COPULA GAUSSIAN GRAPHICAL MODEL FOR DISCRETE DATA

Justin Dauwels, Hang Yu, Shiyan Xu, and Xueou Wang

School of Electrical and Electronics Engineering, School of Physical and Mathematical Sciences
Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798

ABSTRACT

Copula Gaussian graphical models are capable of describing dependencies between a large number of heterogeneous variables. In this paper, low-complexity algorithms are proposed for learning copula Gaussian graphical models from discrete data. The proposed approach is Monte-Carlo expectation maximization: in the E-step, an efficient Gibbs sampler is applied, and in the M-step, the sparse graphical model is inferred by solving a penalized maximize likelihood problem. The regularization parameter is determined through the BINCO method proposed by Li *et al.* Numerical results for both synthetic and real data demonstrate the effectiveness of the proposed approach.

Index Terms— copula, discrete data, glasso, Gibbs sampling, expectation maximization

1. INTRODUCTION

Sparse graphical models provide an effective way to describe and exploit statistical patterns in data, especially for high-dimensional datasets such as gene expression data, multi-electrode brain recordings, and stock market data. Gaussian graphical models are commonly used in this context, since inference in such models is often tractable, either exactly or approximately. The structure of a Gaussian graphical model is characterized by its precision matrix (the inverse covariance matrix): zero elements in the precision matrix correspond to the absence of edges in the graphical model and the conditional independence between pairs of variables. A common method of finding a sparse precision matrix from Gaussian data is to maximize the log-likelihood with an ℓ_1 penalty on the precision matrix. The problem can be solved by a simple and fast algorithm, i.e., graphical lasso or glasso [1].

Real-world datasets, however, are often non-Gaussian, for instance, geophysics data or digital images. For non-Gaussian continuous data, Liu *et al.* [2] proposed to use copula Gaussian graphical models; non-Gaussian observed variables are transformed to Gaussian latent variables, and next a sparse graphical model is learned from the Gaussian data. In earlier work, we have extended this approach to hidden-variable graphical models [3], multiscale graphical models [4], and extreme-value graphical models [5], which are more powerful models that can be applied to a wide range of applications.

For discrete data, however, the situation is more convoluted; discrete data cannot be transformed directly into Gaussian data, since the mapping is one-to-many. A common approach is to apply Markov chain Monte Carlo method (MCMC) to simulate both the latent Gaussian variables and the posterior distribution of the precision matrix [6, 7, 8]. Different priors have been selected for the precision matrix (or the covariance matrix), including covariance selection prior [6], the inverse-Wishart distribution [7] and the G-Wishart distribution [8]. On the other hand, the framework of factor graph is also used to learn the dependence structure with the generalized double Pareto prior [9].

In this paper, we propose an algorithm to directly infer point estimates of the precision matrix. The proposed method is reliable yet much more efficient than the full MCMC approach of [6, 7, 8, 9], as it avoids the costly Monte Carlo simulations for the posterior distribution of the precision matrix. Instead, we apply Monte Carlo expectation maximization [10]. In the E-step, we draw samples for the latent Gaussian variables by efficient Gibbs sampling [11]. In the M-step, we learn the sparse Gaussian graphical model through the glasso method [1]. The sparsity of the resulting graphical model is determined by the regularization parameter associated with the glasso method [1]. Standard approaches for regularization selection are known to result in overly dense graphs [12]. In this work, we divide the problem of learning the precision matrix into two steps: we first apply the BINCO method [13] to learn the sparsity structure of the graphical model; next we determine the non-zero elements in the precision matrix via Iterative Proportional Fitting (IPF) [14].

Our numerical results for synthetic data show that the proposed approach can recover the statistical dependency between discrete variables more reliably than glasso [1] and copula glasso [2]. For real datasets, the proposed method produces very similar results to alternative approaches (see, e.g. [7, 8, 15, 16]), which are more computationally complex and less widely applicable.

This paper is organized as follows. In Section 2, we briefly introduce the Gaussian copula graphical model, and outline our learning algorithm for discrete data. In Section 3, we deal with the problem of regularization selection, and in Section 4, we present numerical results on synthetic and real datasets; we offer concluding remarks in Section 5.

