Referee Response – Algorithm xxx: A Covariate-Dependent Approach to Gaussian Graphical Modeling in R

JACOB HELWIG, SUTANOY DASGUPTA, PENG ZHAO, BANI K. MALLICK, and DEBDEEP PATI, Texas A&M Department of Statistics, USA

ACM Reference Format:
Jacob Helwig, Sutanoy Dasgupta, Peng Zhao, Bani K. Mallick, and Debdeep Pati. 2022. Referee Response – Algorithm xxx: A Covariate-Dependent Approach to Gaussian Graphical Modeling in R. 1, 1 (January 2022), 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 REFEREE 1

> I suggest authors add more discussions on recent literature of heterogeneous Gaussian graphical models [1,2,3], especially GGM with Covariates [4], in Introduction.
>
> [1] Gao, C., Zhu, Y., Shen, X., & Pan, W. (2016). Estimation of multiple networks in gaussian mixture models. Electronic journal of statistics, 10, 1133.
>
> [2] Hao, B., Sun, W. W., Liu, Y., & Cheng, G. (2018). Simultaneous clustering and estimation of heterogeneous graphical models. Journal of Machine Learning Research.
>
> [3] Ren, M., Zhang, S., Zhang, Q., & Ma, S. (2022). Gaussian graphical model-based heterogeneity analysis via penalized fusion. Biometrics, 78(2), 524-535.
>
> [4] Zhang, J., & Li, Y. (2022). High-Dimensional Gaussian Graphical Regression Models with Covariates. Journal of the American Statistical Association, 1-13.

Thank you for bringing these works to our attention. In Section 1, we have added Das et al. [2020]; Gao et al. [2016]; Hao et al. [2018]; Peterson et al. [2015]; Ren et al. [2022]; Shaddox et al. [2018] to our discussion of methods that assume the data may be partitioned into groups in which the precision matrix is homogeneous. We have also added Liu et al. [2022]; Qiu et al. [2016]; Yang and Peng [2020] to our discussion of works which assume that the precision matrix varies continuously as a function of time. Finally, we have discussed the work by Ni et al. [2019] for covariate-dependent modeling of DAGs and the works by Ni et al. [2022] and Zhang and Li [2023] done concurrently with Dasgupta et al. [2023] for covariate-dependent Gaussian Graphical Models. However, we note that these covariate-dependent GGMs do not currently provide their methods in a software package.

Authors' address: Jacob Helwig, jacob.a.helwig@tamu.edu; Sutanoy Dasgupta, sutanoy@stat.tamu.edu; Peng Zhao, pzhao@stat.tamu.edu; Bani K. Mallick, bmallick@stat.tamu.edu; Debdeep Pati, debdeep@stat.tamu.edu, Texas A&M Department of Statistics, 155 Ireland Street, College Station, Texas, USA, 77843-3143.

Manuscript submitted to ACM

2 REFEREE 2

> In the abstract, you write "There are many existing software packages for Gaussian graphical modeling, however, these packages make the restrictive assumption that the data are identically distributed or can be partitioned into identically distributed subgroups." Again, on page 1 you write "There are more than 35 existing R packages for GGM...all make the restrictive assumption that the precision matrix is homogeneous throughout the data, or that the data may be partitioned into homogeneous subgroups." Later on page two, you write "Currently available implementations for heterogeneous graphical modeling are limited in that they either treat the data as homogeneous within subgroups, corresponding to a discrete covariate (e.g., JGL [Danaher et al. 2014]), or as varying continuously with time, corresponding to a single-dimensional covariate (e.g., mgm [Haslbeck and Waldorp 2020])." As you remark, it is not that all the existing packages are homogeneous/block-homogeneous, rather in the heterogeneous case they do not handle the more general covariate structures you are able to account for. The language in the abstract and bottom of page 1 should be amended to account for this.

Thank you for pointing out this mistake. We have updated the sentence in the abstract and added a more precise characterization of existing packages and methods in the first paragraph of Section 1.

> The indexing of variables is a bit confusing as presented – For example. should β_k also be indexed by l and j in Eqs. (3) and (4)? The same comment holds for other parameters (γ, μ, s_k, etc.) as well. This would greatly clarify the notation and would be more in line with the presentation in the accompanying theory paper.

