GRAPH LEARNING INFORMATION CRITERION

Koki Yamada[1] *and Yuichi Tanaka*[1,2]

[1]Department of EECS, Tokyo University of Agriculture and Technology, Tokyo, Japan
[2]PRESTO, Japan Science and Technology Agency, Saitama, Japan

ABSTRACT

In this paper, we propose a parameter selection method for graph learning. Graph learning, a technique of learning graphs from observations, is required in many applications, e.g., classification, prediction, and clustering. However, there is no established method to determine hyperparameters that control the strength of the regularization reflecting prior knowledge. To resolve the problem, we consider a model selection criterion for the graph learning problem based on Laplacian constrained Gaussian Markov random field. The proposed criterion is the value based on *model evidence*, which is used for model selection in Bayesian statistics. It can be estimated by averaging the negative log-likelihood over the posterior distribution of a graph learning model. To compute this criterion, we present an efficient sampler of the posterior distribution. In the experiment with random graphs, we demonstrate that the proposed method can select hyperparameters having a good trade-off between F-measure and relative error.

***Index Terms*—** Graph learning, network topology inference, model evidence, information criterion

1. INTRODUCTION

Signal processing and machine learning on graphs have become a popular tool to analyze signals and features taking into account underlying networks. They are also a hot research topic in various application fields [1, 2].

In many applications, we face a situation that the underlying network for the observed signals is unknown or we only have limited information on it. Examples of such situations are estimating brain functional connectivity from EEGs or fMRI [3], identification of biological networks, e.g., protein, RNA, and DNA [4], and inference of relationships among companies from historical stock price data [5], to name a few.

To tackle such a situation, we need to estimate or learn graphs from observations. Traditionally, simple graph estimation methods like k-nearest-neighbor have been widely utilized. However, such a simple graph estimation does not result in good performance, especially in the noisy case. Therefore, *graph learning*, techniques of learning graphs from observations, has been developed [6, 7]. The main motivation of graph learning is to provide structural information useful for analysis and to discover the relationships among observations.

Most studies on graph learning formulate their problem as convex optimization that minimizes some criteria (e.g., signal smoothness on graphs) [8–13]. Since their formulations generally contain regularization terms based on prior knowledge, we need to tune hyperparameters that control the strength of the regularization to obtain the desired graph. A typical example of the regularization used in graph learning is the ℓ_1 regularization, and its hyperparameter controls the edge sparsity of the learned graph [9]. However, we do not know how sparse the actual graph is. As a result, it is difficult to determine the optimal hyperparameters in real applications and we often decide the parameters in an ad hoc manner.

The question we consider in this paper is: *How should we choose the optimal hyperparameter(s) for graph learning?* Although the choice of hyperparameters strongly affects the performance of graph learning, the research on hyperparameter selection still remains insufficient. When graph learning is applied for supervised learning problems such as prediction and classification, the hyperparameters could be determined by cross-validation (but they do not have a theoretical guarantee in general). In contrast, for unsupervised learning such as clustering, there is few methods to determine hyperparameters.

Some studies of graph learning determine hyperparameters using Bayesian information criterion (BIC) [14, 15]. BIC is a criterion based on *model evidence* and is used for model selection among a finite set of models, i.e., a set of graphs learned under different parameters. However, BIC is only a rough approximation of model evidence and fails to choose a good model in graph learning (we later show it in the experiment in Section 4).

In this paper, we propose a new model selection criterion specifically designed for graph learning: *Graph learning information criterion* (GLIC). The strategy of our method is to introduce the Bayesian model equivalent to the formulation based on Laplacian constrained Gaussian Markov random field (LGMRF) model [9] and to compute the approximation of the model evidence of this Bayesian model using the result in [16]. Since this approximation computation requires sampling from the posterior distribution of the Bayesian model, we also present an efficient sampling method using block Gibbs sampling scheme. Experiments on random graphs demonstrate that the proposed method can successfully determine the hyperparameter and gives a good trade-off between F-measure (edge correctness) and relative error (edge weight correctness).

