SpiderLearner: An Ensemble Method for Gaussian Graphical Model Estimation

Kate Hoff Shutta

ENAR 2021 Spring Meeting

March 15, 2021

UMassAmherst

School of Public Health
& Health Sciences

Biostatistics and Epidemiology

Undirected Graphical Models

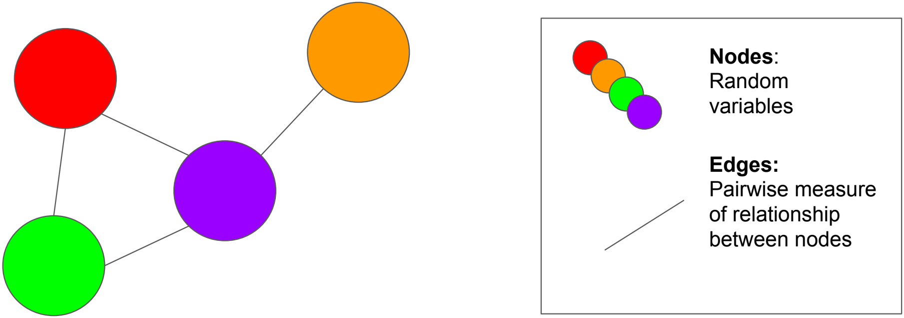


Figure 1: An undirected graphical model consists of a set of nodes and edges capturing relationships between the nodes.

The Gaussian Graphical Model

- Suppose our random variables have the distribution

$$\underline{X} = (X_1, \ldots, X_P) \sim MVN(\underline{\mu}, \Sigma) \tag{1}$$

The Gaussian Graphical Model

- Suppose our random variables have the distribution

$$\underline{X} = (X_1, \ldots, X_P) \sim MVN(\underline{\mu}, \Sigma) \tag{1}$$

- Construct an undirected graphical model where edge weights correspond to partial correlations:

$$\rho_{X_i, X_j | V \setminus \{X_i, X_j\}} = \frac{Cov[X_i, X_j | X_{-ij}]}{\sqrt{Var[X_i | X_{-ij}]}\sqrt{Var[X_j | X_{-ij}]}} \tag{2}$$

The Gaussian Graphical Model

- Suppose our random variables have the distribution

$$\underline{X} = (X_1, \ldots, X_P) \sim MVN(\underline{\mu}, \Sigma) \tag{1}$$

- Construct an undirected graphical model where edge weights correspond to partial correlations:

$$\rho_{X_i, X_j | V \setminus \{X_i, X_j\}} = \frac{Cov[X_i, X_j | X_{-ij}]}{\sqrt{Var[X_i | X_{-ij}]}\sqrt{Var[X_j | X_{-ij}]}} \tag{2}$$

- Absence of an edge means zero partial correlation $\iff$ conditional independence in the Gaussian setting

Conditional Independence in GGMs

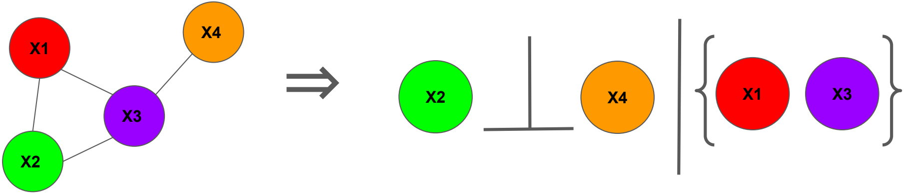


Figure 2: In GGMs, we have the property that $X_i \perp X_j | V \backslash \{X_i, X_j\}$. For example, in this model, X_2 and X_4 are conditionally independent given X_1 and X_3.

