XLV Konferencja Statystyka Matematyczna

Adaptive Bayesian SLOPE — High-dimensional Model Selection with Missing Values

Wei Jiang
CMAP, Ecole Polytechnique

Malgorzata Bogdan, Julie Josse, Blazej Miasojedow, Veronika Rockova

3rd Dec.

Motivation: Paris Hospital

- *Traumabase*® data:
  20000 major trauma patients × 250 measurements.

Accident type	**Age**	**Sex**	**Blood pressure**	**Lactate**	**Temperature**	**Platelet** (G/L)
Falling	50	M	140	NA	35.6	150
Fire	28	F	NA	4.8	36.7	250
Knife	30	M	120	1.2	NA	270
Traffic accident	23	M	110	3.6	35.8	170
Knife	33	M	106	NA	36.3	230
Traffic accident	58	F	150	NA	38.2	400

Motivation: Paris Hospital

- *Traumabase*® data:
  20000 major trauma patients × 250 measurements.

Accident type	Age	Sex	Blood pressure	Lactate	Temperature	Platelet (G/L)
Falling	50	M	140	NA	35.6	150
Fire	28	F	NA	4.8	36.7	250
Knife	30	M	120	1.2	NA	270
Traffic accident	23	M	110	3.6	35.8	170
Knife	33	M	106	NA	36.3	230
Traffic accident	58	F	150	NA	38.2	400

- **Objective:**
  Develop models to help emergency doctors make decisions.
  Measurements $\xrightarrow{\text{Predict}}$ Platelet $\Rightarrow$ $X \xrightarrow{\text{Regression}} y$

Motivation: Paris Hospital

- *Traumabase*® data:
  20000 major trauma patients × 250 measurements.

Accident type	Age	Sex	Blood pressure	Lactate	Temperature	Platelet (G/L)
Falling	50	M	140	NA	35.6	150
Fire	28	F	NA	4.8	36.7	250
Knife	30	M	120	1.2	NA	270
Traffic accident	23	M	110	3.6	35.8	170
Knife	33	M	106	NA	36.3	230
Traffic accident	58	F	150	NA	38.2	400

- **Objective:**
  Develop models to help emergency doctors make decisions.
  Measurements $\xrightarrow{\text{Predict}}$ Platelet $\Rightarrow$ $X \xrightarrow{\text{Regression}} y$
- **Challenge :**
  How to **select** relevant measurements with **missing values**?

Model selection in high-dimension

Linear regression model: $y = X\beta + \varepsilon,$

- $y = (y_i)$: vector of response of length n
- $X = (X_{ij})$: a standardized design matrix of dimension $n \times p$
- $\beta = (\beta_j)$: regression coefficient of length p
- $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$

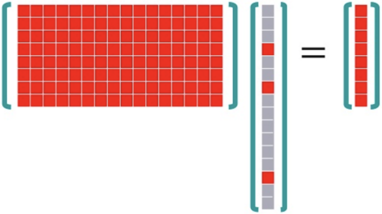
Model selection in high-dimension

Linear regression model: $y = X\beta + \varepsilon,$

- $y = (y_i)$: vector of response of length n
- $X = (X_{ij})$: a standardized design matrix of dimension $n \times p$
- $\beta = (\beta_j)$: regression coefficient of length p
- $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_n)$

Assumptions:

- high-dimension: p large (including $p \geq n$)
- β is **sparse** with $k < n$ nonzero coefficients



l_1 penalization methods

- LASSO (Tibshirani, 1996)

$$\hat{\beta}_{LASSO} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \|y - X\beta\|^2 + \lambda \|\beta\|_1,$$

detects important variables with high probability but includes many **false positives**.

l_1 penalization methods

- LASSO (Tibshirani, 1996)
  $$\hat{\beta}_{LASSO} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1,$$
  detects important variables with high probability but includes many **false positives**.

- SLOPE (Bogdan et al., 2015) penalizes larger coefficients more stringently
  $$\hat{\beta}_{SLOPE} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2}\|y - X\beta\|^2 + \sigma \sum_{j=1}^{p} \lambda_j |\beta|_{(j)},$$
  where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ and $|\beta|_{(1)} \geq |\beta|_{(2)} \geq \cdots \geq |\beta|_{(p)}$.

l_1 penalization methods

- LASSO (Tibshirani, 1996)
$$\hat{\beta}_{LASSO} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1,$$
detects important variables with high probability but includes many **false positives**.

