbf{\Theta}) + \|\mathbf{\Lambda}\odot\mathbf{\Theta}\|_1 \right\}, \tag{14a}$$

$$\text{subject to } \mathbf{\Theta}_{ij} \leqslant 0 \text{ for all } i \neq j. \tag{14b}$$

In order to approximate (14) locally, we employ again the second-order Taylor expansion of f around $\mathbf{\Theta}$ as in (5). According to $\mathcal{I}_{free}$ in (7), problem (14) is restricted to entries $\mathbf{\Theta}_{ij}$ which are potentially nonzero and for this reason they can be assumed to have either positive sign ($i = j$) or negative sign ($i \neq j$). Thus the local regularized quadratic objective function can be rewritten as

$$\begin{aligned} q(\mathbf{\Delta}) = \operatorname{tr}[(\mathbf{S}-\mathbf{W})\mathbf{\Delta}] + \frac{1}{2}\operatorname{tr}[\mathbf{W}\mathbf{\Delta}\mathbf{W}\mathbf{\Delta}] \\ + \sum_i \mathbf{\Lambda}_{ii}(\mathbf{\Theta}_{ii}+\mathbf{\Delta}_{ii}) - \sum_{i\neq j}\mathbf{\Lambda}_{ij}(\mathbf{\Theta}_{ij}+\mathbf{\Delta}_{ij}). \end{aligned} \tag{15}$$

Similarly to SQUIC-fit, prior knowledge on the latent graphical structure can be incorporated in the objective function through a matrix sparsity parameter of the form (12). The constrained M-matrix estimation problem (14) is substituted by

$$\widehat{\mathbf{\Theta}} = \underset{\mathbf{\Delta}}{\operatorname{argmin}}\ q(\mathbf{\Delta}) \tag{16a}$$

$$\text{subject to } \mathbf{\Theta}_{ij} + \mathbf{\Delta}_{ij} \leqslant 0 \text{ for all } i \neq j. \tag{16b}$$

Now the local approximate function (15) is quadratic and differentiable with linear inequality constraints (16b). Therefore it can be easily solved by sequential quadratic programming [46], [47]. The SQP method distinguishes successively between active constraints (which refer to $\mathbf{\Delta}_{ij}$ such that $\mathbf{\Delta}_{ij} \approx -\mathbf{\Theta}_{ij}$) and inactive constraints (i.e. $\mathbf{\Delta}_{ij} \ll -\mathbf{\Theta}_{ij}$). The gradient of the quadratic function $q(\mathbf{\Delta})$ in (15) reads

$$\nabla q(\mathbf{\Delta}) = \mathbf{W}\mathbf{\Delta}\mathbf{W} + \mathbf{S} - \mathbf{W} + \mathbf{\Lambda}\odot(2\mathbf{I} - \mathbf{e}\mathbf{e}^\top). \tag{17}$$

Since (16b) corresponds to box constraints, the KKT system for the unconstrained variables can be solved using a straight projection, i.e., systems (16) and (17) are restricted to the diagonal entries $\mathbf{\Delta}_{ii}$ (which are always unconstrained) and the inactive off-diagonal entries $\mathbf{\Delta}_{ij}$, whereas for active constraints, the entries $\mathbf{\Delta}_{ij}$ enter as inhomogeneity. We denote the affiliated index subsets of $\mathcal{I}_{free}$ by $\mathcal{I}_u$ for the unconstrained indices and by $\mathcal{I}_c$ for the active constraints. The minimization problem is therefore reduced to solving the projected system $\nabla q(\mathbf{\Delta}) = 0$ restricted to $\mathcal{I}_u$. Note that the matrix associated with the term $\mathbf{W}\mathbf{\Delta}\mathbf{W}$ in (17) is equivalent to $\mathbf{W}\otimes\mathbf{W}$, where $\otimes$ refers to the Kronecker product (i.e. the size of the resulting matrix is squared compared with $\mathbf{W}$). Since $\mathbf{W}$ is positive definite, so is $\mathbf{W}\otimes\mathbf{W}$. This requires solving systems with the submatrix of $\mathbf{W}\otimes\mathbf{W}$ belonging to the unconstrained indices $\{i,j\} \in \mathcal{I}_u$. With respect to the linear system $\nabla q(\mathbf{\Delta}) = 0$, the sought matrix $\mathbf{\Delta} \in \mathbb{R}^{p\times p}$ is treated as a vector in a subspace of $\mathbb{R}^{p^2}$ defined via $\mathcal{I}_u$. Because of the shear size of this system, even when projected to the set of unconstrained variables, the only viable option is to use an iterative method. In our case we use the preconditioned conjugate gradient (PCG) method [47], [48] with diagonal preconditioning, implemented in a parallel fashion. It should be noted that the vectors used in the PCG method are sparse symmetric matrices stored in compressed column storage format, and as a result, the data is naturally partitioned. The parallelization is then performed along the set of columns of the underlying matrices. As an example consider $\operatorname{tr}[\mathbf{X}\mathbf{Y}]$ of two sparse symmetric matrices which takes over the role of the scalar product of two vectors. To do so, perform scalar products of the columns of $\mathbf{X}$ and $\mathbf{Y}$ in parallel and finally accumulate these independent scalar products to a single number. The SQP method successively evaluates the computed $\mathbf{\Delta}_{ij}$, checks the constraints, activates and de-activates constraints until eventually the solution is computed. We sketch the SQP part in Algorithm 2.

Algorithm 2 SQP-loop of SQUIC-sqp

input objective function $g(\boldsymbol{\Delta})$ from (15)
$\boldsymbol{\Theta}_{old} \leftarrow \boldsymbol{\Theta}$
while not satisfied **do**
  Compute set $\mathcal{I}_u \subset \mathcal{I}_{free}$ of inactive constraints and diagonal indices
  $\mathcal{I}_c \leftarrow \mathcal{I}_{free} \setminus \mathcal{I}_u$, $\boldsymbol{\Delta} \leftarrow 0$
  Call PCG with diagonal preconditioning for solving $\nabla g(\boldsymbol{\Delta}) = 0$ from (17) restricted to variables associated with $\mathcal{I}_u$ as unknowns and $\mathcal{I}_c$ as constants
  if $\boldsymbol{\Delta} \approx 0$ and there is no further descent direction **then**
    break
  else if $\boldsymbol{\Delta} \approx 0$ but there exists a further descent direction **then**
    de-activate a promising constraint and update $\mathcal{I}_u, \mathcal{I}_c$
  else
    Compute $\nu \in (0, 1]$ such that $\boldsymbol{\Theta}_{ij} + \nu\boldsymbol{\Delta}_{ij} \leqslant 0$ for all $\{i, j\} \in \mathcal{I}_u, i \neq j$
    $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \nu\boldsymbol{\Delta}$
  end if
end while
$\boldsymbol{\Delta} \leftarrow \boldsymbol{\Theta} - \boldsymbol{\Theta}_{old}$
output $\boldsymbol{\Delta}$