2. COPULA GAUSSIAN GRAPHICAL MODEL

We denote the non-Gaussian observed variables and Gaussian latent variables as $Y = Y_1, \ldots, Y_P$ and $Z = Z_1, \ldots, Z_P$ respectively, and a copula Gaussian graphical model is defined as [6, 8]:

$$Z \sim \mathcal{N}(0, K^{-1}) \tag{1}$$

$$Y_k = F_k^{-1}(\Phi(Z_k)), \tag{2}$$

where K is the precision matrix whose inverse (the covariance matrix) has normalized diagonal, Φ is the CDF (cumulative distribution function) of the standard Gaussian distribution, and F_k is the CDF of Y_k. The latter is often approximated by the empirical distributions $\hat{F}_k$. Note that $F^{-1}(y) = \inf_{x \in \mathcal{X}}\{F(x) \geq y\}$. F_k^{-1} is a one-to-one mapping for a continuous distribution, but a one-to-many mapping from the observed discrete Y_k to the latent Gaussian Z_k [7]. Therefore, Z_k cannot be determined uniquely from the observed discrete Y_k.

In order to learn the graphical model, our objective is to infer K from L independent observations $y^{(1:L)} = y^{(1)}, \ldots, y^{(L)}$, where $y^{(\ell)} = y_1^{(\ell)}, \ldots, y_P^{(\ell)}$. We denote the associated latent variables as $z^{(1:L)} = z^{(1)}, \ldots, z^{(L)}$, where $z^{(\ell)} = z_1^{(\ell)}, \ldots, z_P^{(\ell)}$. A reasonable approach to infer K is maximum a posteriori (MAP) estimation:

$$\hat{K} = \operatorname*{argmax}_{K \succ 0} p(K|y^{(1:L)}), \tag{3}$$

where the marginal $p(K|y^{(1:L)})$ is given by:

$$\begin{aligned} p(K|y^{(1:L)}) &\propto \int_z p(y^{(1:L)}, z^{(1:L)}, K; \hat{F}) dz^{(1:L)} \\ &= p(K) \prod_{\ell=1}^{L} \int_{z^{(\ell)}} \left[\prod_{k=1}^{N} p(y_k^{(\ell)}|z_k^{(\ell)}; \hat{F}_k) \cdot \right. \\ &\quad \left. \mathcal{N}(z^{(\ell)}; 0, K^{-1}) \right] dz^{(\ell)}. \end{aligned} \tag{4}$$

If Y is continuous, the conditionals $p(y_k|z_k; \hat{F}_k)$ are Dirac deltas, and the integrals in the right-hand side (RHS) of (4) are trivial. If Y is discrete, those integrals are intractable, and instead we solve (3) by expectation maximization [10]:

$$\hat{K}_{(\kappa+1)} = \operatorname*{argmax}_{K} \log p(K) + Q(K; \hat{K}_{(\kappa)}), \tag{5}$$

where

$$\begin{aligned} Q(K; \hat{K}_{(\kappa)}) = \sum_{\ell=1}^{L} \int_{z^{(\ell)}} \left[p(z^{(\ell)}|y^{(\ell)}; \hat{K}_{(\kappa)}) \cdot \right. \\ \left. \log \mathcal{N}(z^{(\ell)}; 0, K^{-1}) \right] dz^{(\ell)}, \end{aligned} \tag{6}$$

and κ is the iteration number.

Since the integrals in the RHS of (6) are intractable, we calculate them numerically by Monte Carlo integration $1/M \sum_{i=1}^{M} \log \mathcal{N}(\hat{z}_{(\kappa)}^{(\ell,i)}; 0, K^{-1})$, where the M samples $\hat{z}_{(\kappa)}^{(\ell,i)}$ with $i = 1, \ldots, M$, are drawn from $p(z^{(\ell)}|y^{(\ell)}; \hat{K}_{(\kappa)})$. The latter is a truncated Gaussian in the case of discrete Y. Efficient block Gibbs sampling methods have been developed for truncated Gaussian distributions [11]. Those methods are also practical when some data is missing: If a sample is missing for an observed variable Y_k, it is still possible to draw Gibbs samples for Z_k, as the corresponding conditional distribution is Gaussian. In summary, we approximate $Q(K; \hat{K}_{(\kappa)})$ as:

$$\begin{aligned} \hat{Q}(K; \hat{K}_{(\kappa)}) &\approx \frac{1}{M} \sum_{\ell=1}^{L} \sum_{i=1}^{M} \log \mathcal{N}(\hat{z}_{(\kappa)}^{(\ell,i)}; 0, K^{-1}) \\ &= \frac{L}{2} \big(\log\det K - \operatorname{tr}(\Sigma_{(\kappa)} K) \big) + C, \end{aligned} \tag{7}$$

