Learning to Discover Sparse Graphical Models

ICML 2017 (Belilovsky et al., 2017)

Eugene Belilovsky
ESAT-PSI, KU Leuven
INRIA
University of Paris-Saclay

Kyle Kastner
University of Montreal

Gael Varoquaux
INRIA

Matthew B. Blaschko
ESAT-PSI, KU Leuven

Presented by:
Haluk Dogan
https://haluk.github.io/
hlk.dogan@gmail.com

Department of Computer Science
University of Nebraska-Lincoln

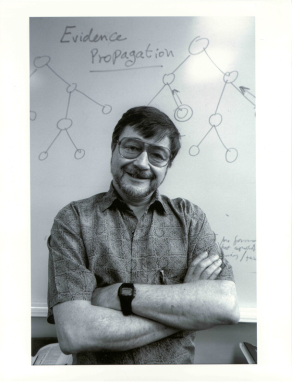
February 11, 2019

Graphical Models

Geoffrey Hinton and Chris Stephenson

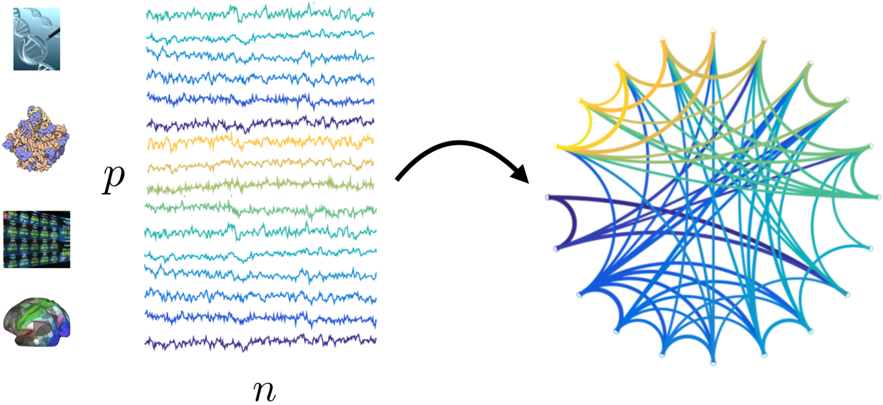
Learning to Discover Sparse Graphical Models

Eugene Belilovsky[1,2,3] Kyle Kastner[4] Gael Varoquaux[2] Matthew B. Blaschko[1]



Judea Pearl

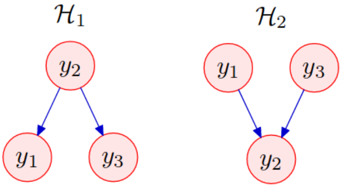
Graphical Models (cont'd)



Outline

Preliminaries for Gaussian Graphical Models

- $X = [X_1, X_2, \ldots, X_p]$ and $X \sim N(\mu, \mathbf{\Sigma})$
- $p(X; \mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)\right)$
- Mean $\mu \in \mathbb{R}^p$
- Covariance matrix $\mathbf{\Sigma} \in \mathbb{S}^p_{++}$
- Precision matrix $\Theta = \mathbf{\Sigma}^{-1}$



Belief Network[1]

A
$$\begin{bmatrix} 9 & 3 & 1 \\ 3 & 9 & 3 \\ 1 & 3 & 9 \end{bmatrix}$$

B
$$\begin{bmatrix} 8 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 8 \end{bmatrix}$$

C
$$\begin{bmatrix} 9 & 3 & 0 \\ 3 & 9 & 3 \\ 0 & 3 & 9 \end{bmatrix}$$

D
$$\begin{bmatrix} 9 & -3 & 0 \\ -3 & 10 & -3 \\ 0 & -3 & 9 \end{bmatrix}$$

[1] "The Humble Gaussian Distribution" by David MacKay

Graphical Lasso

- $\mathbf{X}_{n \times p}$, $\mu = 0^p$ and $\mathbf{\Sigma} \in \mathbb{S}^p_{++}$
- Our task is to estimate $\mathbf{\Sigma}$
  - Challenging problem when $n \ll p$
  - Ordinary MLE does not exits
    - Poorly behaved
    - Regularization is needed (ℓ_1 norm)
  - Assumption: $\Theta = \mathbf{\Sigma}^{-1}$ is sparse