The remainder of this paper is organized as follows. We summarize graph learning based on LGMRF and BIC in Section 2. Section 3 presents GLIC with the approximation computation of the model evidence and the efficient sampling method from the posterior distribution. Experimental results on random graphs are provided in Section 4. Conclusions are given in Section 5.

Notation and Definitions: Lowercase normal, lowercase bold, and uppercase bold letters denote scalars, vectors, and matrices, respectively. Calligraphic capital letters denote sets. The Moore–Penrose pseudo inverse of $\mathbf{X}$ is denoted by $\mathbf{X}^\dagger$. $\mathrm{gdet}(\mathbf{X})$ is the generalized determinant of $\mathbf{X}$ [17].

$\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian distribution with the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$. The inverse Gaussian distribution is denoted by $\mathrm{IGau}(\bar{\mu}, \bar{\lambda})$, where $\bar{\mu}$ and $\bar{\lambda}$ are the mean and the shape parameter,

This work was supported in part by JST PRESTO under grant JPMJPR1935 and JSPS KAKENHI under grant 20H02145 and 20J13647.

respectively. $\mathrm{Gam}(\bar{\alpha}, \bar{\beta})$ represents the gamma distribution with a shape parameter $\bar{\alpha}$ and an inverse scale parameter $\bar{\beta}$. The uniform distribution in the interval $[x, y]$ is denoted by $\mathrm{U}(x, y)$.

A weighted graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges, respectively, and $\mathbf{W}$ is a weighted adjacency matrix. The number of nodes is given by $N = |\mathcal{V}|$. The graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix. The set of valid graph Laplacians is given by:

$$\mathcal{L} = \left\{ \mathbf{L} \in \mathbb{R}^{N \times N} : L_{ij} = L_{ji} \leq 0 \ (i \neq j), L_{ii} = -\sum_{i \neq j} L_{ij} \right\}. \tag{1}$$

2. GRAPH LEARNING AND BAYESIAN INFORMATION CRITERION

2.1. Graph Learning with LGMRF

Graph learning is a problem of learning graph Laplacian(s) from observations $\mathbf{X} \in \mathbb{R}^{N \times K} = [\mathbf{x}_1, \dots \mathbf{x}_K]$, where K is the number of observations. In this paper, we assume the following signal observation model based on Laplacian constrained Gaussian Markov random field (LGMRF) [8–10]:

$$p(\mathbf{x} \mid \mathbf{\Theta}) = \frac{1}{(2\pi)^{N/2} \left(\mathrm{gdet}(\mathbf{\Theta}^\dagger)\right)^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{\Theta} \mathbf{x}\right), \tag{2}$$

where $\mathbf{\Theta} \in \mathcal{L}$ is the precision matrix satisfying graph Laplacian constraints. The negative log-likelihood function $L(\mathbf{\Theta})$ of (2) is given by

$$\begin{aligned} L(\mathbf{\Theta}) &= -\log(\prod_{k=1}^{K} p\left(\mathbf{x}_k \mid \mathbf{\Theta}\right)) \\ &= \frac{1}{2}\sum_{k=1}^{K} \mathrm{Tr}\left(\mathbf{x}_k^\intercal \mathbf{\Theta} \mathbf{x}_k\right) - \frac{K}{2}\log \mathrm{gdet}(\mathbf{\Theta}). \end{aligned} \tag{3}$$

Furthermore, suppose that the prior distribution $p(\mathbf{\Theta})$ is the following Laplace distribution.