GGM Estimation with the Graphical LASSO

- Hastie, Friedman, and Tibshirani, 2008 [1]
- Assume $X_1, \ldots, X_n \sim MVN(\mu, \Theta^{-1})$, where Θ is the inverse covariance (precision) matrix
- Estimate Θ, then convert to GGM using the well-known relationship:

$$\rho_{X_i, X_j | V \setminus \{X_i, X_j\}} = -\frac{\Theta_{ij}}{\sqrt{\Theta_{ii}\Theta_{jj}}} \tag{3}$$

- Consider the penalized log likelihood for parameter Θ and sample covariance S:

$$\ell(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda||\Theta||_1 \tag{4}$$

- Maximization of (4) with respect to Θ yields a sparse estimated precision matrix

[1] Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. Biostatistics, 9(3), 432-441.

No shortage of ways to fit a model!

R Package	Algorithm	Criterion	Hyperparameters	Regularized
`huge`	`glasso`	eBIC	$\gamma = 0.5$	yes
`huge`	`glasso`	eBIC	$\gamma = 0$	yes
`huge`	`glasso`	RIC	n/a	yes
`huge`	`glasso`	StARS	$\texttt{thres} = 0.1$	yes
`huge`	`glasso`	StARS	$\texttt{thres} = 0.05$	yes
`hglasso`	`hglasso`	BIC-type	n/a	yes
`qgraph`	`EBICglasso`	eBIC	$\gamma = 0.5$	yes
`qgraph`	`EBICglasso`	eBIC	$\gamma = 0$	yes
`base`	MLE	n/a	n/a	no

Table 1: An example of some of the available methods for GGM estimation.

Motivation for Ensemble Model

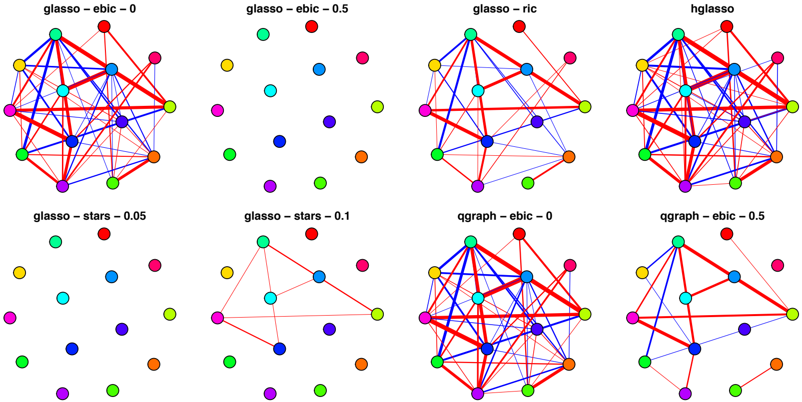


Figure 3: 8 different methods were used to fit GGMs to 14 genes from an ovarian carcinoma gene set[2] using data from the `curatedOvarianData` R package[3].

[2] Köhler et al. (2021), HPO term HP:0025318
[3] Ganzfried et al. (2013)

So ... What's a researcher to do?

- In our experience, no single estimation method seems to systematically outperform the others
- Performance of an estimation method appears to depend on network topology, which is *a priori* unknown
- Results may be highly variable to the selected method, which has implications for practical interpretation
- A researcher is left to make their best guess about which method to use

Can we find a data-adaptive way to combine the methods, forming an **ensemble** network estimate and removing the guesswork from the process?

Combining Methods to Improve Model

- Suppose we try M different network estimation methods and attain estimates $\hat{\Theta}_1, \ldots, \hat{\Theta}_M$
- Our general question is: can a convex combination of these can do at least as well as each method alone, and perhaps do even better?

$$E[\Theta] = \alpha_1 \hat{\Theta}_1 + \alpha_2 \hat{\Theta}_2 + \cdots + \alpha_M \hat{\Theta}_M ; \sum_{i=1}^{M} \alpha_i = 1, \alpha_i \geq 0 \tag{5}$$

- Van der Laan et al. have shown that the answer to this question is, in certain cases, yes! [4]

[4] Van der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super Learner.

