Fast and Scalable Learning of Sparse Changes in High-Dimensional Gaussian Graphical Model Structure

Beilun Wang[1]   Arshdeep Sekhon[1]   Yanjun Qi[1]

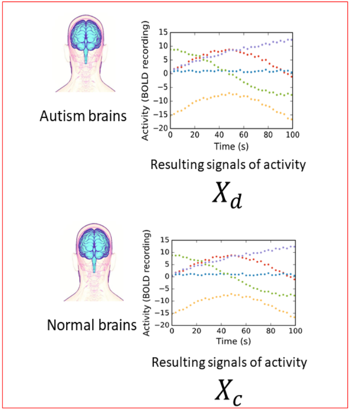
[1]University of Virginia
http://jointggm.org/

Published @ AISTAT18;
2018

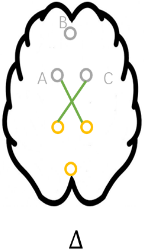
Outline

Motivation: Structure Difference Learning from two Datasets

- Two Datasets $\mathbf{X}_c$, $\mathbf{X}_d$ $\rightarrow$ Differential Network Δ.
  - Case vs. Control;
  - Autism vs. Normal;



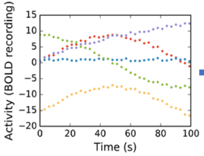
Data



Differential Network

Motivation: Estimating Graph from Dataset via sparse Gaussian Graphical Model:

- A pipeline to infer Graph from one homogeneous dataset X_d.



Resulting signals of activity

X_d

1.05	-0.23	0.05	-0.02	0.05
-0.23	1.45	-0.25	0.10	-0.25
0.05	-0.25	1.10	-0.24	0.10
-0.02	0.10	-0.24	1.10	-0.24
0.05	-0.25	0.10	-0.24	1.10

Covariances between signals

Σ_d

sGGM

1	0.2	0	0	0
0.2	1	0.2	0	0.2
0	0.2	1	0.2	0
0	0	0.2	1	0.2
0	0.2	0	0.2	1

Sparse inversion (Precision Matrix)

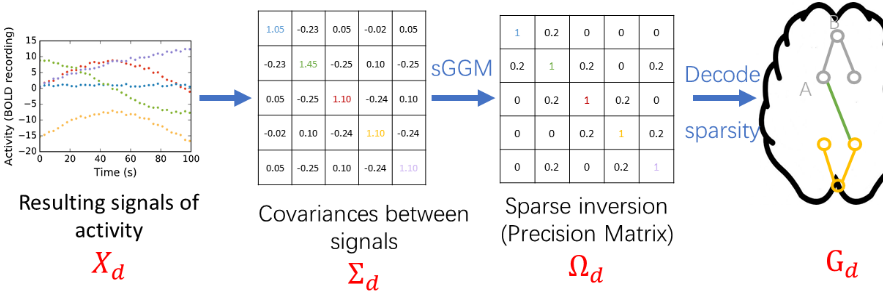
Ω_d

Decode sparsity

G_d

Motivation: Estimating Graph from Dataset via sparse Gaussian Graphical Model:

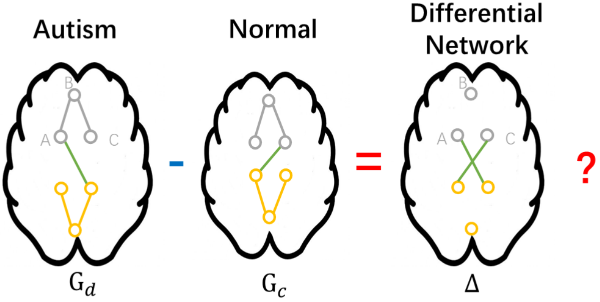
- A pipeline to infer Graph from one homogeneous dataset X_d.



- $X_c \rightarrow G_c$ is the same.
- We are more interested in the structure changes between two different but related datatsets.

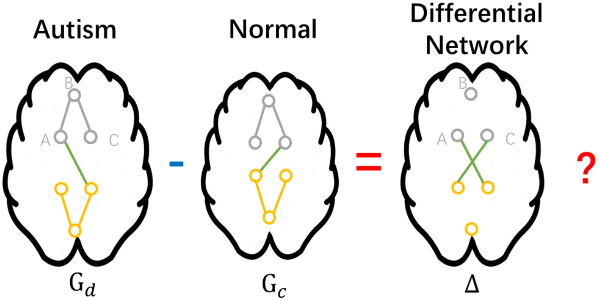
Motivation: Estimating the Difference by separately Learning Two Graphs from two datasets has Limitations

- Sparsity Assumption:
  - If estimating two graphs separately, we need to enforce sparsity assumption on both graphs
  - However, in some real-world applications, G_c, G_d are not sparse.



