ESTIMATING SCALE-FREE NETWORKS VIA THE EXPONENTIATION OF MINIMAX CONCAVE PENALTY

Kei Hirose∗, Yukihiro Ogura† and Hidetoshi Shimodaira‡

ABSTRACT

We consider the problem of sparse estimation of undirected graphical models via the $L_1$ regularization. The ordinary lasso encourages the sparsity on all edges equally likely, so that all nodes tend to have small degrees. On the other hand, many real-world networks are often scale-free, where some nodes have a large number of edges. In such cases, a penalty that induces structured sparsity, such as a log penalty, performs better than the ordinary lasso. In practical situations, however, it is difficult to determine an optimal penalty among the ordinary lasso, log penalty, or somewhere in between. In this paper, we introduce a new class of penalty that is based on the exponentiation of the minimax concave penalty. The proposed penalty includes both the lasso and the log penalty, and the gap between these two penalties is bridged by a tuning parameter. We apply cross-validation to select an appropriate value of the tuning parameter. Monte Carlo simulations are conducted to investigate the performance of our proposed procedure. The numerical result shows that the proposed method can perform better than the existing log penalty and the ordinary lasso.

1. Introduction

Gaussian graphical models have been widely used to investigate the conditional independence between two variables given other variables. It is well-known that the conditional independence between two variables corresponds to the zero entries of the inverse covariance matrix under a Gaussian assumption (Edwards, 2000). In recent years, many researchers have applied an $L_1$ regularization, such as the lasso (Tibshirani, 1996), to estimate the high-dimensional sparse inverse covariance matrix. Meinshausen and Bühlmann (2006) proposed fitting a lasso regression to each node with other nodes being predictors. We call that method MB. Peng et al. (2009) introduced a joint regression, which also fits the lasso regression but ensures the symmetry of the estimated inverse covariance matrix. Several authors have considered the problem of maximizing the penalized log-likelihood function via the lasso (e.g., Yuan and Lin, 2007; Banerjee et al., 2008; Rothman et al., 2008; Friedman et al., 2008; Witten et al., 2011, hereinafter referred to as graphical lasso).

It is well-known that the ordinary lasso implies the Laplace prior to each parameter independently, which means the probability that two nodes are connected is equal for any pair of nodes (Erdös and Rényi, 1960). In this case, the degree distribution becomes binomial, so that all nodes tend to have small degrees. However, in many real-world networks,
some nodes have a large number of edges. An example is citations between scientific papers; most papers are not referred, but there exist a few papers which are referred to by a large number of papers. In such cases, the degree distribution should be heavy-tailed such as the power-law distribution rather than binomial distribution. In fact, it is well-known that many real-world networks are often scale-free (Barabási and Albert, 1999), in which the degree distribution follows the power-law distribution.

The estimation of the scale-free networks is realized by specification of a prior distribution that the underlying network is likely to be scale-free. Because the scale-free prior indirectly imposes sparseness on networks (Del Genio et al., 2011), the sparse solutions can be obtained even if a sparse estimation procedure such as $L_1$ regularization is not employed. Some authors have estimated scale-free networks without the use of the $L_1$ regularization. Sheridan et al. (2010) took a full-Bayesian approach, and generated samples from the posterior distribution using Markov chain Monte Carlo (MCMC) methods. More recently, Maruyama and Shikita (2014) proposed a prior distribution for the inverse covariance matrix based on their work on protein complex prediction, and estimated the posterior modes by the simulated annealing algorithm. However, both the MCMC and the simulated annealing require heavy computational loads, resulting in difficulties in the estimation of high-dimensional data.

On the other hand, $L_1$ regularization procedures, such as the graphical lasso (Friedman et al., 2008), are fast and stable even if the number of variables is large. A few researchers have proposed methods for estimating scale-free networks via $L_1$ regularization. Liu and Ihler (2011) introduced a log penalty, which is viewed as a prior distribution for the scale-free network. Defazio and Caetano (2012) introduced submodular priors on the degree distribution of the networks, and applied a Lovász extension to obtain a convex relaxation. In the two approaches, only Liu and Ihler’s (2011) procedure can be directly applied to both the MB and the penalized maximum likelihood procedures. Defazio and Caetano’s (2012) procedure is based only on the penalized maximum likelihood procedure. Therefore, we focus attention on Liu and Ihler’s (2011) method to construct a general procedure.

In practical situations, Liu and Ihler’s (2011) method has two different difficulties. The first issue is that we do not know the degree distribution of the true network. If the true degree distribution follows a binomial distribution, the ordinary lasso may perform better than a log penalty. The second issue is that even if the true network is scale-free, their penalty may not estimate the scale-free network when the values of nonzero-parameters vary greatly, because their prior distribution is valid only when all of the nonzero elements have equal values. In this case, a nonconvex penalty such as an $L_0$ penalty performs better than the lasso.

