Bayesian Learning of Sparse Gaussian Graphical Models

Minhua Chen, Hao Wang, Xuejun Liao and Lawrence Carin

Electrical and Computer Engineering Department
Duke University
Durham, NC, USA

Statistics Department
University of South Carolina
Columbia, SC, USA

Abstract

Sparse inverse covariance matrix modeling is an important tool for learning relationships among different variables in a Gaussian graph. Most existing algorithms are based on \( \ell_1 \) regularization, with the regularization parameters tuned via cross-validation. In this paper, a Bayesian formulation of the problem is proposed, where the regularization parameters are inferred adaptively and cross-validation is avoided. Variational Bayes (VB) is used for the model inference. Results on simulated and real datasets validate the proposed approach. In addition, a graph extension algorithm is proposed to include a new variable in an existing graph, which can be used when separate testing data are available.

I. INTRODUCTION

There is significant interest in using limited data to infer relationships between different variables. A Gaussian Graphical Model (GGM) [1] is an effective way to model such dependencies. Let \( X_i \in \mathbb{R}^{p \times 1} \) be a \( p \)-dimensional random vector drawn from a multivariate Gaussian distribution, i.e., 
\[ p(X_i) = \mathcal{N}(X_i; 0, J^{-1}) , \quad i = 1, 2, \cdots, n. \]
Without loss of generality, we assume the mean is subtracted in a pre-processing step. Here \( J \) is the inverse covariance matrix, also called a precision matrix or concentration matrix.

According to the Hammersley-Clifford Theorem [1], the conditional dependency of the variables in \( X_i \) is encoded in the sparsity pattern of the inverse covariance matrix \( J \). To see this more clearly, the distribution of a certain variable \( X_{\bar{s}i} \) conditioned on the remaining variables \( X_{\bar{s}i} \) can be written as
\[
p(X_{si} | X_{\bar{si}}) = \mathcal{N}(X_{si}; -J^{-1}_{ss}J_{ss}^{\top}X_{\bar{si}}, J^{-1}_{ss}),
\]
where \( \bar{s} \) denotes all variable indices excluding \( s \), and \( J_{ss} \) denotes the \( s \)th column in \( J \) excluding

\( s \)th column in \( J \) excluding.
the element $J_{ss}$. If some elements in $J_{ss}$ are zero, then the corresponding variables are conditionally independent of variable $s$ given all other variables, according to (1).

Inspired by the $l_1$-based sparse modeling in signal processing, sparse inverse covariance matrix estimation algorithms based on $l_1$ regularization have been proposed recently [2]–[7]. In [5], a neighborhood selection algorithm with LASSO regression [8] is applied to each node in the graph, reducing the original problem to multiple sparse linear regression problems. In [2], an $l_1$ penalty is directly imposed on the inverse covariance matrix to favor sparsity, so that the conditional independencies of the variables can be effectively identified. Motivated by the work on adaptive LASSO for regression [9], an adaptive LASSO penalty has also been considered for sparse inverse covariance matrix estimation [10]. It is shown [10] that the adaptive version enjoys some desirable properties and gives better results. The idea of using adaptive LASSO penalty can also be found in a recent paper [11]. The difference between [10], [11] and our work is that we take a Bayesian approach, and our algorithm is stand-alone, while the above-mentioned methods require usage of the basic graphical LASSO algorithm from [2].

In another line of research, sparse Bayesian learning has been applied successfully in many areas, for example, compressive sensing and dictionary learning. The $\ell_1$ regularization is equivalent to imposing a Laplacian prior on the variables to be estimated, which yields the Bayesian LASSO model [12] for linear regression. In addition, a Bayesian adaptive LASSO model was proposed in [13] to enable adaptive shrinkage and avoid biased estimation.

We propose a Bayesian hierarchical model for sparse inverse covariance matrix learning. Bayesian adaptive LASSO priors are imposed on off-diagonal elements of the inverse covariance matrix with different regularization parameters.

The positive-definiteness constraint makes the estimation of a sparse inverse covariance matrix challenging. A first-order condition was derived in [14] under which positive definiteness is guaranteed. This condition arises naturally in our Bayesian inference algorithm, hence positive definiteness is guaranteed in our algorithm. The objective function in [14] is similar to our model. The main difference is that a Bayesian framework is used in our model, while an optimization
approach is used in [14]. As for most optimization approaches, regularization parameters have to be chosen through cross-validation in [14]. This tuning process is avoided in Bayesian models, where sparseness-promoting priors with non-informative hyper-parameters are imposed.

Bayesian hierarchical models have been adopted to infer a sparse GGM. In [15] a dependency network was constructed by predicting each variable using a small number of other variables. A Bayesian variable selection technique was used to select the predictors. This approach is equivalent to sparse Cholesky decomposition modeling of the inverse covariance matrix. A similar idea was exploited in [16]. In contrast, the Bayesian model proposed in this paper directly models the inverse covariance matrix. Due to the property of the Gaussian distribution, the dependency network can be expressed directly as in (1), without the need for the Cholesky component.