$$p(\mathbf{\Theta} \mid \lambda) = \prod_{i<j} p(\theta_{ij}) = \prod_{i<j} \frac{\lambda}{2} e^{-\lambda|\theta_{ij}|}, \tag{4}$$

where λ is a scale parameter of the Laplace distribution. The maximum a posteriori (MAP) estimation of $\mathbf{\Theta}$ with (3) and (4) leads to the following optimization problem [9]:

$$\underset{\mathbf{\Theta} \in \mathcal{L}}{\text{minimize}} \ \frac{1}{K}\mathrm{Tr}(\mathbf{\Theta}\mathbf{S}) - \log \mathrm{gdet}(\mathbf{\Theta}) + \alpha \|\mathbf{\Theta}\|_{1,\mathrm{off}}, \tag{5}$$

where $\mathbf{S} = \mathbf{X}\mathbf{X}^\intercal$, $\alpha = \lambda/K$, and $\|\mathbf{\Theta}\|_{1,\mathrm{off}}$ represents the absolute sum of off-diagonal elements in $\mathbf{\Theta}$. The optimization problem in (5) can be solved using the block coordinate descent algorithm [9].

In (5), the hyperparameter α controls the sparsity of the learned graph Laplacian $\mathbf{\Theta}$. If we do not have prior information on the graph sparsity, selecting the optimal α becomes a difficult problem.

2.2. Bayesian Information Criterion

In Bayesian statistics, model evidence is often used for model selection among a finite set of models [18]. The model evidence is a likelihood function in which some parameters are marginalized.

The model evidence of the graph learning model introduced in Section 2.1 is given by

$$p(\mathbf{X} \mid \lambda) = \int_{\mathbf{\Theta}} p(\mathbf{X} \mid \mathbf{\Theta}) p(\mathbf{\Theta} \mid \lambda) \mathrm{d}\mathbf{\Theta}. \tag{6}$$

Unfortunately, (6) cannot be analytically computed. Hence, the approximate computation of the model evidence is required.

Bayesian information criterion (BIC) is a criterion for the model evidence-based model selection [15]. BIC is an approximation of the negative logarithmic model evidence, which is computed using the Laplace method. It is defined as follows:

$$\mathrm{BIC} = L(\mathbf{\Theta}) + \frac{d}{2}\log K, \tag{7}$$

where $L(\cdot)$ is the negative log-likelihood function of the estimated parameter, d is the number of estimated parameters in the model, and K is the number of observations.

The BIC of the graph learning in (5) can be computed as follows:

$$\mathrm{BIC}(\alpha) = \frac{1}{2}(\mathrm{Tr}(\widehat{\mathbf{\Theta}}\mathbf{S}) - K \log \mathrm{gdet}(\widehat{\mathbf{\Theta}}) + \|\widehat{\mathbf{\Theta}}\|_0 \log K), \tag{8}$$

where $\widehat{\mathbf{\Theta}}$ is the solution of (5) with the parameter α and $\|\widehat{\mathbf{\Theta}}\|_0$ is the number of nonzero elements in $\widehat{\mathbf{\Theta}}$.

When selecting a model among a finite model set, the ones with small BIC are preferred. Thus, the hyperparameter α of (5) can be determined by the following steps: 1) Estimating $\mathbf{\Theta}$ with different α's; 2) Computing $\mathrm{BIC}(\alpha)$ of the estimated $\mathbf{\Theta}$; and 3) Selecting the one with the smallest BIC.

However, BIC is only a rough approximation of model evidence [16] and often fails to choose a good model for graph learning. We will show it later in Section 4.

3. GLIC

In this section, we present a computation method of GLIC: A model selection criterion for graph learning. The strategy of the proposed method is to estimate the negative logarithmic model evidence in a different way from BIC.

For GLIC, we use an approximation of the negative logarithmic model evidence based on WBIC [16]:

$$-\log p(\mathbf{X} \mid \lambda) \approx \frac{\int L(\mathbf{\Theta}) \prod_{k=1}^{K} p\left(\mathbf{x}_k \mid \mathbf{\Theta}\right)^\eta p(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta}}{\int \prod_{k=1}^{K} p\left(\mathbf{x}_k \mid \mathbf{\Theta}\right)^\eta p(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta}}, \tag{9}$$