In order to handle these issues, this paper introduces a new class of penalty that is based on the exponentiation of the minimax concave penalty (MCP; Zhang, 2010). The proposed penalty includes two different features. First, the proposed penalty includes the ordinary lasso and the log penalty Liu and Ihler (2011), and the gap between these two penalties is bridged by a tuning parameter. Second, the MCP, which bridges the gap between the $L_1$ and $L_0$ penalization, allows us to estimate a scale-free network even if some of the nonzero values vary greatly. We apply cross-validation to select an appropriate value of the tuning parameter, which allows us to select an optimal penalty among the lasso, MCP, log penalty, and somewhere in between. We conduct Monte Carlo simulations to investigate the effectiveness of the proposed procedure. The result shows that the proposed method can perform better than the existing log penalty and the ordinary lasso.
2. Estimating scale-free networks

2.1. Sparse Gaussian graphical modeling

Let $X = (X_1, \ldots, X_p)^T$ be a $p$-dimensional multivariate-normal distributed random variable with mean vector $0$ and variance-covariance matrix $\Sigma$. A nonzero element of the inverse covariance matrix $\Omega = \Sigma^{-1}$ corresponds to conditional dependence given other variables. We review two popular methods for estimating sparse inverse covariance matrices. The first method is the neighborhood selection proposed by Meinshausen and Bühlmann (2006). Under a Gaussian assumption, each variable is written as $X_i = \sum_{j \neq i} \beta_{ij} X_j + \delta_i$, where $\beta_{ij} = -\frac{\omega_{ij}}{\omega_{ii}}$ and $\delta_i \sim N(0, 1/\omega_{ii})$, so that $B = (\beta_{ij})$ and $\Omega$ share the same nonzero pattern. For this reason, Meinshausen and Bühlmann (2006) propose to conduct a lasso regression to each variable with all other variables being predictors. Suppose that we have $n$ random observations $x_1, \ldots, x_n$ from the $p$-dimensional normal population $N(0, \Sigma)$. Let $X_n$ be the $n \times p$ data matrix $X_n = (x_1, \ldots, x_n)^T$. Define $x_{(i)}^T$ as the $i$th row vector of $X_n$. In this case, we can write $X_n = (x_{(1)}, \ldots, x_{(p)})$. The following penalized squared loss function is minimized:

$$\frac{1}{2} \| x_{(i)} - \sum_{j \neq i} \beta_{ij} x_{(j)} \|^2 + P(\beta_{-i}),$$

where $\beta_{-i} = (\beta_{i1}, \ldots, \beta_{i,i-1}, \beta_{i,i+1}, \ldots, \beta_{ip})^T$ is the $(p-1)$-dimensional regression coefficient vector, and $P(\cdot)$ is the penalty term which produces the sparse solution. For example, $P(\beta_{-i}) = \lambda \sum_{j \neq i} |\beta_{ij}|$ is the lasso penalty.

In practice, it is possible that $\hat{\beta}_{ij} = 0$ but $\hat{\beta}_{ji} \neq 0$ for $i \neq j$, so that we usually use the following “AND rule” or “OR rule” to estimate the network:

**AND rule:** If $\hat{\beta}_{ij} \neq 0$ and $\hat{\beta}_{ji} \neq 0$, the nodes between the $i$th variable and $j$th variable are connected.

**OR rule:** If $\hat{\beta}_{ij} \neq 0$ or $\hat{\beta}_{ji} \neq 0$, the nodes between the $i$th variable and $j$th variable are connected.

In the second approach, we directly maximize the following $L_1$ penalized log-likelihood function

$$\log |\Omega| - \text{tr}(\Omega S) - P(\Omega),$$

where $S$ is the sample variance-covariance matrix. Here the first two terms correspond to the log-likelihood function, and the third term, $P(\Omega)$, is a penalty term. Yuan and Lin, (2007) proposed to use the lasso penalty in order to make some of the off-diagonal elements of $\Omega$ exactly zero

$$P(\Omega) = \lambda \|\Omega - \text{diag}(\Omega)\|_1.$$
The problem of Eq. (2) using the lasso penalty given by (3) is referred to as the graphical lasso. As mentioned in Zhao et al. (2012), although the graphical lasso ensures the symmetry of the inverse covariance matrix, it sometimes requires a large computational load compared with the MB.

Both the MB in (1) and the graphical lasso in (2) can be written in the following form:

$$\ell(\Theta) - P(\Theta),$$

where $$\Theta = (\theta_{ij})$$ is a $$p \times p$$ parameter matrix; $$\Theta = B$$ for the MB and $$\Theta = \Omega$$ for the graphical lasso. Here $$\ell(\Theta)$$ is the loss function, and $$P(\Theta)$$ is the penalty term. Hereinafter we use the general form given by (4) in order to construct a methodology that can be applied to both the MB and the penalized maximum likelihood procedure.

2.2. Estimating scale-free networks via the exponentiation of the minimax concave penalty (MCP)

In the Bayesian framework, the lasso penalty corresponds to an independent double-exponential prior on each parameter

$$\pi(\Theta) \propto \prod_{i=1}^{p} \exp(-\lambda ||\theta_{-i}||_1) = \prod_{i=1}^{p} \prod_{j \neq i} \exp(-\lambda |\theta_{ij}|),$$

where $$\theta_{-i}$$ is the $$(p-1)$$-dimensional vector $$(\theta_{i1}, \ldots, \theta_{i,i-1}, \theta_{i,i+1}, \ldots, \theta_{ip})^T$$. This prior distribution implicitly assumes that each pair of nodes is connected with equal probability, so that the degree distribution becomes binomial. In many real-world networks, however, the degree distribution follows the power-law distribution (Barabási and Albert, 1999). Because the number of edges for node $$i$$ is given by $$||\theta_{-i}||_0$$, the prior distribution for a power-law distribution would be

$$\pi(\Theta) \propto \prod_{i=1}^{p} (||\theta_{-i}||_0)^{-\lambda}.$$ 

The penalty term is then the logarithm of the prior distribution

$$P(\Theta) = \lambda \sum_{i=1}^{p} \log (||\theta_{-i}||_0).$$

The penalty in (5) has two problems. The first problem is that if $$\theta_{-i} = 0$$, the penalty becomes minus infinity. The second problem is that it is difficult to obtain a numerical solution with the non-convex penalty $$||\cdot||_0$$. Liu and Ihler (2011) proposed the following penalty term that handles the above issues:

$$P(\Theta) = \lambda \sum_{i=1}^{p} \log (||\theta_{-i}||_1 + a_i),$$

where $$a_i$$ is a positive value. However, there are two issues using the penalty in (6) as follows:

1. In practical situations, we do not know the degree distribution of the true network. If the true degree distribution follows a binomial distribution, the ordinary lasso may perform better than a log penalty in (6). Furthermore, as shown in our simulation study presented in Section 4.1.1, even if the true degree distribution follows the power-law distribution, a penalty that lies somewhere in between the lasso and the log penalty can perform better than the log penalty when the MB method (OR rule) is used.
2. Even if we know that the true degree distribution follows a power law, the penalty in (6) may not approximate the true network when some of the nonzero-parameters vary greatly, because (6) uses the $L_1$ norm instead of $L_0$ norm.

In order to handle the above issues, we extend the penalty in (6) as follows:

$$P(\Theta) = \lambda \alpha^{-1} C^{1-\alpha} \sum_{i=1}^{p} \left( \eta \sum_{j \neq i} p_{\eta, \gamma}(|\theta_{ij}|) + C \right)^{\alpha} - C^{\alpha},$$

(7)

where $\alpha \in (0, 1)$ is a tuning parameter between the MCP and log penalty, $C = (1 - \alpha)/\beta$ with $\beta > 0$, and $p_{\eta, \gamma}(|\theta_{ij}|)$ is the minimax concave penalty (MCP; Zhang, 2010)

$$\eta p_{\eta, \gamma}(|\theta|) = \eta \int_0^{|\theta|} \left( 1 - \frac{x}{\eta \gamma} \right)^+ dx$$

$$= \eta \left( |\theta| - \frac{\theta^2}{2\eta \gamma} \right) I(|\theta| < \eta \gamma) + \frac{\eta^2 \gamma}{2} I(|\theta| \geq \eta \gamma).$$

(8)

Here $\eta > 0$ and $\gamma > 0$ are tuning parameters in MCP. The MCP bridges the $L_1$ and $L_0$ penalizations, and the gap between $L_1$ and $L_0$ penalizations is bridged by the tuning parameter $\gamma$. When the MCP approaches $L_0$ penalty, the MCP is able to estimate the scale-free network even if some of the nonzero values vary greatly, because the $L_0$ penalty does not depend on the magnitude of nonzero values.

The proposed penalty includes 4 tuning parameters $\alpha, \beta, \gamma, \eta$. We describe the role of each.

- The tuning parameter $\alpha$ bridges the gap between the MCP and log penalty. When $\alpha \to 1$, the penalty term becomes ordinary MCP, i.e., $P(\Theta) = \lambda \sum_{i=1}^{p} \left( \eta \sum_{j \neq i} p_{\eta, \gamma}(|\theta_{ij}|) \right)$. On the other hand, when $\alpha \to 0$, we obtain the log penalty expressed as $P(\Theta) = \lambda \beta^{-1} \sum_{i=1}^{p} \left( \log(\eta \sum_{j \neq i} p_{\eta, \gamma}(|\theta_{ij}|) + \beta^{-1}) - \log(\beta^{-1}) \right)$ (the proof is given in Appendix A).

- The value of $\gamma$ controls the degrees of sparsity: if $\gamma \to \infty$, we have $\eta p_{\eta, \gamma}(|\theta|) = \eta |\theta|$, so that the MCP becomes the lasso. On the other hand, when $\gamma \to +0$, the MCP becomes an $L_0$ penalty.

- $\eta$ is a tuning parameter of the MCP. Typically $\eta = 1$.

- $\beta$ is a tuning parameter of the log penalty. Typically $\beta = 1$.

As a result, the penalty in (7) includes the ordinary lasso in (3), log penalty in (6), and $L_0$ penalty in (5).

**Remark 1** In this modeling procedure, the tuning parameter $\alpha$, along with $\lambda$, must be selected based on a given set of data. These parameters cannot be selected by the maximum likelihood procedure, because it estimates the most complex model (i.e., $\lambda = 0$). In the lasso-type penalization procedure, a commonly-used tuning parameter selection procedure is cross-validation (Friedman et al., 2010). Thus, we apply cross-validation to select the tuning parameters. Other tuning parameter selection procedures such as the empirical Bayes method should be considered but are beyond the scope of this paper. We would like to take this as a future research topic.
3. Algorithm

In this section, we describe a parameter estimation algorithm for maximizing the function in (4) using the proposed penalty in (7). It is difficult to directly deal with the penalty term, because it includes the exponential function. Thus, we apply the Minorize-Maximization algorithm (MM algorithm; Hunter and Lange, 2004) to obtain the weighted penalized likelihood function based on the MCP.