We agree that updating variable indices would improve clarity and consistency with Dasgupta et al. [2023], and thus have made the updates detailed in the table below. In this table, the (j, l)-th regression refers to the regression with the j-th variable fixed as the response and weighted with respect to the l-th observation. Thank you for raising this point.

Original	Updated	Description
β	β_j^l	Coefficients for (j, l)-th regression
γ	γ_j^l	Vector of Bernoulli variables for (j, l)-th regression
$L_{l,j}$	L_j^l	Likelihood function for (j, l)-th regression
$v_{l,j}$	v_j^l	Posterior for (j, l)-th regression
$\text{PIP}_{l,j}$	PIP_j^l	Posterior Inclusion Probability for (j, l)-th regression
μ	μ_j^l	Variational approximation to posterior regression coefficient means for (j, l)-th regression
s^2	$(s_j^l)^2$	Variational approximation to posterior regression coefficient variances for (j, l)-th regression
α	α_j^l	Variational approximation to posterior inclusion probabilities for (j, l)-th regression
ϕ	ϕ_j^l	Variational parameters for (j, l)-th regression
`ssq`	$\texttt{ssq}_\text{j}$	Prior residual variance candidates for j-th regressions
`sbsq`	$\texttt{sbsq}_\text{j}$	Prior slab variance candidates for j-th regressions
`pip`	$\texttt{pip}_\text{j}$	Prior inclusion probability candidates for j-th regressions
Ω	Ω_j	Hyperparameter candidates for j-th regressions
ω	ω_j^l	Model averaging weights for (j, l)-th regression

> Please expound on how the upper bound for π is estimated in Section 2.3.3. The Lasso gives a rough estimate of π which you then use to compute an upper bound. Please include the details of this last step.

The estimate for π provided by LASSO is the upper endpoint for the π candidate grid – there is no additional step beyond this. This may not have been clear since in Section 2.3.3, we mistakenly referred to the largest values in the hyperparameter candidate grids for the prior inclusion probability π and the residual noise variance σ^2 as "upper bounds". We have updated this section to more accurately describe these values as "upper endpoints" of the candidate grids instead. Thank you for pointing out this mistake.

> More motivation for the experimental parameter choices is needed. For example, why is the precision matrix off-diagonal chosen equal to 0 for all indices greater than 3? Why these three "groups" of precision matrices? This is not to say these are bad choices, just that explanation would help clarify.

The precision matrices in all settings have 3 defining characteristics: (1) they are sparse, (2) the non-zero entries are all adjacent and are in the upper left corner, and (3) they are a continuous function of the extraneous covariate z_l. We now justify each choice, and have added details regarding each choice to our revised manuscript.

(1) Sparsity of the precision matrix, equivalent to only a subset of the variables exhibiting conditional dependence, is a common assumption in the literature which has led to the wide-spread use of sparsity-inducing priors and loss terms [Danaher et al. 2014; Friedman et al. 2008; Ren et al. 2022; Yang and Peng 2020]. This assumption is realistic in many cases, yields more interpretable estimates than dense graphs, and offers a means of dealing with the ill-posedness arising in $n < p$ settings. This is our motivation for choosing a spike-and-slab prior, and thus, we have added further discussion on this to the first paragraph of Section 2.2.

(2) The choice for the non-zero, off-diagonal entries to be adjacent and in the upper left corner of the precision matrix was for ease of visualization. The indexing of these non-zero entries does not have any effect on the estimates other than in how they are indexed. That is, an identical setting which instead permutes the non-zero entries to no longer be adjacent would result in identically permuted graph estimates. To see that this is true, consider regressing y on the covariates x_1 and x_2 using 2 linear models which are equivalent to one another up to a permutation of the covariate orderings. That is, the fitted models are given by $y = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\epsilon}$ and $y = \tilde{\beta}_1 x_2 + \tilde{\beta}_2 x_1 + \tilde{\epsilon}$. Observe that $\hat{\epsilon} = \tilde{\epsilon}$, $\hat{\beta}_1 = \tilde{\beta}_2$, and $\hat{\beta}_2 = \tilde{\beta}_1$. This shows that if we permute the ordering of the variables in a linear regression, the coefficient estimates are equally permuted, a result which also holds for our case, wherein we are fitting multiple linear regressions. This is an important point for which we have added a clarification in the final paragraph of the introduction for Section 3.2.