Motivation: Estimating the Difference by separately Learning Two Graphs from two datasets has Limitations

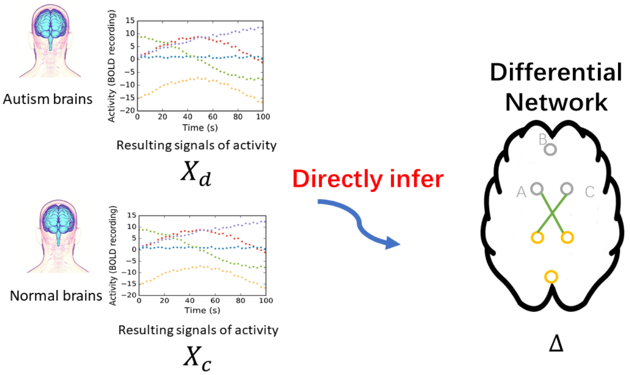
- Sparsity Assumption:
  - If estimating two graphs separately, we need to enforce sparsity assumption on both graphs
  - However, in some real-world applications, G_c, G_d are not sparse.



- Difficulty in the computation: Current methods can not scale-up. In applications like neuroscience, the number of regions (nodes) p for connectivity analysis in the human brain ranges from 160 to 800,000

Our Aim: To Learn Differential Network from two Datasets in a large-scale

- Our focus: How to directly estimate / learn Differential Network (Δ) from Two datasets ($\mathbf{X}_c$, $\mathbf{X}_d$) about the same set of features in a large scale.



Notations

X_c, X_d	Data matrix.
$\Sigma_{c,d}$	Covariance matrix.
Ω_c, Ω_d	Inverse of covariance matrix (precision matrix).
p	The total number of feature variables.
n_c, n_d	The number of samples.
Δ	The Differential Network.

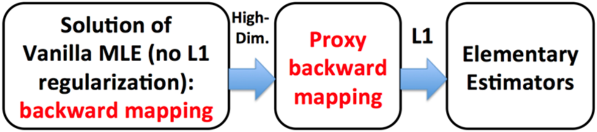
Outline

Background: Elementary Estimator for Exponential Family

- The canonical parameter θ of an exponential family distribution can be learned by the following equation.

Elementary Estimator

$$\begin{aligned} &\underset{\theta}{\operatorname{argmin}} ||\theta||_1 \\ &\text{Subject to: } ||\theta - \mathcal{B}^*(\widehat{\phi})||_\infty \leq \lambda_n \end{aligned} \tag{1.1}$$



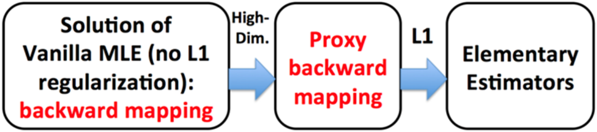
Background: Elementary Estimator for Exponential Family

- The canonical parameter θ of an exponential family distribution can be learned by the following equation.

Elementary Estimator

$$\underset{\theta}{\operatorname{argmin}} ||\theta||_1$$
$$\text{Subject to: } ||\theta - \mathcal{B}^*(\widehat{\phi})||_\infty \le \lambda_n \tag{1.1}$$

- For high-dimensional case, Vanilla MLE solutions are mostly not available. Therefore, we choose Proxy backward mapping.



Background: Elementary Estimator for sGGM

- θ is the canonical parameter of the exponential distribution.
- $\mathcal{B}^*(\widehat{\phi})$ is the backward mapping of θ. Normally, it is the solution of Vanilla MLE.
- For example, for sGGM:

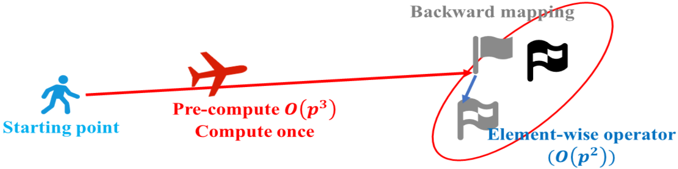
EE	θ	$\mathcal{B}^*$	$\widehat{\phi}$
EE-sGGM	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\widehat{\Sigma}$