- SLOPE (Bogdan et al., 2015) penalizes larger coefficients more stringently
$$\hat{\beta}_{SLOPE} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2}\|y - X\beta\|^2 + \sigma \sum_{j=1}^{p} \lambda_j |\beta|_{(j)},$$
where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ and $|\beta|_{(1)} \geq |\beta|_{(2)} \geq \cdots \geq |\beta|_{(p)}$.

  To control **False Discovery Rate (FDR)** at level q:
  $\lambda_{BH}(j) = \phi^{-1}(1 - q_j), \quad q_j = \frac{jq}{2p}, \quad X^T X = I,$ then
$$FDR = \mathbb{E}\left[\frac{\#\text{False rejections}}{\#\text{Rejections}}\right] \leq q$$

Bayesian SLOPE

Problem: λ for SLOPE leading to FDR control are typically large.
SLOPE often returns **an inconsistent estimation.**

$\Rightarrow$ improve?

Bayesian SLOPE

Problem: λ for SLOPE leading to FDR control are typically large. SLOPE often returns **an inconsistent estimation.**

$\Rightarrow$ improve?

SLOPE estimate = MAP of a Bayesian regression with SLOPE prior.

$$\hat{\beta}_{SLOPE} = \arg\max_{\beta} \mathtt{p}(y \mid X, \beta, \sigma^2; \lambda) \propto \mathtt{p}(y \mid X, \beta)\mathtt{p}(\beta \mid \sigma^2; \lambda)$$

where the SLOPE prior:

$$\mathtt{p}(\beta \mid \sigma^2; \lambda) \propto \prod_{j=1}^{p} \exp\left(-\frac{1}{\sigma}\lambda_j |\beta|_{(j)}\right)$$

Adaptive Bayesian SLOPE

We propose an adaptive version of Bayesian SLOPE (ABSLOPE), with the prior for β as

$$\mathtt{p}(\beta \mid \gamma, c, \sigma^2; \lambda) \propto c^{\sum_{j=1}^{p} \mathbb{I}(\gamma_j = 1)} \prod_j \exp\left\{ -w_j |\beta_j| \frac{1}{\sigma} \lambda_{r(W\beta, j)} \right\},$$

Interpretation of the model:

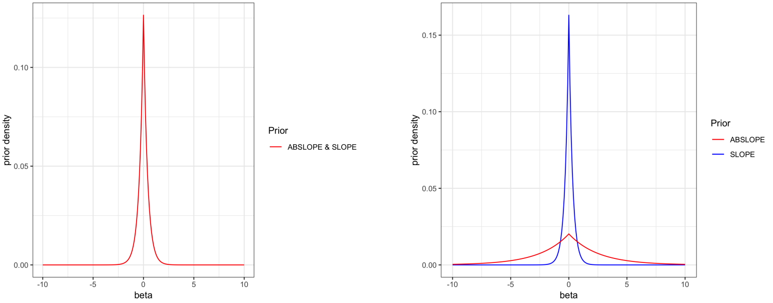
- β_j is large enough $\Rightarrow$ **true signal**; 0 $\Rightarrow$ noise.
- $\gamma_j \in \{0, 1\}$ signal indicator. $\gamma_j | \theta \sim \mathit{Bernoulli}(\theta)$ and θ the **sparsity**.
- $c \in [0, 1]$: the inverse of **average signal magnitude**.
- $W = \operatorname{diag}(w_1, w_2, \cdots, w_p)$ and its diagonal element:

$$w_j = c\gamma_j + (1 - \gamma_j) = \begin{cases} c, & \gamma_j = 1 \\ 1, & \gamma_j = 0 \end{cases}.$$

Adaptive Bayesian SLOPE

Advantage of introducing W:

- when $\gamma_j = 0$, $w_j = 1$, i.e., the null variables are treated with the regular SLOPE penalty
- when $\gamma_j = 1$, $w_j = c < 1$, i.e, **smaller penalty** $\lambda_{r(W\beta,j)}$ for true predictors than the regular SLOPE one



(a) Null β

(b) Non-null β

Figure: comparison of SLOPE prior and ABSLOPE prior

Model selection with missing values

Decomposition: $X = (X_{\text{obs}}, X_{\text{mis}})$

Pattern: matrix M with $M_{ij} = \begin{cases} 1, & \text{if } X_{ij} \text{ is observed} \\ 0, & \text{otherwise} \end{cases}$

Assumption 1: Missing at random (MAR)

$\mathrm{p}(M \mid X_{\text{obs}}, X_{\text{mis}}) = \mathrm{p}(M \mid X_{\text{obs}}) \quad \Rightarrow \quad$ ignorable missing patterns
e.g. People at older age didn't tell his income at larger probability.

