Efficient Robust Graph Learning Based on Minimax Concave Penalty and γ-Cross Entropy

Tatsuya Koyakumaru Masahiro Yukawa
Department of Electronics and Electrical Engineering, Keio University, Japan

***Abstract*—This paper presents an efficient robust method to learn sparse graphs from contaminated data. Specifically, the convex-analytic approach using the minimax concave penalty is formulated using the so-called γ-lasso which exploits the γ-cross entropy. We devise a weighting technique which designs the data weights based on the ℓ_1 distance in addition to the Mahalanobis distance for avoiding possible failures of outlier rejection due to the combinatorial graph Laplacian structure. Numerical examples show that the proposed method significantly outperforms γ-lasso and $t$lasso as well as the existing non-robust graph learning methods in contaminated situations.**

***Index Terms*—graph learning, minimax concave penalty, robust statistics, γ-cross entropy**

I. Introduction

Graph learning aims to infer potential relationships among data, and it has many applications in a variety of fields such as financial analysis [1], molecular biology [2], and network anomaly detection [3]. The Gaussian graphical model approaches [4]–[8] assume that data emerge from a multivariate Gaussian distribution and that the edge weights are designed based on partial correlation coefficients. A popular example is the graphical lasso [4], which imposes positive definiteness as well as sparseness on the matrix representing the graph, where the ℓ_1 norm is used for sparsification. The Gaussian graphical model approaches are highly versatile because they are based on strong statistical foundations without any physical constraints on the graph. Unfortunately, however, the use of the ℓ_1 norm tends to yield estimation biases, causing significant degradation of interpretability. Alternative approaches using nonconvex penalties such as the minimax concave (MC) penalty have been proposed to alleviate this issue while maintaining the benefit of variance reduction [9]–[12]. Among those approaches, the one proposed in [9], [10] uses the classical Moreau's decomposition as well as the Tikhonov regularization to convexify the overall cost function. Simultaneous use of such a nonconvex regularization and the Tikhonov regularization successfully enhances interpretability, as well as sparseness, of graphs with convergence guarantee. This approach employs the primal-dual (PD) splitting method, and we thus refer to it as PD-MC henceforth.

While such methods tend to yield interpretable graphs in many situations, its performance may degrade significantly when the data are contaminated by outliers. Robust graph learning methods based on graphical lasso have been studied independently [13]–[16]. For instance, $t$lasso [15] exploits the heavy-tailness of the t-distribution to suppress the deterioration of estimation accuracy caused by outliers. Unfortunately, this method does not perform well when the outliers are concentrated on one side of the real axis [16]. This limitation has been overcome by γ-lasso which is based on the γ-cross entropy to remove outliers while taking into account the structure of the graphical model. Its interpretability, however, is limited due again to the use of the ℓ_1 norm. Now, our primitive idea is the following: a robust method to learn highly interpretable graphs will be attained by blending the advantages of γ-lasso and PD-MC.

In this paper, we present an efficient robust graph learning method extending γ-lasso based on the idea of PD-MC. In the original γ-lasso framework, a weight to each data vector is computed based on the Mahalanobis distance from the estimated mean. Specifically, outlier is distant from the mean by its nature, and this vanishes its weight. As a result, the impacts of outliers to the estimates of the mean and the covariance matrix (which are involved in the PD-MC formulation) become negligible, and hence the estimate becomes robust against outliers. However, due to the combinatorial graph Laplacian (CGL) structure on which PD-MC is based, outlier rejection by γ-lasso may fail when outliers have certain structures. To solve this issue, we devise a weighting technique which designs the weights based not only on the Mahalanobis distance but also on the ℓ_1 distance. We show the robustness of the proposed γ-PD-MC method based on the approximate Pythagorean relation. Numerical examples show the remarkable advantages of the proposed method over γ-lasso and $t$lasso for several types of graph.

II. Preliminaries

This section presents the notation and mathematical tools used in the present work.

