2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2015, BOSTON, USA

VARIATIONAL BAYES LEARNING OF GRAPHICAL MODELS WITH HIDDEN VARIABLES

Hang Yu and Justin Dauwels

School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, 639798

ABSTRACT

Hidden variable graphical models are powerful tools to describe high-dimensional data; they capture dependencies between observed variables by introducing a suitable number of hidden variables. Present methods for learning the dependence structure of hidden variable graphical models are derived from the idea of maximizing penalized likelihood, and hence are associated with the troublesome problem of regularization selection. In this paper, we show that this problem can be successfully circumvented by treating the penalty parameters as random variables and describing the hidden variable graphical models in a Bayesian formulation. An efficient variational Bayes algorithm is further developed to adaptively learn the graphical model as well as the distribution of penalty parameters. Numerical results from both synthetic and real data show that the proposed variational Bayes method yields comparable or better performance than the stability selection based maximum penalized likelihood method, yet it requires several orders of magnitude less computational time.

Index Terms— Gaussian graphical models, hidden variables, variational Bayes, regularization selection

1. INTRODUCTION

Graphical models can describe complicated systems with few parameters by learning or imposing a sparse dependency structure. Moreover, the structure can in turn be leveraged to derive highly efficient inference algorithms. Thus, the application of graphical models has grown in prominence in different fields, including signal processing [1], image processing and computer vision [2], computational biology and neuroscience [3, 4], geophysics and earth science [5, 6], and sociology [7].

One typically constructs a sparse graphical model by discovering the most important interactions between observed variables [8], as shown in Fig. 1a. We refer to this graphical model as a standard graphical model in this context, as opposed to the graphical models with hidden variables to be introduced. Standard graphical models, however, fail to deal with the case where there exist hidden variables. Now suppose that the yellow nodes in Fig. 1a represent hidden variables; we only observe samples of the observed variables (green nodes in Fig. 1a) and no information is provided about the hidden variables. Under such scenario, approaches to inferring standard graphical models would yield a dense graph (see Fig. 1b), including both interactions originally between observed variables as well as those coming from hidden variables. Instead, we may consider the effect of hidden variables and obtain a sparse graph that only characterizes the direct interdependencies between observed variables. In practice, it is quite common that data is unavailable for some relevant variables. For instance, when inferring functional brain networks given brain signals, the signals may only be measured from some specific brain areas (e.g., cortex in the case of scalp electroencephalograms). However, those signals may be affected by brain areas from which no measurements are available (e.g., deeper areas such as hippocampus). The latter may then be treated as hidden variables in a statistical model.

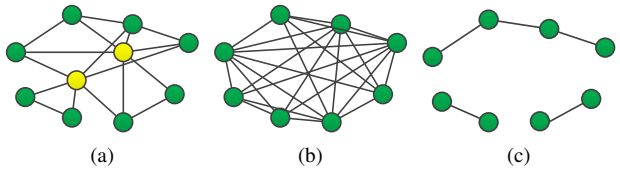

Fig. 1: A graphical model with hidden variables (yellow nodes): (a) the joint graphical model; (b) the marginal graphical model of observed variables (green nodes); (c) the conditional graphical model of observed variables.

When the observed variables $\boldsymbol{z}_O$ (green nodes in Fig. 1a) and hidden variables $\boldsymbol{z}_H$ (yellow nodes in Fig. 1a) are jointly Gaussian distributed, the structure of the graphical model can be defined by the precision matrix (inverse covariance matrix) of the observed and hidden variables. Recently, Chandraskaran et al. [9] decomposed the (marginal) precision matrix of $\boldsymbol{z}_O$ into a sparse matrix K_O (conditional precision matrix) and a low-rank matrix L, which describes the coupling between the observed and hidden variables. The conditional graphical model K_O and the number of hidden variables (rank of L) are inferred by solving a convex penalized maximum-likelihood problem; an ℓ_1 norm is imposed on K_O to favor sparsity while a nuclear norm on L to favor low rank. The conditional precision matrix K_O is represented as a graph (Fig. 1c), where nodes $\mathcal{V}_i$ and $\mathcal{V}_j$ are connected by an edge if and only if the corresponding element (i, j) in K_O is nonzero. That graph visualizes the dependence among the observed variables, conditioned on the hidden variables.

978-1-4673-7454-5/15/$31.00

Unfortunately, the result of the penalized maximum likelihood problem is quite sensitive to the penalty parameters, which measure the trade-off between model fidelity and the number of parameters and therefore determine the sparsity of K_O and the rank of L. Standard methods for regularization selection, such as cross validation, Akaike information criterion, and Bayesian information criterion, are shown to overfit the data for high dimensional problems, leading to dense graphs [10]. As an alternative, stability based methods [11, 12] can reliably infer the graphical model structure by selecting a "stable" graph from bootstrapped sample sets. Although such methods have given encouraging results for learning hidden variable graphical models in [3], they suffer from two immediate issues. First, they present a daunting computational task, since the learning algorithm has to be run on each sample set for every possible combination of regularization parameters. Moreover, they are not directly applicable to determine the rank of L.