(3) Lastly, in all settings, the precision matrix is a continuous function of the extraneous covariate. Without continuity, we cannot expect that observations with similar extraneous covariate values will have similar precision matrices, and thus, information sharing based on the extraneous covariate is not well-motivated. In the piece-wise linear, $q = 1$ setting, this continuity implies that as the covariate value approaches the left-most interval from the right, the precision matrix converges to the precision matrix for observations in the left-most interval. Similarly, as the covariate value approaches the right-most interval from the left, the precision matrix converges to the precision matrix for observations in the right-most interval. We formalize this in Equations 20 and 21 in Section 3.2.1, and in the following sentences have added an intuitive interpretation of the equations.

Additionally, we have added Figure 2 to visualize this continuity for all settings by plotting the values of the covariate-dependent entries as a function of the covariate.

> Also, these are the same experimental conditions included in the theory paper, though the results in the present submission are much better (e.g., see table 1 in the present and table 1 int he theory paper). Including additional experimental conditions and examples would be helpful, as would more explanation for the better results contained herein.

To more thoroughly evaluate `covdepGE`, we have added 8 challenging experimental settings to Section 3 which do not appear in Dasgupta et al. [2023]. Specifically, we have added experiments with a 4-dimensional extraneous covariate $z \in \mathbb{R}^4$ and experiments where the precision matrix $\Omega(z)$ is a non-linear function of z, and in each case consider $p \in \{10, 25, 50, 100\}$. We formalize each setting in Sections 3.2.2 and 3.2.3, and have added visualizations of the precision matrices in these settings to Figures 1 and 2. Each of these settings present unique challenges which we discuss in Sections 3.3.3 and 3.3.4 and visualize in Appendices C.4 and C.6. We have additionally updated the extended experiments and results in Appendix C to include these settings where appropriate, including the hypothesis tests which we discuss further in the subsequent response (Appendix C.5), median/IQR aggregated results (Appendix C.7), and `Mclust` subgroup counts (Appendix C.8.)

The improvement in results relative to those presented in Dasgupta et al. [2023] can be attributed to the larger sample size. The sample size used in Dasgupta et al. [2023] is $n = 150$ for the $q = 1$, PWL setting, and $n = 225$ for the $q = 2$ setting. 225 was chosen since it is divisible by 9, which is the number of sets in our partition of $[-3, 3]^2$ discussed in the first paragraph of Section 3.2 of the present work, thus allowing each set to have 25 observations. In the present work, we set $n = 225$ for both settings for a more direct comparison. We have added a note to the end of the first paragraph of Section 3 more clearly stating the sample size of $n = 225$ that we used here.

> Perhaps more care needs to be taken with the presentation of the results. I see that the mean (especially in Sensitivity) and variance (again, especially in Sensitivity) are improved almost across the board for covdepGE. However, there is often overlap in the mean+-sd intervals for each of the algorithms. Similar comments hold for the median(IQR) presented results. Clarifying comments like "performance is significantly reduced" (page 12 of the submission) would be helpful in this situation; e.g., what is meant by "significant" here.

Thank you for pointing out the potentially misleading use of “significant” – we have removed all such usages. We have also added hypothesis tests for all 16 settings to Tables 8-11 in Appendix C.5 to ascertain the statistical significance of the performance gains offered by `covdepGE`. We observe that in 13 of 16 settings, these gains are statistically significant on the 0.001 level, and on the 0.05 level in 15 of 16 settings. Although the standard deviations presented in Tables 2-5 of our unrevised manuscript may not suggest this significance, it is instead the standard error, which includes a pooling of the standard deviations and a normalization by $\sqrt{n}$, that is used for hypothesis testing and confidence intervals. Since these hypothesis tests analyze the difference in sample means to make inferences regarding the difference in the population means, the normalization, which is the reason for the apparent discrepancy, accounts for the law of large numbers, which says that the sample means will converge to the population means as n (here the number of trials) grows large. Therefore, we have replaced standard deviation with the standard error of the mean in all results. Additionally, we have replaced all tables in the main text with line plots of the mean with error bars showing 2 times

the standard error, which corresponds to roughly a 95% confidence interval for the mean. Tables 1-5 from the main text have been relocated to Appendix C.1.