where tr denotes trace, C is an irrelevant constant, and $\Sigma_{(\kappa)}$ is the empirical covariance matrix:

$$\Sigma_{(\kappa)} = \frac{1}{L} \sum_{\ell=1}^{L} \Sigma_{(\kappa)}^{(\ell)} = \frac{1}{ML} \sum_{\ell=1}^{L} \sum_{i=1}^{M} \hat{z}_{(\kappa)}^{(\ell,i)} (\hat{z}_{(\kappa)}^{(\ell,i)})^T, \tag{8}$$

where $\Sigma_{(\kappa)}^{(\ell)}$ may be viewed as the empirical covariance matrix computed from the ℓ-th observation $y^{(\ell)}$.

A common choice for the prior $p(K)$ is the Laplace distribution:

$$p(K) = \frac{\tilde{\lambda}^N}{2} \exp\left(-\tilde{\lambda} \|K\|_1 \right). \tag{9}$$

That prior favors sparse K and hence a sparse latent Gaussian graphical model. The resulting Monte-Carlo EM algorithm iterates the following step until convergence:

$$\hat{K}_{(\kappa+1)} = \operatorname*{argmax}_{K \succ 0} \log\det K - \operatorname{tr}(\Sigma_{(\kappa)} K) - \lambda \|K\|_1, \tag{10}$$

where $\lambda = 2\tilde{\lambda}/L$. This convex optimization problem can be solved efficiently in various ways [1, 17]. Convergence of the EM algorithm for penalized likelihood problems (cf. (10)) has been proven in [18]. Our experimental study shows that the algorithm usually converges after several iterations (< 10). Note that the regularization parameter λ needs to be chosen appropriately to recover the correct precision matrix $\hat{K}_{(\kappa+1)}$ in each EM iteration, which is a critical issue to be addressed in Section 3.

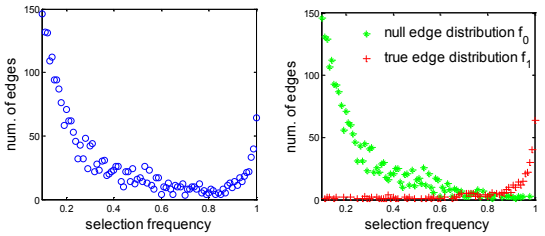
3. REGULARIZATION SELECTION

A suitable choice of regularization parameters λ in (10) can produce the graphical model with true sparsity pattern of K. However, standard procedures for selecting λ are known to overfit the data and result in graphs that are too dense [12]. As an alternative, we employ a two-step procedure of structure learning and parameter learning.

3.1. Structure Learning

We use the BINCO procedure [13] to infer the sparsity pattern of the precision matrix $K_{(\kappa+1)}$, from $N = ML$ Gibbs samples drawn in the E-step.

We randomly divide all the N samples into M sample sets S_m, $m = 1, \cdots, M$, each with L samples. For each λ, we estimate one precision matrix K_m for each sample set S_m by solving (10), resulting in M precision matrices. For each element (i, j) in the matrix K_m, the number of times it is non-zero for each sample set S_m is counted and divided by M. As a result, we obtain the selection frequency (or stability) x of the corresponding edge associated with λ. Typically, a proper selection of λ will result in a U-shaped empirical density function $f^\lambda(x)$ of selection frequencies of all the candidate edges, as illustrated in Fig. 1(a).



(a) (b)

Fig. 1. The distribution of selection frequencies based on a simulated dataset (a) all the edges; (b) null edges and true edges separately.

