Adaptive Variable Clustering in Gaussian Graphical Models

Siqi Sun*
TTI Chicago

Yuancheng Zhu*
University of Chicago

Jinbo Xu
TTI Chicago

Abstract

Gaussian graphical models (GGMs) are widely-used to describe the relationship between random variables. In many real-world applications, GGMs have a block structure in the sense that the variables can be clustered into groups so that inter-group correlation is much weaker than intra-group correlation. We present a novel nonparametric Bayesian generative model for such a block-structured GGM and an efficient inference algorithm to find the clustering of variables in this GGM by combining a Gibbs sampler and a split-merge Metropolis-Hastings algorithm. Experimental results show that our method performs well on both synthetic and real data. In particular, our method outperforms generic clustering algorithms and can automatically identify the true number of clusters.

1 INTRODUCTION

Gaussian graphical models (GGMs) [11] are widely used to describe real world data and have important applications in various fields such as computational biology, spectroscopy, climate studies, etc. Learning the structure of GGMs is a fundamental problem since it helps uncover the relationship between random variables and allows further inference. It is well known that the structure of a GGM, i.e., the conditional dependence of the underlying Gaussian vector, is encoded only by the zero pattern of its precision matrix. A straightforward method to estimate the precision matrix is to invert the empirical covariance matrix. In addition to the singularity issue when the dimension p is larger than the number of samples n, the precision matrix resulting from this method is usually not sparse and thus, the learned structure may greatly deviate from the real one. Graphical Lasso (Glasso) is a popular approach for the estimation of the structure of a GGM. Glasso maximizes the log-likelihood while penalizing the L_1 norm of the precision matrix [2] [15] [3], which is used to favor a sparse graph.

In many real-world applications the underlying graph or network that we want to estimate has block structure such that it can be divided into blocks where the inter-block dependence is much weaker than the intra-block dependence. For example, in protein-protein interaction networks, proteins with similar functions are more likely to form a pathway or a complex [12]. Therefore it is of great interest to learn such a block-structured graph, which is also equivalent to clustering the variables into disjoint groups. Actually, the clustering would not be hard as long as we could estimate the graph accurately since we could simply use the connected components of the estimated graph as a clustering of variables. However, almost all the graph estimation methods such as Glasso require some predefined parameters controlling the sparsity of the graph and different values of the parameters may lead to quite different clustering results. We may also apply those generic clustering algorithms such as k-means to the variables. However, these clustering algorithms are mainly designed for clustering observations rather than variables and they cannot differentiate direct couplings of variables from indirect couplings.

Some studies have been done to simultaneously infer the block structure of GGMs and estimate the precision matrix [6] [7] [1]. Such methods, however, require a predefined parameter for the number of clusters and the inference is based on some variational approximation.

In this paper, we present a method to estimate the block structure of GGMs and cluster the variables from a Bayesian point of view. Our method is attractive in several aspects. First of all, our model is parameter-free in that we do not have to tune any parameter, especially the number of clusters. Secondly, using a

Appearing in Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

*These two authors contributed equally to this work

MCMC sampling method, we directly sample from the posterior distribution rather than its approximation obtained by variational methods. In addition, we have also described an efficient greedy strategy to find the finest clustering of the variables.

The rest of the paper is organized as follows. In section 2 we introduce some background and related work. Our model and methods for inference are described in section 3 and 4, respectively. Experimental results for both synthetic and real data are presented in section 5, followed by a conclusion.

2 RELATED WORK

Suppose that $X = (X_1, X_2, ..., X_p)$ follows a p-dimensional multivariate Gaussian distribution. For simplicity we assume $X \sim N(0, \Sigma)$, and let $\Omega = [\Omega_{ij}]_{p\times p} = \Sigma^{-1}$ be its precision matrix. It is easy to prove that X_i and X_j are conditionally independent given all the other random variables if and only if $\Omega_{ij} = 0$. Therefore, estimating the structure of a Gaussian graphical model (GGM) is equivalent to estimating the zero pattern in Ω.

Banerjee et al. [2] and Yuan and Lin [15] independently proposed a technique that can estimate the sparse precision matrix. They achieved this by maximizing the L_1 penalized log likelihood, i.e.

$$\Omega = \arg\max_{\Omega \succ 0} \log\det(\Omega) - \mathrm{tr}(\Omega\hat{\Sigma}) - \lambda\|\Omega\|_1$$

where λ is the tuning parameter, $\|\Omega\|_1 = \sum |\Omega_{ij}|$ and $\hat{\Sigma} = \frac{1}{n}X^T X$ is the empirical covariance matrix. The problem can then be solved by a block coordinate descent algorithm called graphical Lasso (Glasso) [3].