Elementary Estimator for sGGM

$$\underset{\Omega}{\operatorname{argmin}} ||\Omega||_1$$
$$\text{Subject to: } ||\Omega - [T_v(\widehat{\Sigma})]^{-1}||_\infty \leq \lambda_n \tag{1.2}$$

Background: Elementary Estimator – visualization

- Elementary Estimator:



- Closed-form solution (non-iterative algorithm)

Background: Elementary Estimator – Advantages

- Closed-form solution (non-iterative algorithm)
- Fast computation

Background: Elementary Estimator – Advantages

- Closed-form solution (non-iterative algorithm)
- Fast computation
- Easy to prove the theoretical error bound

Background: Exponential Family Distribution

- Learn an exponential family distribution $\Longleftrightarrow$ To learn the canonical parameter θ
  - e.g., Ω is the canonical parameter of the sparse Gaussian Graphical Model

Background: Exponential Family Distribution

- Learn an exponential family distribution $\iff$ To learn the canonical parameter θ
  - e.g., Ω is the canonical parameter of the sparse Gaussian Graphical Model
- The density ratio of two Gaussian distributions is naturally an exponential family distribution.
  - We prove Δ is the canonical parameter of the density ratio distribution.
  - Therefore we can apply Elementary Estimator to estimate Δ.

Outline

Goals

Our model aims to have the following properties:

- closed-form solution.

Goals

Our model aims to have the following properties:

- closed-form solution.
- Fast and scalable algorithm.

Goals

Our model aims to have the following properties:

- closed-form solution.
- Fast and scalable algorithm.
- It provides a strong theoretical guarantee.

Proposed Method: estimating DIFFerential networks via an Elementary Estimator (DIFFEE) when high-dimensional

We model the differential network Δ as:

$$\Delta = \Omega_d - \Omega_c \tag{2.1}$$

We apply the elementary estimator to the differential network.

DIFFEE

$$\begin{gathered} \underset{\Delta}{\operatorname{argmin}} ||\Delta||_1 \\ \text{Subject to: } ||\Delta - \mathcal{B}^*(\widehat{\Sigma}_d, \widehat{\Sigma}_c)||_\infty \le \lambda_n \end{gathered} \tag{2.2}$$

Proposed method: DIFFEE – Solution

EE	θ	$\mathcal{B}^*$	$\widehat{\phi}$
EE-sGGM	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\widehat{\Sigma}$
DIFFEE	Δ	$[\mathrm{T}_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$	$\widehat{\Sigma}_d, \widehat{\Sigma}_c$

- We choose $[T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$ as the proxy backward mapping for Δ.

Proposed method: DIFFEE – Solution

EE	θ	$\mathcal{B}^*$	$\widehat{\phi}$
EE-sGGM	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\widehat{\Sigma}$
DIFFEE	Δ	$[\mathrm{T}_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$	$\widehat{\Sigma}_d, \widehat{\Sigma}_c$

- We choose $[T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$ as the proxy backward mapping for Δ.
- It is theoretical guaranteed (will talk later).

Proposed method: DIFFEE – Solution

EE	θ	$\mathcal{B}^*$	$\widehat{\phi}$
EE-sGGM	Ω	$[T_v(\widehat{\Sigma})]^{-1}$	$\widehat{\Sigma}$
DIFFEE	Δ	$[\mathrm{T}_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$	$\widehat{\Sigma}_d, \widehat{\Sigma}_c$

- We choose $[T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}$ as the proxy backward mapping for Δ.
- It is theoretical guaranteed (will talk later).

- Closed-form solution:

$$\widehat{\Delta} = S_{\lambda_n}([T_v(\widehat{\Sigma}_d)]^{-1} - [T_v(\widehat{\Sigma}_c)]^{-1}) \tag{2.3}$$

Here $[S_\lambda(A)]_{ij} = \mathrm{sign}(A_{ij}) \max(|A_{ij}| - \lambda, 0)$.

Why DIFFEE is better

- It has closed-form solution.