First, for given \(x_0 \geq 0\), let \(f(x)\) be

\[
f(x) = \frac{1}{(x_0 + C)^{1-\alpha}}(x - x_0) - \frac{1}{\alpha}(x + C)^\alpha - (x_0 + C)^\alpha, \quad x \geq 0.
\]

Because \(f(x_0) = f'(x_0) = 0\), \(f'(x) < 0\) for \(x < x_0\), and \(f'(x) > 0\) for \(x > x_0\), we obtain \(f(x) \geq 0\) for any \(x \geq 0\). Thus, we have the following basic inequality:

\[
\frac{1}{\alpha}(x + C)^\alpha - (x_0 + C)^\alpha \leq \frac{1}{(x_0 + C)^{1-\alpha}}(x - x_0), \quad x \geq 0, \ x_0 \geq 0. \tag{9}
\]

Let \(\Theta^{(t)}\) be the current value of \(\Theta\). From (9), we have

\[
P(\Theta) - P(\Theta^{(t)}) \leq \lambda C^{1-\alpha} \sum_{i=1}^{p} \frac{1}{\{\eta \sum_{j \neq i} p_{\eta,\gamma}(|\theta_{ij}|) + C\}^{1-\alpha}} \left\{\eta \sum_{j \neq i} p_{\eta,\gamma}(|\theta_{ij}|) - \eta \sum_{j \neq i} p_{\eta,\gamma}(|\theta_{ij}^{(t)}|)\right\}
\]

\[
= \sum_{i=1}^{p} \sum_{j \neq i} \lambda_{ij}^{(t)} \left\{\eta p_{\eta,\gamma}(|\theta_{ij}|)\right\} + \text{const.,}
\]

where

\[
\lambda_{ij}^{(t)} = \lambda \left(\frac{C}{\eta \sum_{i' \neq i} p_{\eta,\gamma}(|\theta_{i'j}|) + C}\right)^{1-\alpha}. \tag{10}
\]

Then, we obtain

\[
\{\ell(\Theta) - P(\Theta)\} - \{\ell(\Theta^{(t)}) - P(\Theta^{(t)})\} \geq \ell(\Theta) - \sum_{i=1}^{p} \sum_{j \neq i} \lambda_{ij}^{(t)} \left\{\eta p_{\eta,\gamma}(|\theta_{ij}|)\right\} + \text{const.} \tag{11}
\]

The equality in (11) holds if \(\Theta = \Theta^{(t)}\). Thus, the objective function increases monotonically (i.e., \(\ell(\Theta^{(t+1)}) - P(\Theta^{(t+1)}) \geq \ell(\Theta^{(t)}) - P(\Theta^{(t)})\)) by updating \(\Theta\) as follows:

\[
\Theta^{(t+1)} = \arg\max_{\Theta} \left[\ell(\Theta) - \sum_{i=1}^{p} \sum_{j \neq i} \lambda_{ij}^{(t)} \left\{\eta p_{\eta,\gamma}(|\theta_{ij}|)\right\}\right]. \tag{12}
\]

The optimization of (12) is given as follows: in the MB (i.e., the regression via the lasso-type penalization is conducted separately), we can use the sparsenet algorithm (Mazumder et al., 2011) to obtain the nonconcave penalized least squares estimates. In the penalized maximum likelihood procedure, it is difficult to directly apply the standard graphical lasso algorithm (Friedman et al., 2008). Therefore, the alternating direction method of multipliers algorithm (ADMM algorithm; Boyd et al., 2011) is applied. The detail of the ADMM algorithm for our penalty is provided in Appendix B. We have not shown any convergence properties for our algorithm. We would like to take this as a future research topic.
Remark 2 Liu and Ihler (2011) also applied the MM algorithm, and our algorithm generalizes Liu and Ihler’s. In fact, when $\alpha = 0$, the weight in (10) coincides with Liu and Ihler’s weight.

4. Monte Carlo simulation

Monte Carlo simulation was conducted to investigate the effectiveness of the proposed penalty. This simulation study consisted of two different parts. In the first simulation study, we compared the performance among various $\alpha$. In the second simulation study, we investigated whether the MCP can yield good performance if the values of nonzero-parameters vary greatly.

4.1. Comparison of performance among various $\alpha$

In the first simulation study, we investigated the performance among the lasso, the log penalty, and somewhere in between. Because the gap between the lasso and the log penalty is controlled by $\alpha$, we change only $\alpha$ ($\alpha = 0.0, 0.2, 0.5, 0.8, 1.0$) and other tuning parameters are fixed by $\beta = 1, \eta = 1,$ and $\gamma = \infty$.

The networks were generated according to the scale-free network and Erdős-Rényi model (Erdős and Rényi, 1960) as follows:

**Scale-free network** We first generated $p = 1000$ i.i.d. sample $D_1, \ldots, D_p$ according to the power-law distribution $p(k) \propto k^{-s}$ with $s = 2.4$. Then, $M = 2000$ edges between node $i$ and node $j$ were randomly connected with the probability $D_i D_j / \sum_{i', j'} D_{i'} D_{j'}$. Finally, several edges corresponding to the isolated vertices were randomly added, resulting in $M = 2185$ edges. Note that it is not necessary to assume that the minimum degree of the graph is one. However, if isolated nodes exist, it is important to investigate the cluster of variables (Tan et al., 2013), which is beyond the scope of this research. Thus, we eliminated such isolated nodes.

This network corresponds to a static model of scale-free networks (Goh et al., 2001), in which the number of nodes $p$ is considered to be constant.

**Erdős-Rényi model** In the Erdős-Rényi model (Erdős and Rényi, 1960), all pairs of two nodes are connected with equal probability. The number of edges was $M = 2000$. As is the case with the scale-free network, we added 20 more edges to eliminate the isolated vertices.