where $\eta = 1/\log(K)$. This indicates that the negative logarithmic model evidence can be estimated by the expectation of $L(\mathbf{\Theta})$ in (3) over the posterior distribution $p(\mathbf{\Theta} \mid \mathbf{X}, \lambda, \eta)$ using Markov chain Monte Carlo (MCMC) method. Hence, (9) is rewritten as follows:

$$\begin{aligned} &-\log p(\mathbf{X} \mid \lambda) \approx \frac{1}{M}\sum_{m=1}^{M} L(\mathbf{\Theta}_m), \ \mathbf{\Theta}_m \sim p(\mathbf{\Theta} \mid \mathbf{X}, \lambda, \eta), \\ &p(\mathbf{\Theta} \mid \mathbf{X}, \lambda, \eta) = \frac{\prod_{k=1}^{K} p\left(\mathbf{x}_k \mid \mathbf{\Theta}\right)^\eta p(\mathbf{\Theta})}{\int \prod_{k=1}^{K} p\left(\mathbf{x}_k \mid \mathbf{\Theta}\right)^\eta p(\mathbf{\Theta}) \mathrm{d}\mathbf{\Theta}}. \end{aligned} \tag{10}$$

To compute the approximation of $-\log p(\mathbf{X} \mid \lambda)$ efficiently, we need to sample the posterior distribution $p(\mathbf{\Theta} \mid \mathbf{X}, \lambda, \eta)$. In the following, we present an effective sampler of the posterior distribution.

3.1. Block Gibbs Sampler for GLIC

It is generally difficult to directly sample from the posterior distribution under the graph Laplacian constraint. Hence, we generate pseudo-samples $\{\mathbf{L}_m\}_{m=1}^M$ ($\mathbf{L}_m \in \mathcal{L}$) from $p(\mathbf{\Theta} \mid \mathbf{X}, \lambda, \eta)$ by the following steps:

1. Sampling $\{\mathbf{\Theta}_m\}_{m=1}^M$ from a posterior distribution using Gaussian Markov random field *without* the Laplacian constraint (i.e., $\mathrm{N}(0, \mathbf{\Theta}^{-1})$).
2. Constructing the graph Laplacian $\mathbf{L}_m$ closest to $\mathbf{\Theta}_m$.

Here, we describe the sampling method in Step 1 using block Gibbs sampling. First, the Laplace distribution in (4) is rewritten by scale mixture of Gaussians [19, 20] to derive a hierarchical representation to which block Gibbs sampling can be applied:

$$\frac{\lambda}{2} e^{-\lambda|\theta|} = \int_0^\infty \frac{1}{\sqrt{2\pi\tau}} e^{-\theta^2/(2\tau)} \frac{\lambda^2}{2} e^{-\lambda^2\tau/2} \mathrm{d}\tau, \quad \lambda > 0, \quad (11)$$

where τ is a latent scale parameter. By using $p(\mathbf{x} \mid \mathbf{\Theta}) = \mathrm{N}(0, \mathbf{\Theta}^{-1})$ and (11), it can be derived that $p(\mathbf{\Theta}, \boldsymbol{\tau} \mid \mathbf{X}, \lambda, \eta)$ is proportional to the following quantity:

$$p(\mathbf{\Theta}, \boldsymbol{\tau} \mid \mathbf{X}, \lambda, \eta) \propto |\mathbf{\Theta}|^{\frac{\eta K}{2}} \exp\left\{-\eta \operatorname{Tr}\left(\frac{1}{2}\mathbf{S}\mathbf{\Theta}\right)\right\} \times \prod_{i<j} \left\{\tau_{ij}^{-\frac{1}{2}} \exp\left(-\frac{\theta_{ij}^2}{2\tau_{ij}}\right) \exp\left(-\frac{\lambda^2}{2}\tau_{ij}\right)\right\}. \quad (12)$$

Hereafter, we describe our block Gibbs sampler based on (12). We denote block matrix representations of $\mathbf{\Theta}$, $\mathbf{S}$, and $\mathbf{T}$ as follows:

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Theta}_{11} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{12}^\mathsf{T} & \theta_{22} \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} \mathbf{S}_{11}, \mathbf{s}_{12} \\ \mathbf{s}_{12}^\mathsf{T}, s_{22} \end{bmatrix}, \ \mathbf{T} = \begin{bmatrix} \mathbf{T}_{11}, \boldsymbol{\tau}_{12} \\ \boldsymbol{\tau}_{12}^\mathsf{T}, 0 \end{bmatrix}, \quad (13)$$

where $\mathbf{T} \in \mathbb{R}^{N\times N}([\mathbf{T}]_{ij} = \tau_{ij})$ be a symmetric matrix whose all diagonal elements are zero. The block Gibbs sampler iterates sampling of $\boldsymbol{\theta}_{12}$, θ_{22} and τ_{ij} from their conditional posterior distributions to obtain samples $\{\mathbf{\Theta}_m\}_{m=1}^M$.

First, we consider sampling of $\boldsymbol{\theta}_{12}$ and θ_{22}. From (12) and (13), the conditional posterior of $\boldsymbol{\theta}_{12}$ and θ_{22} are given by

$$\begin{aligned} &p\left(\boldsymbol{\theta}_{12}, \theta_{22} \mid \mathbf{\Theta}_{11}, \mathbf{T}, \mathbf{X}, \lambda, \eta\right) \\ &\propto \left(\theta_{22} - \boldsymbol{\theta}_{12}^\mathsf{T}\mathbf{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}\right)^{\frac{\eta K}{2}} \\ &\quad \times \exp\left[-\frac{1}{2}\left\{\boldsymbol{\theta}_{12}^\mathsf{T}\mathbf{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\theta}_{12} + 2\eta\mathbf{s}_{12}^\mathsf{T}\boldsymbol{\theta}_{12} + (\eta s_{22} + \lambda)\theta_{22}\right\}\right], \end{aligned} \quad (14)$$

where $\mathbf{D}_{\boldsymbol{\tau}} = \operatorname{diag}(\boldsymbol{\tau}_{12})$. We use the following Schur complement lemma [21] to derive (14).

$$\det\left(\begin{bmatrix} \mathbf{\Theta}_{11} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{12}^\mathsf{T} & \theta_{22} \end{bmatrix}\right) = \det(\mathbf{\Theta}_{11}) \det\left(\theta_{22} - \boldsymbol{\theta}_{12}^\mathsf{T}\mathbf{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}\right). \quad (15)$$

After change of variables as $\boldsymbol{\beta} := \boldsymbol{\theta}_{12}$ and $\gamma := \theta_{22} - \boldsymbol{\theta}_{12}^\mathsf{T}\mathbf{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}$, (14) is reduced to the following form:

$$\begin{aligned} &p\left(\boldsymbol{\beta}, \gamma \mid \mathbf{\Theta}_{11}, \mathbf{T}, \mathbf{X}, \lambda, \eta\right) \\ &\propto \gamma^{\frac{\eta K}{2}} \exp\left(-\frac{\eta s_{22} + \lambda}{2}\gamma\right) \\ &\quad \times \exp\left(-\frac{1}{2}\left[\boldsymbol{\beta}^\mathsf{T}\left\{\mathbf{D}_{\boldsymbol{\tau}}^{-1} + (\eta s_{22} + \lambda)\mathbf{\Theta}_{11}^{-1}\right\}\boldsymbol{\beta} + 2\eta\mathbf{s}_{12}^\mathsf{T}\boldsymbol{\beta}\right]\right). \end{aligned} \quad (16)$$

This indicates that γ and $\boldsymbol{\beta}$ can be sampled from the following gamma and multivariate Gaussian distributions:

$$\begin{aligned} \gamma \mid (\mathbf{\Theta}_{11}, \mathbf{T}, \mathbf{X}, \lambda, \eta) &\sim \mathrm{Gam}\left(\frac{\eta K}{2} + 1, \frac{s_{22} + \lambda}{2}\right), \\ \boldsymbol{\beta} \mid (\mathbf{\Theta}_{11}, \mathbf{T}, \mathbf{X}, \lambda, \eta) &\sim \mathrm{N}\left(-\eta\mathbf{C}\mathbf{s}_{12}, \mathbf{C}\right), \end{aligned} \quad (17)$$

where $\mathbf{C} = ((\eta s_{22} + \lambda)\mathbf{\Theta}_{11} + \mathbf{D}_{\boldsymbol{\tau}})$.

