# ESTIMATING NORMALIZED GRAPH LAPLACIANS IN FINANCIAL MARKETS

*José Vinícius de M. Cardoso*[1], *Jiaxi Ying*[1], *Sandeep Kumar*[2], *Daniel P. Palomar*[1]

The Hong Kong University of Science and Technology[1],
Indian Institute of Technology Delhi[2],
Clear Water Bay, Hong Kong SAR, China,
New Delhi, Delhi, India

## ABSTRACT

Gaussian Markov random fields, a class of graphical models, play an increasingly important role in real-world problems, where they are often applied to uncover conditional correlations between pairs of entities in a network. Motivated by recent applications of graphs in financial markets, we investigate the problem of learning undirected, weighted, normalized, graphical models. More precisely, we design an optimization algorithm to learn precision matrices that are modeled as normalized graph Laplacians. The proposed algorithm takes advantages of frameworks such as the alternating direction method of multipliers and projected gradient descent, which allows us to decompose the original problem into subproblems that can be solved efficiently. We demonstrate the empirical performance of the proposed algorithm, in comparison to state-of-the-art benchmark models, in a number of datasets involving financial time-series.

***Index Terms***— graphical models, normalized Laplacian, financial markets, time series, estimation theory

## 1. INTRODUCTION

Graph learning from data has been a problem of critical importance for the statistical graph learning and graph signal processing fields [1, 2], with direct impact on applied areas such as unsupervised learning, clustering [3, 4, 5], applied finance [6], network topology inference [7], and community detection [8]. In addition, graph matrices play a fundamental role for graph neural networks [8].

In financial markets, graphical models were initially investigated in [9] that concluded that learning topological arrangements in a stock market context reveals economic factors that may affect price data evolution. Other benefits that graphical models bring to applications in financial markets have been extensively discussed in the literature. A representative, yet not exhaustive, list of examples include: (*i*) identifying "business as usual" and "crash" periods via asset tree graphs [10], (*ii*) constructing networks of companies based on graphs [10], (*iii*) leveraging properties of graphs into follow-up tasks such as hierarchical portfolio designs [11], (*iv*) exploring graph properties such as degree and eigenvector centralities for market crash detection and portfolio construction [12], and (*v*) community detection in financial stock markets [13].

The basic idea behind graph learning is to answer the following question: given a data matrix whose columns represent observations measured at the graph nodes, how can one design a graph that best fits such data matrix without possibly any (or with at most partial) knowledge of the underlying graph structure?

While estimators for connected graphs have been well studied [2, 14], some of their properties, such as sparsity, are yet being investigated [15]. Recently, [3, 5, 4, 16] proposed optimization programs for learning graphs with structure such as bipartite and $k$-component graphs, as this classes of graph are appealing models for clustering tasks due to the spectral properties of the Laplacian and adjacency matrices.

In finance, prior information about stocks is available via sector classification systems such as the Global Industry Classification Standard (GICS) [17] or the Industry Classification Benchmark (ICB) [18]. However, stocks may have impact on multiple industries, *e.g.*, the evident case of technology companies, whose influence on prices affect stocks not only in their own sector, but spans across multiple sectors.

Motivated by challenging applied finance tasks such as estimation of conditional correlations, we investigate the problem of learning undirected, weighted, normalized graphs from a Gaussian Markov random field (GMRF) framework [15]. In addition, the wide success of graphical models in applied areas can be arguably attributed to the development of fast, scalable optimization algorithms pioneered by [1, 19]. From that perspective, the main contributions of this paper are as follows:

1. We propose a novel formulation to learn normalized Laplacian matrices from a GMRF perspective.

2. We design an optimization algorithm for such formula-

This work was supported by the Hong Kong GRF 16207820 research grant. Sandeep's work is supported by the DST Inspire faculty grant MI02322G.

tion based on the alternating direction method of multipliers (ADMM) [20] and projected gradient descent (PGD) [21].

3. We present empirical results that reveal the outperformance of the proposed estimation scheme when compared to state-of-the-art methods.