Another Bayesian model proposed recently for sparse GGM learning is [17]. This is a two-step approach. In the first step, a Bayesian hierarchical regression model was proposed to learn the dependency network, with sparseness-promoting priors imposed on the edges. In the second step, with the inferred graph structure, a standard optimization method (graphical LASSO [2]) was used to learn the inverse covariance matrix. There are two main differences between this approach and our approach. First, as mentioned in their paper, the Bayesian model used in the first step is a pseudo-likelihood model rather than a proper generative model for the data. However, our model is a valid generative model for the data. Secondly, as mentioned at the end of their paper, a future direction of their work is to combine the two-step process into one. Our model addresses this challenge, as we learn the structure of the graph and the inverse covariance matrix simultaneously, in one step.

A unique aspect of our paper is that we study the graph-extension problem. Suppose a \( p \times p \) inverse covariance matrix is inferred using training data. Now given an additional testing variable or node, the question is how to grow the matrix to dimension \( (p+1) \times (p+1) \). In order for the extended graph to be consistent with the original graph, the covariance matrix for the original \( p \) variables is fixed as the training result, while a sparseness constraint is imposed on the extended part of the inverse covariance matrix. The resulting algorithm is a sparse regression for the additional variable using the original variables, with positive definiteness constraints.
The remainder of the paper is organized as follows. In Section II we provide detailed specifications for the Bayesian Gaussian Graphical Model, and discuss variational Bayesian inference and computational details. In Section III we discuss model properties, extensions, and relationships to existing models. An extensive set of experimental results are presented in Section IV, followed by conclusions in Section V.

II. MODEL SPECIFICATION AND VARIATIONAL BAYESIAN INFERENCE

A. Model construction

Define $X = [X_1, X_2, \cdots, X_n] \in \mathbb{R}^{p \times n}$ to be the data matrix with $p$ variables and $n$ sample points. A joint density function for the problem can be expressed as

$$p(X, J) = p(X|J)p(J) = \prod_{i=1}^{n} N(X_i; 0, J^{-1})p(J).$$

A Laplacian prior is imposed on the off-diagonal elements $J_{st}$ ($s \neq t$) to induce sparsity [12]:

$$p(J) = \prod_{s=1}^{p} \prod_{t<s} \frac{\sqrt{\tau \gamma_{ts}}}{2} \exp(-\sqrt{\tau \gamma_{ts}}|J_{st}|)$$

with symmetry constraint $J_{st} = J_{ts}$; $\tau$ is a global regularization parameter, and it is generally fixed to be one [10]. It is important to note that a separate parameter $\gamma_{ts}$ is used on each component of $J$, and therefore the components of $J$ are not drawn i.i.d. from a single Laplace prior, but rather non-i.i.d., with component dependent parameters $\gamma_{ts}$. This non-i.i.d. construction mitigates recent observations about the inconsistency of $\ell_1$ regularizer (Lasso) [18] and the fact that an i.i.d. Laplace prior does not promote sparsity. The analog to using separate $\gamma_{ts}$ on each component of $J$ is related to weighted (or adaptive) Lasso [9].

The log likelihood of the model in (2) can be expressed as

$$\log p(X, J) \propto \frac{n}{2} \left( \log \det(J) - \text{tr}(\Sigma J) - \sum_{s=1}^{p} \sum_{t<s} \frac{2}{n} \sqrt{\tau \gamma_{st}}|J_{st}| \right)$$

with $\Sigma = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T$ the sample covariance. This shows that our model is effectively imposing a weighted $\ell_1$ regularization on the off-diagonal elements of $J$ [2], [7]. Different
regularization parameters \( (\gamma_{st}) \) are used for \( J \) to foster adaptive regularization [9], [10]. Besides the Laplacian shrinkage prior, other priors could also be used, for example, the Student-t prior [19] and spike-and-slab prior [20].

The Laplacian prior in (3) is not analytic for Bayesian inference. Following [12], we use a scale mixture of normals representation to avoid this problem:

\[
\frac{\sqrt{\tau_2}}{2} \exp(-\sqrt{\tau_2} |J_t|) = \int \mathcal{N}(J_t; 0, \tau^{-1} \alpha^{-1}) \text{InvGa}(\alpha; 1, \frac{\gamma_2}{2}) d\alpha
\]

where \( \text{InvGa}(x; g, h) = \frac{h^g}{\Gamma(g)} x^{-g-1} \exp\left(-\frac{h}{x}\right) \) \( (x > 0) \) denotes the inverse gamma distribution. In addition, a gamma prior is imposed on \( \gamma_{ts} \) to infer this regularization parameter adaptively within the Bayesian framework. In contrast, this parameter is found via cross-validation in the optimization based approaches [2], [10]. The full model can be expressed as

\[
p(X, J, \alpha, \gamma) = p(X|J)p(J|\alpha)p(\alpha|\gamma)p(\gamma) \times 1(J \in S_+)/C
\]