Objective Function

$$f_{gl}(\hat{\mathbf{\Sigma}}) = \arg\min_{\mathbf{\Theta} \succ 0} -\log|\Theta| + \mathrm{Tr}(\hat{\mathbf{\Sigma}}\Theta) + \lambda\|\Theta\|_1$$

Graphical Lasso (cont'd)

Why do we put sparsity constraint?

- $\mathbf{\Sigma}$ is $p \times p$ matrix
- Estimating $\mathbf{\Sigma}$ requires $O(p^2)$ measurements
- Each observation $X_i \in \mathbb{R}^p$ provides p scalar values. $O(p)$ is not enough to estimate $\mathbf{\Sigma}$
- Sparse graphical model:
  - $|E| \ll O(p^2)$
  - Suppose each vertex (variable) is connected to at most $d \ll p$ other vertices, there are only $O(dp)$ edges in the graph

Graphical Lasso (cont'd)

Alternative penalties:

- Fused graphical lasso, Group graphical lasso (Danaher et al., 2014)
- Elastic net penalty (combines ℓ_1 and ℓ_2) (Ryali et al., 2012)
- Mixed norm ℓ_{21} (Varoquaux et al., 2010)

Challenges:

- Novel surrogates for structured-sparsity assumptions on MRF structures
- Priors need to be formulated
- Regularization parameters are often unintuitive
  - Model selection becomes difficult

Proposed Approach

- Learn the estimator
  - Select a function from a large flexible function class by risk minimization for edge estimation
- Sampling from a distribution of graphs and empirical covariances with desired properties
- Polynomial function
  - Neural network
  - Function class is CNN

Learning the Estimator

- $\mathbf{X} \in \mathbb{R}^{n \times p}$
- $G = (V, E)$ be an undirected and unweighted graph
- $\mathcal{L} = \{0, 1\}$ and $N_e = \frac{p(p-1)}{2}$ the maximum possible edges
- $Y^{ij} = \begin{cases} 0 & x_i \perp x_j | x_{V \setminus i,j} \\ 1 & x_i \not\perp x_j | x_{V \setminus i,j}. \end{cases}$
- $\hat{Y} = g_w(\mathbf{X})$ is an approximate structure discovery method
  - $\hat{\mathbf{\Sigma}}$: $g_w(\mathbf{X}) := f_w(\hat{\mathbf{\Sigma}})$
- We want to minimize expected risk: $R(f) = \mathbb{E}_{(\hat{\mathbf{\Sigma}}, Y) \sim \mathbb{P}}[l(f(\hat{\mathbf{\Sigma}}), Y)]$
  - $\mathbb{P}$ on $\mathbb{R}^{p \times p} \times \mathcal{L}^{N_e}$
  - $l : \mathcal{L}^{N_e} \times \mathcal{L}^{N_e} \to \mathbb{R}^+$ (0/1 loss function)

Learning the Estimator (cont'd)

- Distribution $\mathbb{P}$ may not be tractable:
- Empirical risk minimization:
  - N samples $\{Y_k, \mathbf{\Sigma}_k\}_{k=1}^N$ drawn from $\mathbb{P}$
  - $\min\limits_w \frac{1}{N} \sum_{k=1}^N l(f_w(\hat{\mathbf{\Sigma}}_k), Y_k)$
  - $\hat{l} : \mathbb{R}^{N_e} \times \mathcal{L}^{N_e}$
    - 0/1 loss is not convex
    - $\sum\limits_{i \neq j} \left(Y^{ij} \log(f_w^{ij}(\hat{\mathbf{\Sigma}})) + (1 - Y^{ij}) \log(1 - f_w^{ij}(\hat{\mathbf{\Sigma}}))\right).$