## 2. BACKGROUND

We denote an undirected, weighted graph as a triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where $\mathcal{V} = \{1, 2, \ldots, p\}$ is the vertex (or node) set, $\mathcal{E} \subseteq \{\{u, v\} : u, v \in \mathcal{V}, u \neq v\}$ is the edge set, that is, a subset of the set of all possible unordered pairs of $p$ nodes such that $\{u, v\} \in \mathcal{E}$ iff nodes $u$ and $v$ are connected. $\boldsymbol{W} \in \mathbb{R}_{+}^{p\times p}$ is the symmetric weighted adjacency matrix that satisfies $W_{ii} = 0, W_{ij} > 0$ iff $\{i, j\} \in \mathcal{E}$ and $W_{ij} = 0$, otherwise. The combinatorial, *unnormalized* graph Laplacian matrix $\boldsymbol{L}$ is defined, as usual, as $\boldsymbol{L} \triangleq \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D} \triangleq \mathsf{Diag}(\boldsymbol{W}\mathbf{1})$ is the degree matrix.

The data generating process is assumed to be a zero-mean, GMRF $\boldsymbol{x} \in \mathbb{R}^{p}$, such that $x_i$ is the random variable generating a signal measured at node $i$, whose rank-deficient precision matrix is modeled as a graph Laplacian matrix. This model is also known as Laplacian-constrained Gaussian Markov Random Field (LGMRF) [15]. Assume we are given $n$ observations of $\boldsymbol{x}$, *i.e.*, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^{\top}$, $\boldsymbol{X} \in \mathbb{R}^{n\times p}$. The goal of graph learning algorithms is to learn a Laplacian matrix given only the data matrix $\boldsymbol{X}$, *i.e.*, often without any knowledge of $\mathcal{E}$.

To that end, the classical penalized Maximum Likelihood Estimator (MLE) of the Laplacian-constrained precision matrix of $\boldsymbol{x}$, on the basis of the observed data $\boldsymbol{X}$, may be formulated as the following optimization program [2, 14]:

$$\begin{array}{ll} \underset{\boldsymbol{L}\succeq\mathbf{0}}{\mathsf{minimize}} & \mathsf{tr}\left(\boldsymbol{L}\boldsymbol{S}\right) - \log\det^{*}\left(\boldsymbol{L}\right), \\ \mathsf{subject\ to} & \boldsymbol{L}\mathbf{1} = \mathbf{0},\ L_{ij} = L_{ji} \leq 0, \end{array} \tag{1}$$

where $\det^{*}(\boldsymbol{L})$ is the generalized determinant, *i.e.*, the product of the positive eigenvalues of $\boldsymbol{L}$, $\boldsymbol{S}$ is a similarity matrix, *e.g.*, the sample covariance (or correlation) matrix $\boldsymbol{S} \propto \boldsymbol{X}^{\top}\boldsymbol{X}$.

## 3. PROPOSED FORMULATION AND ALGORITHM

In this section we describe the proposed formulation for learning a normalized graph Laplacian matrix from data.

The symmetric normalized graph Laplacian, denoted as $\boldsymbol{\Theta}$ , and the unnormalized adjacency matrix are related as follows:

$$\boldsymbol{\Theta} \triangleq \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}. \tag{2}$$

Assuming that the data generating process is now a GMRF whose precision matrix is modeled as a normalized Laplacian of a connected graph, we write the MLE of $\boldsymbol{\Theta}$ as the following optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{\Theta}\succeq\mathbf{0},\boldsymbol{D},\boldsymbol{W}}{\mathsf{minimize}} & \mathrm{tr}\left(\boldsymbol{\Theta}\boldsymbol{S}\right) - \log\det^{*}\left(\boldsymbol{\Theta}\right), \\ \mathsf{subject\ to} & \boldsymbol{\Theta} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}, \\ & \boldsymbol{\Theta}_{ij} = \boldsymbol{\Theta}_{ji} \leq 0,\ i \neq j,\ \mathrm{rank}(\boldsymbol{\Theta}) = p - 1, \\ & \boldsymbol{\Theta}\boldsymbol{D}^{1/2}\mathbf{1} = \mathbf{0},\ \boldsymbol{D} = \mathsf{Diag}(\boldsymbol{W}\mathbf{1}). \end{array} \tag{3}$$

Leveraging the linear operators for the adjacency [4] and degree [16] matrices, we are able to express the normalized Laplacian matrix as

$$\boldsymbol{\Theta} = \boldsymbol{I} - (\mathfrak{D}\boldsymbol{w})^{-1/2}\mathcal{A}\boldsymbol{w}(\mathfrak{D}\boldsymbol{w})^{-1/2}, \tag{4}$$

which relate those quantities to the vector of edge weights $\boldsymbol{w} \in \mathbb{R}_{+}^{p(p-1)/2}$.