\[
= \prod_{i=1}^{n} \mathcal{N}(X_i; 0, J^{-1}) \prod_{s=1}^{p} \prod_{t<s} \mathcal{N}(J_{ts}; 0, \tau^{-1} \alpha^{-1}) \text{InvGa}(\alpha; 1, \frac{\gamma_{ts}}{2}) \text{Ga}(\gamma_{ts}; a_0, b_0) \times 1(J \in S_+)/C
\]

where \( 1(J \in S_+) \) defines a feasible region of the set of positive definite matrices, and \( C = \int \int \int p(J|\alpha)p(\alpha|\gamma)p(\gamma) \times 1(J \in S_+) dJ d\alpha d\gamma \) is the normalizing constant. In practice, we do not need to evaluate this constant when performing VB inference. The Normal-InvGa-Ga prior on \( J_{st} \) is called Bayesian adaptive LASSO, or Normal-Exponential-Gamma (NEG) prior [13]. Hyperparameters are set as \( a_0 = b_0 = 10^{-6} \), thereby imposing a “flat” (essentially non-informative) prior.

**B. Variational Bayesian Inference**

We seek a variational Bayesian (VB) approximation to the posterior distribution [21]. In a variational Bayesian analysis, one approximates the posterior distribution as the product (factorization) of distributions with associated hyperparameters, and one optimizes these hyperparameters to minimize the Kullback-Leibler divergence between the actual posterior distribution and the factorized approximation. This minimization can be achieved by employing a variational
representation, with details discussed in [21]. In this paper we employ the following factorization

\[ p(J, \alpha, \gamma | X) = \prod_{s=1}^{p} \delta(J_{ss} - J_{ss}^*) \delta(J_{ss} - J_{ss}^*) q(\alpha_{ss}) q(\gamma_{ss}) \]  

(7)

where again \( \tilde{s} \) denotes all variable indices except \( s \). A point estimate (hence the delta functions in (7)) is used for the inverse covariance matrix, for convenience of inference. The density functions \( q(\alpha_{ss}) \) and \( q(\gamma_{ss}) \) correspond to

\[ q(\alpha_{ss}) = \prod_{t<s} \text{InvGa}(\alpha_{ts}; a_{ts}, b_{ts}) \]  

(8)

\[ q(\gamma_{ss}) = \prod_{t<s} \text{Ga}(\gamma_{ts}; c_{ts}, d_{ts}) \]  

(9)

and we seek to infer the hyperparameters \( \{a_{ts}, b_{ts}, c_{ts}, d_{ts}\} \), performed via an iterative procedure [21] that is guaranteed to converge.

Using identity

\[ \mathcal{N}(X_i; 0, J^{-1}) = \mathcal{N}(X_{\tilde{s}i}; -J_{\tilde{s}s}^{-1} J_{s\tilde{s}} X_{si}, J_{\tilde{s}s}^{-1}) \times \mathcal{N}(X_{si}; 0, (J_{ss} - J_{s\tilde{s}}^\top J_{\tilde{s}s}^{-1} J_{\tilde{s}s})^{-1}) \]  

(10)

the VB update equations are

1) Update for \( J \):

\[ (J_{ss}^*, J_{\tilde{s}s}^*) = \arg \max_{J_{ss}, J_{\tilde{s}s}} \int \log \left( \prod_{i=1}^{n} \mathcal{N}(X_{\tilde{s}i}; -J_{\tilde{s}s}^{-1} J_{s\tilde{s}} X_{si}, J_{\tilde{s}s}^{-1}) \times \mathcal{N}(X_{si}; 0, (J_{ss} - J_{s\tilde{s}}^\top J_{\tilde{s}s}^{-1} J_{\tilde{s}s})^{-1}) \times \mathcal{N}(\alpha_{ss}; 0, \tau^{-1} \text{diag}^{-1}(\alpha_{\tilde{s}s})) \times 1(J \in S_+) \right) \]

\[ \times q(\alpha_{ss}) \delta(J_{\tilde{s}s} - J_{\tilde{s}s}^*) \, d\alpha_{ss} \, dJ_{\tilde{s}s} \]  

(11)

We first consider the unconstrained optimization problem without the term \( 1(J \in S_+) \).

By taking derivatives with respect to \( (J_{ss}, J_{\tilde{s}s}) \) simultaneously, we can obtain

\[ J_{ss}^* = (n \Sigma_{ss} J_{\tilde{s}s}^{-1} + \tau \text{diag}(\langle \alpha_{\tilde{s}s} \rangle))^{-1} (-n \Sigma_{ss}) \]  

(12)

\[ J_{ss}^* = \Sigma_{ss}^{-1} + J_{ss}^\top J_{\tilde{s}s}^{-1} J_{\tilde{s}s} \]  

(13)
We then find that the above solution always lies in the feasible region \( J \in S_+ \). See Section III-A for details.