Second, we consider sampling of τ_{ij} in (12). The conditonal distribution of $u_{ij} = 1/\tau_{ij}$ is given by

$$u_{ij} \mid (\theta_{ij}, \lambda) \sim \mathrm{IGau}(\sqrt{(\lambda^2/\theta_{ij}^2)}, \lambda^2). \quad (18)$$

As a result, the block Gibbs sampler procedure of (17) and (18) allows us to sample $\{\mathbf{\Theta}_m\}_{m=1}^M$ from the posterior distribution (12).

In Step 2, the sampled $\mathbf{\Theta}_m$ is replaced by the graph Laplacian $\mathbf{L}_m$ closest to $\mathbf{\Theta}_m$. This graph Laplacian $\mathbf{L}_m$ is obtained by solving the problem:

$$\underset{\mathbf{L}_m \in \mathcal{L}}{\text{minimize}} \ \|\mathbf{\Theta}_m - \mathbf{L}_m\|_1. \quad (19)$$

This problem can be easily solved using the result in [22].

3.2. Computation of GLIC

Let $\{\mathbf{L}_m\}_{m=1}^M$ be samples from the block Gibbs sampler presented in Section 3.1. Based on (10), we define the GLIC as follows:

$$\mathrm{GLIC} = \frac{1}{2M}\left(\sum_{m=1}^M \operatorname{Tr}(\mathbf{L}_m\mathbf{S}) - K \log \operatorname{gdet}(\mathbf{L}_m)\right). \quad (20)$$

Although GLIC seems similar to (8), it is computed based on WBIC and MCMC and can yield a more reliable model evidence approximation. The computational complexity of GLIC is $\mathcal{O}(MN^3)$.

The algorithm of the proposed method is summarized in Algorithm 1. From a set of graphs learned from different hyperparameters, one with the smallest GLIC is selected as the optimal graph.

Algorithm 1 GLIC

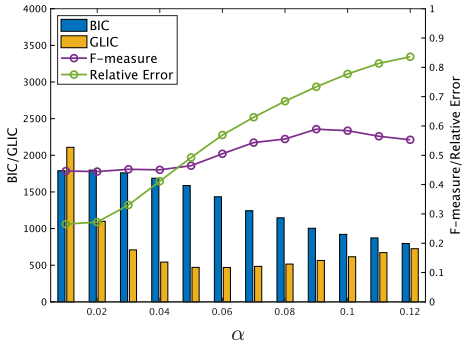
Input: α, $\mathbf{S}$, M
Output: GLIC
  Initialize $\mathbf{\Theta} = \mathbf{I}$
  for $m = 0$ to M **do**
    for $i = 0$ to N **do**
      Sample γ and $\boldsymbol{\beta}$ from (17).
      Update $\boldsymbol{\theta}_{12} = \boldsymbol{\beta}$ and $\theta_{22} = \gamma + \boldsymbol{\theta}_{12}^\mathsf{T}\mathbf{\Theta}_{11}^{-1}\boldsymbol{\theta}_{12}$.
      Rearrange row/columns of $\mathbf{\Theta}$, $\mathbf{S}$, and $\mathbf{T}$.
    end for
    for $i < j$ **do**
      Sample u_{ij} from (18).
      Update $\tau_{ij} = 1/u_{ij}$.
    end for
    Obtain $\mathbf{L}_m$ by solving (19).
  end for
  Compute GLIC in (20).