Let $A$ be a graph generated in the two previous steps of making scale-free networks and Erdős-Rényi models. An inverse covariance matrix $\Omega$ was then generated using the $p \times p$ adjacency matrix $A$ as follows: we first created a matrix $E = (E_{ij})$

$$
E_{ij} = \begin{cases} 
U[D] & \text{if } A_{ij} = 1 \\
0 & \text{otherwise} 
\end{cases}
$$

where $U[D]$ is a random sample from a uniform distribution with $D = [-0.75, -0.25] \cup [0.25, 0.75]$. We calculated $\bar{E} := (E + E^T)/2$ and set $\Omega = \bar{E} + (\delta - \lambda_{\text{min}})I$ with $\delta = 0.1$, where $\lambda_{\text{min}}$ is the smallest eigenvalue of $\bar{E}$ and $I$ is the identity matrix. Then, we set $\Omega = L^{1/2} \Omega L^{1/2}$, where $L = \text{diag}(\Omega^{-1})$.

Based on the above covariance matrix, we generated 10 datasets with $n = 2000$, and applied the MB and the penalized maximum likelihood procedure based on the proposed penalty. For the scale-free network, the log penalty (i.e., $\alpha = 0.0$) would be desirable. On
the other hand, for the Erdős-Reényi graph, the ordinary lasso ($\alpha = 1.0$) may perform better than the log penalty.

4.1.1. Comparison of true positive rate

We first compared the performance among various $\alpha$. Figure 1 shows the true positive rate (TPR) for each $\alpha$ averaged over 10 datasets. TPR indicates the proportion of cases where nonzero elements correctly set to nonzero. For each $\alpha$, we chose $\lambda$ so that the estimated number of edges is equal to 1500, $M_0$, and 3000. Here $M_0$ is the true number of edges; $M_0 = 2185$ for scale-free networks and $M_0 = 2020$ for Erdős-Reényi model. From Figure 1, we observe that

- For both scale-free and Erdős-Rényi graphs, the results were essentially independent of the number of edges.
- The penalized maximum likelihood procedure resulted in a satisfactory result, i.e., $\alpha = 0.0$ (log penalty) performed the best for the scale-free network, and $\alpha = 1.0$ (lasso) performed the best for the Erdős-Reényi graph.
- For the MB method, the OR rule performed better than the AND rule on scale-free networks.

Fig. 1: True positive rate (TPR) for $\alpha = 0.0, 0.2, 0.5, 0.8, 1.0$ averaged over 10 datasets. The dashed line of the “PML” indicates the penalized maximum likelihood procedure. The value of $\lambda$ was chosen so that the estimated number of edges is equal to 1500, $M_0$, and 3000. Here $M_0$ is the true number of edges; $M_0 = 2185$ for scale-free networks and $M_0 = 2020$ for Erdős-Reényi model.
Estimating Scale-free Networks

- The AND rule and the OR rule yielded completely different results. By definition of the AND rule and OR rule, the graph estimated by the AND rule is identical to that estimated by the OR rule only when \( B \) is symmetric. Therefore, the estimated regression coefficient matrix \( B \) obtained by the MB was far from symmetric.

- The result of OR rule was not largely affected by \( \alpha \), resulting in stable estimates for all \( \alpha \).

- For the MB, the change of TPR with respect to \( \alpha \) was not what we have expected. For example, the ordinary lasso somehow performed better than the log penalty for OR rule when the true network was scale-free.

We investigated the unreasonable change of TPR with respect to \( \alpha \) of the MB for the scale-free network. First, we computed the true degrees for all nodes \( d_1, \ldots, d_p \), and arranged them in a descending order, say, \( d_{i_1}, \ldots, d_{i_p} \). Then, we picked one of the 10 datasets, and counted the number of edges for nodes \( i_1, \ldots, i_p \). Figure 2 depicts the number of edges for

---

Fig. 2: The number of edges for nodes \( i_1, \ldots, i_{20} \) when \( \alpha = 0.0 \) (log penalty) and \( \alpha = 0.5 \). Both AND rule (left side) and OR rule (right side) are displayed.
nodes $i_1, \ldots, i_{20}$ when $\alpha = 0.0$ (log penalty) and $\alpha = 0.5$. The value of $\lambda$ was chosen so that the estimated number of edges was approximately equal to the true number of edges. The OR rule selected too many edges for $i_2, i_3,$ and $i_4$ when $\alpha = 0.0$, but the AND rule did not. Thus, the $i_2, i_3$ and $i_4$th rows of the estimated coefficient matrix $B$ were extremely dense but the corresponding columns were sparse (for example, the $i_4$th row of $B$ had 511 nonzero values but the $i_4$th column had only 15 nonzero values). As a result, it is better to use $\alpha = 0.5$ rather than $\alpha = 0.0$ for OR rule, even if the true network is scale-free.

The result of Fig. 1 shows that the MB with the OR rule seems to be worse than the penalized maximum likelihood procedure for the scale-free network when $\alpha = 0.0$. However, if an appropriate value of $\alpha$ is selected, both the penalized maximum likelihood procedure and the MB (OR rule) yielded similar TPR.

In general, it is important to investigate the false positive and negative rates. However, in this case, the selected number of edges was fixed for all $\alpha$. Therefore, if the TPR is large, the FPR becomes small.