2) Update for \( \alpha \):

\[
q(\alpha_{\cdot s}) \propto \exp \left( \int \log(\mathcal{N}(J_{\cdot s}; 0, \tau^{-1}\text{diag}^{-1}(\alpha_{\cdot s}))) \right.
\times \prod_{t \neq s} \text{InvGa}(\alpha_{ts}; 1, \frac{\gamma_{ts}}{2}) \times q(\gamma_{\cdot s}) \delta(J_{\cdot s} - J^*_{\cdot s}) \text{d}J_{\cdot s} \text{d}J_{\cdot s}
\]

\[
= \prod_{t \neq s} \text{InvGaussian}(\alpha_{ts}; \sqrt{\frac{\langle \gamma_{ts} \rangle}{\tau J_{ts}^2}}, \langle \gamma_{ts} \rangle)
\]

where

\[
\text{InvGaussian}(x; g, h) = \sqrt{\frac{h}{2\pi x^3}} \exp\left(-\frac{h(x-g)^2}{2g^2x}\right) \quad (x > 0)
\]

\( (x > 0) \) denotes the inverse Gaussian distribution with mean \( \langle x \rangle = g \) and \( \langle x^{-1} \rangle = g^{-1} + h^{-1} \).

3) Update for \( \gamma \):

\[
q(\gamma_{\cdot s}) \propto \exp \left( \int \log(\prod_{t \neq s} \text{InvGa}(\alpha_{ts}; 1, \frac{\gamma_{ts}}{2}) \times \text{Ga}(\gamma_{ts}; a_0, b_0))q(\alpha_{\cdot s}) \text{d}\alpha_{\cdot s} \right.
\]

\[
= \prod_{t \neq s} \text{Ga}(\gamma_{ts}; a_0 + 1, b_0 + \frac{1}{2} \langle \alpha_{ts}^{-1} \rangle)
\]

where \( \text{Ga}(x; g, h) = \frac{h^a}{\Gamma(a)} x^{a-1} \exp(-hx) \quad (x > 0) \) denotes the Gamma distribution with \( \langle x \rangle = \frac{g}{h} \).

The above iterations only update the \( s \)th column of \( J \), and we sweep all columns to update the whole inverse covariance matrix. The * symbol on \( J \) is omitted for brevity in the following discussion.

C. Computational details

Computation of \( J_{\cdot s}^{-1} \) is required in (12) in each iteration. However, by maintaining a global covariance matrix \( W = J^{-1} \), this matrix inversion can be avoided. The following matrix
Algorithm 1 Bayesian Learning of Sparse GGM

Input: Sample covariance $\Sigma$, sample size $n$.
Initialize $J$, $W = J^{-1}$, $\alpha$ and $\gamma$.

repeat
\[ s = 1 \text{ to } p \]
\[ J^{-1}_{ss} = W_{ss} - W_{ss}W_{ss}^{-1}W_{ss}^{T}, \]
\[ J_{ss} = (n\Sigma_{ss}J_{ss}^{-1} + \tau \text{diag}(\langle \alpha_{ss} \rangle))^{-1}(-n\Sigma_{ss}). \]
\[ J_{ss} = \Sigma_{ss}^{-1} + J_{ss}^{-1}J_{ss}. \]
\[ W_{ss} = (J_{ss} - J_{ss}^{T}J_{ss}^{-1}J_{ss})^{-1}. \]
\[ W_{ss} = -J_{ss}^{-1}J_{ss}^{T}W_{ss}. \]
\[ W_{ss} = J_{ss}^{-1} + W_{ss}W_{ss}^{-1}W_{ss}^{T}. \]
\[ q(\alpha_{ss}) = \prod_{t \neq s} \text{InvGaussian}(\alpha_{ts}; \sqrt{\langle \gamma_{ts} \rangle \tau J_{ts}^{-1}}, \langle \gamma_{ts} \rangle). \]
\[ q(\gamma_{ss}) = \prod_{t \neq s} \text{Ga}(\gamma_{ts}; a_{0} + 1, b_{0} + \frac{1}{2}(\alpha_{ts}^{-1})). \]
end for
until Maximum number of iteration is reached.

inversion identity is used to achieve this:
\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}^{-1} =
\begin{pmatrix}
B^{-1} & -A_{11}^{-1}A_{12}D^{-1} \\
-A_{22}^{-1}A_{21}B^{-1} & D^{-1}
\end{pmatrix}
\]
with $B = A_{11} - A_{12}A^{-1}_{22}A_{21}$ and $D = A_{22} - A_{21}A^{-1}_{11}A_{12}$. We can obtain $J_{ss}^{-1}$ from the global covariance matrix $W$ via identity
\[ J_{ss}^{-1} = W_{ss} - W_{ss}W_{ss}^{-1}W_{ss}^{T} \]
and after applying (12) and (13), we can efficiently update $W$ using the new $J_{ss}$ and $J_{ss}$ via identities:
\[ W_{ss} = (J_{ss} - J_{ss}^{T}J_{ss}^{-1}J_{ss})^{-1}; \quad W_{ss} = -J_{ss}^{-1}J_{ss}W_{ss}; \quad W_{ss} = J_{ss}^{-1} + W_{ss}W_{ss}^{-1}W_{ss}^{T} \]
Although the inversion is avoided, we do need to solve the linear systems in (12). The overall algorithm is summarized in Algorithm 1.
III. PROPERTIES AND EXTENSIONS

A. Proof of Positive Definiteness and Convergence

In the inference, matrix $J$ is always symmetric since we imposed this symmetry constraint in the model. Equation (13) in the VB inference coincides with the first-order condition obtained in [14], which guarantees positive definiteness of $J$. This is proved in the following theorem.