> In the provided experiments, covdepGE seems to be performing roughly at the same speed in the q=1 and q=2 settings. How well does the algorithm scale with q in general? Where (if at all) would the computational bottleneck come into play in high-dimensional covariate settings?

In Figure 1, we show the computational graph for `covdepGE`, where the first step in the pipeline is to calculate the similarity weights using the covariate $\mathbf{Z} \in \mathbb{R}^{n \times q}$. After this stage, $\mathbf{Z}$ is not used again, and thus, q has no effect on the time complexity beyond this stage, although differences in real time may arise due to different data leading to different rates of convergence for CAVI.

For estimating the similarity weights, `covdepGE` first estimates the observation-specific bandwidths, which we detail in Appendix A.2. From Equations 31-34, this estimation has linear time complexity in q. Next, the bandwidths are used to calculate the similarity weight for each observation, which from Equations 1 and 2 is again linear in q. Thus, the overall time complexity of `covdepGE` only scales linearly with q.

However, while there is not a computational bottleneck that depends on q, there is an information bottleneck arising from compressing q-dimensional vectors into a scalar similarity weight. Specifically, from Equation 1, the similarity weight for observation i with respect to observation l is calculated by taking the distance between $z_l \in \mathbb{R}^q$ and $z_i \in \mathbb{R}^q$. The curse of dimensionality says that the variance in these distances will decrease as q grows large. This will cause the similarity weights for all observations to become increasingly similar, reducing the degree of heterogeneity in the estimated graphs.

As larger q is a setting of interest, we explore this in greater depth in Section 3.3.3, where we have added experiments with $q = 4$. In Appendix C.6, we analyze the number of unique graph estimates in all settings and find that although the number of unique ground-truth CDS is largest in the $q = 4$ setting, the number of graphs estimated by `covdepGE` is smallest. For more lossless compression, future work may consider dimensionality reduction algorithms such as PCA.

> The comparison algorithms chosen in the present manuscript and in the theoretical companion paper differ. Both papers use mgm for comparison, while the theory paper also uses loggle and this version uses JGL. In the theory paper, loggle seems to outperform mgm, and it may be useful to include loggle here for comparison as well.

Although `loggle` and `mgm` are both regularized kernel-smoothing methods, and `loggle` is no longer available on CRAN, we agree that the results from Dasgupta et al. [2023] motivate its inclusion. We therefore have added `loggle` to all 16 settings in Section 3, as well as updating the extended experiments and results (Appendix C) to include `loggle` where appropriate.

> covdepGE is the only one of the three algorithms that includes parallelization. It would be helpful, at least in one of the simulation settings considered, to include a comparison of cpu time as well (or include a parallelization of the competitor algorithms; I think mgm should also be embarrassingly parallelizable). This would also help clarify how efficient the parallel implementation of covdepGE is here.

We agree that this is an interesting analysis, and thus have added runtimes for `covdepGE` executed sequentially in Appendix C.2, which we refer to in the final paragraph of the introduction to Section 3. We find that for $p \in \{25, 50, 100\}$, parallel execution is more than 20× faster than sequential execution.

> It would be helpful to include code to execute real data examples in your package rather than just simulations (perhaps the FOXC2 example from the theory paper?)

We have added an example using The Cancer Genome Atlas (TCGA) data for patients with Breast Invasive Carcinoma (BRCA) to the GitHub for `covdepGE` (https://github.com/JacobHelwig/covdepGE/blob/master/examples/TCGA_analysis.md), which is the same setting considered by Dasgupta et al. [2023]. So that this example is easy to find, we have referenced and linked it in the final paragraph of the GitHub introduction (https://github.com/JacobHelwig/covdepGE/).

> The github link to the simulations on page 8 is not working.

Thank you for pointing this out – we have corrected the link.

> Please clarify the notation $\mathcal{N}(;,)$ in equation (2). Here, I believe it is the Gaussian density, while in the companion theory paper it denotes the normal distribution.

We use $\mathcal{N}$ to denote the Gaussian/normal density function, for which we have added a clarification to Section 2.1 following Equation 2. Additionally, in the first paragraph of Section 2.2, our previous draft included the sentence "...the regression coefficients are independently drawn from a $\mathcal{N}(0, \sigma^2\sigma_\beta^2)$ distribution...", which we have now updated for consistency with other usages of $\mathcal{N}$. Thank you for pointing out this inconsistency.