4. EXPERIMENTS

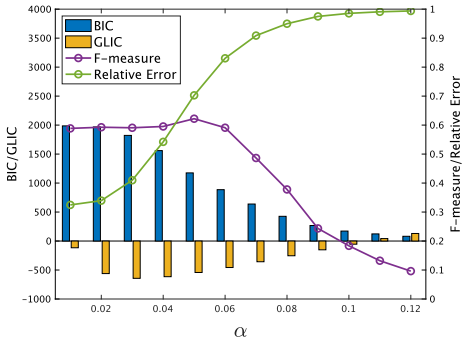
In this section, we conduct graph learning experiments using random graphs to validate the efficiency of the proposed method.

Table 1. Average performance of graph learning under hyperparameters selected by BIC and GLIC.

	Dataset	Relative error	F-measure
BIC	ER	0.867	0.518
	RM	0.968	0.231
GLIC	ER	**0.541**	**0.546**
	RM	**0.451**	**0.572**



(a) ER graph



(b) RM graph

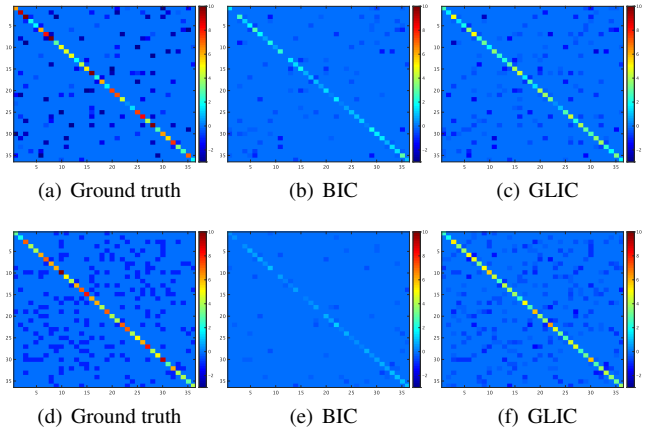
Fig. 1. BIC and GLC with different α. (a) BIC: $\hat{\alpha} = 0.12$, GLIC: $\hat{\alpha} = 0.06$. (b) BIC: $\hat{\alpha} = 0.12$, GLIC: $\hat{\alpha} = 0.03$.

4.1. Dataset and Setup

We construct datasets by the following two steps: 1) Constructing random graphs and 2) generating signals from the constructed graph. In this experiment, we use two types of random graphs:

- Erdős–Rényi (ER) graph $\mathcal{G}_{\mathrm{ER}}^{(p)}$ where p is the edge connection probability.
- Random modular (RM) graph $\mathcal{G}_{\mathrm{RM}}^{(p_1,p_2)}$ (also known as graphs with stochastic block model), where p_1 and p_2 are intra-cluster and inter-cluster edge connection probabilities, respectively.

All graphs have $N = 36$, and the edge weights are selected randomly from the uniform distribution $\mathrm{U}(0.1, 3)$. We construct 30 graphs for each graph model: p of ER graph is chosen from $\mathrm{U}(0.08, 0.12)$; p_1 and p_2 of RM graph are chosen from $\mathrm{U}(0.25, 0.3)$ and $\mathrm{U}(0.08, 0.12)$, respectively. We use LGMRF in (2) to generate 100 graph signals from each of the constructed graphs. For both datasets, graphs are learned with different α and compute BIC and GLIC.



(a) Ground truth (b) BIC (c) GLIC

(d) Ground truth (e) BIC (f) GLIC

Fig. 2. Visualization of the learned graph Laplacian with hyperparameters selected by BIC and GLIC. The top rows and bottom rows depict the learned graphs from ER graph dataset and RM graph dataset, respectively.

The performance of graph learning is evaluated by F-measure and relative error. The F-measure, which is the harmonic average of the precision and recall, represents the accuracy of the estimated graph structure. Relative error is given by:

$$\mathrm{RE}\left(\widehat{\boldsymbol{\Theta}}, \boldsymbol{\Theta}^*\right) = \frac{\left\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\right\|_F}{\left\|\boldsymbol{\Theta}^*\right\|_F} \tag{21}$$

where $\widehat{\boldsymbol{\Theta}}$ is the estimated graph Laplacian, $\boldsymbol{\Theta}^*$ is the ground truth, and $\|\cdot\|_F$ is the Frobenius norm.