Not much work has been done for learning the block structure in a GGM. When the block structure information is not known a priori, all the existing studies employ a Bayesian approach, partially because it is hard to design a penalty term to enforce the block structure without leading to a computationally intractable problem. An example of such work is by Marlin and Murphy [6], who propose a Bayesian model, use a stochastic block model as prior and then use variational Bayes to do inference. Further, they employ a heuristic method to determine the number of clusters. This method starts by putting all the variables in a single cluster, and then split clusters iteratively to increase the free energy. After computing the marginal MAP clustering information, they use group Lasso [14] to infer the precision matrix.

In another two similar approaches to learn a block-structured GGM, Marlin et al. [7] and Ambroise et al. [1] use latent variables to indicate group membership and Laplace distributions as the priors for the precision matrix entries. The group membership information is used to choose the hyperparameters of the prior distributions. An EM algorithm and a variational algorithm are then used, respectively, to learn the structure and estimate the graph.

Another relevant method is Dirichlet process variable clustering (DPVC) proposed by Palla et al. [9]. This work considers the variable clustering problem in a factor model setting and uses nonparametric Bayesian methods to cluster the variables. Specifically, they consider the model where the p variables can be generated as follows.

$$X_j = g_j Y_{z_j} + \epsilon_j,\ j = 1, \ldots, p$$

where z_j is the membership of the jth variable, Y_z is a Gaussian distributed latent factor for group z, g_j is the factor loading, and ϵ_j is a Gaussian noise. In fact, X generated by this model forms a block-structured GGM and thus can be viewed as a special case of the model to be presented below.

3 THE NONPARAMETRIC BAYESIAN MODEL

We consider the problem of clustering the variables of a Gaussian graphical model. Suppose that Ω, the precision matrix of $X = (X_1, \ldots, X_p)$, is block diagonal after some permutation. This is equivalent to assuming that the variables can be grouped into several clusters, and that the edges in the underlying graph only exist within each cluster. The clustering structure can be relaxed to a more general setting where a relatively small number of edges exist between clusters or the inter-cluster edges carry much smaller weight. We now propose a nonparametric Bayesian approach to model such settings.

3.1 Model

Suppose that $Z = (Z_1, Z_2, \ldots, Z_p)$ are hidden variables indicating the membership of $X_1, \ldots, X_p$, i.e., the X_i and X_j are in the same cluster if and only if $Z_i = Z_j$. In fact, Z defines a partition over the set $\{1, \ldots, p\}$. We assume that $Z_1, \ldots, Z_p$ are generated by a Chinese restaurant process CRP(α) [10] where α is the concentration parameter, controlling how diverse the clustering tends to be. The Chinese restaurant process defines a distribution over random partitions of positive integers, with the possible number of clusters being infinite. Specifically, $Z_1, \ldots, Z_p$ are exchangeable and can be sampled sequentially by the following conditional probability.

$$P(Z_i | Z_{1:i-1}, \alpha) = \begin{cases} \frac{\sum_{j<i} \mathbb{1}_{Z_j = Z_i}}{i-1+\alpha} & \exists j < i : Z_j = Z_i \\ \frac{\alpha}{i-1+\alpha} & \forall j < i : Z_j \neq Z_i \end{cases}.$$

Further, when only considering the first p elements, a specific partition $\varrho = (z_1, \ldots, z_p)$ is assigned with the following probability.

$$P(\varrho) = \frac{\alpha^{\#\text{clusters}}\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{\text{cluster}\in\varrho} \Gamma(|\text{cluster}|).$$

For a given clustering Z, we assume that the precision matrix Ω is from a Wishart distribution defined over symmetric positive semidefinite matrices. As a prior distribution for the precision matrix, the Wishart distribution is conjugate to the multivariate Gaussian likelihood. The density function of $\Omega \sim \text{Wishart}_p(V, \nu)$ is

$$P(\Omega|V,\nu) = \frac{|\Omega|^{\frac{\nu-p-1}{2}} \exp\{-\frac{1}{2}\text{tr}(V^{-1}\Omega)\}}{2^{\frac{\nu p}{2}}|V|^{\frac{\nu}{2}}\Gamma_p(\frac{\nu}{2})}$$

where $\Gamma_p(\cdot)$ is the multivariate Gamma function, V is known as the scale matrix and ν the degree of freedom. The expectation of $\text{Wishart}_p(V,\nu)$ is νV. Here, to reflect our knowledge about the clustering pattern based on Z, we set the scale matrix V to have a block diagonal structure. In particular, we let

$$V = V(Z,W) = \begin{cases} W_{ij}/\nu & \text{if } Z_i = Z_j \\ 0 & \text{if } Z_i \neq Z_j \end{cases}$$

where W is a prior guess of the precision matrix and we scale it by a factor of $1/\nu$ so that the expectations of remaining entries will be the same as in W.