The SpiderLearner Estimator

$$\hat{\Theta}_{SL} = \hat{\alpha}_1\hat{\Theta}_1 + \hat{\alpha}_2\hat{\Theta}_2 + \cdots + \hat{\alpha}_M\hat{\Theta}_M; \sum_{i=1}^{M} \hat{\alpha}_i = 1; \hat{\alpha}_i \geq 0 \tag{6}$$

- The estimates $\hat{\Theta}_1, \ldots, \hat{\Theta}_M$ have already been learned from the data, using one of the M candidate methods
- To construct the estimator, we just need to obtain the estimated coefficients $\hat{\alpha}_1, \ldots, \hat{\alpha}_M$
- This is done through a likelihood-based cross-validation approach

5-fold CV Example

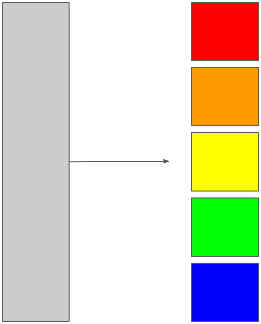


Figure 4: We begin by partitioning the data into five non-overlapping folds.

5-fold CV Example

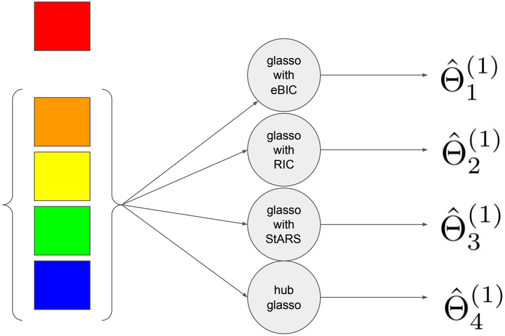
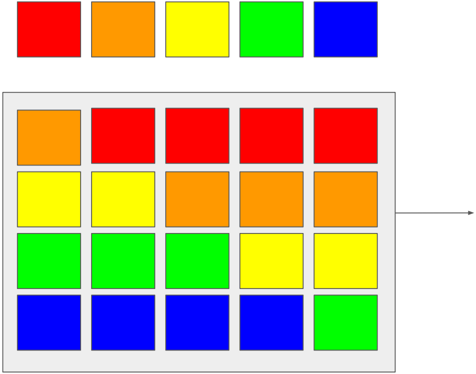


Figure 5: Next, we hold Fold 1 out of the data and train network estimates on Folds 2-5. The notation $\hat{\Theta}_2^{(1)}$, for example, refers to the network estimated by Method 2 (glasso with RIC) when leaving Fold 1 out of the dataset.

5-fold CV Example



4 methods x 5 folds = 20 estimates

$$\begin{pmatrix} \hat{\Theta}_1^{(1)} & \hat{\Theta}_1^{(2)} & \hat{\Theta}_1^{(3)} & \hat{\Theta}_1^{(4)} & \hat{\Theta}_1^{(5)} \\ \hat{\Theta}_2^{(1)} & \hat{\Theta}_2^{(2)} & \hat{\Theta}_2^{(3)} & \hat{\Theta}_2^{(4)} & \hat{\Theta}_2^{(5)} \\ \hat{\Theta}_3^{(1)} & \hat{\Theta}_3^{(2)} & \hat{\Theta}_3^{(3)} & \hat{\Theta}_3^{(4)} & \hat{\Theta}_3^{(5)} \\ \hat{\Theta}_4^{(1)} & \hat{\Theta}_4^{(2)} & \hat{\Theta}_4^{(3)} & \hat{\Theta}_4^{(4)} & \hat{\Theta}_4^{(5)} \end{pmatrix}$$

Figure 6: We repeat this process across all five folds, obtaining a 4×5 array of network estimates $\Theta_i^{(k)}$; $i = 1, \ldots, 4$; $k = 1, \ldots, 5$.