Plugging (4) into (3), we rewrite Problem (3) as follows:

$$\begin{array}{ll} \underset{\boldsymbol{\Theta},\boldsymbol{w}\geq\mathbf{0}}{\mathsf{maximize}} & \mathrm{tr}\left((\mathfrak{D}\boldsymbol{w})^{-1/2}\mathcal{A}\boldsymbol{w}(\mathfrak{D}\boldsymbol{w})^{-1/2}\boldsymbol{S}\right) \\ & + \log\det^{*}\left(\boldsymbol{\Theta}\right), \\ \mathsf{subject\ to} & \boldsymbol{\Theta} = \boldsymbol{I} - (\mathfrak{D}\boldsymbol{w})^{-1/2}\mathcal{A}\boldsymbol{w}(\mathfrak{D}\boldsymbol{w})^{-1/2}, \\ & \mathsf{rank}(\boldsymbol{\Theta}) = p - 1. \end{array} \tag{5}$$

We note that the constraints $\boldsymbol{\Theta}\boldsymbol{D}^{1/2}\mathbf{1} = \mathbf{0}$ and $\boldsymbol{D} = \mathsf{Diag}(\boldsymbol{W}\mathbf{1})$, originally part of Problem (3), have been dropped from Problem (5) as a result of the following lemma:

**Lemma 1** *Any* $\boldsymbol{\Theta}$ *in the feasible set of Problem* (5) *satisfies the constraint* $\boldsymbol{\Theta}\left(\mathfrak{D}\boldsymbol{w}\right)^{1/2}\mathbf{1} = \mathbf{0}$.

**Proof** The proof follows directly from the definition of the normalized Laplacian matrix as well as the fact that $\mathcal{A}\boldsymbol{w}\mathbf{1} = \mathfrak{D}\boldsymbol{w}\mathbf{1}$ for any $\boldsymbol{w} \geq \mathbf{0}$. ■

We further simplify Problem (5) by introducing a slack variable $\boldsymbol{\Psi} \in \mathbb{D}$, $\boldsymbol{\Psi} \triangleq (\mathfrak{D}\boldsymbol{w})^{-1/2}$, where $\mathbb{D}$ is defined as

$$\mathbb{D} \triangleq \left\{\boldsymbol{\Psi} \in \mathbb{R}^{p\times p} \,|\, \boldsymbol{\Psi}_{ii} > 0\,\forall\, i \in [p],\ \boldsymbol{\Psi}_{ij} = 0\,\forall\, i \neq j\right\}, \tag{6}$$

*i.e.*, the set of $p \times p$ positive definite diagonal matrices.

Then, our proposed formulation for learning a normalized Laplacian matrix is given as follows:

$$\begin{array}{ll} \underset{\boldsymbol{\Theta},\boldsymbol{w}\geq\mathbf{0},\boldsymbol{\Psi}}{\mathsf{minimize}} & -\log\det^{*}\left(\boldsymbol{\Theta}\right) - \mathrm{tr}\left(\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\boldsymbol{S}\right), \\ \mathsf{subject\ to} & \boldsymbol{\Theta} = \boldsymbol{I} - \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi},\ \boldsymbol{\Psi}^{-2} = \mathfrak{D}\boldsymbol{w}, \\ & \mathsf{rank}(\boldsymbol{\Theta}) = p - 1,\ \boldsymbol{w} \geq \mathbf{0},\ \boldsymbol{\Psi} \in \mathbb{D}. \end{array} \tag{7}$$

*Remark I:* We note that, in practice, we only need to consider the diagonal elements of $\boldsymbol{\Psi}$ (as its off-diagonal elements are zero) and we could rewrite $\boldsymbol{\Psi} = \mathsf{Diag}(\boldsymbol{\psi})$, $\boldsymbol{\psi} \in \mathbb{R}_{++}^{p}$, but we will stick with the current style for the sake of notational simplicity.

### 3.1. ADMM Solution

In this subsection, we use ADMM to design an iterative algorithm for the optimization program stated in Problem (7).

