Estimating graph-based regression coefficients in high-dimensional linear models

Duzhe Wang

Ph.D. Preliminary Exam
March 2, 2020

Joint work with Po-Ling Loh

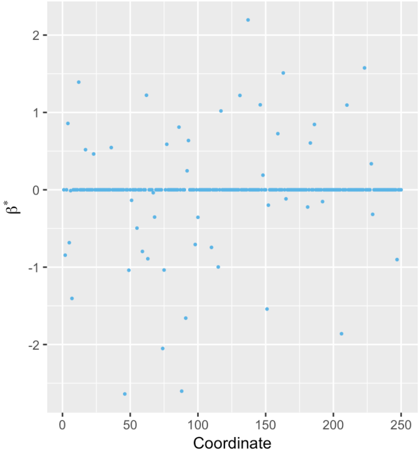
High-dimensional linear models

$$y = X\beta^* + \varepsilon$$

- $X = (X_1, \ldots, X_N)^T \in \mathbb{R}^{N \times n}$: design matrix with $X_i \in \mathbb{R}^n$
- $y = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$: response vector
- $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_N)^T \in \mathbb{R}^N$: additive noise, each component is i.i.d.
- $\beta^* \in \mathbb{R}^n$: true regression coefficients
- High-dimensional setting: $N \ll n$
- Goal: estimate the unknown regression coefficients β^*

Structural assumptions

- β^* is sparse: very common in real-world applications with high-dimensional settings



Lasso

- Tibshirani '96:

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N} \| y - X\beta \|_2^2 + \lambda \|\beta\|_1$$

- Theory of Lasso: For s-sparse regression coefficients, and sub-Gaussian random design and noise,

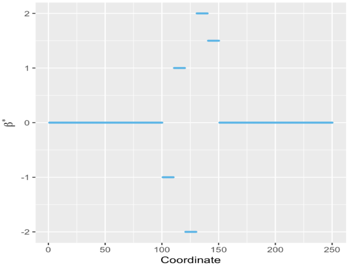
$$\|\hat{\beta}^{\text{lasso}} - \beta^*\|_2 \leq \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{s \log n}{N}}\right), \quad \|\hat{\beta}^{\text{lasso}} - \beta^*\|_1 \leq \mathcal{O}_{\mathbb{P}}\left(s\sqrt{\frac{\log n}{N}}\right),$$

and

$$\frac{1}{N}\|X(\hat{\beta}^{\text{lasso}} - \beta^*)\|_2^2 \leq \mathcal{O}_{\mathbb{P}}\left(\frac{s \log n}{N}\right)$$

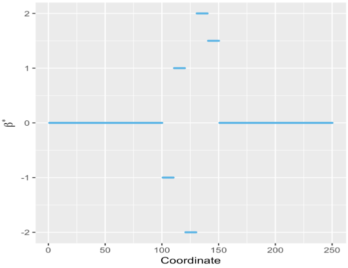
Structural assumptions

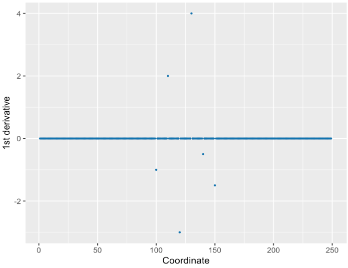
- β^* is both sparse and locally constant: common in biology applications, e.g., comparative genomic hybridization data (Tibishirani and Wang '08)



Structural assumptions

- β^* is both sparse and locally constant: common in biology applications, e.g., comparative genomic hybridization data (Tibishirani and Wang '08)





Fused Lasso

Tibshirani et al. '04:

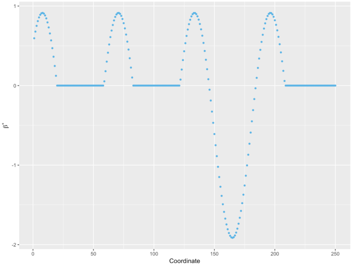
$$\hat{\beta}^{\text{fl}} = \underset{\beta \in \mathbb{R}^n}{\text{argmin}} \quad \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\Delta_u^{(1)} \beta\|_1,$$

where

$$\Delta_u^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}$$

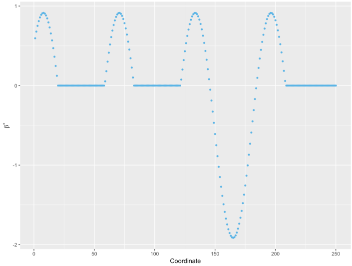
Structural assumptions

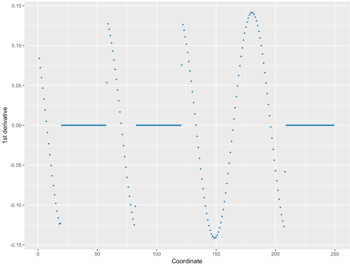
- β^* is both sparse and smooth: common in macroeconomics, financial time series analysis, and medical sciences (Kim et al. '09)