Due to the simplicity of box constraints in (16b), the Lagrangian multiplier $\mathbf{M}_{ij}$ of the augmented system $g(\boldsymbol{\Delta}) + \sum_{i \neq j} \mathbf{M}_{ij}(\boldsymbol{\Theta}_{ij} + \boldsymbol{\Delta}_{ij})$ is easily obtained as $\mathbf{M} = -\nabla g(\boldsymbol{\Delta})$ for all $\{i, j\} \in \mathcal{I}_c$. If there exist negative components in $\mathbf{M}$, then there must be further descent directions and the index pair $\{i, j\}$ associated with the most negative entry of $\mathbf{M}$ is a promising candidate to be de-activated. We do not go into further details of the SQP method but kindly refer to the literature [46]. In what follows we denote the SQP-based variant of SQUIC as SQUIC-sqp.

4 Analysis and Validation on Synthetic Data

In this section, we present experimental results on synthetic data for an extensive evaluation of the accuracy and efficacy of the proposed M-matrix estimation routines SQUIC-fit, as outlined in Section 3.2, and SQUIC-sqp as summarized in Section 3.3. We will compare our methods against the following state-of-the-art graph learning packages:[2]

1) Combinatorial Graph Laplacian (CGL) [20]: Graph Laplacian estimation via an iterative block-coordinate descent algorithm. The authors here decompose the original problem into a series of lower-dimensional subproblems. We use a cycle of 100 row/column updates for the minimization of the objective function.
2) Structured Graph Learning (SGL) [22]: The graph Laplacian is estimated by converting combinatorial structural constraints into spectral constraints, and the resulting optimization problem is solved with an algorithm based on quadratic methods. The parameter controlling the quadratic approximation term is set at $\beta = 20$, as suggested by the authors.

The following subsection 4.1 summarizes our experimental setup, and then 4.2 is devoted in a comparison of the accuracy of the methods under consideration. Then in subsection 4.3 we present timing comparisons between the methods for datasets of an increasing size. Last, in subsection 4.4, we shift our attention to the way incorporating prior knowledge of the graphical structure of a dataset influences the accuracy of the M-matrix retrieval.

2. The CGL code is available at: https://github.com/STAC-USC/Graph_Learning. The code for SGL is available as an R package at: https://cran.r-project.org/web/packages/spectralGraphTopology/index.html.

4.1 Experimental setup

Since both external algorithms directly estimate the combinatorial graph Laplacian in the set (2), we compare the accuracy of our proposed methods by estimating the M-matrix $\widehat{\boldsymbol{\Theta}}$ and then building the combinatorial graph Laplacian as

$$\widehat{\mathbf{L}}_{ij} = \begin{cases} \widehat{\boldsymbol{\Theta}}_{ij}, & \text{for all } i \neq j \\ -\sum_{r:r\neq i}^{p} \widehat{\boldsymbol{\Theta}}_{ir}, & \text{for all } i = j \end{cases} \tag{18}$$

The accuracy in the estimation of $\widehat{\mathbf{L}}$ is measured in terms of F-score and relative error (RE). The F-score is defined as

$$\text{F-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{19}$$

where precision is defined as precision $= TP/(TP + FP)$ and recall as recall $= TP/(TP + FN)$. TP stands for true positive entries, i.e. actual edges that are detected by the algorithm; FP corresponds to the false positives, i.e. edges that are falsely detected, and FN stands for the edges that the algorithm failed to detect. A score of $F = 1$ suggests that the matrix has been fully recovered, while smaller values of F suggest worse recovery success. The relative error is defined as

$$\text{RE} = \frac{\|\widehat{\mathbf{L}} - \mathbf{L}_{\text{true}}\|_F}{\|\mathbf{L}_{\text{true}}\|_F}, \tag{20}$$

where $\widehat{\mathbf{L}}$ is the estimated matrix and $\mathbf{L}_{\text{true}}$ the true reference graph Laplacian matrix.

We base our results on two synthetic datasets generated from Gaussian distributions with a mean of zero and the following types of predefined graphical structures:

- A grid graph structure denoted as $\mathcal{G}^{(p)}_{\text{grid}}$, where p is the number of nodes. Each node is connected to its four nearest neighbors (except the nodes at the boundaries).
- A random structured matrix denoted as $\mathcal{G}^{(p)}_{\text{clust}}$ representing a graphical structure of $p/100$ balanced clusters with an average node degree of 20 and with 90% of the edges contained within the clusters [49].

Edge weights are then randomly selected based on a uniform distribution from the interval $[0.1, 3]$. From these structures we generate an IGMRF model parametrized by the true graph Laplacian $\mathbf{L}_{\text{true}}$. From this IGMRF model n samples are drawn from the degenerate zero-mean multivariate Gaussian distribution $\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{L}^{\dagger}_{\text{true}}\right)$, where $\mathbf{L}^{\dagger}_{\text{true}}$ is the Moore-Penrose pseudoinverse of $\mathbf{L}_{\text{true}}$. The sample covariance matrix $\boldsymbol{\Sigma}$ is computed as

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}_i})\,(\bar{\mathbf{x}_i} - \mathbf{x}_i)^T, \quad \text{with } \bar{\mathbf{x}_i} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i. \tag{21}$$

We follow the approach of [50] and define the regularization parameter as

$$\lambda = c \cdot \sqrt{\log(p)/n}, \tag{22}$$

where the scaling term $\sqrt{\log(p)/n}$ adjusts the regularization according to p and n, and $c \in \mathbb{R}$ is based on experimental results.