The partial augmented Lagrangian of Problem (7) is:

$$\begin{aligned} L_\rho(\boldsymbol{\Theta}, \boldsymbol{w}, \boldsymbol{\Psi}, \boldsymbol{Y}, \boldsymbol{Z}) &= -\log\det^*(\boldsymbol{\Theta}) - \operatorname{tr}(\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\boldsymbol{S}) \\ &+ \frac{\rho}{2}\|\boldsymbol{\Theta} - \boldsymbol{I} + \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\|_{\mathrm{F}}^2 + \frac{\rho}{2}\|\boldsymbol{\Psi}^{-2} - \mathfrak{D}\boldsymbol{w}\|_{\mathrm{F}}^2 \\ &+ \langle \boldsymbol{\Theta} - \boldsymbol{I} + \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}, \boldsymbol{Y}\rangle + \langle \boldsymbol{\Psi}^{-2} - \mathfrak{D}\boldsymbol{w}, \boldsymbol{Z}\rangle, \end{aligned} \tag{8}$$

where $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are dual variables associated to the equality constraints.

#### 3.1.1. Subproblem for $\boldsymbol{\Theta}$

The subproblem for $\boldsymbol{\Theta}$ can be written as

$$\begin{aligned} \boldsymbol{\Theta}^\star = \; & \underset{\boldsymbol{\Theta}}{\text{arg min}} \; \tfrac{\rho}{2}\|\boldsymbol{\Theta} - \boldsymbol{I} + \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\|_{\mathrm{F}}^2 + \langle \boldsymbol{\Theta}, \boldsymbol{Y}\rangle \\ & \qquad - \log\det^*(\boldsymbol{\Theta}), \\ & \text{subject to } \operatorname{rank}(\boldsymbol{\Theta}) = p - 1. \end{aligned} \tag{9}$$

**Lemma 2** *The global minimizer of Problem (9) is [22]*

$$\boldsymbol{\Theta}^\star = \frac{1}{2\rho}\boldsymbol{U}\left(\boldsymbol{\Gamma} + \sqrt{\boldsymbol{\Gamma}^2 + 4\rho\boldsymbol{I}}\right)\boldsymbol{U}^\top, \tag{10}$$

*where $\boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^\top$ is the eigenvalue decomposition of $\rho(\boldsymbol{I} - \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}) - \boldsymbol{Y}$ with $\boldsymbol{\Gamma} \in \mathbb{R}^{p-1\times p-1}$ containing the largest $p-1$ eigenvalues along its diagonal and $\boldsymbol{U} \in \mathbb{R}^{p\times p-1}$ containing the corresponding eigenvectors.*

#### 3.1.2. Subproblem for $\boldsymbol{w}$

The subproblem with respect to $\boldsymbol{w}$ can be written as

$$\boldsymbol{w}^\star = \underset{\boldsymbol{w}\geq\boldsymbol{0}}{\text{arg min}} \; f(\boldsymbol{w}; \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{Y}, \boldsymbol{Z}), \tag{11}$$

where

$$\begin{aligned} f(\boldsymbol{w}; \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{Y}, \boldsymbol{Z}) &\triangleq \frac{\rho}{2}\|\boldsymbol{\Theta} - \boldsymbol{I} + \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\|_{\mathrm{F}}^2 \\ &+ \frac{\rho}{2}\|\mathfrak{D}\boldsymbol{w} - \boldsymbol{\Psi}^{-2}\|_{\mathrm{F}}^2 + \langle \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}, \boldsymbol{Y} - \boldsymbol{S}\rangle - \langle \mathfrak{D}\boldsymbol{w}, \boldsymbol{Z}\rangle. \end{aligned} \tag{12}$$

Problem (11) is a nonnegative, convex quadratic program (QP), however, due to the large number of variables, *i.e.*, $O(p^2)$, using a disciplined convex programming language such as cvxpy [23] or even calling directly a QP solver is prohibitive. Instead, we update $\boldsymbol{w}$ via a simple projected gradient descent, with backtracking line search, designed as in Algorithm 1.