### 4.1.2. Selection of $\alpha$ by cross-validation

In practical situations, the values of $\alpha$ and $\lambda$ are unknown. In order to choose these tuning parameters, we applied 10-fold cross-validation (CV) to the MB. In the 10-fold cross-validation procedure, we split the data into 10 parts, $X^{(1)}, \ldots, X^{(10)}$, that consist of almost equal-size sample sizes. Let $X^{(-k)} = (X^{(1)}, \ldots, X^{(k-1)}, X^{(k+1)}, \ldots, X^{(10)})$. The loss functions constructed by $X^{(k)}$ and $X^{(-k)}$ are denoted by $\ell^{(k)}(\Theta)$ and $\ell^{(-k)}(\Theta)$, respectively. Then, we estimate the parameter by $\Theta^{(k)} := \arg\min_\Theta -\ell^{(-k)}(\Theta) + P(\Theta)$ for $k = 1, \ldots, 10$, and minimize the following CV value with respect to $\alpha$ and $\lambda$:

$$CV = -\frac{1}{10} \sum_{k=1}^{10} \ell^{(k)}(\Theta^{(k)}). \quad (13)$$

We did not apply the 10-fold cross validation to the penalized maximum likelihood procedure, because this procedure has a large number of parameters (in this simulation, we have $p(p+1)/2 \approx 500,000$ parameters) and then requires heavy computational load (e.g., Zhao et al., 2012). Figure 3 shows the value of the CV, and Figure 4 shows the true positive
Fig. 4: The value of F-measure, true positive (TP) and the false positive (FP) for each α. The left column correspond to the scale-free network, and the right column corresponds to the Erdős-Reényi graph.

(TP), the false positive (FP), and the F-measure for each α. Here the number of positive examples which are predicted correctly is denoted by TP (true positive). The F-measure is defined by

\[ \text{F-measure} = \frac{2TP}{2TP + FN + FP} \]

For the scale-free network, the CV value was minimized at \( \alpha = 0.8 \), in which the TP on
AND rule attained the maximum value. The FP increases as \( \alpha \) increases. At least the true positive on \( \alpha = 0.5 \) and \( \alpha = 0.8 \) was larger than that on \( \alpha = 1.0 \). Therefore, the 10-fold cross validation did not select an inappropriate value of tuning parameter.

For the Erdős-Rényi graph, the CV value was minimized at \( \alpha = 0.8 \). The TP on \( \alpha = 0.8 \) was better than that on \( \alpha = 0.0, 0.2, 0.5 \), but worse than that on \( \alpha = 1.0 \). However, the FP on \( \alpha = 1.0 \) was relatively larger than that on \( \alpha = 0.8 \). Therefore, the tuning parameter selected by the 10-fold cross validation seems to be reasonable.

On the both scale-free networks and Erdős-Rényi graph, the minimum value of CV did not yield a large F-measure. However, this result does not mean that the cross-validation performed poorly; i.e., the change in false positive due to \( \alpha \) was much larger than that in true positive and false negative, which indicates that the F-measure was strongly affected by the false positive.

### 4.2. Performance of the minimax concave penalty (MCP)

If all of the nonzero components have the same values, we have \( \| \theta_i \|_1 = c \cdot \| \theta_i \|_0 \). In this case, Liu and Ihler’s (2011) penalty becomes the log penalty based on the \( L_0 \) norm given in (5), so that their procedure may perform well. However, if the nonzero components of the \( \Theta \) have different values, Liu and Ihler’s (2011) may not perform well. In such cases, our penalty based on MCP may perform better than Liu and Ihler’s (2011) penalty, because the MCP approaches an \( L_0 \) penalty as \( \gamma \) decreases. In order to investigate this, we generated the following two types of inverse covariance matrix \( \Omega \).

**Type (A):** Same as the generation of inverse covariance matrix presented by Section 4.1., but with \( M = 400 \) and \( \delta = 0.01 \) instead of 0.1. Note that arbitrary \( \delta > 0 \) ensures the positive definiteness of \( \Omega \), but if \( \delta \) is small, the nonzero elements of \( \Omega \) vary greatly.

**Type (B):** We set \( \Omega = I + 0.05A \), in which the nonzero components of \( \Omega \) have the same value. Here, \( A \) is an adjacency matrix.

We created a scale-free network with \( s = 2.1 \) and \( p = 100 \), and generated \( n = 500 \) observations based on this network. The parameters \( \beta \) and \( \eta \) were fixed by \( \beta = 1.0 \) and \( \eta = 1.0 \). Figure 5 shows a receiver operating characteristic (ROC) curve averaged over 10 datasets for \((\alpha, \gamma) = (1.0, 1.5), (\alpha, \gamma) = (1.0, \infty), \text{ and } (\alpha, \gamma) = (0.0, \infty)\). We also applied the ordinary lasso (i.e., \( \alpha = 0.0 \) and \( \gamma = \infty \)). The \( x \) axis shows the number of edges of the graph derived from \( \Omega \), and the \( y \) axis shows the TPR. The “logMC” corresponds to our proposed penalty, “log” is the Liu and Ihler’s penalty, and the lasso corresponds to the ordinary lasso. The vertical line corresponds to the true number of edges. The result shows that

- Because the true network was scale-free, the log penalty performed better than the ordinary lasso.