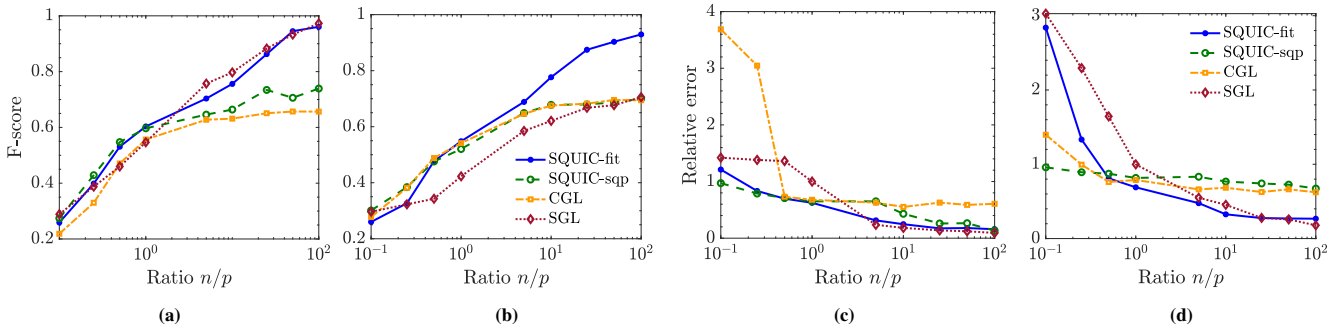


Fig. 2: Accuracy comparisons between the different combinatorial graph Laplacian estimation methods measured in terms of F-score (19) and relative error (20). a) F-scores for the lattice grid graph $\mathcal{G}_{\text{grid}}^{(64)}$. b) F-scores for the random clusters graph $\mathcal{G}_{\text{clust}}^{(60)}$. c) RE for the lattice grid graph $\mathcal{G}_{\text{grid}}^{(64)}$. d) RE for the random clusters graph $\mathcal{G}_{\text{clust}}^{(60)}$.

The convergence tolerance for all the methods is set to $\tau = 10^{-4}$, the threshold parameter for SQUIC-fit to $\kappa = 0$, and all the results reported hereby correspond to their mean value after 10 runs.

4.2 Accuracy estimation

Our first round of numerical experiments is designed to evaluate and compare the accuracy of the two proposed algorithms in the retrieval of the synthetic combinatorial graph Laplacian matrices emerging from the graphical structure of $\mathcal{G}_{\text{grid}}^{(64)}$ and $\mathcal{G}_{\text{clust}}^{(60)}$.

We generate 10 instances of each synthetic graph, and present in Figure 2 the mean accuracy results in terms of F-score (19) and the associated RE (20). The performance of the algorithms is compared for different ratios of sample sizes $n/p = \{0.1, 0.25, 0.5, 1, 5, 10, 25, 50, 100\}$. The parameter c in (22) is selected independently at each level for each method, and corresponds to the one that maximizes the F-score. Additionally, for SQUIC-fit we set in (12) $\eta = \lambda/10$.

For the lattice grid graphs (Figures 2a, 2c) SQUIC-fit achieves very high F-scores and low RE for higher sampling ratios $n/p > 1$, while still remaining competitive in the low sampling regimes $n/p \leq 1$ in terms of F-score. The accuracy of our post processing approach is similar to that of SGL both in terms of F-score and RE. For low sampling ratios $n/p \leq 1$ SQUIC-sqp reports the best F-scores and RE, as exploiting M-matrix constraints satisfies the model assumptions of attractive GMRFs. For the random clusters graphs (Figures 2b, 2d) SQUIC-fit achieves the highest F-scores for sampling ratios $n/p > 1$, with the RE reported being comparable with that of SGL. Our constrained approach SQUIC-sqp reports here similar F-score and RE with CGL for all sampling regimes.

4.3 Timing comparisons

We proceed with a comparison of the runtimes of the methods under question when learning M-matrices. To this end, we consider a sequence of 6 true combinatorial graph Laplacian matrices $\mathbf{L_{true}}$ of increasing size. In particular, for the lattice grid graph $\mathcal{G}_{\text{grid}}^{(p)}$ we consider graphs of dimension $p = \{16, 64, 256, 1024, 4096, 16384\}$ and for the random clusters graph $\mathcal{G}_{\text{clust}}^{(p)}$ of $p = \{100, 200, 1000, 2000, 5000, 10000\}$. The number of samples is fixed in both cases at $n = 500$ and the parameter c in (22) is set for each method such that the best solution in terms of F-score is reported at each p level. We report these timing results in Figure 3.

The timing results for CGL and SGL are excluded if the runtimes exceed 10^4 seconds. For the lattice grid graph (Figure 3a) SGL exceeds this time limit for $p \geq 4096$ and CGL for $p = 16384$. For the random clusters graph (Figure 3b) the time limit is exceeded by SGL at $p \geq 5000$ and by CGL at $p = 10^4$. In Figure 3 we additionally observe that SQUIC-fit outperforms all competing algorithms across all dimensions for both graphical structures. This is an expected behaviour, as SQUIC-fit is the only unconstrained method included in the comparisons. SQUIC-sqp is outperformed by CGL only in the lattice grid experiments for the low dimensional cases $p \leq 64$. In all experiments both SQUIC variants are up to 3 orders of magnitude faster that the competing methods for $p \geq 256$.

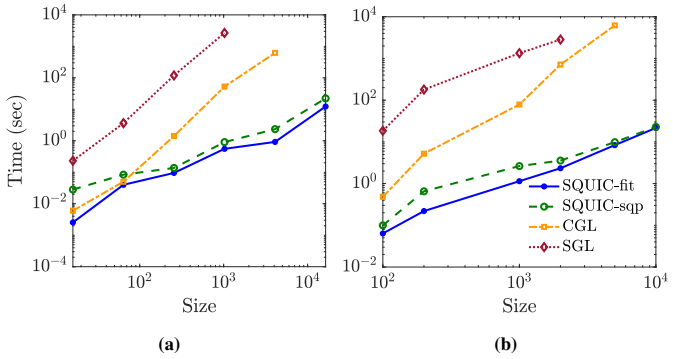


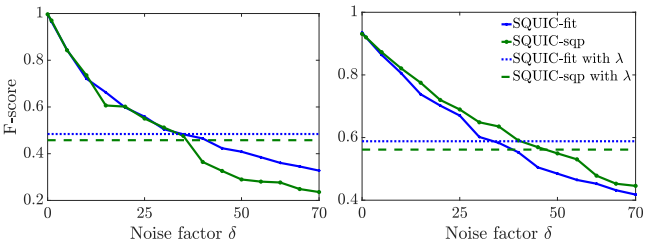
Fig. 3: Timing comparisons between the different graph Laplacian estimation methods when learning $\widehat{\mathbf{\Theta}}$ from synthetic graphs with an increasing number of p. a) Results for the lattice grid graph $\mathcal{G}_{\text{grid}}^{(p)}$ with $p \in \{16, \ldots, 16384\}$. b) Results for the random clusters graph $\mathcal{G}_{\text{clust}}^{(p)}$ with $p \in \{100, \ldots, 10000\}$.