In this paper, we sidestep these problems with the help of a Bayesian formulation of graphical models with hidden variables. In particular, we focus on the case where both the observed and hidden variables are jointly Gaussian distributed. We further impose sparsity priors on off-diagonal elements of the conditional precision matrix K_O, and enforce the low-rank constraint on L as sparsity priors. The parameters of the resulting model is inferred using a variational Bayes (VB) algorithm. Numerical results demonstrate that the proposed approach can select the proper amount of regularization in an automated manner, and therefore provides a good fit with few parameters, while requiring several orders of magnitude less computational time.

The paper is organized as follows. In Section 2, we briefly introduce hidden variable graphical models and their Bayesian formulation. We then derive the VB algorithm in Section 3. We present experimental results in Section 4 and close the paper with concluding remarks in Section 5.

2. HIDDEN VARIABLE GRAPHICAL MODELS

Suppose we have P observable variables $\boldsymbol{z}_O$ (green nodes in Fig. 1a) and R hidden variables $\boldsymbol{z}_H$ (yellow nodes in Fig. 1a) which are jointly Gaussian distributed. The joint precision matrix of $\boldsymbol{z}_O \cup \boldsymbol{z}_H$, $K_{O\cup H}$, which characterizes the graphical model in Fig. 1a, can be expressed as:

$$K_{O\cup H} = \begin{bmatrix} K_{OO} & K_{OH} \\ K_{HO} & K_{HH} \end{bmatrix}. \tag{1}$$

Then according to the Schur complement, the marginalized precision matrix $\tilde{K}_{OO}$ of $\boldsymbol{z}_O$ can be written as:

$$\tilde{K}_{OO} = K_{OO} - K_{OH}K_{HH}^{-1}K_{HO} = K_{OO} - L, \tag{2}$$

where $L = K_{OH}K_{HH}^{-1}K_{HO}$. Note that given the joint covariance matrix $\Sigma_{O\cup H}$, the marginal precision matrix of observed variables $\tilde{K}_{OO} = ([\Sigma_{O\cup H}]_{OO})^{-1}$.

The two components of $\tilde{K}_{OO}$ have their own properties [9]. K_{OO} is the conditional precision matrix of $\boldsymbol{z}_O$, conditioned on $\boldsymbol{z}_H$ (see Fig. 1c). It is supposed to be sparse as it describes the interactions only between observed variables. L summarizes the effect of marginalization over the hidden variables. The rank of L is equal to the number of hidden variables, and it is assumed to be low since the number of hidden variables is supposed to be small. Since hidden variables $\boldsymbol{z}_H$ are connected to many observed variables $\boldsymbol{z}_O$, K_{OH} and K_{HO} are not sparse, thus making the product matrix dense. Resulting from the subtraction, $\tilde{K}_{OO}$ is also dense, as shown in Fig. 1b. Standard Gaussian graphical models [8] cannot yield a sparse graph in this case, since these models can only estimate $\tilde{K}_O$.

Given N $i.i.d.$ samples of $\boldsymbol{z}_O$, our objective is to estimate K_{OO} and L; we are especially interested in the rank R of L since it equals the number of hidden variables. Those matrices may be recovered by solving the convex relaxation [9]:

$$\begin{aligned}(\hat{K}_{OO}, \hat{L}) = &\underset{K_{OO}\succ 0, L\succeq 0}{\operatorname{argmin}} \operatorname{tr}((K_{OO} - L)S_{OO}) \\ &- \log\det(K_{OO} - L) + \lambda|K_{OO}\|_1 + \gamma \operatorname{tr}(L),\end{aligned} \tag{3}$$

where $\hat{K}_{OO}$ and $\hat{L}$ are estimates of K_{OO} and L respectively, and S_{OO} is the empirical marginal covariance of $\boldsymbol{z}_O$. In the above expression, $\operatorname{tr}((K_{OO} - L)S_{OO}) - \log\det(K_{OO} - L)$ is the divergence between the observed data and the estimated model. The two penalty or regularization parameters λ and γ can be interpreted as follows. λ is the penalty parameter of the ℓ_1 norm; it controls the trade-off between the sparsity of K_{OO} and the fidelity of the data. On the other hand, γ is the penalty parameter of the nuclear norm (which reduces to the trace norm for symmetric, positive-semidefinite matrices), and therefore it controls the rank of L. Note that the parameters λ and γ need to be chosen appropriately in order to recover the correct K_{OO} and L,