A. Notation

The sets of real numbers and nonnegative real numbers are denoted by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. The transpose of vector/matrix is denoted by $(\cdot)^\mathsf{T}$. Given a vector $\boldsymbol{x} := [x_1, x_2, \cdots, x_n]^\mathsf{T} \in \mathbb{R}^n$, we define the ℓ_1 and the ℓ_2 norms by $\|\boldsymbol{x}\|_1 := \sum_{i=1}^n |x_i|$ and $\|\boldsymbol{x}\|_2 := \left(\sum_{i=1}^n x_i^2\right)^{\frac{1}{2}}$, respectively. Let $\boldsymbol{I}$, $\boldsymbol{0}$, and $\boldsymbol{1}$ denote the identity matrix, the vector of zeros, and the vector of ones, respectively. Let $\boldsymbol{J} := \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\mathsf{T} \in \mathbb{R}^{n\times n}$. Let $\mathrm{diag}(\boldsymbol{x})$ denote the diagonal matrix with its diagonal entries given by the components of a vector $\boldsymbol{x}$.

This work was supported by the Grants-in-Aid for Scientific Research (KAKENHI) under Grant JP18H01446.

We consider undirected weighted graphs with nonnegative edge weights. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ is composed of a set of nodes $\mathcal{V}$, edges $\mathcal{E}$, and a weight matrix $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, where $n = |\mathcal{V}|$ is the number of nodes. Here, $\boldsymbol{W}$ is symmetric with $w_{ii} = 0$ by convention, and hence it is characterized completely by its upper triangular part of which the vectorized version is denoted by $\boldsymbol{w} \in \mathcal{C} := \mathbb{R}_+^{\frac{n(n-1)}{2}}$. The combinatorial graph Laplacian (CGL) is a function of $\boldsymbol{w}$ defined by $L(\boldsymbol{w}) := \mathrm{diag}(\boldsymbol{W}\mathbf{1}) - \boldsymbol{W} \in \mathbb{R}^{n\times n}$. Here, $L : \mathcal{C} \to \mathbb{R}^{n\times n}$ is a linear operator with its adjoint operator denoted by L^*. It is clear that $L(\boldsymbol{w})\mathbf{1} = \mathbf{0}$, and the zero eigenvalue is simple when the graph is connected.

B. Mathematical tools

The projection of $\boldsymbol{w}$ onto the nonnegative cone $\mathcal{C}(:= \mathbb{R}_+^{\frac{n(n-1)}{2}})$ is denoted by $P_{\mathcal{C}}(\boldsymbol{w}) := \operatorname{argmin}_{\boldsymbol{y}\in\mathcal{C}} \|\boldsymbol{w} - \boldsymbol{y}\|_2$. The soft thresholding operator $\mathrm{soft}_\lambda : \mathcal{C} \to \mathcal{C}$ for $\lambda > 0$ is defined by $\mathrm{soft}_\lambda(\boldsymbol{w}) := P_{\mathcal{C}}(\boldsymbol{w} - \lambda\mathbf{1})$ for any nonnegative vector $\boldsymbol{w} \in \mathcal{C}$. The MC penalty [17] of index $\eta > 0$ is defined by

$$\Phi_\eta^{\mathrm{MC}}(\boldsymbol{w}) := \|\boldsymbol{w}\|_1 - \min_{\boldsymbol{y}\in\mathbb{R}^n} \left(\|\boldsymbol{y}\|_1 + \frac{1}{2\eta}\|\boldsymbol{w} - \boldsymbol{y}\|_2^2 \right), \quad (1)$$

which is a weakly convex function. The MC penalty $\Phi_\eta^{\mathrm{MC}}(\boldsymbol{w})$ induces a sparse estimate as well as alleviating underestimation compared to the ℓ_1 penalty, because it becomes constant above the threshold η.

C. γ-cross entropy

Let $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^n$, be the underlying probability density function of signals, and $\delta(\boldsymbol{x})$ be the density of outliers. Given a contamination ratio $\varepsilon \in (0, 1)$, the density of contaminated data is given by

$$g(\boldsymbol{x}) = (1 - \varepsilon)f(\boldsymbol{x}) + \varepsilon\delta(\boldsymbol{x}). \quad (2)$$

Given a $\gamma > 0$, the γ-cross entropy between the g and an estimate $\widehat{f}$ of f is defined as follows [18]:

$$d_\gamma(g, \widehat{f}) = -\frac{1}{\gamma}\log\int g(\boldsymbol{x})\widehat{f}^{\,\gamma}(\boldsymbol{x})d\boldsymbol{x} + \frac{1}{1+\gamma}\log\int \widehat{f}^{\,1+\gamma}(\boldsymbol{x})d\boldsymbol{x}.$$

Minimizing $d_\gamma(g, \widehat{f})$ w.r.t. $\widehat{f}$ leads to robust estimation of the density f in the presence of outliers (see Section III-B).