4.4 Incorporating graphical bias

In this unit test we study the recovery accuracy of the two introduced SQUIC algorithms when using prior graphical knowledge in the estimation of the sparse M-matrix $\widehat{\mathbf{\Theta}}$. The graphical structure of the bias $\mathbf{G}$ is defined as a corrupted version of the structure of the true graph Laplacian matrix $\mathbf{L}_{\text{true}}$. We control the degree of this corruption with a random symmetric sparse matrix $\mathbf{Z} \in \mathbb{R}^{p \times p}$ with $\delta \cdot |\mathbf{L}_{\text{true}}|/p$ number of nonzeros per row. The structure of $\mathbf{G}$ is then defined as

$$\mathbf{G}_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \mathbf{I}\left(\mathbf{L}_{ij}^{\text{true}} > 0\right) + \mathbf{I}\left(\mathbf{Z}_{ij} > 0\right), & \text{if } i \neq j. \end{cases} \tag{23}$$

Notice that for $\delta = 0$ we retrieve the exact structure of $\mathbf{L}_{\text{true}}$, while for an increasing $\delta > 0$ the structure of $\mathbf{G}$ has an increasing number of noisy entries. Then the matrix tuning parameter is composed in similar fashion to (12) as



(a) (b)

Fig. 4: Studying the effect that incorporating the structure of $\mathbf{G}$ in the matrix tuning parameter $\mathbf{\Lambda}$ has on the retrieval accuracy of $\widehat{\mathbf{L}}$. a) Results for the lattice grid graph $\mathcal{G}_{\text{grid}}^{(1024)}$. b) Results for the random clusters graph $\mathcal{G}_{\text{clust}}^{(1000)}$. We use $n = 500$ samples in both cases.

$$\mathbf{\Lambda}_{ij} = \begin{cases} \lambda_{\text{opt}}/b & \text{for} \quad \mathbf{G}_{ij} \neq 0, \\ b \cdot \lambda_{\text{opt}} & \text{for} \quad \mathbf{G}_{ij} = 0, \end{cases} \tag{24}$$

with λ_{opt} being the scalar regularization parameter resulting in the highest F-score, and $b \in \mathbb{R}$ a scalar parameter controlling the effect of the matrix bias on the regularization. Larger values of b in (24) result in the matrix bias $\mathbf{G}$ being more strictly enforced. We select $b = 2$ for a moderate influence of $\mathbf{G}$ on the estimated $\widehat{\mathbf{L}}$.

We consider two true graph Laplacian matrices emerging from the graphical structure of $\mathcal{G}_{\text{grid}}^{1024}$ and $\mathcal{G}_{\text{clust}}^{1000}$ with $n = 500$ number of samples. In Figure 4 we present the effect that an increasing noise factor $\delta = \{0, 1, \ldots, 70\}$ has on the retrieval accuracy of both SQUIC-fit and SQUIC-sqp in terms of F-score, and compare it with the retrieval accuracy achieved by the algorithms when using a scalar regularization parameter λ with no graphical bias. Note that the corruption matrix $\mathbf{Z}$ has no effect on the retrieval accuracy when using a scalar regularization parameter, as no graphical bias is utilized in the composition of the matrix penalty term $\mathbf{\Lambda}$.

The best F-scores achieved at the optimal scalar λ_{opt} are represented with the horizontal dashed lines. The performance of the SQUIC algorithms when taking into account a noisy graphical bias $\mathbf{G}$ in the matrix sparsity parameter $\mathbf{\Lambda}$ (solid lines) greatly outperforms the scalar counterparts. In particular, for the lattice grid graph $\mathcal{G}_{\text{grid}}^{(1024)}$ (Figure 4a) utilizing the graphical structure with SQUIC-fit improves the achieved F-score of 0.49 for noise factors of $\delta \leq 35$. For SQUIC-sqp improvements over the baseline of $\text{F} - \text{score} = 0.46$ are observed for $\delta \leq 40$. For the random clusters graph $\mathcal{G}_{\text{clust}}^{(1000)}$ (Figure 4b) the baseline of SQUIC-fit is $\text{F} - \text{score} = 0.59$, and is improved for $\delta \leq 30$, while for SQUIC-sqp the best F-score of 0.56 is improved when considering a graphical bias with $\delta \leq 45$.

5 Experiments with Real-World Data

In this section, we illustrate the applicability and efficiency of SQUIC-sqp and SQUIC-fit in the estimation of sparse M-matrices emerging from real-world problems. In subsection 5.1 we identify the largest connected components of a graph emerging from the COVID-19 daily cases in the USA, and perform spectral clustering with the M-Matrix of the largest component. Subsequently, in subsection 5.2 we classify image datasets of up to $p = 7 \cdot 10^4$ dimensions based on the eigenvectors of the estimated M-matrices.

5.1 Clustering of COVID-19 daily cases

We consider the publicly available [3] data for the US confirmed daily cases, reported at the county level [51]. We emphasize that the case study presented here is intended to highlight the capabilities of the proposed algorithms and not propose any course of COVID-19 related actions.

The dataset under consideration consists of $p = 3342$ counties and reports the number of daily COVID-19 cases C for $n = 671$ days for the time window 22 January 2020 to 23 November 2021. Counties with a total number of cases $\sum_n C < 100$ are discarded, resulting in $p = 3209$, and these cases are normalized by the number of residents per county [4] in order to obtain information on the infection rate per capita.

Subsequently, the M-matrix $\widehat{\mathbf{\Theta}}$ of the positively correlated counties is constructed with SQUIC-fit in 72 seconds and in 211 seconds with SQUIC-sqp, and the largest connected components of the resulting graphical structure are identified. For the SQUIC-sqp variant we use a scalar regularization parameter of $\lambda = 0.7$, and for the SQUIC-fit algorithm we set in (9) $\lambda = 0.7$ and in (12) $\eta = 2\lambda/3$. The matrices retrieved from both algorithms are almost identical, thus in what follows we report the results obtained with SQUIC-fit.

We illustrate in Figure 5 the six largest connected components of $\widehat{\mathbf{\Theta}}$. The largest component (Figure 5a) includes 1774 counties from the entire USA, the second one (Figure 5b) captures 165 counties from the states of Oklahoma and Iowa, the third one (Figure 5c) 113 counties from Missouri, the fourth one (Figure 5d) 81 counties from Michigan, the fifth one (Figure 5e) 79 counties from Nebraska, and the sixth largest connected component of $\widehat{\mathbf{\Theta}}$ (Figure 5f) includes 66 counties from the state of Florida. The clear geographic partition of the components $2 - 6$ demonstrates that SQUIC-fit successfully captures the positively correlated variables of the dataset.