The selection frequencies fall into two categories, i.e., "true" or "null", depending on whether the edge exists in the true graphical model. Consequently, the density function $f^\lambda(x)$ can be decomposed as $f^\lambda(x) = (1-\pi)f_0^\lambda(x) + \pi f_1^\lambda(x)$, where π is the proportion of true edges and f_0^λ and f_1^λ are the density functions of selection frequencies when the edge belongs to the "null" and "true" category respectively. As illustrated in Fig. 1(b), there exists V_1 and V_2, $0 < V_1 < V_2 < 1$, such that, $f_1^\lambda \to 0$ on $(V_1, V_2]$, and f_0^λ is monotonically decreasing on $(V_1, 1]$ [13]. These observations motivate us to estimate f_0^λ and π by fitting empirical f^λ on $(V_1, V_2]$ to a parametric model, that is, a binomial distribution with a Beta prior as in [13]. For each λ, we choose the threshold c that minimizes the False Detection Error (FDR) defined as $\text{FDR}^\lambda(c) = \sum_{x \geq c} f_0^\lambda(x) / \sum_{x \geq c} f^\lambda(x)$. We then find the optimal λ that maximizes the estimated number of true edges whose definition is $N_e(\lambda) = (1 - \text{FDR}^\lambda(c)) \sum_{x \geq c} f^\lambda(x)$. Eventually, the sparsity pattern K_p is obtained by retaining those edges with selection frequencies above threshold c.

3.2. Parameter Learning

After learning the structure of the graphical model (cf. Section 3.1), the non-zero entries in the precision matrix can be inferred via Iterative Proportional Fitting [14].

4. NUMERICAL RESULTS

In this section, we benchmark the proposed method with existing methods on both synthetic and real data.

4.1. Synthetic Data

We generate synthetic discrete data as follows:

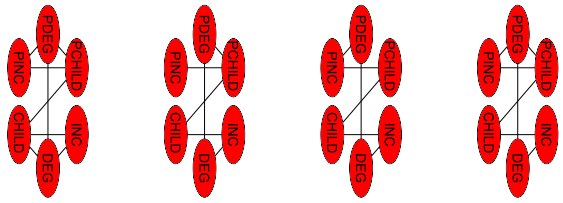
1. We generate a random precision matrix through the method in [2]. Specifically, first we uniformly sample $x_1, ..., x_n$ from a unit square. The precision matrix is initialized as a unit matrix. Next, we set the element $K(i, j) = K(j, i)$ of precision matrix equal to $\rho = 0.245$ with probability $(\sqrt{2\pi})^{-1} \exp(-4\|x_i - x_j\|^2)$, and equal to zero otherwise.
2. From the resulting precision matrix, we generate Gaussian data and then apply Beta copulas with different shape parameters for each variable, leading to non-Gaussian continuous data.
3. We partition the domain of each continuous variable into d equidistant intervals, where d is the number of discrete values for each variable. We replace the samples within the i-th interval by the value i (with $i = 1, \cdots, d$) , leading to discrete data.

We generate 4 groups of 100 datasets with 2, 3, 5, 10 discrete values respectively, all generated from the same precision matrix. The sample size for each dataset is 750. Our results are summarized in Table 1. 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. Moreover, F_1-score = 2·precision·recall/(precision+recall) is a weighted average of the precision and recall. We show the mean value and standard deviation (in brackets) for each criterion averaged over 100 datasets.

Table 1. Quantitative comparison of different methods

	Methods	Precision	Recall	F_1-score
binary	glasso [1]	0.96(0.09)	0.24(0.08)	0.37(0.11)
	continuous copula glasso [2]	0.96(0.09)	0.25(0.09)	0.38(0.11)
	discrete copula glasso	0.96(0.07)	0.40(0.10)	0.56(0.10)
3-ary	glasso [1]	0.99(0.03)	0.54(0.17)	0.68(0.17)
	continuous copula glasso [2]	0.99(0.04)	0.54(0.17)	0.69(0.16)
	discrete copula glasso	0.99(0.04)	0.69(0.18)	0.79(0.15)
5-ary	glasso [1]	0.99(0.02)	0.83(0.09)	0.90(0.05)
	continuous copula glasso [2]	1.00(0.01)	0.85(0.10)	0.91(0.06)
	discrete copula glasso	1.00(0.01)	0.92(0.08)	0.95(0.04)
10-ary	glasso [1]	1.00(0.01)	0.93(0.05)	0.96(0.03)
	continuous copula glasso [2]	1.00(0.01)	0.95(0.05)	0.97(0.03)
	discrete copula glasso	1.00(0.007)	0.97(0.04)	0.99(0.02)

The proposed approach ("discrete copula glasso") always outperforms glasso [1] and copula glasso [2], especially when the alphabet is small. The performance of all three methods improves, when the alphabet size increases and hence more information about the precision matrix is contained in the discrete data. When the number of discrete values increases to 10 and beyond, the discrete distribution becomes a pseudo-continuous distribution, which may explain why glasso [1]