D. PD-MC algorithm

The PD-MC algorithm [10] further improves the learning accuracy of the combinatorial graph Laplacian by using the MC penalty in the regularization term of the CGL estimation method [19]. It also has the features of efficient learning and guaranteed convergence under certain conditions at the same time by representing CGL in the form of $L(\boldsymbol{w})$.

PD-MC is given in Algorithm 1. It requires $\mathcal{O}(n^3)$ complexity, which is the same as the other graph learning methods based on graphical models. For more details, see [10].

Algorithm 1 PD-MC

Input: Initial estimate $(\boldsymbol{w}_0, \boldsymbol{V}_0)$, tolerance $\epsilon_{\mathrm{PD}} > 0$, proximity parameters $\tau > 0$, $\sigma > 0$, covariance matrix $\boldsymbol{S}$, regularization parameters $\lambda_1 \geq 0, \lambda_2 \geq 0$, MC parameter $\eta > 0$, relaxation parameters $\rho_k > 0$. (Set $k := 0$)

while $\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\|_2^2 > \epsilon\|\boldsymbol{w}_k\|_2^2$ $(k \neq 0)$ **do**

1. Compute $\tilde{\boldsymbol{w}}_{k+1} = P_{\mathcal{C}}[\boldsymbol{w}_k - \tau L^*(\boldsymbol{V}_k) - \tau(\lambda_1\mathbf{1} + L^*(\boldsymbol{S})) - \tau(\eta^{-1}\lambda_1\mathrm{soft}_{\mathbf{1}}(\boldsymbol{w}_k) - \eta^{-1}\lambda_1\boldsymbol{w}_k + \lambda_2\boldsymbol{w}_k)]$
2. Find the eigenvalues ν_i and the matrix $\boldsymbol{U} = [\boldsymbol{u}_1 \dots \boldsymbol{u}_n]$ containing all the corresponding (unit-norm) eigenvectors of $\left(\boldsymbol{J} + \sigma^{-1}\boldsymbol{V}_k + L\left(2\tilde{\boldsymbol{w}}_{k+1} - \boldsymbol{w}_k\right)\right)$
3. Compute $\tilde{\boldsymbol{V}}_{k+1} = \boldsymbol{V}_k + \sigma L\left(2\tilde{\boldsymbol{w}}_{k+1} - \boldsymbol{w}_k\right) + \sigma\boldsymbol{J} - \sigma\left[\boldsymbol{U}\mathrm{diag}\left(\frac{\nu_1+\sqrt{\nu_1^2+4\sigma^{-1}}}{2}, \dots, \frac{\nu_n+\sqrt{\nu_n^2+4\sigma^{-1}}}{2}\right)\boldsymbol{U}^{\mathsf{T}}\right]$
4. $(\boldsymbol{w}_{k+1}, \boldsymbol{V}_{k+1}) = \rho_k\left(\tilde{\boldsymbol{w}}_{k+1}, \tilde{\boldsymbol{V}}_{k+1}\right) + (1-\rho_k)(\boldsymbol{w}_k, \boldsymbol{V}_k)$
5. $k \leftarrow k + 1$

end while

return graph Laplacian $L(\boldsymbol{w}_k)$

III. Robust Graph Learning

In this section, we present the proposed graph learning method and its properties.

A. Proposed γ-PD-MC Method

Suppose that m measurement vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m \in \mathbb{R}^n$ are available for learning. Using the empirical γ-cross entropy, the negative γ-likelihood function under the CGL structure is given by (cf. [16])

$$\widetilde{d}_\gamma(\boldsymbol{\theta}) := -\frac{1}{\gamma}\log\left(\frac{1}{m}\sum_{i=1}^m \widehat{f}_{\boldsymbol{\theta}}^{\,\gamma}(\boldsymbol{x}_i)\right) + \frac{1}{1+\gamma}\log\int \widehat{f}_{\boldsymbol{\theta}}^{\,1+\gamma}(\boldsymbol{x})d\boldsymbol{x},$$

where $\boldsymbol{\theta} := (\boldsymbol{\mu}, \boldsymbol{w}) \in \mathbb{R}^n \times \mathcal{C}$, and $\widehat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = (2\pi)^{-n/2}|L(\boldsymbol{w})|^{1/2}\exp\left[-(\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}}L(\boldsymbol{w})(\boldsymbol{x} - \boldsymbol{\mu})/2\right]$ is the density of the multivariate normal distribution. Here, $|L(\boldsymbol{w})|$ is the pseudo determinant (i.e., the product of the nonzero eigenvalues) of the singular matrix $L(\boldsymbol{w})$ [19]. It actually holds that $|L(\boldsymbol{w})| = \det(L(\boldsymbol{w}) + \boldsymbol{J})$, because the eigenvalue of $L(\boldsymbol{w}) + \boldsymbol{J}$ corresponding to the eigenvector $\frac{1}{n}\mathbf{1}$ is one.