5-fold CV Example

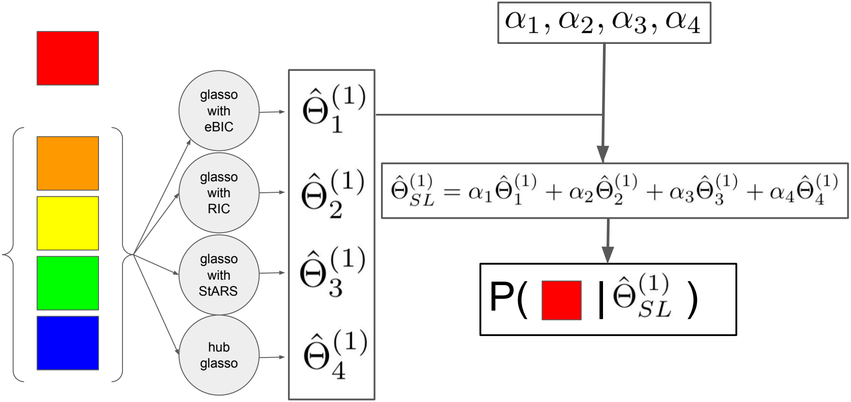


Figure 7: For each input α, we can calculate the likelihood of the test data given the estimates.

5-fold CV Example

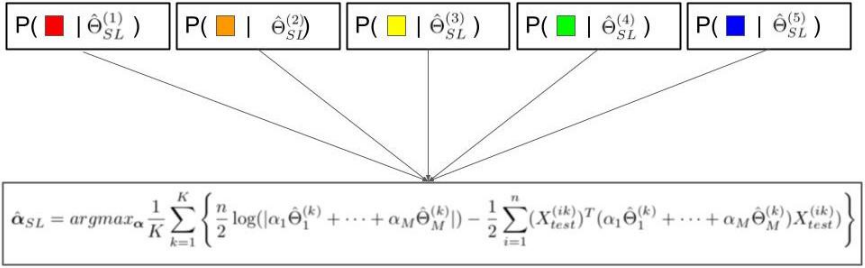


Figure 8: We repeat this process across all folds, and average the results to calculate the objective function. The estimates $\hat{\alpha}$ are then found by maximizing this function with respect to $\boldsymbol{\alpha}$.

Simulation Study Design

- Sample size and number of predictors:

Simulation	n	p	m	$(4/5 * n)/m$	Dimensionality
A	10,000	50	1275	6.275	Low
B	1,600	50	1275	1.004	Low
C	100	50	1275	0.0627	High
D	60	100	5050	0.0079	High

- Network topologies:
  - Random
  - Small world
  - Scale-free
  - Hub-and-spoke
- Densities: Low (6% dense), high (20% dense)

Simulation Study Error Metrics

- Relative Frobenius norm (RFN) of error matrix: how far off is the estimated precision matrix from the true, gold-standard precision matrix?

$$\delta_{ij} = \hat{\theta}_{SL,ij} - \theta_{ij} \tag{7}$$

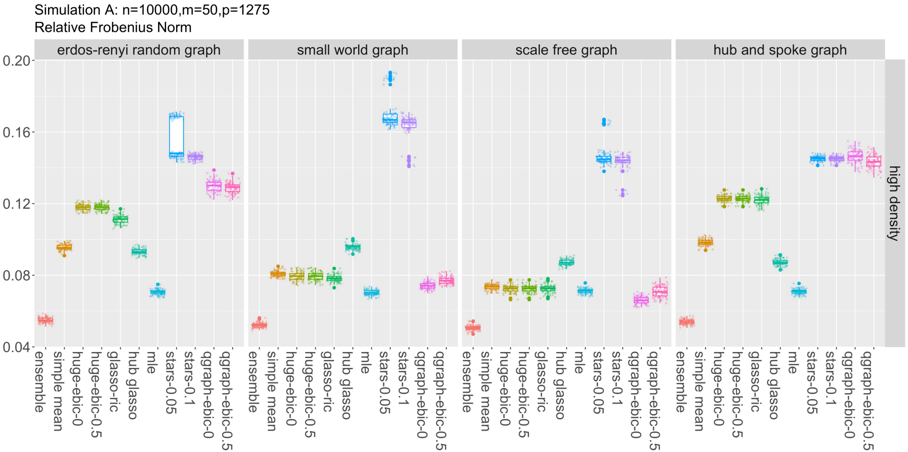
$$||\Delta||_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p}\delta_{ij}^2} \tag{8}$$