Thus, we have introduced a generative model to form a Gaussian graphical model with clustered variables. As shown in Fig. 1, our model can be summarized below.

$$\begin{aligned} Z|\alpha &\sim \text{CRP}(\alpha), \\ \Omega|Z,W,\nu &\sim \text{Wishart}_p(V(Z,W),\nu), \\ X|\Omega &\sim N(0,\Omega^{-1}). \end{aligned}$$

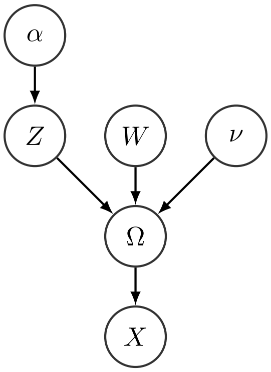

Figure 1: Graphical representation of the generative model

An alternative way to model block-structured GGMs is to assume that the precision matrix Ω, given the clustering information Z, follows a block-wise Wishart distribution. Specifically, suppose that $Z_1, \ldots, Z_p$ take values in $\{1, \ldots, k\}$, and for $z = 1, \ldots, k$, let $\mathcal{I}_z = \{i : Z_i = z\}$ and $p_z = |\mathcal{I}_z|$. Then we can assume the precision matrix Ω is from

$$\Omega_{\mathcal{I}_z} \overset{\text{ind}}{\sim} \text{Wishart}_{p_z}(V_z, \nu_z), \text{ for } z = 1, \ldots, k,$$

$$\Omega_{ij} = 0, \text{ if } Z_i \neq Z_j,$$

where $\Omega_{\mathcal{I}_z}$ is the submatrix of Ω with indices $\mathcal{I}_z$. In other words, the precision matrix is assumed to be block-structured, and each block is assumed to follow a Wishart distribution. Such an approach sets the off-block-diagonal entries of the precision matrix to exactly 0. In practice, this alternative approach also works for the case where weak dependence exists between clusters, and performs similarly as the model we proposed above. Therefore, in this paper, we mainly discuss the model proposed first.

3.2 Hyperparameter

There are three hyperparameters to be specified or tuned in the model, namely, α, W and ν. We discuss below our strategies of choosing them and the underlying reasons. The concentration parameter α of the Chinese restaurant process takes value in $(0,\infty)$. To improve the flexibility of the model, we can put a prior distribution on the hyperparameter α, for which we use Gamma(1,1) throughout this paper. In fact, the inference results are similar with different choices of the priors as long as it has a support $(0,\infty)$.

The Wishart distribution of Ω is characterized by three parameters, Z, W and ν, where Z is obtained from the Chinese restaurant process. Some methods such as empirical Bayes estimation [4] are proposed for the scale matrix without enforcing a block diagonal. We set W to the empirical precision matrix (i.e., $W = \widehat{\Omega}$), which is a widely-used method. For the case when p is smaller than n, we set W to be the Glasso estimator with a small penalization parameter. From now on we will treat W as fixed, and denote $V(Z,W)$ as $V(Z)$.

For the degree of freedom ν, a common choice, which is also the least informative one, would be to set $\nu = p$, the dimension of the matrix. In order to reflect our prior knowledge of the block structure, we set $\nu = \max\{p,n\}$ where n is the sample size. To see why this favors block diagonal structure of the precision matrix, consider the posterior distribution $P(\Omega|Z,\nu,X)$ where X represents n i.i.d. samples. Because of the conjugacy, this is still a Wishart distribution, with expectation

$$\tilde{\Omega} = \left(\frac{\nu V(Z)^{-1} + n\widehat{\Sigma}}{\nu + n}\right)^{-1}$$

where $\widehat{\Sigma}$ is the sample covariance matrix. Notice that $V(Z)^{-1}$ has a block diagonal structure, so the posterior mean somehow preserves the intra-cluster covariance structure while adding some shrinkage on the inter-cluster correlation. By choosing $\nu = \max\{p, n\}$, the shrinkage effect remains consistent for different p and n when $n \geq p$. Besides, when $p < n$, such a choice will introduce more shrinkage on the off-block-diagonal entries, reflecting more strength from the prior knowledge of the block structure when we have insufficient data. Although there are some other sensible choices for the degree of freedom, such as putting a prior with a support on $(p-1, \infty)$, we choose $\nu = \max\{p, n\}$ throughout this paper, which turns out to work well for various settings regardless of p and n.

4 INFERENCE

In this section, we describe the methods we have implemented to achieve variable clustering using the model introduced in section 3.1. Specifically, given the data X, we would like to compute the posterior distribution of the latent variables, with special interest in the clustering information Z. Note that for Z this is a distribution over partitions of $\{1, \ldots, p\}$. Although we can compute the posterior distribution $P(Z|X)$ with other variables integrated out analytically up to a normalization constant, the number of partitions on $\{1, \ldots, p\}$ is known to be the Bell number, which grows faster than exponentially, hence making it computationally intractable to find the normalization constant and to directly sample from the posterior distribution.