We proceed with an analysis of the clusters present in the largest connected component of $\widehat{\mathbf{\Theta}}$. This component is denoted as $\widehat{\mathbf{\Theta}}_{\text{a}}$, and is used to build the random-walk normalized graph Laplacian $\widehat{\mathbf{L}}_{\text{rw}} = \mathbf{T}^{-1}\widehat{\mathbf{L}}$, where $\widehat{\mathbf{L}}$ is defined as in (18) and $\mathbf{T}$ is the diagonal degree matrix satisfying $\mathbf{T}_{ii} = \widehat{\mathbf{L}}_{ii}$ for all i. After computing the eigenvalues λ_k of $\widehat{\mathbf{L}}_{\text{rw}}$ the number of natural clusters present in the dataset is estimated with the relative eigengap

$$\gamma_k = \frac{\lambda_{k+1} - \lambda_k}{\lambda_k}, \quad k \geqslant 2. \tag{25}$$

A high value of γ_k indicates that $\widehat{\mathbf{\Theta}}_{\text{a}}$ admits a natural decomposition into at least k clusters [52]. In order to obtain discrete partitions, the eigenvectors corresponding to the k smallest eigenvalues of $\widehat{\mathbf{L}}_{\text{rw}}$ are clustered with the k-means algorithm with 20 orthogonal and 10 random initializations [53].

We present the clustering results using $\widehat{\mathbf{\Theta}}_{\text{a}}$ in Figure 6. According to the relative eigengap, 8 distinct clusters are present in the subgraph. The locations of the counties present at each cluster are illustrated in Figure 6a, and the cardinality of the respective clusters is presented in Figure 6b. The largest cluster (black) captures 734 counties located mostly in the south and mideast

3. The COVID-19 Data Repository is provided by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University at https://github.com/CSSEGISandData/COVID-19. Puerto Rico municipalities are included.

4. Demographic information of the USA at the county level is available at https://www.census.gov/data/datasets/time-series/demo/popest/2010s-counties-total.html.

Fig. 5: Visualizing the US counties corresponding to the six largest connected components of the estimated $\widehat{\mathbf{\Theta}}$ with $p = 3209$ dimensions. The number of COVID-19 daily instances is represented by the $n = 671$ available samples. The six components are illustrated in a – f in descending order according to their size.

states of Georgia, South and North Carolina, Virgina, Tenessee, Kentucky, and the northwest states of Washington and Oregon and Alaska. The second largest cluster (orange) includes 279 counties in the northeast states of West Virginia, Pennsylvania, New York, Maine, Delaware, the District of Columbia, the southwest state of California and Hawaii. The third largest cluster (cyan) has 214 counties mostly located in the neighboring states of North and South Dakota, Minnesota and Wisconsin. The fourth cluster in size (dark blue) is comprised of 211 nodes in the states of Massachusetts, Ohio and Indiana. The fifth cluster (light blue) includes 169 nodes mostly located in Illinois, Utah, Colorado and New Mexico. The sixth (green) captures 101 counties of Arkansas and Alabama, the seventh (red) 56 counties of Louisiana and finally the eighth (purple) 10 counties of New Hampshire. The clear geographical patterns present in the clusters indicate that the M-matrix $\widehat{\mathbf{\Theta}}_{\mathrm{a}}$ estimated by SQUIC-fit accurately captures the latent graphical structure of the dataset, and that the resulting eigenvectors of $\widehat{\mathbf{L}}_{\mathrm{rw}}$ are well suited for spectral clustering tasks.

5.2 Image classification

In this case study we demonstrate the applicability of the introduced algorithms in the estimation of M-matrices emerging from image applications. We study the problem of classifying facial images and handwritten characters according to their labels by applying a spectral clustering routine on the eigenvectors of the estimated random walk Laplacian $\widehat{\mathbf{L}}_{\mathrm{rw}}$. Classification accuracy is measured in terms of the unsupervised clustering accuracy (ACC $\in [0, 1]$), and the normalized mutual information (NMI $\in [0, 1]$) [54]. For both classification metrics a value of 1 suggests a perfect grouping of the nodes according to the true labels. We consider the following publicly available datasets

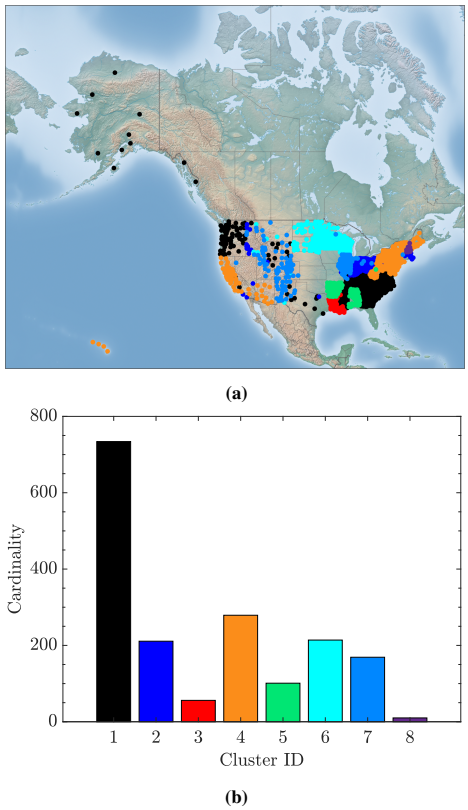


Fig. 6: Spectral clustering of the largest connected component of $\widehat{\mathbf{\Theta}}$ a) Geographical locations of the nodes belonging to each cluster. b) Cardinality of each cluster. (Best viewed in color.)

- YaleA [55]: A collection of $p = 165$ grayscale images of 15 individuals at resolution $n = 64 \times 64$ pixels. There are 11 images per subject, one per different facial expression.
- Olivetti [56]: A set of 10 different facial images of 40 distinct subjects, resulting $p = 400$ instances at resolution 64×64 pixels, taken at different times, varying lighting, facial expressions and facial details.
- USPS [57]: A balanced set of $p = 11000$ images of 10 distinct handwritten digits with $n = 16 \times 16$ pixels.
- KMNIST [58]: The entire Kuzushiji-MNIST balanced dataset with $p = 70000$ images of 10 modern Japanese hiragana characters at resolution $n = 28 \times 28$.