To bypass the delicate issue of regularization selection, we instead formulate the learning problem of hidden variable graphical models from a Bayesian perspective. First, we focus on imposing priors that favor low rank on L. Recall that $L = K_{OH}K_{HH}^{-1}K_{HO}$ and K_{HH} is positive definite according to the definition, therefore, L can be factorized as:

$$L = AA^T, \tag{4}$$

where A is a matrix with the same rank as L. To obtain a low-rank estimate of L, we can equivalently enforce column sparsity in A such that most columns in A are set equal to zero. As a result, we impose a Gaussian prior with zero mean and precision γ_j on each column of A [13, 14]:

$$p(A_{:,j}|\gamma_j) \propto \sqrt{\gamma_j}\exp\left(-\frac{\gamma_j}{2}A_{:,j}^T A_{:,j}\right), \tag{5}$$

where $A_{:,j}$ denotes the jth column of matrix A. We further put conjugate Gamma hyperpriors on the precisions γ_j with shape parameters a_0 and rate parameters b_0:

$$p(\gamma_j) = \text{Gamma}(\gamma_j; a_0, b_0) \propto \gamma_j^{a_0-1}\exp(-b_0\gamma_j). \tag{6}$$

To obtain non-informative hyperpriors, we set a_0 and b_0 to be very small (e.g., 10^{-10}). Note that the corresponding expectation of γ_j can be computed as $\langle\gamma_j\rangle = a_0/b_0$.

Next, let us turn our attention to the priors on the off-diagonal entries of K_O. To simplify notation, we replace K_O with K in the following context. Similar to A, the modeling of the sparsity in K is done by utilizing independent Gaussian priors with zero means on each off-diagonal entry of K, that is,

$$p(K_{ij}|\lambda_{ij}) \propto \sqrt{\lambda_{ij}} \exp\left(-\frac{\lambda_{ij}}{2}K_{ij}^2\right), \quad \text{for} \quad i > j. \quad (7)$$

As before, we employ conjugate Gamma hyperpriors on the precisions λ_{ij}

$$p(\lambda_{ij}) = \text{Gamma}(\lambda_{ij}; c_0, d_0), \quad (8)$$

where the shape and rate parameters (c_0, d_0) are set to be 10^{-10}. Note that

$$\int p(K_{ij}|\lambda_{ij})p(\lambda_{ij})d\lambda_{ij} = \frac{\Gamma\big(a+\frac{1}{2}\big)}{\Gamma(a)\sqrt{2\pi b}}\left(\frac{1}{1+\frac{1}{2b}K_{ij}^2}\right)^{a+\frac{1}{2}},$$

which is a student's t-distribution. Therefore, we essentially put a t-prior on K_{ij}. Such shrinkage prior is often used in the Bayesian framework to promote sparsity [15, 16]. On the other hand, the ℓ_1 norm in Eq. (3) is equivalent to a Laplace prior. Although Laplace priors can also be regarded as a scale mixture of Gaussian, the hyperprior on precisions λ_{ij} is the inverse Gamma distribution that is not conjugate to the Gaussian distributions parameterized by precisions [17]. As a result, we employ t-prior here since it is more tractable for Bayesian inference. In addition, as shown in our numerical experiments, many of the precisions λ_{ij} will take very large values during the learning process, and consequently, the selected prior can successfully shrink most elements of K to zero, thus yielding sparse a conditional graphical model corresponding to $\boldsymbol{z}_O$.

Altogether, the overall joint model can be expressed as:

$$p(\boldsymbol{z}_O, K, A, \Lambda, \boldsymbol{\gamma}) = \prod_{i=1}^{N} p(\boldsymbol{z}_O^{(i)}|K, A) \prod_{j=1}^{P}\prod_{i>j}\Big[p(K_{ij}|\lambda_{ij}) \cdot p(\lambda_{ij})\Big] \prod_{j=1}^{P}\Big[p(A_{:,j}|\gamma_j)p(\gamma_j)\Big], \quad (9)$$

where $p(\boldsymbol{z}_O^{(i)}|K, A)$ is a Gaussian distribution with zero mean and precision matrix $\tilde{K} = \tilde{K}_{OO} = K - AA^T$, $\Lambda = [\lambda_{ij}]$, and $\boldsymbol{\gamma} = [\gamma_j]^T$.

3. VARIATIONAL BAYES INFERENCE

In this section, we devise a VB algorithm to estimate the sparse conditional precision matrix K and the low-rank matrix A. Specifically, we attempt to make the variational distribution $q(K, A, \Lambda, \boldsymbol{\gamma})$ a good approximation to the intractable posterior $p(K, A, \Lambda, \boldsymbol{\gamma}|\boldsymbol{z}_O)$ by minimizing the KL divergence between p and q as measured by $\text{KL}(q|p) = \int q\log(q/p)$. Here, we apply the mean-field approximation, and therefore, the variational distribution can be factorized as:

$$q(K, A, \Lambda, \boldsymbol{\gamma}) = \prod_{i=1}^{P}\delta(K_{i:P,i} - K_{i:P,i}^*)\delta(A_{i,:} - A_{i,:}^*) \cdot \prod_{j=1}^{P}\prod_{i>j} q(\lambda_{ij}) \prod_{j=1}^{P} q(\gamma_j), \quad (10)$$

where $\delta(\boldsymbol{a} - \boldsymbol{a}^*)$ is a delta function, $K_{i:P,i}$ denotes the ith to Pth elements in the ith column of K, and $A_{i,:}$ denotes the ith row of A. We follow [16, 17, 18] to use delta functions as the variational distributions of elements in K and A for the sake of convenience. Furthermore, as the algorithm proceeds, many of the precisions λ_{ij} and γ_j will become very large, and then delta functions can well approximate the true posterior distribution.

The VB update rules can be derived as follows. For the sparse matrix K and the low-rank matrix A, we generate point estimates of $K_{i:P,i}$ and $A_{i,:}$ sequentially for $i = 1, \cdots, P$. Equating the corresponding gradient to zero yields Eq. (11) and (12). Note that there is no guarantee that the KL divergence is decreased every time we update K^* and A^*, due to the non-conjugacy between the corresponding prior and likelihood [19]. However, as shown in [19], such problem can be fixed by damping. As a result, we employ the damping method with a damping factor $\rho < 1$, and find that it successfully ameliorates the problem and helps find a better local maximum in our experiments. More precisely, given $K_{i:P,i}^*$ and $A_{i,:}^*$ resulting from (11) and (12) as well as the estimates in the previous iteration $(K_{i:P,i}^*)^{(\kappa-1)}$ and $(A_{i,:}^*)^{(\kappa-1)}$, the es-

$$\begin{bmatrix} K_{i+1:P,i} \\ A_{i,:}^T \end{bmatrix} = \begin{bmatrix} NS_{ii}[\tilde{K}_{-i,-i}^{-1}]_{i:P-1,i:P-1} + \text{diag}(\langle\Lambda_{i+1:P,i}\rangle) & -NS_{ii}[\tilde{K}_{-i,-i}^{-1}]_{i:P-1,:}A_{-i,:} \\ -NS_{ii}A_{-i,:}^T[\tilde{K}_{-i,-i}^{-1}]_{:,i:P-1} & NS_{ii}\tilde{K}_{-i,-i}^{-1}A_{-i,:} + \text{diag}(\langle\boldsymbol{\gamma}\rangle) \end{bmatrix}^{-1} \cdot \begin{bmatrix} -NS_{i+1:P,i} - NS_{ii}[\tilde{K}_{-i,-i}^{-1}]_{i:P-1,1:i-1}K_{i:i-1,i} \\ A_{-i,:}^T(NS_{-i,i} + NS_{ii}[\tilde{K}_{-i,-i}^{-1}]_{:,1:i-1}K_{1:i-1,i}) \end{bmatrix}, \quad (11)$$

$$K_{ii} = \frac{1}{S_{ii}} + A_{i,:}A_{i,:}^T + (K_{i,-i} - A_{i,:}A_{-i,:}^T)\tilde{K}_{-i,-i}^{-1}(K_{-i,i} - A_{-i,:}A_{i,:}^T), \quad (12)$$

where $-i$ denotes all the indices in $\{1, \cdots, P\}$ except i, $\tilde{K}_{OO}$ is given in Eq. (2), S is the empirical covariance, $\text{diag}(\boldsymbol{a})$ denotes a diagonal matrix with vector $\boldsymbol{a}$ on the diagonal, and $\langle\boldsymbol{a}\rangle$ denotes the expectation of $\boldsymbol{a}$ $w.r.t.$ $q(\boldsymbol{a})$.

timates in the current iteration can be computed as:

$$(K^*_{i:P,i})^{(\kappa)} = \rho(K^*_{i:P,i})^{(\kappa-1)} + (1-\rho)K^*_{i:P,i}, \quad (13)$$

$$(A^*_{i,:})^{(\kappa)} = \rho(A^*_{i,:})^{(\kappa-1)} + (1-\rho)A^*_{i,:}. \quad (14)$$

We further prove that $K \succ 0$ and $L = AA^T \succeq 0$ during the update in the following proposition.