(a) (b) (c) (d)

Fig. 2. Results of different methods on the GSS data: (a) Hoff's method [7]; (b) glasso [1]; (c) continuous copula glasso [2]; (d) discrete copula glasso.

and copula glasso [2] perform well for 10-ary data. We also considered smaller sample sizes (not shown here), and observed that the performance gain of discrete copula glasso over the other two methods becomes larger. We also tested the MCMC algorithm of [8] on the 100 binary datasets (first group). The resulting mean value (standard deviation) for the three criteria are 0.76(0.11), 0.48(0.09), 0.58(0.09) respectively. In our simulations, the running time of the proposed algorithm (MATLAB implementation) is about 5 times shorter than the MCMC approach of [8] (C++ implementation).

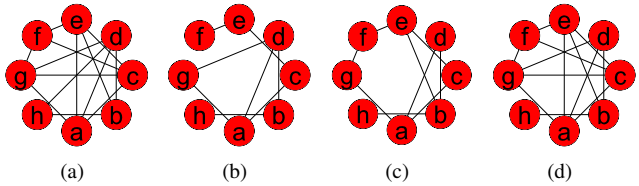
4.2. Real Data

4.2.1. General Social Survey Data

Here we consider data from the 1994 General Social Survey (GSS), concerning 1002 males in the U.S. labor force [19]. The relevant variables are as follows: income of the respondent in 1000s of dollars (INC), highest degree ever obtained (DEG), number of children ever had (CHILD), financial status of respondent's parents when respondent was 16 (PINC), maximum of mother's or father's highest degree (PDEG), and number of siblings of the respondent plus one (PCHILD). Age of the survey respondent is additionally included, as it is typically strongly related to income and number of children. The results for all three methods are shown in Fig. 2, in addition to results for Hoff's method [7], which is a discrete copula Gaussian graphical model learned by standard MCMC. The graphical models obtained from glasso [1] and copula glasso [2] are quite different from the graphical model from Hoff's method [7]. In contrast, the graphical model obtained through the proposed method is identical to the one from Hoff's method (apart from one edge). Hoff's method [7], however, is much more computationally complex than the proposed method. In each iteration, the computational complexity of Gibbs sampling in Hoff's algorithm are $\mathcal{O}\{LP^4\}$, while being $\mathcal{O}\{LP^2\}$ in our algorithm. The complexity of learning dependence structure is not comparable due to different method used in those two algorithms. Moreover, Hoff's method needs 25,000 iterations to simulate the stable posterior distribution while discrete copula glasso converges after only 5 iterations.

4.2.2. The Rochdale Data

Here we consider a social survey data set previously analyzed in [15]. This observational study was conducted in Rochdale



(a) (b) (c) (d)

Fig. 3. Results of different methods on the Rochdale data: (a) Whittaker's method [15]; (b) glasso [1]; (c) continuous copula glasso [2]; (d) discrete copula glasso.

and attempted to explore factors affecting womens economic activity. The corresponding eight variables are as follows: a) Wife economically active (no, yes); b) Age of wife > 38 (no, yes); c) Husband unemployed (no, yes); d) Child ≤ 4 (no, yes); e) Wife's education, high-school+ (no, yes); f) Husband's education, high-school+ (no, yes); g) Asian origin (no, yes); h) Other household member working (no, yes). The results are shown in Fig. 3 for glasso [1], copula glasso [2], and the proposed discrete copula glasso method, in addition to a log-linear model with minimum sufficient statistics, proposed by Whittaker [15]. The result from the proposed discrete copula glasso (see Fig. 3(d)) is almost identical to Whittaker's result (see Fig. 3(a)), as only one edge *dh* is missing. However, glasso [1] and continuous copula glasso [2] fail to infer the graphical model, probably because both models are mostly designed for continuous data.

Interestingly, in the discrete copula glasso, the variables connected to a ("Wife economically active") are c ("Husband unemployed"), d ("Child ≤ 4"), e ("Wife's education"), and g ("Asian origin"), which is identical to Whittaker's results [15] and results in [8, 16], which use MCMC simulations for all variables and hence are computationally complex.

We also investigated the partial correlations for pairs of variables. Whittaker [15] observed that the strongest pairwise interaction is $\{b, d\}$, followed by $\{b, h\}$, $\{e, f\}$ and $\{a, g\}$. The order of those pairs is identical in the proposed discrete copula glasso and the MCMC approach of [8, 16]. The latter two models learn the order of interactions based on the Cramer's V statistic.