Structural assumptions

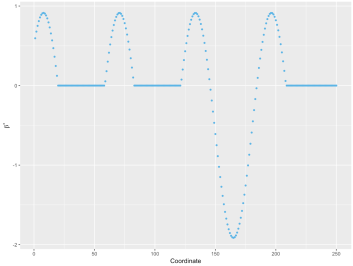
- β^* is both sparse and smooth: common in macroeconomics, financial time series analysis, and medical sciences (Kim et al. '09)

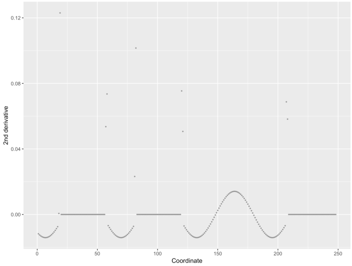




Structural assumptions

- β^* is both sparse and smooth: common in macroeconomics, financial time series analysis, and medical sciences (Kim et al. '09)





Smooth-Lasso and Spline-Lasso

- Hebiri et al. '11:

$$\hat{\beta}^{\text{smooth}} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\Delta_u^{(1)} \beta\|_2^2$$

- Guo et al. '16:

$$\hat{\beta}^{\text{spline}} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\Delta_u^{(2)} \beta\|_2^2,$$

where

$$\Delta_u^{(2)} = \begin{bmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 0 & 1 & -2 & \dots & 0 & 0 \\ \vdots & & & \ddots & & \vdots & \vdots \\ 0 & 0 & & \dots & 1 & -2 & 1 \end{bmatrix} \in \mathbb{R}^{(n-2)\times n}$$

Summary

$$\left.\begin{array}{l}\text{• Fused Lasso}\\ \text{• Smooth-Lasso}\\ \text{• Spline-Lasso}\end{array}\right\}\ \textbf{Adaptive estimation}$$

- Lasso: a baseline method
- Fused Lasso
- Smooth-Lasso
- Spline-Lasso

Summary

- Lasso: a baseline method
- Fused Lasso
- Smooth-Lasso
- Spline-Lasso

$\left.\begin{array}{l}\text{Fused Lasso}\\ \text{Smooth-Lasso}\\ \text{Spline-Lasso}\end{array}\right\}$ **Adaptive estimation**

- But these adaptive estimation methods implicitly assume β^* is formed in a sequence form

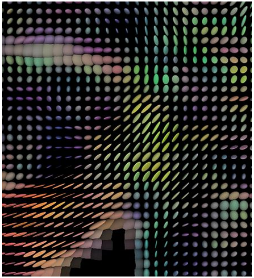
Summary

- Lasso: a baseline method
- Fused Lasso
- Smooth-Lasso
- Spline-Lasso

$\left.\begin{array}{l}\text{Fused Lasso}\\ \text{Smooth-Lasso}\\ \text{Spline-Lasso}\end{array}\right\}$ **Adaptive estimation**

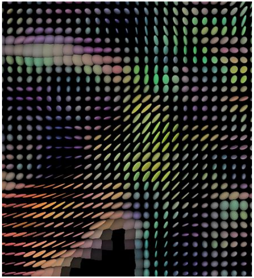
- But these adaptive estimation methods implicitly assume β^* is formed in a sequence form
- What if β^* is modeled in the form of a complex graph?