More precisely, the gradient of the objective function (13) is as follows:

$$\begin{aligned} \nabla_{\boldsymbol{w}} f(\boldsymbol{w}; \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{Y}, \boldsymbol{Z}) &= \rho\mathcal{A}^*\left[\boldsymbol{\Psi}\left(\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi} + (\boldsymbol{\Theta} - \boldsymbol{I})\right.\right. \\ &\left.\left. + \rho^{-1}(\boldsymbol{Y} - \boldsymbol{S})\right)\boldsymbol{\Psi}\right] + \rho\mathfrak{D}^*\left(\mathfrak{D}\boldsymbol{w} - \boldsymbol{\Psi}^{-2} - \rho^{-1}\boldsymbol{Z}\right), \end{aligned} \tag{13}$$

and the stepsize $\alpha^l$ is computed via the backtracking line search as follows:

$$\begin{aligned} f\left(\boldsymbol{w}^{l+1}\right) &\leq f\left(\boldsymbol{w}^l\right) + \left\langle \nabla f\left(\boldsymbol{w}^l\right), \boldsymbol{w}^{l+1} - \boldsymbol{w}^l\right\rangle \\ &+ \frac{1}{2\alpha^l}\|\boldsymbol{w}^{l+1} - \boldsymbol{w}^l\|_{\mathrm{F}}^2. \end{aligned} \tag{14}$$

**Algorithm 1:** PGD for Subproblem (11)

```
while $l \leq$ maxiter do
    $\boldsymbol{w}^{l+1} \leftarrow \left(\boldsymbol{w}^l - \alpha^l \nabla f(\boldsymbol{w}^l)\right)^+$, where $\alpha^l$ is chosen such that (14) is satisfied
    if $\|\boldsymbol{w}^{l+1} - \boldsymbol{w}^l\|_\infty \leq \epsilon$ then
        return $\boldsymbol{w}^{l+1}$
    end
end
```

#### 3.1.3. Subproblem for $\boldsymbol{\Psi}$

The subproblem for $\boldsymbol{\Psi}$ can be written as

$$\boldsymbol{\Psi}^\star \in \underset{\boldsymbol{\Psi}\in\mathbb{D}}{\text{argmin}} \; g(\boldsymbol{\Psi}; \boldsymbol{\Theta}, \boldsymbol{w}, \boldsymbol{Y}, \boldsymbol{Z}), \tag{15}$$

where

$$\begin{aligned} g(\boldsymbol{\Psi}; \boldsymbol{\Theta}, \boldsymbol{w}, \boldsymbol{Y}, \boldsymbol{Z}) &\triangleq \frac{\rho}{2}\|\boldsymbol{\Theta} - \boldsymbol{I} + \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\|_{\mathrm{F}}^2 \\ &+ \frac{\rho}{2}\|\boldsymbol{\Psi}^{-2} - \mathfrak{D}\boldsymbol{w}\|_{\mathrm{F}}^2 + \langle \boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}, \boldsymbol{Y} - \boldsymbol{S}\rangle + \langle \boldsymbol{\Psi}^{-2}, \boldsymbol{Z}\rangle. \end{aligned} \tag{16}$$

Because Problem (15) is nonconvex, we use gradient descent to find a local minima. Note that elements of the set $\mathbb{D}$ satisfy two constraints: (*i*) $\boldsymbol{\Psi}_{ij} = 0$ for $i \neq j$, and (*ii*) $\boldsymbol{\Psi}_{ii} > 0$ for $i \in [p]$. The first constraint can be trivially accounted for by enforcing $\boldsymbol{\Psi}_{ij}^0 = 0$, whereas the second constraint is handled via backtracking line search.

More specifically, we have the iterate

$$\boldsymbol{\Psi}^{l+1} = \boldsymbol{\Psi}^l - \alpha^l \nabla_{\boldsymbol{\Psi}} g\big(\boldsymbol{\Psi}^l\big), \tag{17}$$

where $\nabla g\big(\boldsymbol{\Psi}^l\big)$ is the gradient of the objective function in (15) at $\boldsymbol{\Psi}^l$, which is computed as

$$\begin{aligned} \nabla g\big(\boldsymbol{\Psi}; \boldsymbol{\Theta}, \boldsymbol{w}, \boldsymbol{Y}, \boldsymbol{Z}\big) &= \rho(\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w} + \mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}) \\ &+ \rho\big(-2\boldsymbol{\Psi}^{-5} + 2\mathfrak{D}\boldsymbol{w} \odot \boldsymbol{\Psi}^{-3}\big) \\ &+ \big(\boldsymbol{Y} - \boldsymbol{S} + \rho(\boldsymbol{\Theta} - \boldsymbol{I})\big)\boldsymbol{\Psi}\mathcal{A}\boldsymbol{w} \\ &+ \mathcal{A}\boldsymbol{w}\boldsymbol{\Psi}\big(\boldsymbol{Y} - \boldsymbol{S} + \rho(\boldsymbol{\Theta} - \boldsymbol{I})\big) - 2\boldsymbol{\Psi}^{-3} \odot \boldsymbol{Z}, \end{aligned} \tag{18}$$

where $\odot$ denotes the elementwise product.