The results for this dataset suggest that the proposed method yields very similar results as Whittaker's log-linear model and the approach of [8, 16]. Whittaker's log-linear model, however, is not able to learn the sparsity structure in an automated fashion. The approach of [8, 16] is computationally much more complex than the proposed approach.

5. CONCLUSION

We proposed novel learning algorithms for discrete copula Gaussian graphical models, which can be applied to any combinations of heterogeneous variables. Numerical results for synthetic data show that the proposed method can produce better estimates than glasso [1] and continuous copula glasso [2]. Moreover, results on real data suggest that the proposed method yields similar results as for existing methods, while being more computationally efficient and more generally applicable.

6. REFERENCES

[1] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse Inverse Covariance Estimation with the Graphical Lasso," Biostatistics 9:3, pp. 432-441, 2008.

[2] H. Liu, J. Lafferty, and L. Wasserman, "The Nonparanormal: Semiparametric Estimation of High Dimensional Undirected Graphs," *Journal of Machine Learning Research,* pp. 2295-2328, 2010.

[3] H. Yu, J. Dauwels, and X. O. Wang, "Copula Gaussian Graphical Models with Hidden Variables," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2177-2180, 2012.

[4] H. Yu, J. Dauwels, X. Zhang, S. Y. Xu, and W. I. T. Uy, "Copula Gaussian Multiscale Graphical Models with Application to Geophysical Modeling," *Proceedings of 15th International Conference on Information Fusion (Fusion)*, pp. 1741- 1748, 2012.

[5] H. Yu, Z. Choo, W. I. T. Uy, J. Dauwels, and P. Jonathan, "Modeling Extreme Events in Spatial Domain by Copula Graphical Models," *Proceedings of 15th International Conference on Information Fusion (Fusion)*, pp. 1761-1768, 2012.

[6] M. Pitt, D. Chan, and R. Kohn, "Efficient Bayesian Inference for Gaussian Copula Regression Models," *Biometrika* 93:3, pp. 537-554, 2006.

[7] P. D. Hoff, "Extending the Rank Likelihood for Semiparametric Copula Estimation," *Annals of Applied Statistics*, vol. 1, no. 1, pp. 265-283, 2007.

[8] A. Dobra and A. Lenkoski, "Copula Gaussian Graphical Models and their Application to Modeling Functional Disability Data," *Annals of Applied Statistics,* vol. 5, No. 2A, pp. 969-993, 2011.

[9] J. S. Murray, D. B. Dunson, L. Carin, and J. E. Lucas, "Bayesian Gaussian copula factor models for mixed data," *Arxiv preprint arXiv:1111.0317*, 2011.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society,* vol. 39, Series B, pp. 1-38, 1977.

[11] G. Rodriguez-Yam, R. Davis, and L. Scharf, "Efficient Gibbs Sampling of Truncated Multivariate Normal with Application to Constrained Linear Regression," Technical Report, Department of Statistics, Columbia University, 2004.

[12] H. Liu, K. Roeder, and L. Wasserman, "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models," *Advances in Neural Information Processing Systems,* 2010.

[13] S. Li, L. Hsu, J. Peng and P. Wang, "Bootstrap Inference for Network Construction," *Annals of Applied Statistics*, 2011.

[14] P. F. Xu, J. Guo, and X. He, "An Improved Iterative Proportional Scaling Procedure for Gaussian Graphical Model," *Journal of Computational and Graphical Statistics*, vol. 20, pp. 417-431, 2011.

[15] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*, John Wiley and Sons, 1990.

[16] A. Bhattacharya and D. B. Dunson, "Simplex Factor Models for Multivariate Unordered Categorical Data," *Journal Amer. Stat. Assoc.* 107, pp. 362-377, 2012.

[17] O. Banerjee, L. El Ghaoui, and A. dAspremont "Model Selection Through Sparse Maximum Likelihood Estimation," *Journal of Machine Learning Research* 9, pp. 485-516, March 2008.

[18] P. J. Green, "On Use of the EM for Penalized Likelihood Estimation," *Journal of the Royal Statistical Society*, Series B, vol. 52, No. 3, pp. 443-452, 1990.

[19] `http://webapp.icpsr.umich.edu/GSS/`