Proposition 1. *Let $K^{(0)}$ and $\tilde{K}^{(0)}$ be initially symmetric positive definite matrices, and $S_{ii} > 0\ \forall i$. Then it is guaranteed that $K^{(\kappa)} \succ 0$ and $L^{(\kappa)} \succeq 0$ as the variational Bayes algorithm proceeds.*

Proof. Since $L^{(\kappa)}$ can be factored as AA^T, it is always positive semi-definite. On the other hand, since $K^{(\kappa)} = \tilde{K}^{(\kappa)} + L^{(\kappa)}$, we only need to show that $\tilde{K}^{(\kappa)} \succ 0$ in order to prove the positive definiteness of $K^{(\kappa)}$. It follows from Eq. (12) that in each iteration before the damping step, the updated $\tilde{K}$ can be decomposed as:

$$\begin{aligned}\tilde{K} &= \begin{bmatrix} \tilde{K}_{-i,-i} & \tilde{K}_{-i,i} \\ \tilde{K}_{i,-i} & 1/S_{ii} + \tilde{K}_{i,-i}\tilde{K}^{-1}_{-i,-i}\tilde{K}_{-i,i} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{K}_{-i,-i} \\ \tilde{K}_{i,-i} \end{bmatrix} K^{-1}_{-i,-i} \begin{bmatrix} \tilde{K}_{-i,-i} & \tilde{K}_{-i,i} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1/S_{ii} \end{bmatrix}\end{aligned}$$

Note that $\tilde{K}_{-i,-i}$ is positive definite by induction. In addition, $S_{ii} > 0$ and therefore $1/S_{ii} > 0$. As the sum of two positive semi-definite matrices, $\tilde{K} \succeq 0$. Moreover, $\det(\tilde{K}) = \det(\tilde{K}_{-i,-i})/S_{ii} > 0$. Hence, $\tilde{K}$ is positive definite and so is K. Since the damping step will not change the positive definiteness of K, we can conclude that $K^{(\kappa)} \succ 0$ during the update. □

Finally, for λ_{ij} and γ_j, their variational distribution can be updated as:

$$q(\lambda_{ij}) = \text{Gamma}\Big(\lambda_{ij}; d_0 + \frac{1}{2}, c_0 + \frac{1}{2}(K^*_{ij})^2\Big), \quad (15)$$

$$q(\gamma_j) = \text{Gamma}\Big(\gamma_j; a_0 + \frac{P}{2}, b_0 + \frac{1}{2}{A^*_{:,j}}^T A^*_{:,j}\Big). \quad (16)$$

Damping is also used here when updating the parameters of $q(\lambda_{ij})$ and $q(\gamma_j)$.

Note that as the algorithm proceeds, we have to evaluate the inverse of $\tilde{K}_{-i,-i}$ in (11) and (12) for each $i \in \{1, \cdots, P\}$ in each iteration, whose computational complexity is $\mathcal{O}(P^3)$. On the other hand, according to Schur complement, we find that $\tilde{K}^{-1}_{-i,-i}$ can be equivalently expressed as:

$$\tilde{K}^{-1}_{-i,-i} = \tilde{\Sigma}_{-i,-i} - \frac{\tilde{\Sigma}_{-i,i}\tilde{\Sigma}_{i,-i}}{\tilde{\Sigma}_{ii}}. \quad (17)$$

Hence, to avoid the computation of the inverse, we only compute $\tilde{\Sigma} = \tilde{K}^{-1}$ at the beginning of the algorithm, and further update $\tilde{\Sigma}$ once elements in K^* and A^* is updated. The update rules are listed below [17]:

$$\tilde{\Sigma}_{ii} = \frac{1}{\tilde{K}_{ii} - \tilde{K}_{i,-i}\tilde{K}^{-1}_{-i,-i}\tilde{K}_{-i,i}}, \quad (18)$$

Algorithm 1 VB-HVGM

Input: empirical covariance S of observed variables $\boldsymbol{z}_O$
Initialize $\tilde{\Sigma} = (K - AA^T)^{-1}$
repeat
  for $i = 1, \cdots, P$ **do**
    Compute $\tilde{K}^{-1}_{-i,i} = \tilde{\Sigma}_{-i,-i} - \tilde{\Sigma}_{-i,i}\tilde{\Sigma}_{i,-i}/\tilde{\Sigma}_{ii}$.
    Compute $K^*_{i:P,i}$ and $A^*_{i,:}$ following (11) and (12).
    Update $K^*_{i:P,i}$ and $A^*_{i,:}$ via damping (13)-(14).
    Update $\tilde{\Sigma}$ following (18)-(20).
  end for
  Update $q(\lambda_{ij})$ and $q(\gamma_j)$ following (15) and (16).
until convergence criterion is met

$$\tilde{\Sigma}_{-i,i} = -\tilde{K}^{-1}_{-i,-i}\tilde{K}_{-i,i}\tilde{\Sigma}_{ii}, \quad (19)$$

$$\tilde{\Sigma}_{-i,-i} = \tilde{K}^{-1}_{-i,-i} + \frac{\tilde{\Sigma}_{-i,i}\tilde{\Sigma}_{i,-i}}{\tilde{\Sigma}_{ii}}. \quad (20)$$

The proposed technique is summarized in Algorithm 1.