The high dimensionality p of these datasets renders them computationally unfavorable for the CGL [**?**] and SGL [22] methods, thus here we compare our proposed algorithms against the traditional approach of building adjacency matrices $\mathbf{A}$. This approach consists of initially creating the connectivity matrix $\mathbf{G} \in \mathbb{R}^{p \times p}$ from a k-nearest neighbors routine, with the number of nearest neighbors (NN) set such that the resulting graph is connected. For these datasets the number of nearest neighbors needed for a connected graph is $\mathrm{NN} = 12$ for YaleA and Olivetti and $\mathrm{NN} = 11$ for both USPS and KMNIST. Subsequently, the similarity matrix $\mathbf{H} \in \mathbb{R}^{p \times p}$ between the data points is defined similarly to [59] as $\mathbf{H}_{ij} = \max\{\mathbf{H}_i(j), \mathbf{H}_j(i)\}$ with $\mathbf{H}_i(j) = \exp\left(-4\frac{\|y_i - y_j\|^2}{\sigma_i^2}\right)$, with σ_i standing for the Euclidean distance between the i-th data point and its k-th nearest neighbor. The adjacency matrix $\mathbf{A}$ is then created as

$$\mathbf{A} = \mathbf{G} \odot \mathbf{H}. \tag{26}$$

We utilize the kNN connectivity matrix $\mathbf{G}$ as graphical bias for SQUIC-fit and SQUIC-sqp and find the optimal scalar tuning parameter $\lambda = \lambda_{\text{opt}}$ for each case. The matrix tuning parameter $\mathbf{\Lambda}$ in (12) is then set with $\eta = \lambda_{\text{opt}}/\sqrt{p}$ for both SQUIC-fit and SQUIC-sqp. Our strategy is thus penalizing the graphical bias $\mathbf{G}$ with a decreasing rate for an increasing number of dimensions p. The goal is to obtain within a reasonable amount of time graphical representations of the datasets that are sparser than $\mathbf{G}$ and more accurate, and which will therefore lead to an increase in the classification accuracy metrics after applying a spectral clustering routine. Sparsity in the graph is measured in term of edge density, defined as $\epsilon = |E|/(|V| * (|V| - 1))$, which is a ratio reflecting how close the graph is to a complete graph, with $\epsilon = 1$ for a complete graph.

The M-matrices of the 4 datasets under consideration are retrieved in $t = 0.9, 6.7, 46.3$ and 1255.6 seconds with SQUIC-fit, and in $t = 2.4, 3.6, 31.5$ and 2024 seconds with SQUIC-sqp respectively. We summarize the rest of our results in Table 1. For each dataset we report the edge density, the ACC and the NMI achieved by the best method, and the percentage the remaining methods are inferior to that value. Inferiority in percentage values is defined as $I = 100 \cdot \gamma \cdot (e_{\text{ref}} - e_{\text{best}})/e_{\text{best}}$, where e_{best} is the best value, e_{ref} the value it is compared against, and $\gamma = -1$ for minimization scenarios (ϵ) and $\gamma = 1$ for maximization ones (ACC, NMI). Both SQUIC-fit and SQUIC-sqp improve the classification accuracy of the traditional kNN graph for all the datasets considered. In particular, SQUIC-fit achieves the highest accuracy metrics for all cases, and the lowest edge density for all cases except USPS. The reduction of the edge density is more evident for YaleA and Olivetti, as the tuning parameter $\eta = \lambda_{\text{opt}}/\sqrt{p}$ applied on the entries of the graphical bias $\mathbf{G}$ has a less impact for graphs of low dimensions p. In Figure 7 we illustrate this reduction in ϵ for the YaleA dataset. For visual clarity we select a subset (variables 100 to 155) of the image dataset YaleA, organized in five distinct classes, denoted by letters A to E, with each class composed by eleven variables. We order the variables in a circular layout and compare the graphical structure obtained by SQUIC-fit (398 gray edges in Figure 7a) and SQUIC-sqp (432 gray edges in Figure 7b). The red edges in both figures represent the edges that were removed from the graphical bias $\mathbf{G}$, estimated with a kNN routine, after applying SQUIC-fit (68 edges) and SQUIC-sqp (28 edges). Multiple edges that connect variables belonging to different classes are removed in both cases, thus reducing the interclass connectivity of the graph. The advantages of these sparser graphical structures, with edge weights assigned by solving the MLE problem, are verified by the increased classification scores of Table 1.

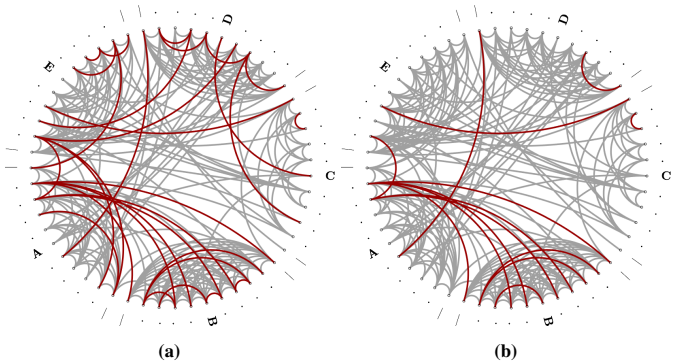


Fig. 7: Comparison of the graphical structure of the adjacency matrix $\mathbf{A}$ for a subset of the dataset YaleA. The coloring indicates the edges that were removed (in red) from the initial kNN graphical bias, and the edges (in gray) that remained after the application of the two proposed algorithms. (a) Graph estimated with SQUIC-fit with 398 remaining and 68 removed edges. (c) Graph estimated with SQUIC-sqp with 432 remaining and 28 removed edges. (Best viewed in color.)

6 Conclusions

In this work, motivated by the effectiveness of the SQUIC package in learning precision matrices of very large dimensions, we developed two algorithms for learning M-matrices that represent graphs whose nodes are non-negatively correlated random variables. Both algorithms are based on the ℓ_1-regularized minimization of the MLE problem, and are able to incorporate available information about the latent graphical structure of the data under question in the form of a matrix regularization parameter. The first one, SQUIC-fit, is an unconstrained approach that performs two consecutive precision matrix estimations, and utilizes the positively correlated variables identified in the first run as graphical bias for the retrieval of the second precision matrix. Subsequent post-processing on its entries guarantees that the resulting matrix is positive definite and an M-matrix. The second one, SQUIC-sqp, is a constrained method that enforces the M-matrix structure during the MLE optimization procedure. The constrained minimization is achieved by means of a sequential quadratic programming algorithm, with the corresponding KKT system being solved with a preconditioned conjugate gradient method.