REFERENCES

Patrick Danaher, Pei Wang, and Daniela M. Witten. 2014. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76, 2 (2014), 373–397. https://doi.org/10.1111/rssb.12033

Priyam Das, Christine B Peterson, Kim-Anh Do, Rehan Akbani, and Veerabhadran Baladandayuthapani. 2020. NExUS: Bayesian simultaneous network estimation across unequal sample sizes. *Bioinformatics* 36, 3 (Feb. 2020), 798–804. https://doi.org/10.1093/bioinformatics/btz636

Sutanoy Dasgupta, Peng Zhao, Jacob Helwig, Prasenjit Ghosh, Debdeep Pati, and Bani K. Mallick. 2023. An Approximate Bayesian Approach to Covariate-dependent Graphical Modeling. https://doi.org/10.48550/arXiv.2303.08979 arXiv:2303.08979 [stat].

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9, 3 (July 2008), 432–441. https://doi.org/10.1093/biostatistics/kxm045

Chen Gao, Yunzhang Zhu, Xiaotong Shen, and Wei Pan. 2016. Estimation of multiple networks in Gaussian mixture models. *Electronic journal of statistics* 10 (2016), 1133–1154. https://doi.org/10.1214/16-EJS1135

Botao Hao, Will Wei Sun, Yufeng Liu, and Guang Cheng. 2018. Simultaneous Clustering and Estimation of Heterogeneous Graphical Models. *Journal of Machine Learning Research* 18, 217 (2018), 1–58. http://jmlr.org/papers/v18/17-019.html

Chunshan Liu, Daniel R. Kowal, and Marina Vannucci. 2022. Dynamic and robust Bayesian graphical models. *Statistics and Computing* 32, 6 (Nov. 2022), 105. https://doi.org/10.1007/s11222-022-10177-0

Yang Ni, Francesco C. Stingo, and Veerabhadran Baladandayuthapani. 2019. Bayesian Graphical Regression. *J. Amer. Statist. Assoc.* 114, 525 (Jan. 2019), 184–197. https://doi.org/10.1080/01621459.2017.1389739 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2017.1389739.

Yang Ni, Francesco C. Stingo, and Veerabhadran Baladandayuthapani. 2022. Bayesian covariate-dependent Gaussian graphical models with varying structure. *The Journal of Machine Learning Research* 23, 1 (Jan. 2022), 242:11049–242:11077.

Christine B. Peterson, Francesco C. Stingo, and Marina Vannucci. 2015. Bayesian Inference of Multiple Gaussian Graphical Models. *J. Amer. Statist. Assoc.* 110, 509 (March 2015), 159–174. https://doi.org/10.1080/01621459.2014.896806

Huitong Qiu, Fang Han, Han Liu, and Brian Caffo. 2016. Joint Estimation of Multiple Graphical Models from High Dimensional Time Series. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 78, 2 (March 2016), 487–504. https://doi.org/10.1111/rssb.12123

Mingyang Ren, Sanguo Zhang, Qingzhao Zhang, and Shuangge Ma. 2022. Gaussian graphical model-based heterogeneity analysis via penalized fusion. *Biometrics* 78, 2 (2022), 524–535. https://doi.org/10.1111/biom.13426

Elin Shaddox, Francesco C. Stingo, Christine B. Peterson, Sean Jacobson, Charmion Cruickshank-Quinn, Katerina Kechris, Russell Bowler, and Marina Vannucci. 2018. A Bayesian Approach for Learning Gene Networks Underlying Disease Severity in COPD. *Statistics in biosciences* 10, 1 (2018), 59–85. https://doi.org/10.1007/s12561-016-9176-6

Jilei Yang and Jie Peng. 2020. Estimating Time-Varying Graphical Models. *Journal of Computational and Graphical Statistics* 29, 1 (Jan. 2020), 191–202. https://doi.org/10.1080/10618600.2019.1647848 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10618600.2019.1647848.

Jingfei Zhang and Yi Li. 2023. High-Dimensional Gaussian Graphical Regression Models with Covariates. *J. Amer. Statist. Assoc.* 118, 543 (July 2023), 2088–2100. https://doi.org/10.1080/01621459.2022.2034632 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2022.2034632.