It is worthwhile to notice that both the variational Bayes and the penalized maximum likelihood approach have the same computational complexity, that is, $\mathcal{O}(P^3)$. However, when utilizing stability selection to determine the proper amount of regularization in MPL-HVGM (3), we have to run the learning algorithm on every bootstrapped sample set for every possible combination of penalty parameters. Thus, the method is time consuming. Instead, the penalty parameters can be determined adaptively together with the sparse and low-rank matrix in the VB framework, successfully reducing the computational burden and resulting in significant efficiency gain as shown in our experimental results.

4. NUMERICAL RESULTS

In this section, we benchmark the proposed method (referred to as VB-HVGM) with the maximum penalized-likelihood method (MPL-HVGM) (cf. Eq. (3)). The graph structure in the latter method is determined by stability selection [3, 11]. Concretely, we bootstrap 100 sample sets from the original data and select 168 pairs of penalty parameters (λ, γ) in order to compute the stability path. We compare the two algorithms by means of accuracy of graphical model estimation, model fitting and computational time. More specifically, for accuracy of graph estimation, we consider three criteria, including precision, recall and F_1-score. Precision is defined as the proportion of correctly estimated edges to all the edges in the estimated graph; recall is defined as the proportion of successfully estimated edges to all the edges in the true graph; F_1-score is defined as 2·precision·recall/(precision+recall), which is a weighted average of the precision and recall. For model fitting, we evaluate the KL divergence between the fitted model and the data, the number of parameters (Prm No.) in the model, and the BIC score.

Table 1: Results from MPL-HVGM and VB-HVGM averaged over 50 trials.

Model		N/P	Accuracy of Graphical Model Estimation								Model Fitting						Computational Time	
			Precision		Recall		F_1-score		Rank R of L		KL divergence		Prm No.		BIC score			
			MPL	VB	MPL	VB	MPL	VB	MPL	VB	MPL	VB	MPL	VB	MPL	VB	MPL	VB
Cycle (R=2)	P=25 (Prm No.=100)	5	0.27	**0.57**	0.05	**0.30**	0.13	**0.39**	4.41	**1.89**	14.56	**14.30**	1.39×10^{2}	$\mathbf{8.61\times10^{1}}$	4.31×10^{3}	$\mathbf{3.99\times10^{3}}$	2.31×10^{3}	**6.00**
		55	0.73	**0.82**	1.00	1.00	0.84	**0.90**	3.17	**2.00**	14.61	**14.56**	1.38×10^{2}	$\mathbf{1.05\times10^{2}}$	4.12×10^{4}	$\mathbf{4.08\times10^{4}}$	2.86×10^{3}	**3.05**
		105	0.72	**0.84**	1.00	1.00	0.84	**0.91**	2.33	**2.00**	14.62	**14.58**	1.19×10^{2}	$\mathbf{1.05\times10^{2}}$	7.77×10^{4}	$\mathbf{7.74\times10^{4}}$	2.90×10^{3}	**2.71**
		155	0.69	**0.83**	1.00	1.00	0.82	**0.91**	2.22	**2.01**	14.63	**14.59**	1.17×10^{2}	$\mathbf{1.06\times10^{2}}$	1.14×10^{5}	$\mathbf{1.13\times10^{5}}$	2.75×10^{3}	**2.59**
	P=100 (Prm No.=400)	5	**1.00**	0.60	0.03	**0.87**	0.07	**0.71**	40.33	**1.30**	**51.84**	53.61	4.14×10^{3}	$\mathbf{3.76\times10^{2}}$	7.75×10^{4}	$\mathbf{5.59\times10^{4}}$	1.19×10^{4}	$\mathbf{4.26\times10^{2}}$
		55	**0.98**	0.84	1.00	1.00	**0.99**	0.91	4.50	**2.03**	54.14	**53.98**	6.53×10^{2}	$\mathbf{4.22\times10^{2}}$	6.00×10^{5}	$\mathbf{5.97\times10^{5}}$	1.49×10^{4}	$\mathbf{1.71\times10^{2}}$
		105	**0.99**	0.83	1.00	1.00	**1.00**	0.90	**2.00**	2.03	54.06	**53.96**	$\mathbf{4.01\times10^{2}}$	4.25×10^{2}	1.139×10^{6}	$\mathbf{1.137\times10^{6}}$	1.57×10^{4}	$\mathbf{7.34\times10^{1}}$
		155	**0.97**	0.84	1.00	1.00	**0.98**	0.91	1.93	**2.00**	54.08	**53.97**	$\mathbf{3.98\times10^{2}}$	4.19×10^{2}	1.681×10^{6}	$\mathbf{1.677\times10^{6}}$	1.26×10^{4}	$\mathbf{7.88\times10^{1}}$
Grid (R=3)	P=25 (Prm No.=140)	5	0.28	**0.45**	0.01	**0.14**	0.06	**0.20**	8.64	**2.20**	**13.97**	14.13	2.42×10^{2}	$\mathbf{9.16\times10^{1}}$	4.66×10^{3}	$\mathbf{3.98\times10^{3}}$	2.45×10^{3}	**7.28**
		55	**0.89**	0.86	0.42	**0.87**	0.56	**0.86**	9.07	**3.43**	14.41	**14.36**	2.71×10^{2}	$\mathbf{1.51\times10^{2}}$	4.16×10^{4}	$\mathbf{4.06\times10^{4}}$	2.43×10^{3}	**4.66**
		105	**0.94**	0.90	0.69	**0.98**	0.79	**0.93**	7.23	**3.00**	14.41	**14.37**	2.35×10^{2}	$\mathbf{1.44\times10^{2}}$	7.75×10^{4}	$\mathbf{7.66\times10^{4}}$	2.59×10^{3}	**4.18**
		155	**0.93**	0.91	0.91	**1.00**	0.92	**0.95**	4.34	**2.91**	14.40	**14.37**	1.73×10^{2}	$\mathbf{1.42\times10^{2}}$	1.30×10^{5}	$\mathbf{1.13\times10^{5}}$	1.52×10^{3}	**8.22**
	P=100 (Prm No.=580)	5	**1.00**	0.67	0.01	**0.56**	0.01	**0.61**	50.73	**3.10**	**50.76**	53.47	5.17×10^{3}	$\mathbf{5.61\times10^{2}}$	8.29×10^{4}	$\mathbf{5.70\times10^{4}}$	1.04×10^{4}	$\mathbf{3.04\times10^{2}}$
		55	**0.99**	0.91	0.55	**1.00**	0.69	**0.95**	24.30	**3.33**	54.12	**53.66**	2.63×10^{3}	$\mathbf{6.29\times10^{2}}$	6.18×10^{5}	$\mathbf{5.96\times10^{5}}$	1.71×10^{4}	$\mathbf{2.75\times10^{2}}$
		105	**1.00**	0.90	0.98	**1.00**	**0.99**	0.95	10.07	**3.00**	53.78	**53.68**	1.28×10^{3}	$\mathbf{5.99\times10^{2}}$	1.14×10^{6}	$\mathbf{1.13\times10^{6}}$	1.41×10^{4}	$\mathbf{1.02\times10^{2}}$
		155	**1.00**	0.91	0.99	**1.00**	**1.00**	0.95	3.03	**3.00**	53.80	**53.68**	$\mathbf{5.83\times10^{2}}$	5.97×10^{2}	1.673×10^{6}	$\mathbf{1.670\times10^{6}}$	1.45×10^{4}	$\mathbf{4.31\times10^{1}}$