- When we used the inverse covariance matrix of Type (A) (i.e., the nonzero elements vary greatly), our penalty performed better than Liu and Ihler’s penalty. On the other hand, in Type (B), our method and Liu and Ihler’s method yielded similar results.
Exponentially Growing Network

Fig. 5: Receiver operating characteristic (ROC) curve averaged over 10 datasets. The x axis shows the number of edges of the graph derived from \( \Omega \), and the y axis shows the TPR. “logMC”, “log”, and “lasso” correspond to \((\alpha, \gamma) = (1.0, 1.5)\), \((\alpha, \gamma) = (1.0, \infty)\), and \((\alpha, \gamma) = (0.0, \infty)\), respectively. The vertical dashed line corresponds to the true number of edges.

5. Conclusion and Discussion

We have introduced a new class of penalty that includes the ordinary lasso and the log-penalty. The value of \( \alpha \) bridges a gap between the lasso and the log penalty, and is selected by CV. However, the value of \( \beta \) also bridges the gap between these two penalties. As a future research topic, it is interesting to compare the performance of our penalty by changing the value of \( \beta \).

Although our nonconvex penalty can be applied when the value of the non-zero elements varies, we observe that the performance of our procedure tends to be quite similar to that of Liu and Ihler’s procedure when the number of variables is large. Another interesting research topic is to investigate whether the nonconvex penalty can be used for high-dimensional data on the penalized maximum likelihood procedure.

Acknowledgement

The authors would like to thank Mr. Thong Pham for helpful discussions about the MM algorithm. We also thank two anonymous reviewers for the constructive and helpful comments that improved the quality of the paper considerably. This work was supported by grants from Japan Society for the Promotion of Science KAKENHI (15K15949 to Kei Hirose, and 26120523 and 24300106 to Hidetoshi Shimodaira).

Appendix A. Proof that our penalty becomes log penalty when \( \alpha \to 0 \)

Let

\[
    h(\alpha) = \alpha^{-1} C^{1-\alpha} \left\{ (r + C)^\alpha - C^\alpha \right\},
\]

— 151 —
HIROSE, OGURA and SHIMODAIRA

where $C = (1 - \alpha)/\beta$, $r$ is a constant which does not depend on $\alpha$. Then, it is sufficient to show that

$$\lim_{\alpha \to 0} h(\alpha) = \frac{1}{\beta} \left\{ \log \left( r + \frac{1}{\beta} \right) - \log \left( \frac{1}{\beta} \right) \right\}. \quad (14)$$

Let $\tilde{h}(\alpha) = \alpha h(\alpha)$. The differential of $\tilde{h}(\alpha)$ is

$$\tilde{h}'(\alpha) = C^{1-\alpha} \left\{ (C + r)^{\alpha} - C^{\alpha} \right\} (-\log C - 1) + C^{1-\alpha} \left[ (C + r)^{\alpha} \left\{ \log (C + r) - \frac{\alpha}{\beta (C + r)} \right\} - C^{\alpha} \left( \log C - \frac{\alpha}{1 - \alpha} \right) \right].$$

Thus, by L'Hôpital's rule, we obtain (14).

Appendix B. ADMM algorithm

In the ADMM algorithm, we prepare a new $p \times p$ matrix $\Omega_0$, and rewrite the problem of (12) as follows:

$$\max \Omega \left[ \ell(\Omega) - \sum_{i=1}^p \sum_{j \neq i} \lambda_{ij}^{(l)} \{ \eta p_{i,j} (|\omega_{ij})_0 |) \right] \text{ subject to } \Omega = \Omega_0, \quad (15)$$

where $(\omega_{ij})_0$ is the $(i,j)$th element of $\Omega_0$. The scaled augmented Lagrangian for (15) takes the form

$$L_\rho(\Omega, \Omega_0, \Lambda) = -\ell(\Omega) + \sum_{i=1}^p \sum_{j \neq i} \lambda_{ij}^{(l)} \{ \eta p_{i,j} (|\omega_{ij})_0 |) \} + \text{tr}(\Lambda^T (\Omega - \Omega_0)) + \frac{\rho}{2} \|\Omega - \Omega_0\|_F^2. \quad (16)$$

In the ADMM algorithm, the parameter is updated as follows (Boyd et al., 2011)

$$\Omega^{(k+1)} = \arg\min_{\Omega} \left\{ -\ell(\Omega) + \frac{\rho}{2} \|\Omega - \Omega_0^{(k)} + U^{(k)}\|_F^2 \right\}, \quad (17)$$

$$\Omega_0^{(k+1)} = \arg\min_{\Omega_0} \left\{ \sum_{i=1}^p \sum_{j \neq i} \lambda_{ij}^{(l)} \{ \eta p_{i,j} (|\omega_{ij})_0 |) \} + \frac{\rho}{2} \|\Omega^{(k+1)} - \Omega_0 - U^{(k)}\|_F^2 \right\}, \quad (18)$$

$$U^{(k+1)} = U^{(k)} + \Omega^{(k+1)} - \Omega_0^{(k+1)}.$$

First we solve the problem (17). By differentiating the augmented Lagrangian $L_\rho(\Omega, \Omega_0, \Lambda)$ in (16) with respect to $\Omega$, we have

$$\rho \Omega - \Omega^{-1} = \rho(\Omega_0^{(k)} - U^{(k)}) - S. \quad (19)$$

In order to solve the problem of (19), we take the orthogonal eigen-value decomposition of the righthand side of (19)

$$\rho(\Omega_0^{(k)} - U^{(k)}) - S = P \Lambda P^T,$$

where $P$ is a orthogonal matrix, and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p)$ with $\lambda_i$ being the eigenvalues. Assuming $\tilde{\Omega} = P \Omega P^T$, we have

$$\rho \tilde{\Omega} - \tilde{\Omega}^{-1} = \Lambda.$$
Since $\hat{\Omega}$ is a diagonal matrix, the solution of $\hat{\Omega}$ is given by

$$\hat{\Omega}_{ii} = \lambda_i + \sqrt{\lambda_i^2 + 2\rho}.$$ 

Then, we obtain the solution $\Omega = P\hat{\Omega}P^T$.