4.1 Gibbs Sampler

To explore the posterior distribution over the latent variables, we propose a Gibbs sampling method as follows. We update the elements of Z one at a time. That is, we sample Z_i according to the conditional distribution $P(Z_i|X, \Omega, Z_{-i}, \alpha)$ where $Z_{-i} = (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_p)$. In particular,

$$\begin{aligned}
&P(Z_i|X, \Omega, Z_{-i}, \alpha)\\
\propto &P(X|\Omega)P(\Omega|Z)P(Z_i|Z_{-i}, \alpha)\\
\propto &\frac{|\Omega|^{\frac{\nu-p-1}{2}}}{|V(Z)|^{\frac{\nu}{2}}} \exp\Big(-\frac{1}{2}\mathrm{tr}\big(V(Z)^{-1}\Omega\big)\Big) P(Z_i|Z_{-i}, \alpha) \quad (1)
\end{aligned}$$

where $P(Z_i|Z_{-i}, \alpha)$ is given by

$$P(Z_i = z|Z_{-i}, \alpha) = \begin{cases} \frac{p_{-i,z}}{p-1+\alpha} & \exists j : Z_j = z \\ \frac{\alpha}{p-1+\alpha} & \forall j : Z_j \neq z \end{cases}$$

with $p_{-i,z}$ being the number of elements in cluster z excluding Z_i, i.e., $p_{-i,z} = \sum_{j \neq i} \mathbb{1}_{Z_j = z}$.

To update Ω, we sample from $P(\Omega|X, Z, \alpha)$ as follows.

$$\Omega|X, Z, \alpha \sim \mathrm{Wishart}_p\Big(\big(V(Z)^{-1} + \sum_{i=1}^{n} X_i X_i^T\big)^{-1}, n+\nu\Big). \quad (2)$$

Alternatively, we can sample one element in Z with Ω integrated out, i.e., using the following probability.

$$\begin{aligned}
P(Z_i|X, Z_{-i}, \alpha) &\propto P(X|Z, \alpha)P(Z_i|Z_{-i}, \alpha)\\
&= \int_{\Omega} P(X, \Omega|Z, \alpha) d\Omega P(Z_i|Z_{-i}, \alpha)\\
&\propto \int_{\Omega} P(X|\Omega)P(\Omega|Z) d\Omega P(Z_i|Z_{-i}, \alpha)\\
&\propto \frac{\Gamma_p(\frac{n+\nu}{2})}{\Gamma_p(\frac{\nu}{2})} \frac{|V(Z)|^{\frac{n}{2}}}{|I_p + V\sum_{i=1}^{n} X_i X_i^T|^{\frac{n+\nu}{2}}}.
\end{aligned} \quad (3)$$

Since Z_i is discrete and the Wishart distribution is conjugate, it is easy to update Z and Ω based on Eqs. (1) and (2), or update Z based on Eq. (3). We will use the latter one as our "default" Gibbs sampler.

To update the hyperparameter α, we compute

$$\begin{aligned}
P(\alpha|X, Z) &\propto P(X|Z)P(Z|\alpha)P(\alpha)\\
&\propto \frac{\alpha^{\#\mathrm{cluster}(Z)}\Gamma(\alpha)}{\Gamma(p+\alpha)} P(\alpha). \quad (4)
\end{aligned}$$

This is a univariate distribution and we sample from it using slice sampling [8].

With the conditional probability defined above, we have a Gibbs sampler for drawing samples from the posterior distribution of the latent variables Z.

4.2 Split-merge Metropolis-Hastings updates

As mentioned in [5], the above-proposed Gibbs sampler may be inefficient. Because the Gibbs sampler updates the cluster membership incrementally, the Markov chain must pass through a series of low-probability states to traverse between two isolated posterior modes. This leads to slow convergence and slow movement between two posterior modes. To tackle this limitation, we incorporate into our Gibbs sampler a split-merge Metropolis-Hastings procedure as proposed in [5] for the updating of the group membership Z. This split-merge Metropolis-Hastings procedure splits or merges the clusters using a restricted Gibbs sampling scan [5]. To exploit the major changes from the Metropolis-Hastings step, and the minor refinement from the Gibbs sampling step, we update Z by

Algorithm 1

```
$\alpha^{(0)} \sim \text{Gamma}(1,1)$
$Z^{(0)} \sim \text{CRP}(\alpha^{(0)})$
for $m = 1$ to $M$ do
    if $m$ is odd then
        for $i = 1$ to $p$ do
            $Z_i^{(m)} \sim P(Z_i|X, Z_{-i}^{(m-1)}, \alpha^{(m-1)})$
        end for
    else
        Update $Z^{(m-1)}$ by split-merge MH procedure
    end if
    $\alpha^{(m)} \sim P(\alpha|X, Z^{(m)})$
end for
```