Why DIFFEE is better

- It has closed-form solution.
- It is faster than the previous studies:

DIFFEE	FusedGLasso	Density Ratio	Diff-CLIME
$O(p^3)$	$O(T * p^3)$	$O((n_c + p^2)^3)$	$O(p^8)$

Why DIFFEE is better

- It has closed-form solution.
- It is faster than the previous studies:

DIFFEE	FusedGLasso	Density Ratio	Diff-CLIME
$O(p^3)$	$O(T * p^3)$	$O((n_c + p^2)^3)$	$O(p^8)$

- $O(p^2)$ to tune different λ_n

Why DIFFEE is better

- It has closed-form solution.
- It is faster than the previous studies:

DIFFEE	FusedGLasso	Density Ratio	Diff-CLIME
$O(p^3)$	$O(T * p^3)$	$O((n_c + p^2)^3)$	$O(p^8)$

- $O(p^2)$ to tune different λ_n
- Theoretical guaranteed

Previous Methods: FusedGLasso for estimating differential network

- Traditionally, we estimate differential network from penalized likelihood formulation.

FusedGLasso [Danaher et al.(2013)Danaher, Wang, and Witten]

$$
\begin{aligned}
\underset{\Omega_c,\Omega_d \succ 0,\Delta}{\operatorname{argmin}} \; & n_c(-\log\det(\Omega_c) + < \Omega_c, \widehat{\Sigma}_c >) \\
& + n_d(-\log\det(\Omega_d) + < \Omega_d, \widehat{\Sigma}_d >) \\
& + \lambda_2(||\Omega_c||_1 + ||\Omega_d||_1) + \lambda_n||\Delta||_1
\end{aligned}
\tag{2.4}
$$

Previous Methods: FusedGLasso for estimating differential network

- Traditionally, we estimate differential network from penalized likelihood formulation.
- FusedGLasso adds a second penalty function fused norm $||\Delta||_1$ into the penalized likelihood formulation.

FusedGLasso [Danaher et al.(2013)Danaher, Wang, and Witten]

$$\begin{aligned} \underset{\Omega_c,\Omega_d \succ 0,\Delta}{\operatorname{argmin}} \; & n_c(-\log\det(\Omega_c) + < \Omega_c, \widehat{\Sigma}_c >) \\ & + n_d(-\log\det(\Omega_d) + < \Omega_d, \widehat{\Sigma}_d >) \\ & + \lambda_2(||\Omega_c||_1 + ||\Omega_d||_1) + \lambda_n||\Delta||_1 \end{aligned} \tag{2.4}$$

Previous Methods: FusedGLasso for estimating differential network

- Traditionally, we estimate differential network from penalized likelihood formulation.
- FusedGLasso adds a second penalty function fused norm $||\Delta||_1$ into the penalized likelihood formulation.
- $||\Delta||_1$ enforces a sparse difference structure between two graphs.

FusedGLasso [Danaher et al.(2013)Danaher, Wang, and Witten]

$$\begin{aligned}\underset{\Omega_c,\Omega_d\succ 0,\Delta}{\operatorname{argmin}}\ & n_c(-\log\det(\Omega_c) + <\Omega_c, \widehat{\Sigma}_c>) \\ & + n_d(-\log\det(\Omega_d) + <\Omega_d, \widehat{\Sigma}_d>) \\ & + \lambda_2(||\Omega_c||_1 + ||\Omega_d||_1) + \lambda_n||\Delta||_1\end{aligned} \tag{2.4}$$

Previous Methods: Diff-CLIME

- Another study to learn the Δ is through a constrained optimization formulation.

Diff-CLIME [Zhao et al.(2014)Zhao, Cai, and Li]

$$\begin{aligned} &\operatorname*{argmin}_{\Delta} ||\Delta||_1 \\ \text{Subject to: } &||\widehat{\Sigma}_c \Delta \widehat{\Sigma}_d - (\widehat{\Sigma}_c - \widehat{\Sigma}_d)||_\infty \leq \lambda_n \end{aligned} \tag{2.5}$$

Previous Methods: Diff-CLIME

- Another study to learn the Δ is through a constrained optimization formulation.
- It reduces the estimation to solve multiple linear programming problems.