An element-wise solution of (18) can be obtained; we only need to solve the following problem:

$$S(\tilde{\theta}) = \arg\min_{\theta} \left\{ \frac{1}{2} (\theta - \tilde{\theta})^2 + \eta^* p_{\eta^*} (|\theta|) \right\}.$$ 

The solution $S(\tilde{\theta})$ can be expressed in a closed form for MCP if $\gamma^* > 1$. In fact, the problem is convex, so that we have thresholding function for MCP

$$S(\tilde{\theta}) = \begin{cases} 
\text{sgn}(\tilde{\theta})(|\tilde{\theta}| - \eta^*) + \frac{1}{1 - 1/\gamma^*} & \text{if } |\tilde{\theta}| \leq \eta^* \gamma^* \\
\tilde{\theta} & \text{if } |\tilde{\theta}| > \eta^* \gamma^*.
\end{cases}$$

If $\gamma^* \leq 1$, the problem becomes non-convex, so that we cannot easily derive an explicit formula. However, the solution is expressed as

$$S(\tilde{\theta}) \in \{0, \tilde{\theta}, \text{sign}(\tilde{\theta})\eta^* \gamma^*\},$$

so that we can easily compute the $S(\tilde{\theta})$ numerically.

**REFERENCES**

Banerjee, O., El Ghaoui, L., and d’Aspremont, A. (2008), Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data, *The Journal of Machine Learning Research*, 9, 485–516.

Barabási, A.-L., and Albert, R. (1999), Emergence of scaling in random networks, *Science*, 286(5439), 509–512.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends® in Machine Learning*, 3(1), 1–122.

Defazio, A., and Caetano, T. S. (2012), A convex formulation for learning scale-free networks via submodular relaxation, in *Advances in Neural Information Processing Systems*, pp. 1250–1258.

Del Genio, C. I., Gross, T., and Bassler, K. E. (2011), All scale-free networks are sparse, *Physical review letters*, 107(17), 178701.

Edwards, D. (2000), *Introduction to graphical modelling*, New York: Springer.

Erdös, P., and Rényi, A. (1960), On the evolution of random graphs, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5, 17–61.

Friedman, J., Hastie, T., and Tibshirani, R. (2008), Sparse inverse covariance estimation with the graphical lasso, *Biostatistics*, 9(3), 432–441.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software*, 33.

Goh, K.-I., Kahng, B., and Kim, D. (2001), Universal behavior of load distribution in scale-free networks, *Physical Review Letters*, 87(27), 278701.

— 153 —
Hunter, D. R., and Lange, K. (2004), A tutorial on MM algorithms, *The American Statistician*, 58(1), 30–37.

Liu, Q., and Ihler, A. T. (2011), Learning scale free networks by reweighted l1 regularization, in *International Conference on Artificial Intelligence and Statistics*, pp. 40–48.

Maruyama, O., and Shikita, S. (2014), A scale-free structure prior for Bayesian inference of Gaussian graphical models, in *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*, IEEE, pp. 131–138.

Mazunder, R., Friedman, J., and Hastie, T. (2011), SparseNet: Coordinate Descent with Nonconvex Penalties, *Journal of the American Statistical Association*, 106, 1125–1138.

Meinshausen, N., and Bühlmann, P. (2006), High-dimensional graphs and variable selection with the lasso, *The Annals of Statistics*, pp. 1436–1462.

Peng, J., Wang, P., Zhou, N., and Zhu, J. (2009), Partial correlation estimation by joint sparse regression models, *Journal of the American Statistical Association*, 104(486).

Rothman, A. J., Bickel, P. J., Levina, E., Zhu, J. et al. (2008), Sparse permutation invariant covariance estimation, *Electronic Journal of Statistics*, 2, 494–515.

Sheridan, P., Kamimura, T., and Shimodaira, H. (2010), A scale-free structure prior for graphical models with applications in functional genomics, *PLoS One*, 5(11), e13580.

Tan, K. M., Witten, D., and Shojaie, A. (2013), The cluster graphical lasso for improved estimation of Gaussian graphical models, *Computational statistics and data analysis*, 85, 23–36.

Tibshirani, R. (1996), Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society, Ser. B*, 58, 267–288.

Witten, D. M., Friedman, J. H., and Simon, N. (2011), New insights and faster computations for the graphical lasso, *Journal of Computational and Graphical Statistics*, 20(4), 892–900.

Yuan, M., and Lin, Y. (2007), Model selection and estimation in the Gaussian graphical model, *Biometrika*, 94(1), 19–35.

Zhang, C. (2010), Nearly Unbiased Variable Selection Under Minimax Concave Penalty, *The Annals of Statistics*, 38, 894–942.

Zhao, T., Liu, H., Roeder, K., Lafferty, J., and Wasserman, L. (2012), The huge package for high-dimensional undirected graph estimation in R, *The Journal of Machine Learning Research*, 13(1), 1059–1062.

(Received: March 2, 2015, Accepted: June 13, 2015)