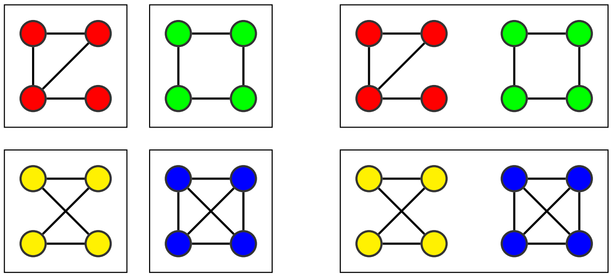

Figure 2: Illustration of different clustering results that both make sense.

alternating between the Gibbs sampler and the split-merge Metropolis-Hastings procedure. The whole procedure is summarized in Algorithm 2. See [5] for more details of the split-merge Metropolis-Hastings procedure.

When the data X is generated from a GGM with variables that can be clustered into disjoint groups, then the posterior distribution is very much likely to have multiple modes, corresponding to different clustering assignments. For example, the graphical model in Figure 2 has 16 variables belonging to 4 groups, shown in 4 different colors. In this figure, the left part shows the most natural way of clustering the variables, while it also makes sense to cluster them in the way as shown on the right part of the figure. For this graphical model, there are 15 reasonable ways to cluster the 16 variables, which are expected to have much higher probabilities than all the others.

Most of the time, we are more interested in such reasonable clusterings, especially the finest clustering, than in the posterior probability of one clustering. By the finest clustering, we mean that the one in which no cluster can be further divided into two disjoint subclusters (e.g. the clustering on the left in Fig. 2). This being said, rather than running the Markov chain for long enough until convergence, finding the posterior mode that corresponds to the finest clustering is good enough for our inference purpose. In practice, we start the Markov chain from a clustering that treats each variable as a single cluster and run the Algorithm 1 without split-merge procedure until it hits a local mode. We then report this state as our clustering of the variables. This method to some extent can be viewed as a greedy algorithm for finding the finest clusters, and we summarize it as Algorithm 2. Although greedy, as we shall see in the following section, it performs pretty well and efficiently on the synthetic data generated by both us and others as well as the real data.

Algorithm 2

```
$\alpha^{(0)} \sim \text{Gamma}(1,1)$
$Z^{(0)} = (1, 2, \ldots, p)$
for $m = 1$ to $M$ do
    for $i = 1$ to $p$ do
        $Z_i^{(m)} \sim P(Z_i|X, Z_{-i}^{(m-1)}, \alpha^{(m-1)})$
    end for
    if $\exists(z_1, \ldots, z_p)$ s.t. for $i = 1, \ldots, p$
$P(Z_i = z_i|X, Z_{-i}^{(m-1)}, \alpha^{(m-1)}) > 1 - \epsilon$ then
        break
    end if
    $\alpha^{(m)} \sim P(\alpha|X, Z^{(m)})$
end for
output $Z^{(m)}$
```

5 EXPERIMENTAL RESULTS

5.1 Synthetic Data

Here we present three experiments on synthetic data. The first experiment illustrates the relationship between posterior modes and clusterings. The second one shows how well our method performs compared to some simple generic methods in a variety of settings. The third experiment evaluates our method using the synthetic data proposed in [9] and compares it with the method in [9].

5.1.1 Modes and Clusterings

Suppose that our model consists of p variables of c clusters. To generate the data, we first assign each variable to one of the c clusters with probability $1/c$. Then, we add an edge between two variables by probability P_{in} if they are in the same cluster or otherwise, by probability P_{out}. For each edge (i, j), we set $\Omega_{ij} = 0.3$. Finally, to make sure that the precision matrix is positive definite, we set its diagonal element to the absolute value of the minimum eigenvalue of the current Ω plus 0.2.

We show a simple example to illustrate that the posterior modes correspond to all reasonable clusterings.