Algorithm 1 Training a GGM edge estimator

for i $\in \{1, .., N\}$ **do**
  Sample $G_i \sim \mathbb{P}(G)$
  Sample $\mathbf{\Theta}_i \sim \mathbb{P}(\mathbf{\Theta}|G = G_i)$
  $\mathbf{X}_i \leftarrow \{x_j \sim N(0, \mathbf{\Theta}_i^{-1})\}_{j=1}^n$
  Construct $(Y_i, \hat{\mathbf{\Sigma}}_i)$ pair from $(G_i, \mathbf{X_i})$
end for
Select Function Class $\mathcal{F}$ (e.g. CNN)
Optimize: $\min\limits_{f \in \mathcal{F}} \frac{1}{N} \sum_{k=1}^N \hat{l}(f(\hat{\mathbf{\Sigma}}_k), Y_k))$

Neural Network Graph Estimator

- If the data is standardized, each entry of $\mathbf{\Sigma}$ corresponds to the correlation $\rho_{i,j}$
- dth-order partial correlation can be obtained from $(d-1)$th-order partial correlation

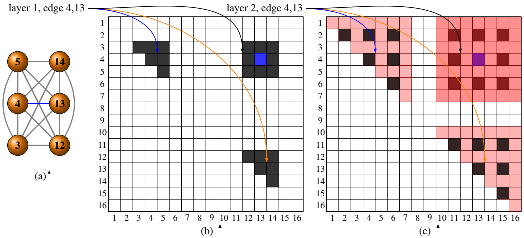
$$\rho(i,j|\mathbf{Z}) = \frac{\rho(i,j|\mathbf{Z}\setminus\{z_0\}) - \rho\left(i,z_0|\mathbf{Z}\setminus\{z_0\}\right)\rho\left(j,z_0|\mathbf{Z}\setminus\{z_0\}\right)}{\sqrt{(1-\rho^2\left(i,z_0|\mathbf{Z}\setminus\{z_0\}\right)\left(1-\rho^2\left(j,z_0|\mathbf{Z}\setminus\{z_0\}\right)\right)}}$$

$$\rho_{i,j|\mathbf{Z}} = (\rho_{i,j|\mathbf{Z}\setminus z_o} - \rho_{i,z_o|\mathbf{Z}\setminus z_o}\rho_{j,z_o|\mathbf{Z}\setminus z_o})\frac{1}{D}$$

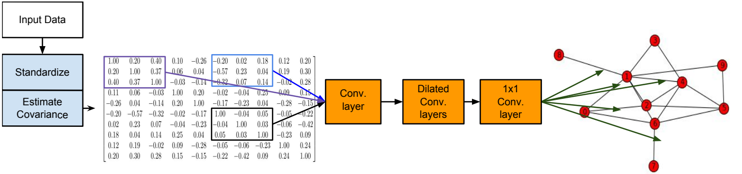
Using gradient descent, a neural network with only two layers can learn a polynomial function of degree d to arbitrary precision given sufficient hidden units (Andoni et al., 2014)

Neural Network Graph Estimator (cont'd)

- DP can yield polynomial computation time and require only low order polynomial computations



$$
\begin{aligned}
o^0_{i,j} &= p_{i,j} \\
o^1_{i,j} &= f_{w^1}(o^0_{i,j}, o^0_{i\text{-}1,j}, o^0_{i,j\text{-}1}, o^0_{i+1,j\text{-}1}, ..) \\
o^2_{i,j} &= f_{w^2}(o^1_{i,j}, o^1_{i\text{-}d_2,j}, o^1_{i,j\text{-}d_2}, o^1_{i+d_2,j\text{-}d_2}, ..) \\
o^l_{i,j} &= f_{w^l}(o^{l\text{-}1}_{i,j}, o^{l\text{-}1}_{i\text{-}d_l,j}, o^{l\text{-}1}_{i,j\text{-}d_l}, o^{l\text{-}1}_{i+d_l,j\text{-}d_l} ..) \\
\hat{y}_{i,j} &= \sigma(w^{l+1} o^l_{i,j})
\end{aligned}
$$