**Theorem.** Let $J^{(0)}$ be initially a symmetric positive definite matrix, and $\Sigma_{ss} > 0$, $\forall s$. Then the variational Bayesian algorithm yields a sequence of updates $J^{(0)}, J^{(1)}, J^{(2)}, ..., J^{(t)}, ...$ that converges to a positive definite matrix $J$.

**Proof:** In each iteration, whenever a certain column of $J$ is updated using (12) and (13), the updated $J$ can be written as

$$J_{\text{new}} = \begin{pmatrix} J_{ss} & J_{s\bar{s}} \\ J_{s\bar{s}}^T \Sigma_{ss}^{-1} + J_{\bar{s}s}^{-1} J_{\bar{s}s} & J_{ss} \end{pmatrix} = \begin{pmatrix} J_{\bar{s}\bar{s}}^{-1} \left( J_{ss} J_{\bar{s}s} \right) + \begin{pmatrix} 0 & 0 \\ 0 & \Sigma_{ss}^{-1} \end{pmatrix} \end{pmatrix} \quad (14)$$

By induction, $J_{\bar{s}\bar{s}}$ is positive definite. Also, $\Sigma_{ss}^{-1} > 0$, as $\Sigma_{ss}$ is the sample variance on the $s$-th coordinate. Hence $J_{\text{new}}$ is nonnegative definite. However, $\det(J_{\text{new}}) = \det(J_{ss}) \Sigma_{ss}^{-1} > 0$. Therefore, $J_{\text{new}}$ is positive definite. This proves that positive definiteness is preserved through all iterations.

The variational Bayesian updates constitute a block coordinate descent that maximizes a lower bound to the marginalized likelihood function. Because the solution of $(J, q(\alpha), q(\gamma))$ in each iteration is unique, by general results on block coordinate descent [22], the solutions must converge. As all updates of $J$ are positive definite, the updates must also converge to a positive definite matrix. Q.E.D.

B. Relation to Other Methods

According to (5) and (6), if we marginalize $\alpha$, the proposed Bayesian model can be expressed as

$$p(X, J, \gamma) = \prod_{i=1}^{n} \mathcal{N}(X_i; 0, J^{-1}) \prod_{s=1}^{p} \prod_{t<s} \frac{\sqrt{\tau \gamma_{ts}}}{2} \exp(-\sqrt{\tau \gamma_{ts}} | J_{ts} |) \text{Ga}(\gamma_{ts}; a_0, b_0) \quad (15)$$
and the log likelihood $\log p(X, J, \gamma)$ is the same as (4) except that we have an additional term for the prior on the weights $\gamma_{ts}$. This underscores the connection of the proposed model with previous adaptive LASSO models [9], [10]. Via the VB representation in (7) we estimate a single (point) approximation to the components of the precision matrix, like in [9], [10]. However, rather than using cross validation to infer (point) estimates on the remaining parameters (weights and Lagrange multipliers in [9], [10], we estimate posterior distributions on $\alpha_{ss}$ and $\gamma_{ss}$. The advantage of this appears to be that the fact that we do not estimate only single values for these latter parameters adds modeling robustness, and the Bayesian formulation has the advantage of not requiring cross validation.

Concerning other previous methods, according to (10) and (11), the likelihood function of $J_{ss}$ can be expressed as

$$p(J_{ss} | -) \propto \exp\left(-\frac{1}{2} (n \Sigma_{ss} J_{ss}^T J_{ss}^{-1} J_{ss} + 2n \Sigma_{ss}^T J_{ss}) \right)$$

Equation (12) is derived based on this likelihood and the prior. This likelihood function is the same as the objective function used in [14]. It can be verified that

$$\exp\left(-\frac{1}{2} (n \Sigma_{ss} J_{ss}^T J_{ss}^{-1} J_{ss} + 2n \Sigma_{ss}^T J_{ss}) \right) = \exp\left(-\frac{1}{2} n J_{ss} (\beta^T W_{ss} \beta - 2 \Sigma_{ss}^T \beta) \right)$$

with $W = J^{-1}$ and $\beta = W_{ss}^{-1} W_{ss} = -J_{ss} J_{ss}^{-1}$. The latter expression is also the objective function used in [2]. The difference of our approach is that Bayesian inference, instead of the $\ell_1$ optimization approach, is used to infer $J_{ss}$. Within the context of the Bayesian analysis we do not require cross-validation, while such is required for the method in [2].