Our methods are compared against various state-of-the-art methods in a series of synthetic tests, showcasing that the introduced algorithms offers significant gains in terms of time-to-solution, while accurately retrieving the underlying M-matrix structure. In

Method	YaleA density	YaleA ACC	YaleA NMI	Olivetti density	Olivetti ACC	Olivetti NMI	USPS density	USPS ACC	USPS NMI	KMNIST density	KMNIST ACC	KMNIST NMI
kNN	-16.81%	-5.13%	-7.26%	-9.39%	-5.57%	-5.29%	-0.28%	-12.63%	-7.73%	-0.24%	-15.72%	-11.54%
SQUIC-fit	**0.083**	**0.613**	**0.650**	**0.04**	**0.646**	**0.7852**	-0.02%	**0.652**	**0.683**	**0.0003**	**0.61**	**0.587**
SQUIC-sqp	-8.32%	-3.06%	-4.62%	-8.16%	-1.87%	-1.81%	**0.002**	-1.89%	-1.85%	-0.05%	-0.23%	-0.37%

TABLE 1. Classification results for the image datasets of subsection 5.2.

particular, for artificial graphs emerging from a grid and a randomly clustered structure the two introduced algorithms attain equivalent retrieval accuracy scores, and are up to 3 orders of magnitude faster for graph dimensions $p \in [256, 10^4]$. Additionally, we see that for these synthetic cases incorporating in the matrix regularization parameter available information regarding the latent graphical structure of the data greatly improves the retrieval accuracy of both SQUIC-fit and SQUIC-sqp.

Furthermore, we provide two case studies that demonstrate the applicability of the introduced algorithms in real-world scenarios. In the first one we identify the largest connected components of the M-matrix emerging from daily 671 COVID-19 cases for 3209 US counties, and observe that these components correspond to clear geographical patterns. Subsequently, we perform spectral clustering on the largest connected component and report that the resulting clusters are also revealing distinct geographic partitions. For the second case study we classify with image datasets with up to $p = 7 \cdot 10^4$ variables based on the eigenvectors of the estimated M-matrices, and report increases in the classification accuracy over the traditional approach of building adjacency matrices for spectral methods.

The consistency of our results, from the artificial tests to the real-world cases, highlights the effectiveness of the introduced graph learning algorithms and the broad applicability of the presented work.

ACKNOWLEDGMENT

We would like to acknowledge the financial support from the Swiss National Science Foundation (SNSF) under the project 182673 entitled "Balanced Graph Partition Refinement Using the Graph p-Laplacian".

REFERENCES

[1] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," IEEE Transactions on Artificial Intelligence, vol. 2, no. 2, pp. 109–127, 2021.

[2] L. Qiao, L. Zhang, S. Chen, and D. Shen, "Data-driven graph construction and graph learning: A review," Neurocomputing, vol. 312, pp. 336–351, 2018.

[3] L. Stanković, D. Mandic, M. Daković, M. Brajović, B. Scalzo, S. Li, and A. G. Constantinides, "Data analytics on graphs part iii: Machine learning on graphs, from graph topology to applications," Foundations and Trends® in Machine Learning, vol. 13, no. 4, pp. 332–530, 2020.

[4] S. Lauritzen, Graphical models, ser. Oxford Statistical Science Series. Clarendon Press, 1996, no. 17.

[5] A. P. Dempster, "Covariance Selection," Biometrics, vol. 28, no. 1, p. 157, mar 1972.

[6] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," Biostatistics, vol. 9, no. 3, pp. 432–441, 12 2007.

[7] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. K. Ravikumar, "Sparse Inverse Covariance Matrix Estimation Using Quadratic Approximation," in Advances in Neural Information Processing Systems 24, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2330–2338.

[8] M. Bollhöfer, A. Eftekhari, S. Scheidegger, and O. Schenk, "Large-scale Sparse Inverse Covariance Matrix Estimation," SIAM Journal on Scientific Computing, vol. 41, no. 1, pp. A380–A401, 2019.

[9] A. Eftekhari, M. Bollhöfer, and O. Schenk, "Distributed memory sparse inverse covariance matrix estimation on high-performance computing architectures," in Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis, ser. SC '18. IEEE Press, 2018.

[10] A. Eftekhari, D. Pasadakis, M. Bollhöfer, S. Scheidegger, and O. Schenk, "Block-enhanced precision matrix estimation for large-scale datasets," Journal of Computational Science, vol. 53, 2021.

[11] C. Wang and B. Jiang, "An efficient ADMM algorithm for high dimensional precision matrix estimation via penalized quadratic loss," Computational Statistics & Data Analysis, vol. 142, no. C, 2020.

[12] R. Zhang, S. Fattahi, and S. Sojoudi, "Large-scale sparse inverse covariance estimation via thresholding and max-det matrix completion," in Proceedings of the 35th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 5766–5775.

[13] H. Pang, H. Liu, and R. Vanderbei, "The fastclime package for linear programming and large-scale precision matrix estimation in R," Journal of Machine Learning Research, vol. 15, no. 14, pp. 489–493, 2014.

[14] E. Bølviken, "Probability inequalities for the multivariate normal with non-negative partial correlations," Scandinavian Journal of Statistics, vol. 9, no. 1, pp. 49–58, 1982.

[15] S. Karlin and Y. Rinott, "M-matrices as covariance matrices of multi-normal distributions," Linear Algebra and its Applications, pp. 419–438, 1983.

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

[17] Y. Wang, U. Roy, and C. Uhler, "Learning high-dimensional Gaussian graphical models under total positivity without adjustment of tuning parameters," in The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy], ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 2020, pp. 2698–2708.

[18] J. K. Tugnait, "Sparse graph learning under Laplacian-related constraints," IEEE Access, vol. 9, pp. 151 067–151 079, 2021.

[19] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graphs," in Proceedings of the 33rd Annual Cognitive Science Conference, 2010.

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

[21] ——, "Graph learning with Laplacian constraints: Modeling attractive gaussian markov random fields," in 2016 50th Asilomar Conference on Signals, Systems and Computers, 2016, pp. 1470–1474.

[22] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," Journal of Machine Learning Research, vol. 21, no. 22, pp. 1–60, 2020.

[23] J. Ying, J. V. de Miranda Cardoso, and D. Palomar, "Nonconvex sparse graph learning under Laplacian constrained graphical model," in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 7101–7113.

[24] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Laplacian matrix learning for smooth graph signal representation," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 3736–3740.