We cast the graph learning task as the problem of finding

$$\hat{\boldsymbol{\theta}} \in \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^n\times\mathcal{C}} \left(\widetilde{d}_\gamma(\boldsymbol{\theta}) + \lambda_1\Phi_\eta^{\mathrm{MC}}(\boldsymbol{w}) + \frac{\lambda_2}{2}\|\boldsymbol{w}\|_2^2\right), \quad (3)$$

where $\lambda_1, \lambda_2 > 0$ are the regularization parameters. Here, the two regularizers are introduced in the work of PD-MC [9], [10] to enhance sparsity and graph interpretability simultaneously with guarantee of global optimality. The proposed method is given in Algorithm 2, based on the algorithm in [16] derived from the majorization-minimization (MM) algorithm. Step 4 of the algorithm is based on the following minimization problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^{\frac{n(n-1)}{2}}} -\mathrm{logdet}(L(\boldsymbol{w}) + \boldsymbol{J}) + \lambda_1'\Phi_\eta^{\mathrm{MC}}(\boldsymbol{w}) + \frac{\lambda_2'}{2}\|\boldsymbol{w}\|_2^2 + \iota_{\mathcal{C}}(\boldsymbol{w}) + \langle(1+\gamma)\boldsymbol{S}_{z^{(t)}}(\boldsymbol{\mu}^{(t+1)}), L(\boldsymbol{w})\rangle, \quad (4)$$

Algorithm 2 γ-PD-MC

Input: Contaminated data vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m \in \mathbb{R}^n$, robustness parameter γ, balancing parameter κ, initial estimate $(\boldsymbol{w}_0, \boldsymbol{V}_0)$, whole loop tolerance $\epsilon > 0$, PD-MC tolerance $\epsilon_{\mathrm{PD}} > 0$, proximity parameters $\tau > 0$, $\sigma > 0$, regularization parameters $\lambda_1 \geq 0, \lambda_2 \geq 0$, MC parameter $\eta > 0$, relaxation parameters $\rho_k > 0$. (Set $t := 0$)

while $\|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}\|_2^2 > \epsilon \|\boldsymbol{w}^{(t)}\|_2^2$ $(t \neq 0)$ **do**

1. Update data weight $z_i^{(t)}$ by (5)
2. $\boldsymbol{\mu}^{(t+1)} := \sum_{i=1}^m z_i^{(t)} \boldsymbol{x}_i$
3. $\boldsymbol{S}_{z^{(t)}}(\boldsymbol{\mu}^{(t+1)}) := \sum_{i=1}^m z_i^{(t)} \big(\boldsymbol{x}_i - \boldsymbol{\mu}^{(t+1)}\big)\big(\boldsymbol{x}_i - \boldsymbol{\mu}^{(t+1)}\big)^{\mathsf{T}}$
4. Update $\boldsymbol{w}^{(t+1)}$ by minimizing (4) using Algorithm 1
5. $t \leftarrow t + 1$

end while

return graph Laplacian $L(\boldsymbol{w}^{(t)})$

where $\lambda_1' = 2(1+\gamma)\lambda_1$, $\lambda_2' = 2(1+\gamma)\lambda_2$, and the indicator function $\iota_{\mathcal{C}}(\boldsymbol{w}) := \begin{cases} 0, & \text{if } \boldsymbol{w} \in \mathcal{C}, \\ +\infty, & \text{otherwise,} \end{cases}$ enforces $\boldsymbol{w}$ to be nonnegative. The weights are given by

$$z_i^{(t)} := \frac{\exp\left[-\big(\kappa\psi_1(\boldsymbol{x}_i; \boldsymbol{\mu}^{(t)}) + \frac{\gamma}{2}\psi_2(\boldsymbol{x}_i; \boldsymbol{\mu}^{(t)}, \boldsymbol{w}^{(t)})\big)\right]}{\sum_{j=1}^m \exp\left[-\big(\kappa\psi_1(\boldsymbol{x}_j; \boldsymbol{\mu}^{(t)}) + \frac{\gamma}{2}\psi_2(\boldsymbol{x}_j; \boldsymbol{\mu}^{(t)}, \boldsymbol{w}^{(t)})\big)\right]}, \quad (5)$$