Motivating examples

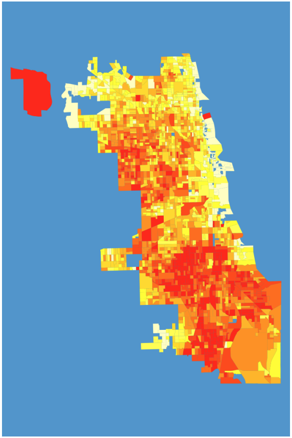


(a) DTI

Motivating examples

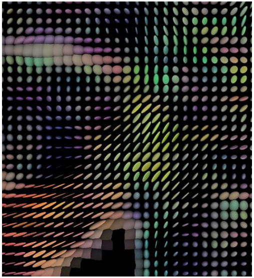


(a) DTI

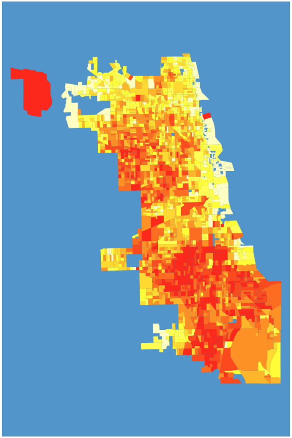


(b) Crime data

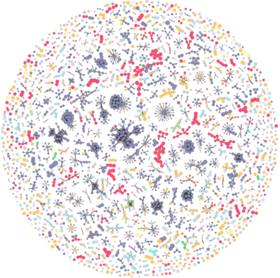
Motivating examples



(a) DTI



(b) Crime data



(c) Gene-pathway network

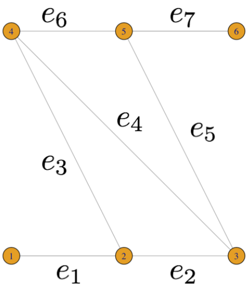
Outline of the remaining talk

1. Graph-based structures
2. Adaptive estimation for high-dimensional graph-based linear models
   - Graph-Smooth-Lasso
   - Graph-Spline-Lasso
   - Graph-Piecewise-Polynomial-Lasso (our focus)
3. Theory of Graph-Piecewise-Polynomial-Lasso
4. Simulation studies
5. Ongoing and future work

Introduction: graph theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $|\mathcal{V}| = n$ and $|\mathcal{E}| = p$.

- Oriented incidence matrix $F \in \{-1, 0, 1\}^{p \times n}$: If the k-th edge is $(i, j) \in \mathcal{E}$ with $i < j$, then the k-th row of F is $(0, ..., -1, ..., +1, ..., 0)$



$$F = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Introduction: graph theory

- Laplacian matrix: $L = F^T F \in \mathbb{R}^{n \times n}$
- Graph difference operator of $k+1$:

$$\Delta^{(k+1)} = \begin{cases} F^T \Delta^{(k)} = L^{\frac{k+1}{2}} \in \mathbb{R}^{n \times n} & \text{for odd k} \\ F \Delta^{(k)} = F L^{\frac{k}{2}} \in \mathbb{R}^{p \times n} & \text{for even k.} \end{cases}$$

  (Wang et al. '16)
- If $\mathcal{G}$ is a path graph, then graph difference operator is similar to the usual difference operator

Graph-based structures

Smoothness

For $k \geq 0$ and $\alpha > 0$, β^* is (k, α)-smooth over the graph $\mathcal{G}$ if

$$\|\Delta^{(k+1)}\beta^*\|_2^2 \leq \alpha$$

Graph-based structures

Smoothness

For $k \geq 0$ and $\alpha > 0$, β^* is (k, α)-smooth over the graph $\mathcal{G}$ if

$$\|\Delta^{(k+1)}\beta^*\|_2^2 \leq \alpha$$

- $k = 0$ recovers the widely used smoothness over graphs (von Luxburg '07)
- Global smoothness
- Analogy of smoothing splines in nonparametric regression (Wahba '90)

Graph-based structures

Piecewise polynomial

For $k \geq 0$ and $s > 0$, β^* is (k, s)-piecewise polynomial over the graph $\mathcal{G}$ if

$$\|\Delta^{(k+1)}\beta^*\|_0 \leq s$$

Graph-based structures

Piecewise polynomial

For $k \geq 0$ and $s > 0$, β^* is (k, s)-piecewise polynomial over the graph $\mathcal{G}$ if

$$\|\Delta^{(k+1)}\beta^*\|_0 \leq s$$

- $k = 0$: piecewise constant; $k = 1$: piecewise linear; $k = 2$: piecewise quadratic
- Local smoothness
- Analogy of trend filtering in nonparametric regression (Tibshirani '14)
- Extension to weakly piecewise polynomial structure (W. and Loh '20)