C. Graph Extension

Suppose the above algorithm is used on a training data set $X^{(1)} \in \mathbb{R}^{p \times n}$, and the inferred covariance matrix is $\tilde{W}_{11}$. Now given data from a new testing variable $X^{(2)} \in \mathbb{R}^{1 \times n}$, we want to extend the original graph to include this new node. The augmented data is $X = \begin{pmatrix} X^{(1)} \\ X^{(2)} \end{pmatrix} \in \mathbb{R}^{(p+1) \times (n+1)}$. 
and the corresponding sample covariance is \( \Sigma = \frac{1}{n}XX^\top \). Denote the covariance and inverse covariance matrix for the augmented graph to be \( J \) and \( W \), with the same partition as \( \Sigma \). Then the likelihood of the augmented data can be expressed as

\[
p(X) = p(X^{(2)}|X^{(1)})p(X^{(1)}) = \prod_{i=1}^{n} \mathcal{N}(X_i^{(2)}; -J^{-1}_{22}J_{12}^\top X_i^{(1)}, J^{-1}_{22}), \mathcal{N}(X_i^{(1)}; 0, W_{11})
\]

In order for the extended graph to be consistent with the original one, the marginal distribution of \( X_i^{(1)} \) should remain the same. Thus \( W_{11} = (J_{11} - J_{12}J^{-1}_{22}J_{12}^\top)^{-1} \) is fixed to be the training result \( \tilde{W}_{11} \). By imposing a Laplacian prior on \( J_{12} \) like (6), the above problem becomes a sparse regression for the additional variable using the original ones. Using VB inference, the point estimate of \( J_{12} \) can be expressed as

\[
J_{12} = (nJ^{-1}_{22}\Sigma_{11} + \tau \text{diag}(\langle \alpha_{12} \rangle))^{-1}(-n\Sigma_{12}) \quad (16)
\]

The update of \( \alpha_{12} \) and \( \gamma_{12} \) are similar to equations in Section II-B.

The same positive definiteness constraint in (13) is imposed here: \( J_{22} = \Sigma^{-1}_{22} + J_{12}J^{-1}_{11}J_{12}^\top \), or equivalently \( W_{22} = (J_{22} - J_{12}J^{-1}_{11}J_{12}^\top)^{-1} = \Sigma_{22} \). Now given \( J_{12}, W_{22} \) and \( W_{11} \), we would like to update \( J_{22} \). Since \( JW = I \), we have \( W_{11}J_{12} + W_{12}J_{22} = 0 \) and \( W_{12}^\top J_{12} + W_{22}J_{22} = 1 \). Using these two equations we obtain \(-J^{-1}_{22}(J_{12}^\top W_{11}J_{12}) + W_{22}J_{22} = 1 \), and finally since \( J_{22} > 0 \),

\[
J_{22} = \frac{1 + \sqrt{1 + 4W_{22}J_{12}^\top W_{11}J_{12}}}{2W_{22}} \quad (17)
\]

We observe that the covariance learned on the training data \( (W_{11}) \) is used for the update of \( J_{22} \).

After \( J_{12} \) and \( J_{22} \) are inferred according to (16) and (17), \( J_{11} \) can be found as \( J_{11} = W^{-1}_{11} + J_{12}J^{-1}_{22}J_{12}^\top \), and it is still sparse.

IV. RESULTS

For all experiments in this section, we first normalize the data matrix \( X \) to zero mean and unit standard deviation for each variable, then apply the sparse GGM learning algorithm. We may recover the inverse covariance matrix for the original data by using the above mean and standard
deviation. The only hyperparameters that must be set are \((a_0, b_0)\) on the gamma prior, and as discussed above we set \(a_0 = b_0 = 10^{-6}\), as recommended in [19]. These same hyperparameters are used in all examples presented below, with no tuning.

**A. Simulation**

In this simulation study (taken from [2]), four kinds of sparse inverse covariance matrices are generated as follows, with \(J \in \mathbb{R}^{p \times p}\) denoting the inverse covariance matrix.

1) \((J^{-1})_{ts} = 0.7^{|t-s|}\).

2) \(J_{ts} = 1\) if \(|t-s| = 0\); \(J_{ts} = 0.4\) if \(|t-s| = 1\); \(J_{ts} = 0.2\) if \(|t-s| = 2\); \(J_{ts} = 0.2\) if \(|t-s| = 3\); \(J_{ts} = 0.1\) if \(|t-s| = 4\).

3) \(J = B + \delta I\), with \(B_{ts} = 0.5\) with probability \(\alpha = 0.1\) and 0 otherwise; \(\delta\) is chosen to make the condition number of \(J\) equal to \(p\).

4) \(J_{ts} = 2\) if \(|t-s| = 0\); \(J_{ts} = 1\) if \(|t-s| = 1\); \(J_{1p} = J_{p1} = 0.9\).

We choose the dimension \(p\) of the matrix \(J\) to be 30, 100 and 200, and the number of data samples for each data matrix to be 200, 500, 800 and 1000. For each \((p, n)\) combination, we generate 50 data matrices from the above four Gaussian models, each of dimension \(p \times n\), and then apply the Bayesian sparse GGM learning algorithm to each data matrix, and average the performance across the 50 runs. Two performance measures are defined: the Kullback-Leibler loss expressed as \(\text{KL}(J, \tilde{J}) = \text{tr}(J^{-1}\tilde{J}) - \log(J^{-1}\tilde{J}) - p\) where \(\tilde{J}\) denotes the estimated inverse covariance matrix, and the \(\ell_2\) loss defined by the Frobenius norm \(\text{L2}(J, \tilde{J}) = \|J - \tilde{J}\|_F\).