where $\kappa \geq 0$ controls the balance between $\psi_1(\boldsymbol{x}_i; \boldsymbol{\mu}^{(t)}) := \|\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)}\|_1$ and $\psi_2(\boldsymbol{x}_i; \boldsymbol{\mu}^{(t)}) := \big(\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)}\big)^{\mathsf{T}} L(\boldsymbol{w}^{(t)}) \big(\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)}\big)$. The weight $z_i^{(t)}$ given to the data $\boldsymbol{x}_i$ vanishes when $\boldsymbol{x}_i$ is an outlier vector (i.e., when it is distant from the mean vector $\boldsymbol{\mu}^{(t+1)}$), thereby removing outliers.

Remark 1. *Letting $\kappa := 0$ in* (5) *produces the weights given in the original work* [16]*, which gives a majorizer used in the MM algorithm. For $\kappa \neq 0$, although the weight $z_i^{(t)}$ certainly gives a majorant, it does not actually give a proper majorizer in the sense of* [16, eq. (5)]*, which means that the monotone decreasing property of the MM algorithm is not guaranteed theoretically. Nevertheless, it works well empirically as shown in Section IV. The reason for introducing ψ_1 in addition to ψ_2 is related to the CGL structure. More specifically, if $\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)} = r\mathbf{1}$ for some $r \in \mathbb{R}$, it follows that $\big(\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)}\big)^{\mathsf{T}} L(\boldsymbol{w}^{(t)}) \big(\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)}\big) = 0$ for any r due to $L(\boldsymbol{w}^{(t)})\mathbf{1} = \mathbf{0}$ coming from the CGL structure. Suppose that there exists an outlier $\boldsymbol{x}_i$ such that $\boldsymbol{x}_i - \boldsymbol{\mu}^{(t)} = r\mathbf{1}$ for some huge r. The weight for such an outlier does not vanish, and thus outlier rejection fails in this case. This may cause a serious performance degradation, and some important links may fail to be detected for instance. The additional function ψ_1 avoids such situation because it grows linearly as r increases, vanishing the weights to reject the corresponding outliers.*

B. Pythagorean relation in graph learning methods

A great advantage of using the γ-divergence [18]

$$D_\gamma(g, \widehat{f}_{\boldsymbol{\theta}}) := d_\gamma(g, \widehat{f}_{\boldsymbol{\theta}}) - d_\gamma(g, g) \quad (6)$$

is that the true distribution can be estimated even when the contamination ratio ε is large [18]. This property relies on the following assumption on the relation between the density $\delta(\boldsymbol{x})$ of outliers and the underlying density $f(\boldsymbol{x})$ of signals:

$$\nu_f := \left(\int \delta(\boldsymbol{x}) f^\gamma(\boldsymbol{x}) d\boldsymbol{x}\right)^{1/\gamma} \approx 0, \quad (7)$$

which means that $\delta(\boldsymbol{x})$ mostly lies on the tail of $f(\boldsymbol{x})$.

In [16, Remark 2], the Pythagorean relation [18] holds true for γ-lasso in the absence of the sparsity promoting regularizer. It can be easily seen that a similar relationship holds for arbitrary values of the regularization parameter as well by transformimg the Pythagorean relation. Assuming (7) for the estimated distribution $\widehat{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$, the approximate Pythagorean relation holds [18]:

$$D_\gamma(g, \widehat{f}_{\boldsymbol{\theta}}) = D_\gamma(f, \widehat{f}_{\boldsymbol{\theta}}) + D_\gamma(g, f) + O\left(\nu^\gamma\right) \quad (8)$$