4.1. Synthetic Data

Here, we simulate samples from predefined hidden variable Gaussian graphical models. We consider two different structures of the conditional graphical models of the observed variables: the first one is a cycle and the second one is a $m \times m$ nearest-neighbor regular grid. The graphical models with cycle and grid structure are associated with $R = 2$ and $R = 3$ hidden variables respectively. In both models, each latent variable is connected to at least 80% of the observed variables. In addition, we also consider different number of observed variables for each structure (i.e., $P = 25$ and 100) as well as different ratios between samples size N and the number of observed variables P (i.e., $N/P = 5, 55, 105$, and 155). The results averaged over 50 trials are summarized in Table 1.

As demonstrated in the table, when N/P is small, VB-HVGM obviously outperforms MPL-HVGM; it can yield better estimates of the conditional graphical model and the number of hidden variables in a much shorter period of time. On the other hand, as the sample size increases, VB-HVGM still yields comparable results to that of MPL-HVGM. More explicitly, it can be seen from Table 1 that the recall given by VB-HVGM is usually higher than that of MPL-HVGM while the precision is lower. This indicates that VB-HVGM produces a relatively dense graph, successfully recovering the true graph at the expense of including a few extra edges. Such slightly dense graphs are often favored in practice; the false positives can be removed in further analysis whereas false negatives can no longer be recovered since they are buried by the sizable number of true negatives. From another perspective, we can see that the F_1-score of VB-HVGM is usually higher than that of MPL-HVGM. Even in the cases where VB-HVGM performs worse, its F_1-score value is very close to that of MPL-HVGM.

In addition, we can find that MPL-HVGM fails to identify the correct number of hidden variables unless the sample size is very large, and sometimes there is a significant difference from the ground truth. This can be explained by the fact that stability selection [18] is proposed to determine the sparsity of a matrix rather than the rank. In contrast, by inferring the penalty parameters, the sparse and the low-rank matrix simultaneously, the proposed VB-HVGM can accurately estimate the number of hidden variables.

On the other hand, in terms of model fitting, the BIC score of VB-HVGM is always smaller than that of MPL-HVGM, suggesting that VB-HVGM can handle the trade-off between data fidelity and model sparsity in an automatic manner.

Last and most importantly, a dramatic decrease in the execution time is achieved for VB-HVGM in comparison with MPL-HVGM, thus making the method more applicable to large-scale problems.