We used the graphical LASSO (GLASSO) algorithm [2] as the baseline, and compared our Bayesian algorithm with it in Figure 2 and 3. Examples of the estimation are shown in Figure 1. The GLASSO algorithm requires tuning of the regularization parameter, so we used a candidate set of \([0.01, 0.02, 0.05, 0.08, 0.1, 0.12, 0.15, 0.18, 0.2]\) for the regularization parameter, and employed ten-fold cross-validation to choose the best candidate in terms of model likelihood, and finally applied the GLASSO algorithm again on the whole dataset using the selected parameter.
From the results we can see that in most cases, the proposed algorithm outperforms the baseline algorithm under both KL and $\ell_2$ measure. The gain over the baseline becomes more obvious as the number of samples increases. The main reason for this performance gain is that in our proposed model, each element of the inverse covariance matrix has a specific regularization parameter, making the shrinkage more flexible and the estimation more consistent [9], [10]. We also observe that the advantage is not obvious when the number of samples $n$ is small and the model dimension $p$ is large (e.g., $n = 200$ and $p = 200$), in which case no algorithm can give a reliable estimation.

Fig. 1. Examples of simulation results ($p = 30$) for model 1 to 4 (from left to right and from top to bottom). Within each model, the true inverse covariance matrix, the empirical (maximum likelihood) estimation, the GLASSO estimation and the proposed estimation are displayed, from left to right and from top to bottom.

B. Senate Data

The Senate Voting Records Data from the 109th congress (2004 – 2006) is studied in this section. It was previously used in [3]. There are 102 senators, 46 who are Democratic and 56
Fig. 2. KL loss (mean and standard deviation) for the baseline algorithm and the proposed algorithm. Each row represents one of the four models with different model dimension, and each subfigure shows the KL loss of the two algorithms as a function of the number of data samples.

Fig. 3. $\ell_2$ loss (mean and standard deviation) for the baseline algorithm and the proposed algorithm. Each row represents one of the four models with different model dimension, and each subfigure shows the $\ell_2$ loss of the two algorithms as a function of the number of data samples.

who are Republican. Each of the senators voted on 645 bills, with yes recorded as 1 and no −1. The missing values are imputed with −1, as was done in [3]. The sample inverse covariance matrix (i.e., the ML estimate of $J$) and the inferred sparse inverse covariance are plotted in
Figure 4. The senators are ordered so that the first 46 are Democratic and the remaining 56 are Republican. From the sample inverse covariance matrix, the partisan structure is not clear, but from the inferred sparse inverse covariance matrix, this partisanship becomes very clear. Hence our algorithm removes weak dependencies between variables, revealing latent structure in the data.

C. Face Data

The Extended Yale Face Database [23] is a standard dataset with 2414 face images from 38 individuals. It was previously used in [24] for face recognition. We randomly select half of the images as training data and randomly project the $192 \times 168$ dimensional image to 120 dimensions. Algorithm 1 is used to learn the face graph. In Figure 5, we show five randomly selected faces in the first column, and the top six faces linked to each of them in columns two to seven, according to $J$. We observe that the strongest links are from the same individuals.

We can also view the linkage of individual groups quantitatively. Denote $X_s$ to be the $s$th image from individual $l_s$. According to (1), each image can be predicted by a sparse combination of the remaining ones, i.e., $p(X_s|X_{\bar{s}}) = \mathcal{N}(X_s; \sum_{t \neq s}(-J_{ss}^{-1}J_{ts})X_t, J_{ss}^{-1}I)$. We define that $X_s$ is predicted to be from individual $c_s$ if the linkages from individual $c_s$ contribute most in predicting it, i.e.,

$$c_s = \arg \min_c \|X_s - \sum_{t \neq s, l_t = c} (-J_{ss}^{-1}J_{ts})X_t\|^2.$$  \hspace{1cm} (18)
Fig. 5. Example of faces (column one) and the top six links (column two to seven) in the inferred Gaussian graph.

Using the true label $l_s$ and the predicted label $c_s$, we define a confusion matrix $G$ as

$$G_{uv} = \sum_{s : l_s = u} 1(c_s = v) / \sum_{s : l_s = u} 1$$

(19)

and display this Hinton map (the size of the boxes are proportional to the associated matrix value) in Figure 6 (left), with $96.6\%$ of the faces grouped correctly.

For a new testing image with unknown label, we apply the graph extension algorithm in
Section III-C on the trained graph, and again use (18) and (19) to classify this new image. Our graph extension approach reduces to the sparse representation approach proposed in [24], when the trained covariance matrix is simply replaced by the sample covariance. Comparable testing performance is obtained using our graph extension approach, as shown in the classification Hinton map in Figure 6 (right) with an accuracy rate of 94.7%. The advantage of our approach is that we can explicitly construct a graph for the training data, as partially depicted in Figure 5. This is important when the primary goal is to learn the relationships between variables, as demonstrated on the cell line data in the next section.