where $\nu := \max\{\nu_f, \nu_{\widehat{f}_{\boldsymbol{\theta}}}\}$ with the $\nu_{\widehat{f}_{\boldsymbol{\theta}}}$ defined in the same way as in (7), and $O(\cdot)$ is Landau's symbol. Note here that $D_\gamma(g, f) + O\left(\varepsilon\nu^\gamma\right)$ is constant in $\widehat{f}_{\boldsymbol{\theta}}$. By (8), it holds that $d_\gamma(\boldsymbol{\theta}) \approx d_\gamma(g, \widehat{f}_{\boldsymbol{\theta}}) = d_\gamma(f, \widehat{f}_{\boldsymbol{\theta}}) + \mathrm{const.}$, which implies that minimizing the cost in (3) leads to minimizing $D_\gamma(f, \widehat{f}_{\boldsymbol{\theta}}) + \lambda_1 \Phi_\eta^{\mathrm{MC}}(\boldsymbol{w}) + \frac{\lambda_2}{2}\|\boldsymbol{w}\|_2^2$ approximately (since $d_\gamma(g, g)$ is constant in $\widehat{f}_{\boldsymbol{\theta}}$). Note here that an empirical estimate of the divergence $D_\gamma(f, \widehat{f}_{\boldsymbol{\theta}})$ from $\widehat{f}_{\boldsymbol{\theta}}$ to the density f of "clean signals" is hardly available since the data available are assumed to be contaminated. The argument given in this part explains the remarkable robustness of γ-PD-MC, as shown in Section IV.

IV. Numerical Examples

We conduct simulations to examine the performances of the proposed γ-PD-MC method, γ-lasso [16], and $t$lasso [15]. For the sake of reference, the non-robust methods (PD-MC [10], graphical lasso [4], and CGL [19]) are also tested.

A. Simulation settings

Dataset generation: We consider two types of graph: (i) grid graph $\mathcal{G}_{\mathrm{grid}}^{(\sqrt{n},\sqrt{n})}$, and (ii) Erdös-Rényi graph $\mathcal{G}_{\mathrm{ER}}^{(n,0,1)}$. The graph weights $\boldsymbol{w}_\star$ are randomly drawn from the uniform distribution over the interval [0.1, 3.0], regarded as the ground-truth graph Laplacian $L(\boldsymbol{w}_\star)$ in this simulation. From each graph generated, normal data are generated from $\mathcal{N}(\mathbf{0}, L(\boldsymbol{w}_\star)^\dagger)$, where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse, and the covariance matrix $\boldsymbol{S}$ is computed from normal data. In addition, 10% of the measurement vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$ are contaminated additively by outlier vectors of size n following the normal distribution $\mathcal{N}(\alpha\mathbf{1}, \boldsymbol{I})$ for $\alpha = 5$. For each type of graph, we randomly generate 15 graphs with $n = 100$ nodes using the toolbox given in [20].

Performance measure: The relative error (RE) and F-score (FS) are used as performance measures: $\mathrm{RE}\,(\widehat{\boldsymbol{w}}, \boldsymbol{w}_\star) := \|L(\widehat{\boldsymbol{w}}) - L(\boldsymbol{w}_\star)\|_{\mathrm{F}}^2 \,/\, \|L(\boldsymbol{w}_\star)\|_{\mathrm{F}}^2$, where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, and $\mathrm{FS}\,(\widehat{\boldsymbol{w}}, \boldsymbol{w}_\star) := 2\mathrm{tp}/(2\mathrm{tp} + \mathrm{fn} + \mathrm{fp})$, where tp, fp, and fn stand for true-positive, false-positive, and false-negative, respectively. Here, the relative error indicates the discrepancy between the ground-truth graph Laplacian $L(\boldsymbol{w}_\star) \in \mathbb{R}^{n\times n}$ and its estimate $L(\widehat{\boldsymbol{w}}) \in \mathbb{R}^{n\times n}$, while the F-score is a measure of accuracy for binary classification (taking values in [0,1]), indicating whether the sparse structures are extracted correctly.

Parameters: For each algorithm, the best parameters are chosen manually. Because the performances in the RE and FS measures are related to each other in the present simulation settings, the parameters are tuned to obtain the smallest relative errors on average. For the proposed method, $\lambda_2 := 0$ gave the best performance together with $\lambda_1 := 5.0 \times 10^{-3}$ and $\gamma := 0.01$ for grid graph, and with $\lambda_1 := 1.0 \times 10^{-2}$ and $\gamma := 0.05$ for Erdös-Rényi graph.

B. Results and discussions

Figures 1 and 2 show the performances for different values of m/n. The γ-PD-MC significantly outperforms the other methods in both measures for all m/n values.* Specifically, the gains compared to γ-lasso and $t$lasso in F-score are approximately up to 0.58 and 0.42, respectively. The remarkable gains come from (i) the use of the weakly convex regularizer instead of the ℓ_1 regularizer and (ii) the exploitation of the CGL structure. The F-scores of γ-lasso are considerably low for small m/n, because the regularization parameter for the ℓ_1 norm cannot be sufficiently large to obtain sparse graphs for avoiding large errors.