Experiments

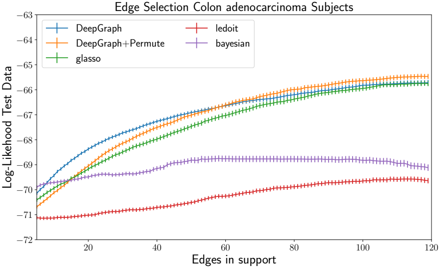
Experimental Setup	Method	Prec@5%	AUC	CE
Gaussian Random Graphs ($n = 35, p = 39$)	Glasso	0.361 $\pm$ 0.011	0.624 $\pm$ 0.006	0.07
	Glasso (optimal)	0.384 $\pm$ 0.011	0.639 $\pm$ 0.007	0.07
	BDGraph	0.441 $\pm$ 0.011	0.715 $\pm$ 0.007	0.28
	DeepGraph-39	0.463 $\pm$ 0.009	0.738 $\pm$ 0.006	0.07
	DeepGraph-39+Perm	$\mathbf{0.487 \pm 0.010}$	$\mathbf{0.740 \pm 0.007}$	0.07
Gaussian Random Graphs ($n = 100, p = 39$)	Glasso	0.539 $\pm$ 0.014	0.696 $\pm$ 0.006	0.07
	Glasso (optimal)	0.571 $\pm$ 0.011	0.704 $\pm$ 0.006	0.07
	BDGraph	$\mathbf{0.648 \pm 0.012}$	$\mathbf{0.776 \pm 0.007}$	0.16
	DeepGraph-39	0.567 $\pm$ 0.009	0.759 $\pm$ 0.006	0.07
	DeepGraph-39+Perm	0.581 ± 0.008	0.771 ± 0.006	0.07
Gaussian Random Graphs ($n = 15, p = 39$)	Glasso	0.233 $\pm$ 0.010	0.566 $\pm$ 0.004	0.07
	Glasso (optimal)	0.263 $\pm$ 0.010	0.578 $\pm$ 0.004	0.07
	BDGraph	0.261 $\pm$ 0.009	0.630 $\pm$ 0.007	0.41
	DeepGraph-39	0.326 $\pm$ 0.009	0.664 $\pm$ 0.008	0.08
	DeepGraph-39+Perm	$\mathbf{0.360 \pm 0.010}$	$\mathbf{0.672 \pm 0.008}$	0.08
Laplace Random Graphs ($n = 35, p = 39$)	Glasso	0.312 $\pm$ 0.012	0.605 $\pm$ 0.006	0.07
	Glasso (optimal)	0.337 $\pm$ 0.011	0.622 $\pm$ 0.006	0.07
	BDGraph	0.298 $\pm$ 0.009	0.687 $\pm$ 0.007	0.36
	DeepGraph-39	0.415 $\pm$ 0.010	0.711 $\pm$ 0.007	0.07
	DeepGraph-39+Perm	$\mathbf{0.445 \pm 0.011}$	$\mathbf{0.717 \pm 0.007}$	0.07
Gaussian Small-World Graphs (n=35,p=39)	Glasso	0.387 $\pm$ 0.012	0.588 $\pm$ 0.004	0.11
	Glasso (optimal)	0.453 $\pm$ 0.008	0.640 $\pm$ 0.004	0.11
	BDGraph	0.428 $\pm$ 0.007	0.691 $\pm$ 0.003	0.17
	DeepGraph-39	$\mathbf{0.479 \pm 0.007}$	0.709 $\pm$ 0.003	0.11
	DeepGraph-39+Perm	0.453 $\pm$ 0.007	$\mathbf{0.712 \pm 0.003}$	0.11
	DeepGraph-39+update	$\mathbf{0.560 \pm 0.008}$	$\mathbf{0.821 \pm 0.002}$	0.11
	DeepGraph-39+update+Perm	0.555 $\pm$ 0.007	0.805 $\pm$ 0.003	0.11