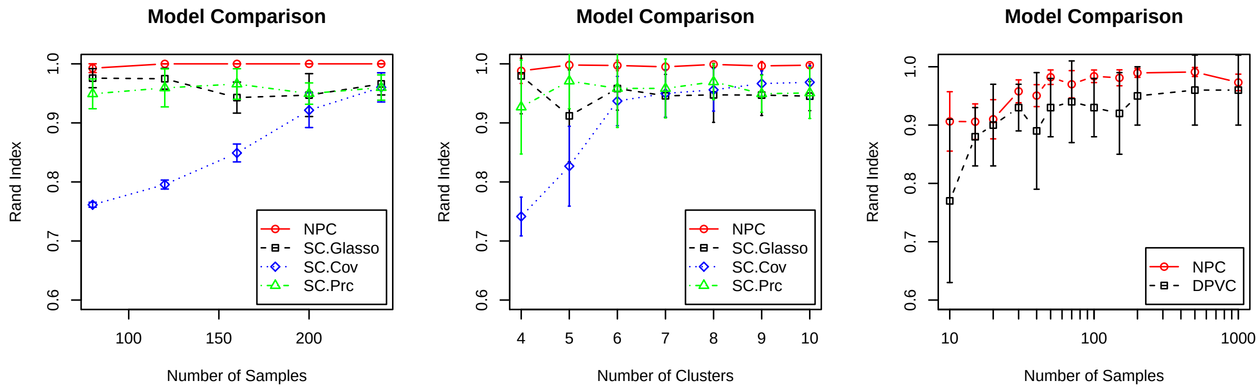

Figure 3: Performance comparison of our method NPC and the others in terms of the average Rand index. From left to right, (a) $p = 50$, $n = 100$, and the number of clusters ranging from 4 to10; (b) $p = 50$, $c = 6$, and the number of samples ranging from 40 to 240; (c) The data is generated according to [9], with $p = 20$, $c = 5$, and the number of samples ranging from 10 to 1000. SC.Glasso, SC.Cov and SC.Prc stand for spectral clustering with three different similarity matrices, and DPVC for the method in [9].

Using the above-mentioned data generation method, we construct a Gaussian graphical model (GGM) with $p = 12$ variables and $c = 4$ clusters with sizes 2, 3, 3, and 4. We set $P_{\text{in}} = 1$ and $P_{\text{out}} = 0$, so the GGM has 4 fully connected components without any inter-component edges. Then we generate $n = 50$ i.i.d. samples from this GGM. We run the Gibbs sampler for 1000 times starting from different starting points of (α, Z) drawn from their prior distributions. At each time we run the Gibbs sampler until it gets trapped at one mode of the posterior distribution, i.e., when the Markov chain has a very small chance (say, < 0.001) to traverse to another state. For all the 1000 simulations, the Markov chain always reaches one of the 15 partitions listed in Table 1, which also lists the frequency the Markov chain dwelling in each mode. The 15 modes are exactly all the possible combinations of the 4 true clusters, showing that the posterior modes and reasonable clusterings are closely related.

Table 1: Frequency of getting trapped at the posterior modes. The first row represents the true clustering according to which we generate our data. Different colors indicate different clusters.

🟥🟥	🟩🟩🟩	🟨🟨🟨	🟦🟦🟦🟦	truth
🟥🟥	🟥🟥🟥	🟥🟥🟥	🟥🟥🟥🟥	48.5%
🟥🟥	🟥🟥🟥	🟥🟥🟥	🟩🟩🟩🟩	3.5%
🟥🟥	🟥🟥🟥	🟩🟩🟩	🟥🟥🟥🟥	10.2%
🟥🟥	🟩🟩🟩	🟥🟥🟥	🟥🟥🟥🟥	3.1%
🟩🟩	🟥🟥🟥	🟥🟥🟥	🟥🟥🟥🟥	7.5%
🟥🟥	🟥🟥🟥	🟩🟩🟩	🟩🟩🟩🟩	3.3%
🟥🟥	🟩🟩🟩	🟥🟥🟥	🟩🟩🟩🟩	3.2%
🟥🟥	🟩🟩🟩	🟩🟩🟩	🟥🟥🟥🟥	2.1%
🟥🟥	🟥🟥🟥	🟩🟩🟩	🟨🟨🟨🟨	1.0%
🟥🟥	🟩🟩🟩	🟥🟥🟥	🟨🟨🟨🟨	2.0%
🟥🟥	🟩🟩🟩	🟨🟨🟨	🟥🟥🟥🟥	3.0%
🟩🟩	🟥🟥🟥	🟥🟥🟥	🟨🟨🟨🟨	2.3%
🟩🟩	🟥🟥🟥	🟨🟨🟨	🟥🟥🟥🟥	3.0%
🟩🟩	🟨🟨🟨	🟥🟥🟥	🟥🟥🟥🟥	2.3%
🟥🟥	🟩🟩🟩	🟨🟨🟨	🟦🟦🟦🟦	3.7%

5.1.2 Finding the Finest Clustering

Now we consider an example where we are interested in recovering the finest clustering. We generate the synthetic data using a GGM with $p = 50$, $P_{\text{in}} = 1$ and $P_{\text{out}} = 0$. We vary the experiment settings with different number of sample and number of clusters to test our method. For comparison, we have also implemented the spectral clustering [13] method. To use spectral clustering, we employ three different similarity measures to define the relationship between variables: the empirical covariance matrix calculated from the sample data, the empirical precision matrix and the precision matrix generated by Glasso. Starting from the spectrum of these matrices, we perform dimensionality reduction and then use k-means to cluster the variables in the transformed space.