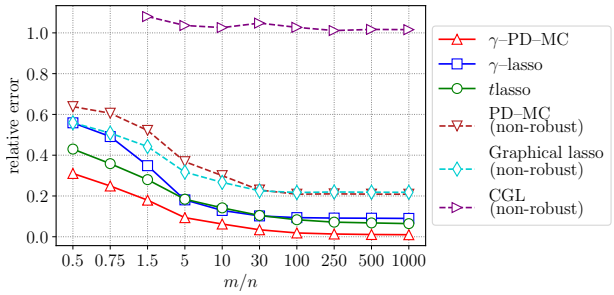
Figure 3 shows the performances across different contamination ratios for Erdös-Rényi graph. The simulation setup and the parameters are the same as in Fig. 2. It is shown that the learning accuracy of the robust methods keeps more or less constant, while that of the non-robust methods deteriorates as the contamination ratio increases. The proposed method achieves remarkably better performances than the existing methods even for large contamination ratios. This is due to the same reasons (i) and (ii) raised in the previous paragraph. The performance of $t$lasso for small outlier ratios is poor, as it assumes heavy tailed distributions despite the existence of no (few) outliers in that case.

V. Concluding Remarks

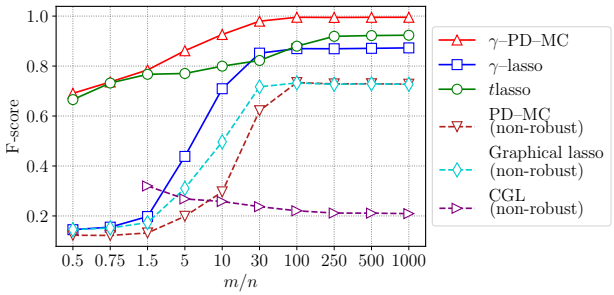
We proposed a robust method to learn sparse and interpretable graphs based both on the γ-lasso framework and PD-MC. We also showed efficient weight-design given to each datum in estimating graphs with the CGL structure. The robustness of the proposed γ-PD-MC method was shown based on the approximate Pythagorean relation. Numerical examples showed that the proposed method achieved remarkably better performance than γ-lasso and $t$lasso both in relative error and F-score for grid graph and Erdös-Rényi graphs.

In the present study, it was mostly assumed that all (or most) components of some data vectors $\boldsymbol{x}_i$ are contaminated. There is also a situation when the data is contaminated in a "cell-wise" manner, meaning that only small portions of many data vectors $\boldsymbol{x}_i$ are contaminated. In such a case, possible issues

*The absence of CGL plots for small m/n is due to numerical errors occurring in solving the quadratic programming.

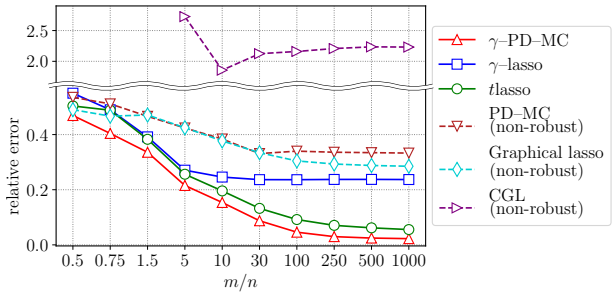


(a) relative error

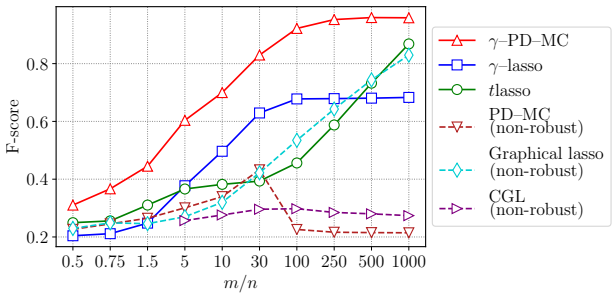


(b) F-score

Fig. 1: Performance across m/n for grid graph $\mathcal{G}_{\mathrm{grid.}}^{(10,10)}$