[25] V. Kalofolias, "How to learn a graph from smooth signals," in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 920–929.

[26] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in 7th International Conference on Learning Representations, ICLR, New Orleans, LA, USA, 6–9 May 2019.

[27] J. Pang and G. Cheung, "Graph Laplacian regularization for image denoising: Analysis in the continuous domain," IEEE Transactions on Image Processing, vol. 26, no. 4, pp. 1770–1785, 2017.

[28] U. Luxburg, "A tutorial on spectral clustering," Statistics and Computing, vol. 17, no. 4, p. 395–416, Dec. 2007.

[29] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, ser. NIPS'01. Cambridge, MA, USA: MIT Press, 2001, p. 849–856.

[30] M. Fiedler, "Algebraic connectivity of graphs," Czechoslovak Mathematical Journal, vol. 23, no. 2, pp. 298–305, 1973.

[31] A. Pothen, H. D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," SIAM J. Matrix Anal. Appl., vol. 11, no. 3, pp. 430–452, May 1990.

[32] I. S. Dhillon and J. A. Tropp, "Matrix nearness problems with bregman divergences," SIAM Journal of Matrix Analysis and Applications (SIMAX), vol. 29, no. 0, nov 2007.

[33] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "BIG & QUIC: Sparse Inverse Covariance Estimation for a Million Variables," in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3165–3173.

[34] T. Cai, W. Liu, and X. Luo, "A constrained l_1 minimization approach to sparse precision matrix estimation," Journal of the American Statistical Association, vol. 106, no. 494, pp. 594–607, 2011.

[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Found. Trends Mach. Learn., vol. 3, no. 1, p. 1–122, Jan. 2011.

[36] W. Liu and X. Luo, "Fast and adaptive sparse precision matrix estimation in high dimensions," Journal of Multivariate Analysis, vol. 135, pp. 153 – 162, 2015.

[37] A. Anandkumar, V. Y. F. Tan, and A. S. Willsky, "High-dimensional graphical model selection: Tractable graph families and necessary conditions," in Proceedings of the 24th International Conference on Neural Information Processing Systems, ser. NIPS'11. Red Hook, NY, USA: Curran Associates Inc., 2011, p. 1863–1871.

[38] H. Rue and L. Held, Gaussian Markov Random Fields: Theory And Applications (Monographs on Statistics and Applied Probability). Chapman & Hall/CRC, 2005.

[39] F. R. K. Chung, Spectral Graph Theory. American Mathematical Society, 1997, vol. 92.

[40] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.

[41] T. Bühler and M. Hein, "Spectral clustering based on the graph p-Laplacian," in Proceedings of the 26th Annual International Conference on Machine Learning, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 81–88.

[42] F. Tudisco and M. Hein, "A nodal domain theorem and a higher-order cheeger inequality for the graph p-Laplacian," Journal of Spectral Theory, Mar. 2017.

[43] D. Pasadakis, C. L. Alappat, O. Schenk, and G. Wellein, "Multiway p-spectral graph cuts on Grassmann manifolds," Machine Learning, nov 2021.

[44] Y. Chen, T. A. Davis, W. W. Hager, S. Rajamanickam, and W. W. Hager, "Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate," ACM Trans. Math. Softw, vol. 35, no. 14, 2008.

[45] T. A. Davis, S. Rajamanickam, and W. M. Sid-Lakhdar, "A survey of direct methods for sparse linear systems," Acta Numerica, vol. 25, p. 383–566, 2016.

[46] J. Nocedal and S. Wright, Numerical Optimization, ser. Operations Research and Financial Engineering. Springer, 2006.

[47] F. Oztoprak, J. Nocedal, S. Rennie, and P. A. Olsen, "Newton-like methods for sparse inverse covariance estimation," Advances in Neural Information Processing Systems, vol. 25, pp. 755–763, 2012.

[48] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed. SIAM Publications, 2003.

[49] J. Ballani and D. Kressner, "Sparse inverse covariance estimation with hierarchical matrices," EPFL Technical Report, Tech. Rep., 2014.

[50] S. Zhou, P. Rütimann, M. Xu, and P. Bühlmann, "High-dimensional covariance estimation based on Gaussian graphical models," Journal of Machine Learning Research, vol. 12, no. 91, pp. 2975–3026, 2011.

[51] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track COVID-19 in real time," The Lancet. Infectious Diseases, vol. 20, no. 5, pp. 533–534, May 2020.

[52] R. J. Sánchez-García, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, and J. W. Bialek, "Hierarchical spectral clustering of power grids," IEEE Transactions on Power Systems, vol. 29, no. 5, pp. 2229–2237, 2014.

[53] D. Verma and M. Meila, "A comparison of spectral clustering algorithms," Department of CSE University of Washington Seattle, WA98195-2350, Tech. Rep., 2005.

[54] H. Dalianis, Evaluation Metrics and Evaluation. Cham: Springer International Publishing, 2018, pp. 45–53.

[55] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711–720, 1997.

[56] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, 1994, pp. 138–142.

[57] J. J. Hull, "A database for handwritten text recognition research," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 5, pp. 550–554, 1994.

[58] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep learning for classical japanese literature," CoRR, vol. abs/1812.01718, 2018.

[59] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in Advances in Neural Information Processing Systems 17, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005, pp. 1601–1608.

Dimosthenis Pasadakis is a Ph.D. candidate at the Faculty of Informatics at Università della Svizzera italiana. He graduated in Physics from the Aristotle University of Thessaloniki, and earned a Msc degree in Computational Science from Università della Svizzera italiana. His research is centered around algorithms for graph learning and combinatorial optimization for graph partitioning and clustering.

Matthias Bollhöfer is professor for Numerical Analysis at TU Braunschweig. His research area covers several aspects at the interface of Numerical Analysis and applications in Computational Science and Engineering. His contributions include numerical methods for partial differential equations, scientific parallel computing, sparse numerical linear algebra, numerical methods for data science applications as well as numerical methods for model order reduction. He has (co-)authored several sparse linear algebra software packages that are frequently used.

Olaf Schenk is a professor for computing at the Faculty of Informatics at Universita della Svizzera italiana. He graduated in Applied Mathematics from Karlsruhe Institute of Technology (KIT), Germany, and earned his PhD from ETH Zurich. Olaf Schenk is an elected Fellow of the Society of Industrial and Applied Mathematics (SIAM), and a senior member of IEEE and ACM. His research interests are centered around the topic of multicore and manycore algorithms for computational science simulations on emerging high-performance computing (HPC) architectures.