First, we set the number of clusters to 6, and then vary the number of samples from 80 to 240. For each set of samples, we conduct 10 independent simulations and compute the average Rand index, which is a widely-used measure for clustering similarity. Rand index ranges from 0 to 1, with 1 indicating the perfect match. As shown in Fig. 3(a), our method outperforms spectral clustering regardless of the number of samples, while the accuracy for both methods im-

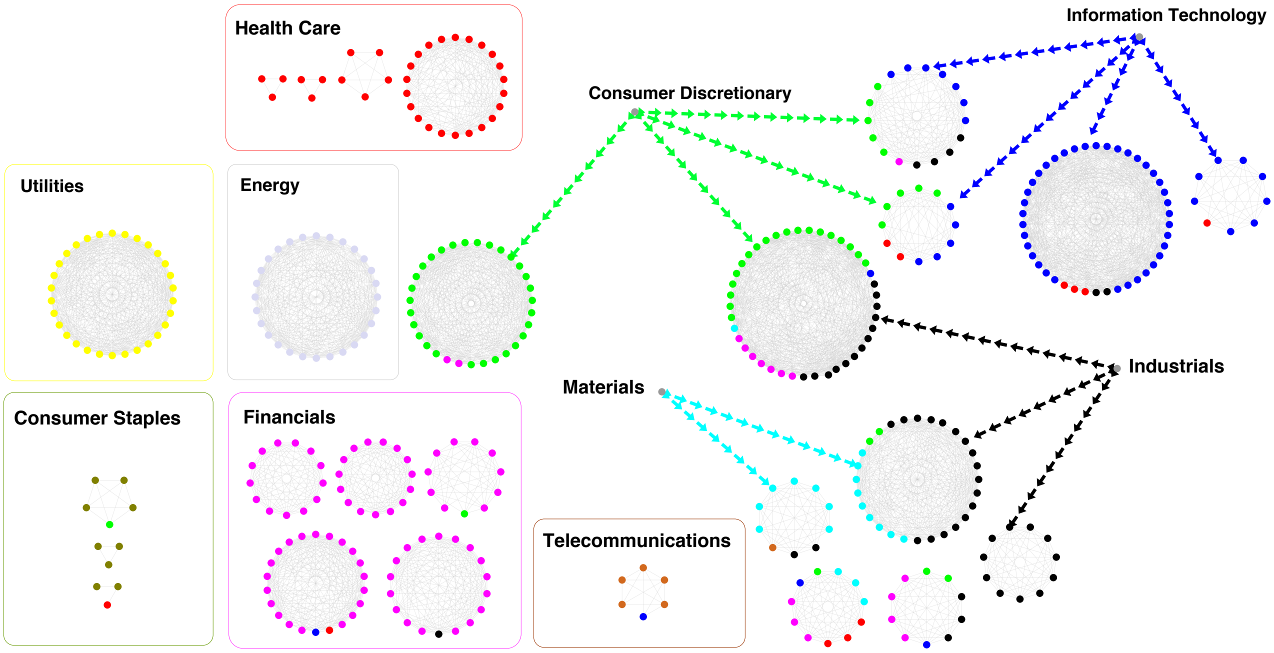

Figure 4: Visualization of the clustering result on equity data. Each stock is colored according to its true sector classification and this figure shows the clustering result obtained by our method.

proves as more samples are used. Note that spectral clustering requires a predefined value for the number of clusters, for which we uses 4, the ground truth here.

Second, we fix the number of samples to 100 and vary the number of clusters from 4 to 10. Spectral clustering is always fed with the true number of clusters as the parameter. As shown in Fig. 3(b), our method still has higher accuracy than spectral clustering in all the experiments, showing that our nonparametric Bayesian method can find the right number of clusters automatically.

5.1.3 Comparison with a Factor Model

As mentioned before, Palla et al. [9] studies variable clustering in a different setting. Although their model is different from ours, the covariance structure is also a block diagonal one. Using the data generation method described in Palla et al's paper, we generate a set of synthetic data with $p = 20$ dimensions and $c = 5$ equally-sized clusters (of 4 variables). For each cluster we sample $Y_{iz} \sim N(0, 1)$ for $i = 1, \dots, n$ and $z = 1, \dots, c$, then $g_j \sim N(0, 1)$ for $j = 1, \dots, p$ and finally sample $X_{ij} \sim N(g_j Y_{iz_j}, 0.1)$ for i and j where z_j denotes the cluster of the jth variable. We generate the test data sets with n, the number of samples, varying from 10 to 1000 and repeat 10 times for each n. As shown in Fig. 3(c), except for some small n, our method always has higher accuracy than the DPVC method proposed in [9].