4.2. Results

Table 1 summarizes the average performance under the hyperparameters selected by BIC and GLIC. For both graphs, the graph selected with GLIC significantly outperforms that with BIC in the F-measure and relative error.

Fig. 1 shows the BIC and GLIC with different α as well as the objective performances. Let us denote $\hat{\alpha}$ as the α having the smallest information criterion. For the ER graph, $\hat{\alpha}$ with BIC is 0.12, and that with GLIC is 0.06; for the RM graph, $\hat{\alpha}$ with BIC is 0.12, and that with GLIC is 0.03. As observed from the figure, GLIC selects the parameter that indicates a good trade-off between F-measure and relative error, while BIC fails to do so.

Fig. 2 shows the visualization of the learned graph Laplacian with hyperparameters selected by BIC and GLIC. While the graph selected by BIC is too sparse compared to the ground truth, GLIC can select graphs close to the ground truth.

5. CONCLUSION

In this paper, we propose a model selection criterion for graph learning using the approximation of the model evidence. It is based on WBIC and is computed with an efficient sampler of the posterior distribution of the graph learning model based on LGMRF. The experimental results demonstrated that the proposed method can successfully select a high quality parameter set among candidates.

6. REFERENCES

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[3] M. G. Preti, T. A. Bolton, and D. Van De Ville, "The dynamic functional connectome: State-of-the-art and perspectives," *NeuroImage*, vol. 160, pp. 41–54, 2017.

[4] Y. Kim, S. Han, S. Choi, and D. Hwang, "Inference of dynamic networks using time-course data," *Brief. Bioinformatics*, vol. 15, no. 2, pp. 212–228, 2014.

[5] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, "Network inference via the time-varying graphical Lasso," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery and Data Mining*. 2017, pp. 205–213, ACM.

[6] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

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

[8] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, 2016.

[9] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.

[10] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from filtered signals: Graph system and diffusion kernel identification," *IEEE Trans. Signal Inf. Process. Netw.*, pp. 1–1, 2018.

[11] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 920–929.

[12] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning based on sparseness of temporal variation," in *Proc. IEEE Int. Conf. Acoust. Speech. Signal Process.*, 2019, pp. 5411–5415.

[13] K. Yamada, Y. Tanaka, and A. Ortega, "Time-varying graph learning with constraints on graph temporal variation," *ArXiv200103346 Cs Eess*, 2020.

[14] J. K. Tugnait, "Sparse-Group Lasso for Graph Learning From Multi-Attribute Data," *IEEE Trans. Signal Process.*, vol. 69, pp. 1771–1786, 2021.

[15] G. Schwarz, "Estimating the Dimension of a Model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[16] S. Watanabe, "A Widely Applicable Bayesian Information Criterion," *J. Mach. Learn. Res.*, vol. 14, no. Mar, pp. 867–897, 2013.

[17] A. Holbrook, "Differentiating the pseudo determinant," *Linear Algebra and its Applications*, vol. 548, pp. 293–304, 2018.

[18] C. Robert, *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*, Springer-Verlag, second edition, 2007.

[19] T. Park and G. Casella, "The Bayesian Lasso," *J. Am. Stat. Assoc.*, vol. 103, no. 482, pp. 681–686, 2008.

[20] H. Wang, "Bayesian Graphical Lasso Models and Efficient Posterior Computation," *Bayesian Anal.*, vol. 7, no. 4, pp. 867–886, 2012.

[21] G. H. Golub and C. F. V. Loan, *Matrix Computations*, JHU Press, 2013.

[22] K. Sato, "Optimal graph Laplacian," *Automatica*, vol. 103, pp. 374–378, 2019.