4.2. Real Data

In this section, we model the interdependencies of monthly stock returns of 84 companies in the S&P 100 stock index. We use the samples of the monthly returns from 1990 to 2007 and disregard 16 companies that have been listed on S&P 100 only after 1990. The resulting number N of samples is 216.

Chandrasekaran *et al.* [9] obtained results of this data set by solving (3); the penalty parameters are tuned manually.

They reported that the KL divergence between the hidden variable graphical model and the distribution specified by the sample covariance is 17.7, while the number of parameters is 639. We test the proposed VB algorithm on the data set, and the resulting KL divergence is 17.1 while the number of parameters is only 498. In other words, VB-HVGM can fit the data better with fewer parameters.

Additionally, we also compare VB-HVGM with MPL-HVGM. The BIC score of the two models are -8.05×10^4 (VB-HVGM) and -3.88×10^4 (MPL-HVGM), whereas the computational time of the two methods is 8.20×10^2 and 6.11×10^3 respectively. Again, VB-HVGM achieves better performance than MPL-HVGM in terms of model fitting with less computational time.

5. CONCLUSION

In this paper, we constructed Gaussian graphical models with hidden variables from a Bayesian perspective and further developed a novel VB algorithm to learn the model. The results showed that the proposed method can reliably infer the conditional precision matrix structure and the number of hidden variables in an automated manner without the need for manually tuning any regularization parameters. Compared with stability selection based maximum penalized likelihood method, the proposed VB algorithm achieves comparable or better performance in significantly less computational time.

6. ACKNOWLEDGMENT

This research was supported by MOE ACRF Tier 2 grant M4020187.

7. REFERENCES

[1] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, P. Li, and F. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295-1322, 2007.

[2] M. Beal, N. Jojic, H. Attias, "A graphical model for audiovisual object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 828-836, 2003.

[3] H. Yu, J. Dauwels, and X. Wang, "Copula Gaussian Graphical Models with Hidden Variables,"in *Proc. ICASSP*, pp. 2177-2180, 2012.

[4] J. Dauwels, H. Yu, X. Wang, F. Vialatte, C. Latchoumane, J. Jeong, and A. Cichocki, "Inferring Brain Networks through Graphical Models with Hidden Variables", *Mach. Learn. & Interpretation in Neuroimaging, Lecture Notes in Comput. Sci., Springer*, pp. 194-201, 2012.

[5] H. Yu, J. Dauwels, X. Zhang, S. Xu, and W. I. T. Uy, "Copula Gaussian Multiscale Graphical Models with Application to Geophysical Modeling," in *Proc. Fusion*, pp. 1741-1748, 2012.

[6] H. Yu, J. Dauwels, and P. Johnathan, "Extreme-Value Graphical Models with Multiple Covariates," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5734-5747, 2014.

[7] J. Dauwels, H. Yu, S. Xu, and X. Wang, "Copula Gaussian Graphical Model for Discrete Data", in *Proc. ICASSP*, pp. 6283-6287, 2013.

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

[9] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Latent Variable Graphical Model Selection via Convex Optimization," *Ann. Stat.*, vol. 40, no. 4, pp. 1935-1967, 2012.

[10] H. Liu, K. Roeder, and L. Wasserman, "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models," *Proc. NIPS*, 2010.

[11] N. Meinshausen, P. Bühlmann, "Stability Selection," *J. Roy. Statist. Soc. B - Stat. Methodol.*, vol. 72, pp. 417-473, 2010.

[12] S. Li, L. Hsu, J. Peng and P. Wang, "Bootstrap inference for network construction with an application to a breast cancer microarray study," *Ann. Appl. Stat.*, vol. 7, no. 1, pp. 391-417, 2013.

[13] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian Methods for Low-Rank Matrix Estimation," *IEEE trans. Signal Process.*, vol. 60, no. 8, pp. 3964-3977, 2012.

[14] Z. Chen, R. Molina, and A. K. Katsaggelos, "A Variational Approach for Sparse Component Estimation and Low-Rank Matrix Recovery," *J. Commun.*, vol. 8, no. 9, pp. 600-611, 2013.

[15] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211-244, 2001.

[16] H. Yu, and J. Dauwels, "Variational Bayes Learning of Multiscale Graphical Models," accepted by *ICASSP*, 2015.

[17] M. Chen, H. Wang, X. Liao, and L. Carin, "Bayesian Learning of Sparse Gaussian Graphical Models", *Technical report*, 2012.

[18] B. M. Marlin, and K. P. Murphy, "Sparse Gaussian Graphical Models with Unknown Block Structure," in *Proc. ICML*, pp. 705-712, 2009.

[19] D. A. Knowles, and T. P. Minka, "Non-conjugate Variational Message Passing for Multinomial and Binary Regression," *Proc. NIPS*, vol. 24, pp. 1701-1709, 2011.