Diff-CLIME [Zhao et al.(2014)Zhao, Cai, and Li]

$$\begin{gathered}\underset{\Delta}{\operatorname{argmin}} \, ||\Delta||_1 \\ \text{Subject to: } ||\widehat{\Sigma}_c \Delta \widehat{\Sigma}_d - (\widehat{\Sigma}_c - \widehat{\Sigma}_d)||_\infty \leq \lambda_n\end{gathered} \tag{2.5}$$

Previous Methods: Density Ratio

- Directly model the sparse differential network with density ratio function $r(x; \Delta)$

Density Ratio [Liu et al.(2013)Liu, Yamada, Collier, and Sugiyama]

$$\operatorname*{argmax}_{\Delta} \mathcal{L}_{\mathsf{KLIEP}}(\Delta) - \lambda_n \parallel \Delta \parallel_1 - \lambda_2 \parallel \Delta \parallel_2 \tag{2.6}$$

Previous Methods: Density Ratio

- Directly model the sparse differential network with density ratio function $r(x; \Delta)$
- Minimizes the KL divergence between $p_d(x)$ and $\widehat{p}_d(x) = r(x; \Delta) p_c(x)$.

Density Ratio [Liu et al.(2013)Liu, Yamada, Collier, and Sugiyama]

$$\underset{\Delta}{\operatorname{argmax}} \mathcal{L}_{\mathsf{KLIEP}}(\Delta) - \lambda_n \parallel \Delta \parallel_1 - \lambda_2 \parallel \Delta \parallel_2 \tag{2.6}$$

Previous Studies: Drawbacks

- The time comparison table:

FusedGLasso	Density Ratio	Diff-CLIME
$O(T * p^3)$	$O((n_c + p^2)^3)$	$O(p^8)$

Previous Studies: Drawbacks

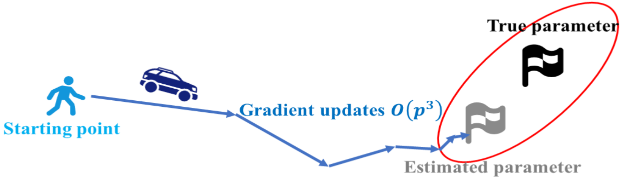
- The time comparison table:

FusedGLasso	Density Ratio	Diff-CLIME
$O(T * p^3)$	$O((n_c + p^2)^3)$	$O(p^8)$

- **Drawbacks:**
  - **I:** all of them are slow when p is large.
  - **II:** Need terative algorithm solution.
  - **III:** No theoretical analysis in the previous FusedGLasso studies.

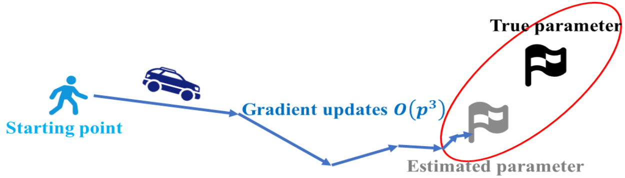
Background: DIFFEE versus Previous studies

- Previous studies:

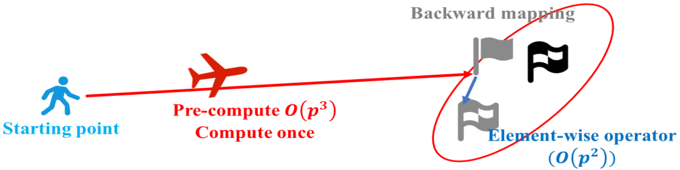


Background: DIFFEE versus Previous studies

- Previous studies:



- DIFFEE:



Outline

Theoretical Results

- error bound: $||\Delta^* - \widehat{\Delta}||$
- DIFFEE achieves similar error bound as the previous studies.

DIFFEE	FusedGLasso	Density Ratio	Diff-CLIME
$\frac{\log p}{\min(n_c, n_d)}$	N/A	$\frac{\log p}{\min(n_c, n_d)}$	$\frac{\log p}{\min(n_c, n_d)}$

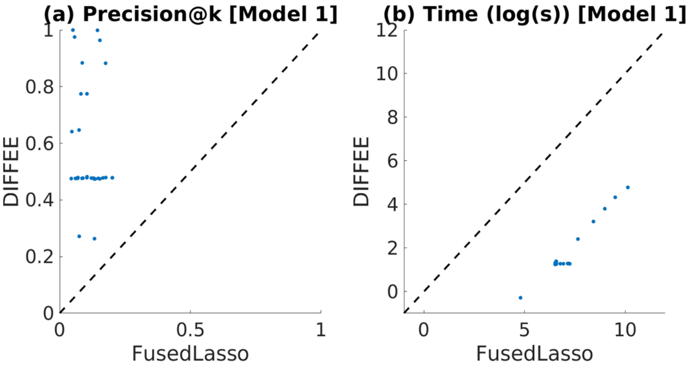
- Sharp convergence rate as the state-of-art