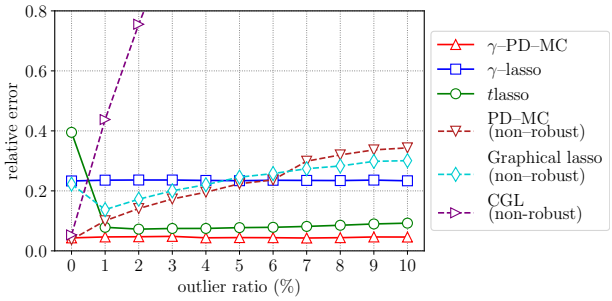


(a) relative error

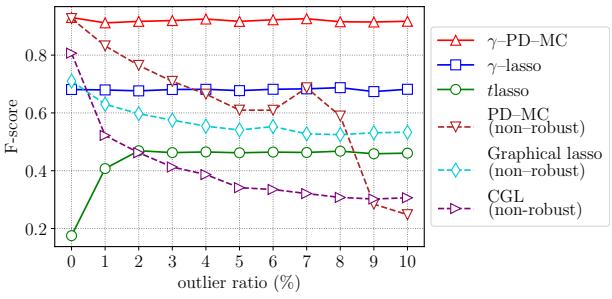


(b) F-score

Fig. 2: Performance across m/n for Erdös-Rényi graph $\mathcal{G}_{\mathrm{ER.}}^{(100,\ 0.1)}$



(a) relative error



(b) F-score

Fig. 3: Performance across the outlier ratio at $m/n = 100$ for Erdös-Rényi graph $\mathcal{G}_{\text{ER.}}^{(100,\ 0.1)}$

may happen because too many data vectors could be removed [21]. There remains a room for further studies on this point.

References

[1] P. Giudici and A. Spelta, "Graphical network models for international financial flows," *Journal of Business & Economic Statistics*, vol. 34, no. 1, pp. 128–138, 2016.

[2] O. Mason and M. Verwoerd, "Graph theory and networks in biology," *IET Systems Biology*, vol. 1, no. 30, pp. 89–119, Mar. 2007.

[3] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection methods, systems and tools," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

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

[5] R. Mazumder and T. Hastie, "The graphical lasso: New insights and alternatives," *Electronic Journal of Statistics*, vol. 6, pp. 2125–2149, 2012.

[6] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *The Annals of Statistics*, vol. 34, pp. 1436–1462, 2006.

[7] 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.

[8] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 467–483, 2017.

[9] T. Koyakumaru, M. Yukawa, E. Pavez, and A. Ortega, "A graph learning algorithm based on Gaussian Markov random fields and minimax concave penalty," in *Proceedings of 46th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5390–5394.

[10] T. Koyakumaru, M. Yukawa, E. Pavez, and A. Ortega, "Learning sparse graph with minimax concave penalty under Gaussian Markov random fields," *IEICE Trans. Fundamentals*, 2022, accepted for publication.

[11] 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*, vol. 33, 2020, pp. 7101–7113.

[12] Y. Zhang, K. C. Toh, and D. Sun, "Learning graph Laplacian with MCP," *ArXiv e-prints*, 2020.

[13] H. Liu, J. Lafferty, and L. Wasserman, "The nonparanormal: Semiparametric estimation of high dimensional undirected graphs," *Journal of Machine Learning Research*, vol. 10, no. 80, pp. 2295–2328, 2009.

[14] H. Sun and H. Li, "Robust Gaussian graphical modeling via ℓ_1 penalization," *Biometrics*, vol. 68, no. 4, pp. 1197–1206, 2012.

[15] M. Finegold and M. Drton, "Robust graphical modeling of gene networks using classical and alternative t-distributions," *The Annals of Applied Statistics*, vol. 5, no. 2A, pp. 1057–1080, 2011.

[16] K. Hirose, H. Fujisawa, and J. Sese, "Robust sparse Gaussian graphical modeling," *Journal of Multivariate Analysis*, vol. 161, pp. 172–190, 2017.

[17] I. Selesnick, "Sparse regularization via convex analysis," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4481–4494, Sep. 2017.

[18] H. Fujisawa and S. Eguchi, "Robust parameter estimation with a small bias against heavy contamination," *Journal of Multivariate Analysis*, vol. 99, no. 9, pp. 2053–2081, 2008.

[19] 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, Sep. 2017.

[20] N. Perraudin, J. Paratte, D. Shuman, *et al.*, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.

[21] S. Katayama, H. Fujisawa, and M. Drton, "Robust and sparse Gaussian graphical modelling under cell-wise contamination," *Stat*, vol. 7, no. 1, e181, 2018.