Graph-based adaptive estimation

If β^* is sparse and (k, α)-smooth over $\mathcal{G}$, then

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\Delta^{(k+1)} \beta\|_2^2$$

Graph-based adaptive estimation

If β^* is sparse and (k, α)-smooth over $\mathcal{G}$, then

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\Delta^{(k+1)}\beta\|_2^2$$

- Graph-Smooth-Lasso:

$$\hat{\beta}^{\text{gsmooth}} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\Delta^{(1)}\beta\|_2^2$$

- Graph-Spline-Lasso:

$$\hat{\beta}^{\text{gspline}} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N}\|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\Delta^{(2)}\beta\|_2^2$$

Graph-based adaptive estimation

If β^* is sparse and (k, s)-piecewise polynomial over $\mathcal{G}$, then

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \quad \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 + \lambda_g \|\Delta^{(k+1)} \beta\|_1$$

- We refer to $\hat{\beta}$ as the k-th order Graph-Piecewise-Polynomial-Lasso
- $k = 0$: Graph-Fused Lasso
- Assume k is known for theory, but tune k in practice
- Our focus in the remaining talk

Adaptivity of Graph-Piecewise-Polynomial-Lasso

- Assume that the graph $\mathcal{G}$ has a single connected component
- $\widehat{S}_1$: support set of $\Delta^{(k+1)}\hat{\beta}$
- $\Delta^{(k+1)}_{-\widehat{S}_1}$: submatrix of $\Delta^{(k+1)}$ after removing the rows indexed by $\widehat{S}_1$
- For even k, let $\mathcal{G}_{-\widehat{S}_1}$ be the subgraph induced by removing the edges indexed by $\widehat{S}_1$
- $C_1, ..., C_j$: connected components of the subgraph $\mathcal{G}_{-\widehat{S}_1}$
- $\mathbb{1}_n = (1, ..., 1)^T \in \mathbb{R}^n$
- $\mathbb{1}_{C_i} \in \mathbb{R}^n$: indicator vector over connected component C_i

Adaptivity of Graph-Piecewise-Polynomial-Lasso

Theorem (W. and Loh '20)

For even k, the null space of $\Delta^{(k+1)}_{-\widehat{S}_1}$ is

$$\mathrm{span}(\mathbb{1}_n) + \mathrm{span}(\mathbb{1}_n)^{\perp} \cap \left(L^{\frac{k}{2}} + \mathbb{1}_n \mathbb{1}_n^T\right)^{-1} \mathrm{span}(\mathbb{1}_{C_1}, \ldots, \mathbb{1}_{C_j}).$$

For odd k, the null space of $\Delta^{(k+1)}_{-\widehat{S}_1}$ is

$$\mathrm{span}(\mathbb{1}_n) + \mathrm{span}(\mathbb{1}_n)^{\perp} \cap \left\{u \in \mathbb{R}^n : u = (L^{\frac{k+1}{2}} + \mathbb{1}_n \mathbb{1}_n^T)^{-1} v, \quad v_{-\widehat{S}_1} = 0\right\}.$$

Adaptivity of Graph-Piecewise-Polynomial-Lasso

Theorem (W. and Loh '20)

For even k, the null space of $\Delta^{(k+1)}_{-\widehat{S}_1}$ is

$$\text{span}(\mathbb{1}_n) + \text{span}(\mathbb{1}_n)^{\perp} \cap \left(L^{\frac{k}{2}} + \mathbb{1}_n \mathbb{1}_n^T\right)^{-1} \text{span}(\mathbb{1}_{C_1}, \ldots, \mathbb{1}_{C_j}).$$

For odd k, the null space of $\Delta^{(k+1)}_{-\widehat{S}_1}$ is

$$\text{span}(\mathbb{1}_n) + \text{span}(\mathbb{1}_n)^{\perp} \cap \left\{u \in \mathbb{R}^n : u = (L^{\frac{k+1}{2}} + \mathbb{1}_n \mathbb{1}_n^T)^{-1} v, \quad v_{-\widehat{S}_1} = 0\right\}.$$

- $k = 0$: $\hat{\beta}$ is piecewise constant over connected components $C_1, \ldots, C_j$
- For general even k, structure of $\hat{\beta}$ is smoothed by multiplying $\text{span}(\mathbb{1}_{C_1}, \ldots, \mathbb{1}_{C_j})$ by $(L^{k/2} + \mathbb{1}_n \mathbb{1}_n^T)^{-1}$
- For odd k, structure of $\hat{\beta}$ is based on the support set $\widehat{S}_1$ and the smoother $(L^{(k+1)/2} + \mathbb{1}_n \mathbb{1}_n^T)^{-1}$