$$RFN = \frac{||\Delta||_F}{||\Theta||_F} \tag{9}$$

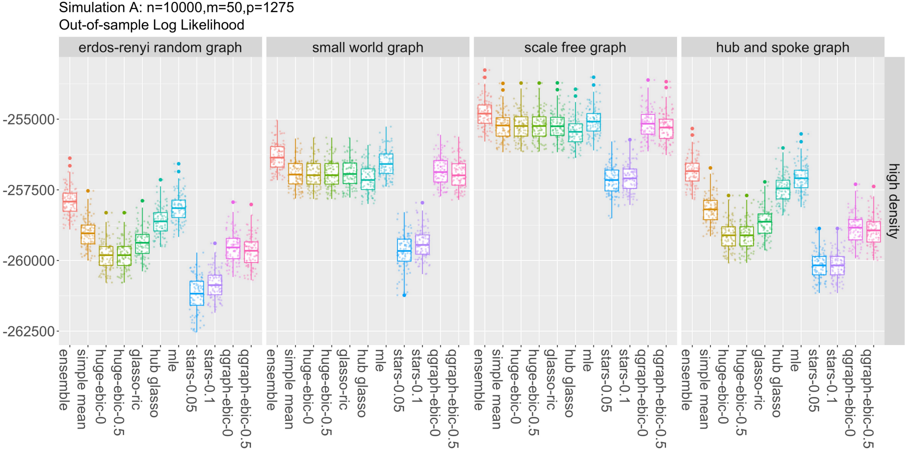
- Out-of-sample likelihood: for a new, independent sample X_{test} from the same data-generating distribution, what is the likelihood of the estimate $\hat{\Theta}_{SL}$?

$$\ell(\hat{\Theta}_{SL}) \propto \frac{n}{2}\log(|\hat{\Theta}_{SL}|) - \frac{1}{2}\sum_{i=1}^{n} X_{test,i}^T \hat{\Theta}_{SL} X_{test,i} \tag{10}$$

Ensemble Model Results: RFN



Ensemble Model Results: Out-of-sample Likelihood



Takeaway from Simulation

- In both low-dimensional and high-dimensional cases, and under a wide range of gold-standard network topologies, SpiderLearner is able to perform as well as, or better than, the best candidate method
- SpiderLearner also outperforms a simple mean of the candidate methods
- Using SpiderLearner is a practical way to optimize the complicated decision-making process of selecting a method for GGM estimation

Implementation as Open-Source Software

- Code and examples (in alpha stage) available at https://github.com/katehoffshutta/SpiderLearner
- SpiderLearner flexibly accommodates user-defined GGM estimation methods with an object-oriented programming structure
  - Any function that takes in multivariate data and outputs a matrix can be implemented as a `Candidate` subclass
- On a Macbook Air using one core, the runtime for estimating one ensemble model with 9 candidate methods on an $n = 260, p = 114, m = 6786$ dataset is around 10 minutes (K=10 folds).
- Parallel processing option is available to decrease runtime of K-fold cross-validation

Work in Progress

Methods

- Investigate variability of estimated networks
- Assess asymptotic properties of the estimator
- Develop rules of thumb for choosing the number of folds and the library of candidate methods

Software

- Add additional features such as bootstrap-based confidence intervals for each estimated edge weight
- Beta testing
- Publish and maintain the code as an R package

Acknowledgements

Many thanks to:

- Balasubramanian Lab: Alexis Edozie, Minsu Kim, Yukun Li, Ryan Sheehan, Yubing Yao, Yibai Zhao
- Administrative Support: Tonya Menard, Deb Osowski, Paula Stamps

Research reported in this presentation was supported by the following funding sources:

- UMass SPHHS Dean's Fellowship
- NIH/NIA R01 AG051600
- NIH/NLM R01 LM013444

Questions?

UMassAmherst

School of Public Health
& Health Sciences
Biostatistics and Epidemiology