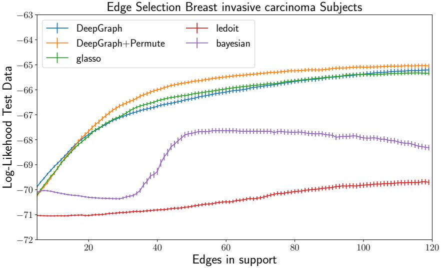
Experiments (cont'd)

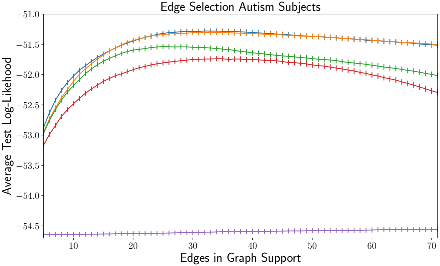
Method	Prec@0.05%	Prec@5%	AUC	CE
random	0.052 $\pm$ 0.002	0.053 $\pm$ 0.000	0.500 $\pm$ 0.000	0.05
Glasso	0.156 $\pm$ 0.010	0.055 $\pm$ 0.001	0.501 $\pm$ 0.000	0.05
Glasso (optimal)	0.162 $\pm$ 0.010	0.055 $\pm$ 0.001	0.501 $\pm$ 0.000	0.05
DeepGraph-500	0.449 $\pm$ 0.018	0.109 $\pm$ 0.002	0.543 $\pm$ 0.002	0.06
DeepGraph-500+Perm	$\mathbf{0.583 \pm 0.018}$	$\mathbf{0.116 \pm 0.002}$	$\mathbf{0.547 \pm 0.002}$	$\mathbf{0.06}$

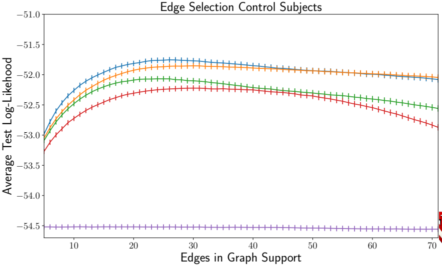
	50 nodes (s)	500 nodes (s)
`sklearn` GraphLassoCV	4.81	554.7
`BDgraph`	42.13	N/A
DeepGraph	***0.27***	***5.6***

Experiments (cont'd)









Experiments (cont'd)

	Gene BRCA	Gene COAD	ABIDE Control	ABIDE Autistic
Graph Lasso	$0.25 \pm .003$	0.34 ± 0.004	$0.21 \pm .003$	$\mathbf{0.21 \pm .003}$
Ledoit-Wolfe	0.12 ± 0.002	0.15 ± 0.003	$0.13 \pm .003$	$0.13 \pm .003$
Bdgraph	0.07 ± 0.002	0.08 ± 0.002	N/A	N/A
DeepGraph	$\mathbf{0.48 \pm 0.004}$	$\mathbf{0.57 \pm 0.005}$	$\mathbf{0.23 \pm .004}$	$0.17 \pm .003$
DeepGraph +Permute	0.42 ± 0.003	0.52 ± 0.006	$0.19 \pm .004$	$0.14 \pm .004$

Questions

Questions?

References I

Andoni, A., R. Panigrahy, G. Valiant, and L. Zhang (2014). Learning polynomials with neural networks. In *International Conference on Machine Learning*, pp. 1908–1916.

Belilovsky, E., K. Kastner, G. Varoquaux, and M. B. Blaschko (2017). Learning to discover sparse graphical models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 440–448. JMLR. org.

Danaher, P., P. Wang, and D. 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), 373–397.

References II

Ryali, S., T. Chen, K. Supekar, and V. Menon (2012). Estimation of functional connectivity in fmri data using stability selection-based sparse partial correlation with elastic net penalty. *NeuroImage 59*(4), 3852–3861.

Varoquaux, G., A. Gramfort, J.-B. Poline, and B. Thirion (2010). Brain covariance selection: better individual functional connectivity models using population prior. In *Advances in neural information processing systems*, pp. 2334–2342.