![Classification Hinton map](image)

**Fig. 6.** Classification Hinton map of the individuals for the training (left) and testing (right) face dataset. The block-diagonal form is consistent with the way the faces were ordered.

### D. Cell Line Data

Human Genome Diversity Cell Line Panel data was collected by [25] to study human genomic variations among geographic regions. It contains 1056 individuals from seven geographic regions, with genotypes sampled at 377 autosomal microsatellite loci. Algorithm 1 is applied to this dataset to learn a $1056 \times 1056$ dimensional sparse inverse covariance matrix. With the geographic labels for the individuals, we apply the same criterion in (18) and (19) to obtain a confusion matrix for the geographical regions, which is displayed in Figure 7. We observe a noticeable sharing among individuals from Central South Asia, Europe and Middle East, consistent with
findings in [26] where a factor model is used for the analysis. This demonstrates that the sharing structure of the variables can be uncovered by the sparse GGM learning algorithm.

Fig. 7. Confusion matrix on cell line data. 1:Africa, 2:America, 3:Central South Asia, 4:East Asia, 5:Europe, 6:Middle East, 7:Oceania.

E. Telephone Call Center Data

The telephone call center data was used as a real data example for inverse covariance matrix estimation in [10]. The data is stored in matrix $\mathbf{Y}$ with $Y_{ti} = \sqrt{N_{ti} + \frac{1}{4}}$, where $N_{ti}$ is the number of calls arriving at the call center for the interval $t$ on day $i$ ($t = 1, 2, \ldots, 102; i = 1, 2, \ldots, 239$). Each data sample $\mathbf{Y}_i \in \mathbb{R}^{102 \times 1}$ is modeled as a multivariate Gaussian as $\mathbf{Y}_i \sim \mathcal{N}(\mathbf{Y}_i; \mu, \mathbf{J}^{-1})$. The data matrix is split into a training and a testing set. The training set contains data for the first 205 days, and it is used for model estimation. The sample mean estimate is used for $\mu$ and the proposed Bayesian model estimate is used for $\mathbf{J}$. For each data sample $\mathbf{Y}_i \in \mathbb{R}^{102 \times 1}$ in the testing set ($i = 206, \ldots, 239$), the last 51 dimensions are assumed missing, and only the first 51 dimensions are observed. To recover the missing data using the observed data and the trained
model, the following equation (similar to (1)) is used

\[ p(Y_{2i}|Y_{1i}) = \mathcal{N}(Y_{2i}; \mu_2 - J_{22}^{-1} J_{21} (Y_{1i} - \mu_1), J_{22}^{-1}) \]

where \( Y_{2i} \) denotes the missing data and \( Y_{1i} \) denotes the observed data. The averaged performance measure is

\[ \text{Error} = \frac{1}{51 \times 34} \sum_{i=206}^{239} \| Y_{2i} - \tilde{Y}_{2i} \|_1 \]

where \( \| \cdot \|_1 \) denotes the \( \ell_1 \) norm, \( Y_{2i} \) is the ground truth and \( \tilde{Y}_{2i} = \mu_2 - J_{22}^{-1} J_{21} (Y_{1i} - \mu_1) \) is the imputation according to the model.

The inverse covariance matrix estimated using the proposed Bayesian method is plotted in Figure 8. This shows that data in nearby temporal intervals within one day are very dependent, and data temporally far away are less dependent. The prediction error in the testing set is 1.336, which is comparable to the adaptive LASSO result (1.34) in [10] and better than the graphical LASSO result (1.39). In the adaptive LASSO method, firstly the shrinkage parameters are estimated using a consistent estimate for \( J \), and then with the shrinkage parameters fixed, the final adaptive LASSO estimate for \( J \) is obtained. In contrast, our method infers \( J \) and the shrinkage parameters simultaneously in one step.

Fig. 8. Inverse covariance matrix learned on the telephone call data.
V. Conclusions

A new Bayesian formulation has been proposed for sparse inverse covariance matrix estimation. The Laplacian prior is imposed on the off-diagonal elements of the inverse covariance matrix, and a Variational Bayesian (VB) inference algorithm is derived; the Laplacian prior is implemented in a hierarchical Bayesian manner. Properties of symmetry and positive definiteness are guaranteed in the estimation. The inferred sparse inverse covariance matrix can be interpreted as a sparse Gaussian graph, with conditional dependencies among variables expressed by the elements of the inferred matrix. In addition, a graph-extension algorithm is proposed to include a new variable into an inferred graph, which can be used for learning on unseen testing data. Experiments on simulated and real datasets show that the proposed algorithm yields accurate and interpretable results.

The main advantage of this Bayesian learning algorithm is that no cross-validation is needed for the regularization parameters, as we learn them adaptively in the Bayesian framework. This means that more data samples can be used for the estimation. This tuning-free algorithm can be readily used in a broad range of real-world applications, where the major goal is to learn relationships among different variables. It could also be used as a building block for more complicated hierarchical models. Our future work will apply the proposed method for missing data imputation [27] and manifold embedding problems.

References