$$
\begin{aligned}
||\widehat{\Delta} - \Delta^*||_\infty &\leq \frac{16\kappa_1 a}{\kappa_2}\sqrt{\frac{\log p}{\min(n_c, n_d)}} \\
||\widehat{\Delta} - \Delta^*||_F &\leq \frac{32\kappa_1 a}{\kappa_2}\sqrt{\frac{k \log p}{\min(n_c, n_d)}} \\
||\widehat{\Delta} - \Delta^*||_1 &\leq \frac{64\kappa_1 a}{\kappa_2} k\sqrt{\frac{\log p}{\min(n_c, n_d)}}
\end{aligned} \tag{3.1}
$$

Outline

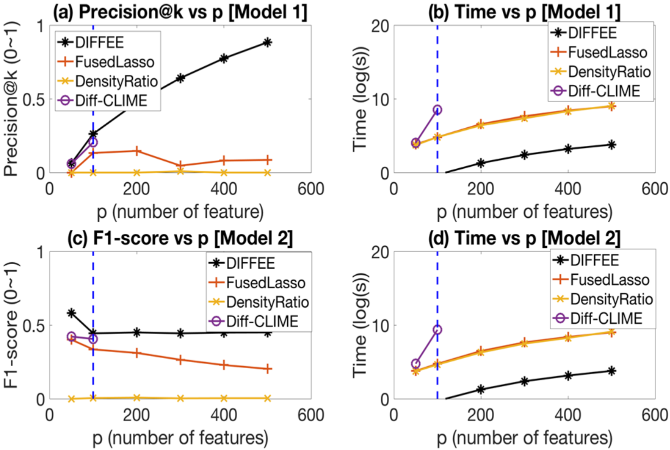
Results on Synthetic Datasets: Vs. FusedGlasso

- Comparison with the best baseline – FusedGLasso:

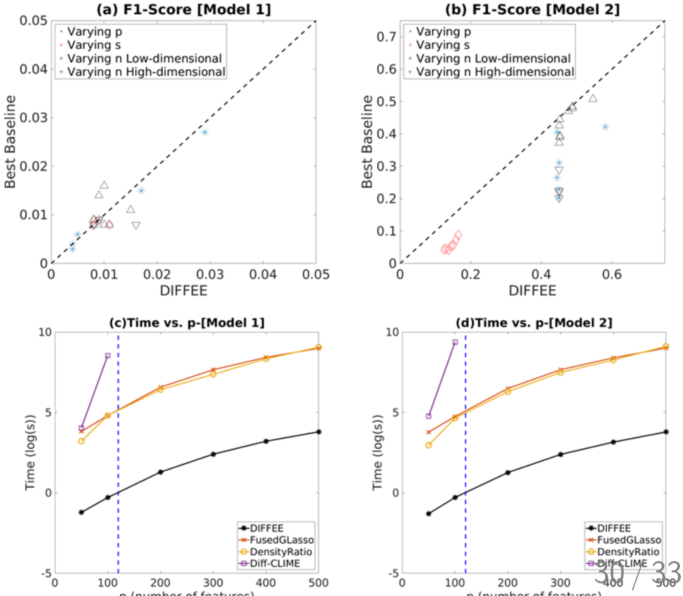


Results on Synthetic Datasets: Accuracy and Computational time When Varying p

- Compare the estimation accuracy and scalabilty of different methods



Results on Synthetic Data Results: More Hyper-parameter Variations



Results on fMRI Datasets: the Classification Accuracy

- (1) ABIDE dataset
- (2) Train the differential network and use it as the parameter of a LDA classifier

Method	DIFFEE	FusedGLasso	Diff-CLIME
Accuracy (%)	**57.58%**	56.90%	53.79%

R Package is Available !!!

- The project website: `http://jointggm.org/`
- R package "diffee":
  - `install.packages("diffee")`
  - `demo(diffeeDemo)` !
  - `https://cran.r-project.org/web/packages/diffee/index.html`

References

P. Danaher, P. Wang, and D. M. Witten.
The joint graphical lasso for inverse covariance estimation across multiple classes.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2013.

S. Liu, M. Yamada, N. Collier, and M. Sugiyama.
Change-point detection in time-series data by relative density-ratio estimation.
Neural Networks, 43:72–83, 2013.

S. D. Zhao, T. T. Cai, and H. Li.
Direct estimation of differential networks.
Biometrika, 101(2):253–268, 2014.