Convergence analysis

- X: sub-Gaussian (good) random design matrix
- ε: sub-Gaussian noise
- $\varepsilon \perp\!\!\!\perp X$
- Assume β^* is (k, s_1)-piecewise polynomial and s_2-sparse over a graph $\mathcal{G}$ with the maximum degree d
- $\lambda \asymp \sqrt{\frac{\log n}{N}}$
- $\lambda_g = \lambda\sqrt{\frac{\nu}{(2d)^{k+1}}}$, where $0 \leq \nu < 1$ is a constant

Convergence analysis

Theorem (W. and Loh '20)

Assume $s_2/s_1 \geq \nu$. Then

$$\|\hat{\beta} - \beta^*\|_2 \leq \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{s_2 \log n}{N}}\right), \quad \|\hat{\beta} - \beta^*\|_1 \leq \mathcal{O}_{\mathbb{P}}\left(s_2\sqrt{\frac{\log n}{N}}\right),$$

and

$$\frac{1}{N}\|X(\hat{\beta} - \beta^*)\|_2^2 \leq \mathcal{O}_{\mathbb{P}}\left(\frac{s_2 \log n}{N}\right).$$

Convergence analysis

Theorem (W. and Loh '20)

Assume $s_2/s_1 \geq \nu$. Then

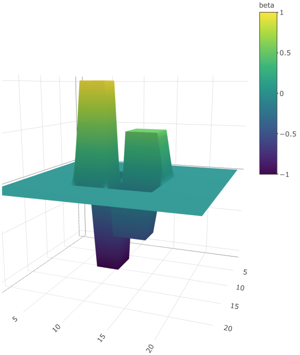
$$\|\hat{\beta} - \beta^*\|_2 \leq \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{s_2 \log n}{N}}\right), \quad \|\hat{\beta} - \beta^*\|_1 \leq \mathcal{O}_{\mathbb{P}}\left(s_2\sqrt{\frac{\log n}{N}}\right),$$

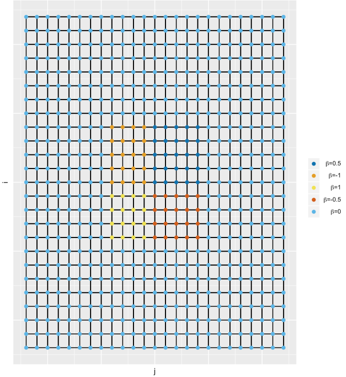
and

$$\frac{1}{N}\|X(\hat{\beta} - \beta^*)\|_2^2 \leq \mathcal{O}_{\mathbb{P}}\left(\frac{s_2 \log n}{N}\right).$$

- Proof sketch: start with the optimization problem, build the basic inequality, then show restricted eigenvalue condition and requirements for tuning parameters
- Same convergence rates with Lasso
- But our approach is adaptive while Lasso is not

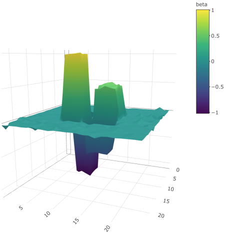
Simulation for structure recovery



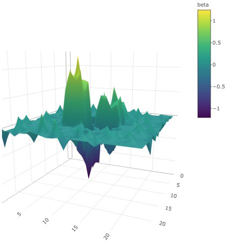


Coordinates of β^* correspond to a 2d grid graph with 25 rows and 25 columns. The right figure shows the value of β^* in each node

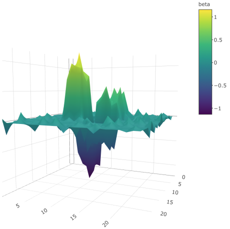
Simulation for structure recovery (N=250)



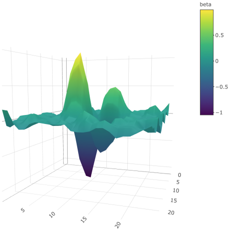
(a) Our approach



(b) Lasso



(c) Graph-Smooth-Lasso



(d) Graph-Spline-Lasso

Simulation for estimation error

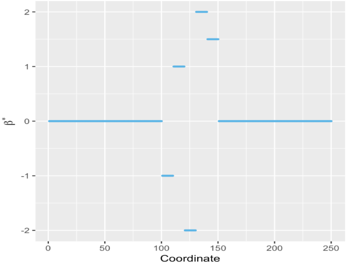
Averages (standard errors) of ℓ_2 estimation error $\|\hat{\beta} - \beta^*\|_2$