The backtracking line search condition is

$$\begin{aligned} g\big(\boldsymbol{\Psi}^{l+1}\big) &\leq g\big(\boldsymbol{\Psi}^l\big) + \big\langle \nabla g\big(\boldsymbol{\Psi}^l\big),\ \boldsymbol{\Psi}^{l+1} - \boldsymbol{\Psi}^l\big\rangle \\ &+ \frac{1}{2\alpha^l}\left\|\boldsymbol{\Psi}^{l+1} - \boldsymbol{\Psi}^l\right\|_{\mathrm{F}}^2, \left[\boldsymbol{\Psi}^{l+1}\right]_{ii} > 0,\ \forall\, i \in [p]. \end{aligned} \tag{19}$$

Algorithm 2 describes the gradient descent scheme to update $\boldsymbol{\Psi}$. The updates for the dual variables $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are done via gradient ascent as in the ADMM framework. Algorithm 3 summarizes our proposed approach for learning a normalized Laplacian matrix.

**Algorithm 2:** Gradient descent for Subproblem (15)

```
while $l \leq$ maxiter do
    $\boldsymbol{\Psi}^{l+1} \leftarrow \boldsymbol{\Psi}^l - \alpha^l \nabla g(\boldsymbol{\Psi}^l)$, where $\alpha^l$ is chosen
      such that (19) is satisfied
    if $\|\boldsymbol{\Psi}^{l+1} - \boldsymbol{\Psi}^l\|_\infty \leq \epsilon$ then
        return $\boldsymbol{\Psi}^{l+1}$
    end
end
```

**Algorithm 3:** Normalized Laplacian Learning

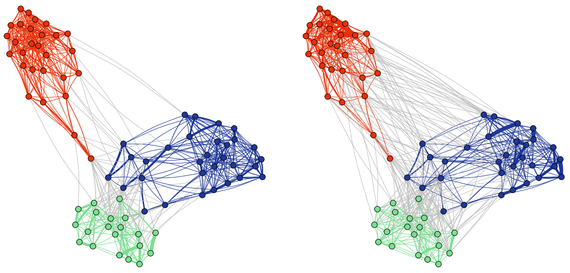
```
while convergence criteria not met do
    ▷ update $\boldsymbol{\Theta}^{l+1}$ via Lemma 2
    ▷ update $\boldsymbol{w}^{l+1}$ via Algorithm 1
    ▷ update $\boldsymbol{\Psi}^{l+1}$ via Algorithm 2
    $\boldsymbol{Z}^{l+1} = \boldsymbol{Z}^l + \rho\left(\left(\boldsymbol{\Psi}^{l+1}\right)^{-2} - \mathfrak{D}\boldsymbol{w}^{l+1}\right)$
    $\boldsymbol{Y}^{l+1} = \boldsymbol{Y}^l + \rho\left(\boldsymbol{\Theta}^{l+1} - \boldsymbol{I} + \boldsymbol{\Psi}^{l+1}\mathcal{A}\boldsymbol{w}^{l+1}\boldsymbol{\Psi}^{l+1}\right)$
end
```

*Remark II:* In spite of significant advancements in convergence theory of ADMM-like algorithms in nonconvex scenarios, see *e.g.* [24], a thorough convergence proof must be done for Algorithm 3. We left this analysis as future work.

## 4. EXPERIMENTAL RESULTS

### 4.1. US Stock Data

In this experiment, we compare the performance of the normalized Laplacian estimated via Algorithm 3 and the unnormalized version estimated via the methods in [2, 14] in learning graphs of financial stocks. Performance is measured in terms of the modularity of the estimated graph [25] – the higher the modularity, the higher is the separation of the graph in clusters, which is a stylized fact in stock markets [6]. More precisely, we select 79 stocks from three sectors of the S&P500 and use as data their daily returns over a period of 1006 trading days from Jan 2nd 2014 to Dec 29th 2017. We can observe from Fig. 1 that the graph estimated from the normalized Laplacian has a higher modularity and as a result it is likely a better representation of the actual financial network than that of the graph estimated from the unnormalized Laplacian.



(a) Normalized Laplacian, proposed (modularity = 0.50).