5.2 Real Data

To test the performance of our method on a real data set, we apply our method to an equity dataset in the "huge" package [16], which consists of 1245 daily closing prices from January 1, 2003 to January 1, 2008 for 452 equities in the S&P 500 index. The stocks are divided into ten sectors including *health care*, *utilities*, *energy*, *consumer staples*, *materials*, *telecommunications*, *industrials*, *consumer discretionary*, and *financials*. Each sector has 6 to 70 stocks. Stocks in the same sector are expected to be more correlated with each other, and therefore tend to form a cluster. We run our method to cluster these stocks based upon their closing prices. We obtain 26 clusters with size larger than 2, in total covering 413 stocks. Compared to the crude manual 10-sector classification, our clustering is more fine-grained. As shown in Fig. 4, each stock is colored according to its true sector classification. Many clusters generated by our method consist of stocks sharing the same color. Our algorithm identifies 7 sectors with very little misclassification. Further examination shows that our clustering result is not only consistent with the true sector classification, but can also provide finer-grain classification. For example, our method divides the financials sector (in pink) into five small clusters, corresponding to five sub-sectors: *property & casualty insurance*, *real estate investment trust*, *banks*, *diversified financial service*, and *other insurance companies*. Our method also clusters some stocks of different sectors into the same group. For example, one of our clusters contains stock in both the *materials* and *industrials* sectors. This is not due to bad clustering. Instead it is because some stocks indeed belong to two different sectors. For example, many stocks in in the *industry* sector belong to industrial materials or industrial conglomerates.

In addition, our clustering result is very stable and also accurate in terms of the Rand index. Running our method 100 times starting from different initial clusterings, the mean and the standard deviation of the Rand Index are 0.89 and 0.007, respectively.

For comparison, we have also implemented the spectral clustering using the precision matrix estimated by Glasso as the similarity measure. This reflects the basic idea of clustering the variables based on the estimated graph. This procedure requires specifying two parameters, namely, the number of clusters and the penalty parameter for Glasso. Among numerous trials with the number of clusters ranging from 10 to 30 and different levels of sparsity of the estimated graph, the clustering results vary substantially. The Rand index ranges from 0.17 to 0.88, which are obtained with $K = 10$ and an estimated graph of 5074 edges, and $K = 29$ and a graph of 8600 edges, respectively. This comparsion clearly shows the advantage of our method: parameter-free and self-adaptive to the data.

6 CONCLUSIONS

We have presented a nonparametric model that can cluster variables in a GGM into correlated groups, by exploiting block structure in a GGM and making use of an efficient MCMC algorithm. Our method performs well on both synthetic and real data and can successfully identify the underlying block structure. In particular, our method does not need a predefined value for the number of clusters. Instead it can automatically determine this based upon the data. In the future we will introduce sparsity-induced prior on the precision matrix, so that we can estimate the block structure and graph structure simultaneously.

7 ACKNOWLEDGEMENTS

The authors thank Dahua Lin for his useful comments and suggestions. This work is supported by the Alfred P. Sloan Research Fellowship to JX and the NSF CAREER award to JX.

References

[1] Christophe Ambroise, Julien Chiquet, and Catherine Matias. Inferring sparse Gaussian graphical models with latent structure. *Electronic Journal of Statistics*, 3:205–238, 2009.

[2] Onureena Banerjee, Laurent El Ghaoui, Alexandre d'Aspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96. ACM, 2006.

[3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[4] LR Haff. Empirical bayes estimation of the multivariate normal covariance matrix. *The Annals of Statistics*, 8(3):586–597, 1980.

[5] Sonia Jain and Radford M Neal. A split-merge Markov Chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1), 2004.

[6] Benjamin M Marlin and Kevin P Murphy. Sparse Gaussian graphical models with unknown block structure. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 705–712. ACM, 2009.

[7] Benjamin M Marlin, Mark Schmidt, and Kevin P Murphy. Group sparse priors for covariance estimation. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 383–392. AUAI Press, 2009.

[8] Radford M Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

[9] Konstantina Palla, David Knowles, and Zoubin Ghahramani. A nonparametric variable clustering model. In *Advances in Neural Information Processing Systems 25*, pages 2996–3004, 2012.

[10] Jim Pitman. *Combinatorial stochastic processes*, volume 1875. Springer-Verlag, 2006.

[11] Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications.* CRC Press, 2005.

[12] Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Molecular systems biology*, 3(1), 2007.

[13] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[14] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[15] Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[16] Tuo Zhao, Han Liu, Kathryn Roeder, John Lafferty, and Larry Wasserman. The huge Package for high-dimensional undirected graph estimation in R. *The Journal of Machine Learning Research*, 98888:1059–1062, 2012.