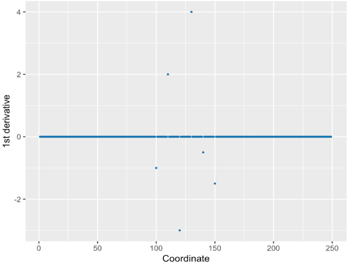
	$(N, n) = (250, 625)$	$(N, n) = (375, 625)$	$(N, n) = (500, 625)$
Our approach	**0.433** (0.007)	**0.345** (0.003)	**0.364** (0.002)
Lasso	3.145 (0.077)	0.538 (0.012)	0.381 (0.003)
Graph-Smooth-Lasso	2.288 (0.063)	0.618 (0.016)	0.384 (0.004)
Graph-Spline-Lasso	3.439 (0.017)	3.191 (0.018)	2.990 (0.011)

- Our approach achieved smaller estimation error across all sampling schemes
- Similar performance in other simulation scenarios (W. and Loh '20)

Ongoing and future work

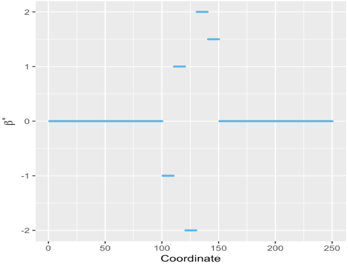
- Variable selection and changepoint detection consistency:

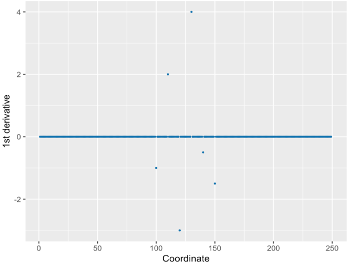




Ongoing and future work

- Variable selection and changepoint detection consistency:





Open question

Let S_1 and S_2 be support sets of $\Delta^{(k+1)}\beta^*$ and β^*. We also denote support sets of $\Delta^{(k+1)}\hat{\beta}$ and $\hat{\beta}$ by $\widehat{S}_1$ and $\widehat{S}_2$. Then when are the support sets $\widehat{S}_1$ and $\widehat{S}_2$ exactly equal to the true support sets S_1 and S_2?

Ongoing and future work

- Statistical inference: unable to directly use Graph-Piecewise-Polynomial-Lasso for statistical inference

Ongoing and future work

- Statistical inference: unable to directly use Graph-Piecewise-Polynomial-Lasso for statistical inference
- Highly correlated design

Ongoing and future work

- Statistical inference: unable to directly use Graph-Piecewise-Polynomial-Lasso for statistical inference
- Highly correlated design
- Numerical algorithm for fast computation: R package

Ongoing and future work

- Statistical inference: unable to directly use Graph-Piecewise-Polynomial-Lasso for statistical inference
- Highly correlated design
- Numerical algorithm for fast computation: R package
- Theory for Graph-Smooth-Lasso and Graph-Spline-Lasso

Ongoing and future work

- Statistical inference: unable to directly use Graph-Piecewise-Polynomial-Lasso for statistical inference
- Highly correlated design
- Numerical algorithm for fast computation: R package
- Theory for Graph-Smooth-Lasso and Graph-Spline-Lasso
- Real-world applications

Summary

Takeaway points:

- Significant room for developing new efficient adaptive estimation methods for the graph setting
- Defined Graph-based smoothness and piecewise polynomial structure
- Proposed several adaptive estimation methods for different graph-based structures

Summary

Takeaway points:

- Significant room for developing new efficient adaptive estimation methods for the graph setting
- Defined Graph-based smoothness and piecewise polynomial structure
- Proposed several adaptive estimation methods for different graph-based structures

Thank you!

Simulation setup

- Generated each row of the design matrix X from $N(0, I_{n\times n})$
- Generated each ε_i from $N(0, 0.1)$
- Generated y via the linear model
- Tuning parameters: 5-fold cross-validation procedure, which minimized the cross-validated prediction error
- Repeated the simulation 50 times