(b) Unnormalized Laplacian ([2, 14]) (modularity = 0.40).

**Fig. 1**: Performance of the estimators for S&P500 stocks.

### 4.2. Empirical Convergence

Convergence is a critical aspect of any iterative algorithm. In order to verify the empirical convergence of Algorithm 3, we generate synthetic data following the GMRF model where the underlying graph is a random graph with 128 nodes. We then measure the convergence trend in terms of the residuals over iterations. We repeat this process over 100 realizations. Fig. 2 shows that Algorithm 3 presents a good empirical convergence by achieving low residuals in just a few iterations.

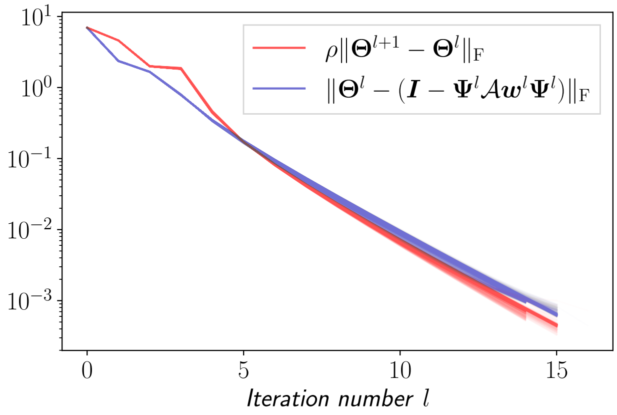


**Fig. 2**: Convergence trend in synthetic data.

## 5. CONCLUSION

In this paper we proposed an estimator for the normalized Laplacian matrix of a graph under the GMRF assumption. Results using financial time-series data from stock markets reveal that the normalized Laplacian provides an improved representation of financial networks than the conventional Laplacian matrix. Future work will analyse the convergence of the proposed algorithm as well as extend the proposed formulation to settings such as $k$-component and bipartite graphs as well as heavy tail assumptions on the data generating process.

## 6. REFERENCES

[1] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, pp. 432–41, 2008.

[2] 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.

[3] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, AAAI'16, pp. 1969–1976.

[4] 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, pp. 1–60, 2020.

[5] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Structured graph learning via laplacian spectral constraints," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[6] G. Marti, F. Nielsen, M. Bińkowski, and P. Donnat, "A review of two decades of correlations, hierarchies, networks and clustering in financial markets," in *arXiv: 1703.00485*, 2017.

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

[8] Z. Chen, L. Li, and J. Bruna, "Supervised community detection with graph neural networks," in *International Conference on Learning Representations*, 2019.

[9] R. N. Mantegna, "Hierarchical structure in financial markets," *The European Physical Journal B*, vol. 11, no. 1, pp. 193–197, 1999.

[10] J.-P. Onnela, K. Kaski, and J. Kertész, "Clustering and information in correlation based financial networks," *The European Physical Journal B*, vol. 38, pp. 353–362, 2004.

[11] T. Raffinot, "Hierarchical clustering-based asset allocation," *The Journal of Portfolio Management*, vol. 44, 2018.

[12] T. Millington and M. Niranjan, "Partial correlation financial networks," *Applied Network Science*, vol. 5, 2020.

[13] R. Ramakrishna, H. Wai, and A. Scaglione, "A user guide to low-pass graph signal processing and its applications," *arXiv e-prints: 2008.01305*, 2020.

[14] L. Zhao, Y. Wang, S. Kumar, and D. P. Palomar, "Optimization algorithms for graph laplacian estimation via ADMM and MM," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4231–4244, 2019.

[15] J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Nonconvex Sparse Graph Learning under Laplacian-structured Graphical Model," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[16] J. V. M. Cardoso, J. Ying, and D. P. Palomar, "Graphical models in heavy-tailed markets," in *Advances in Neural Information Processing Systems (NeurIPS'21)*, 2021.

[17] Morgan Stanley Capital International and S&P Dow Jones, "Revisions to the global industry classification standard (gics) structure," 2018.

[18] A. Schreiner, *Equity Valuation Using Multiples: An Empirical Investigation*, Springer, 2019.

[19] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, no. 15, pp. 485–516, 2008.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[21] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[22] D. M. Witten and R. Tibshirani, "Covariance-regularized regression and classification for high dimensional problems," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 615–636, 2009.

[23] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[24] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, pp. 29–63, 2019.

[25] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, 2006.