Assumption 2: Distribution of covariates

$X_i \sim_{\text{i.i.d.}} \mathcal{N}_p(\mu, \Sigma), \quad i = 1, \cdots, n.$

Model selection with missing values

Decomposition: $X = (X_{\text{obs}}, X_{\text{mis}})$

Pattern: matrix M with $M_{ij} = \begin{cases} 1, & \text{if } X_{ij} \text{ is observed} \\ 0, & \text{otherwise} \end{cases}$

Assumption 1: Missing at random (MAR)

$\mathrm{p}(M \mid X_{\text{obs}}, X_{\text{mis}}) = \mathrm{p}(M \mid X_{\text{obs}}) \quad \Rightarrow \quad$ ignorable missing patterns
e.g. People at older age didn't tell his income at larger probability.

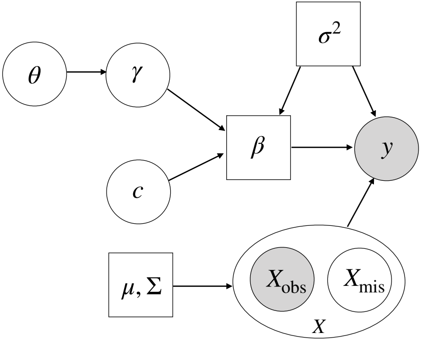
Assumption 2: Distribution of covariates

$X_i \sim_{\text{i.i.d.}} \mathcal{N}_p(\mu, \Sigma), \quad i = 1, \cdots, n.$

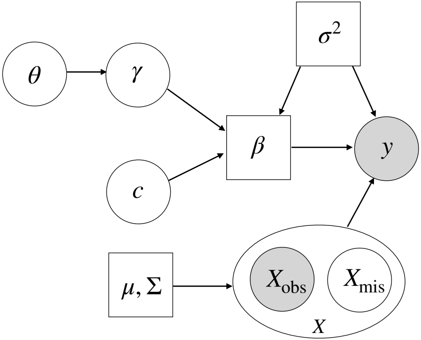
Problem: With NA, only a few methods are available to select a model, and their performances are limited. For example,

- (Claeskens and Consentino, 2008) adapts AIC to missing values ⇒ Impossible to deal with high dimensional analysis.
- (Loh and Wainwright, 2012) LASSO with NA
  ⇒ Non-convex optimization; requires to know bound of $\|\beta\|_1$
  ⇒ difficult in practice

ABSLOPE with missingness: Summary



ABSLOPE with missingness: Summary



$$\begin{aligned}\ell_{\text{comp}} &= \log \mathtt{p}(y, X, \gamma, c;\ \beta, \theta, \sigma^2) + pen(\beta)\\ &= \log \left\{ \mathtt{p}(X;\ \mu, \Sigma)\, \mathtt{p}(y \mid X;\ \beta, \sigma^2)\, \mathtt{p}(\gamma;\ \theta)\, \mathtt{p}(c) \right\} + pen(\beta)\end{aligned}$$

Objective: Maximize $\ell_{\text{obs}} = \iiint \ell_{\text{comp}}\, dX_{\text{mis}}\, dc\, d\theta\, d\gamma$.

EM algorithm

- *E step:* evaluate

$$Q^t = \mathbb{E}(\ell_{\text{comp}}) \quad \mathtt{wrt} \quad \mathtt{p}(X_{\text{mis}}, \gamma, c, \theta \mid y, X_{\text{obs}}, \beta^t, \sigma^t, \mu^t, \Sigma^t).$$

- *M step:* update

$$\beta^t, \sigma^t, \mu^t, \Sigma^t = \arg\max Q^t$$

Problem: The function Q is not tractable. ⇒

1. Monte Carlo EM ? (Wei and Tanner 1990)

EM algorithm

- *E step:* evaluate

$$Q^t = \mathbb{E}(\ell_{\text{comp}}) \quad \text{wrt} \quad \mathrm{p}(X_{\text{mis}}, \gamma, c, \theta \mid y, X_{\text{obs}}, \beta^t, \sigma^t, \mu^t, \Sigma^t).$$

- *M step:* update

$$\beta^t, \sigma^t, \mu^t, \Sigma^t = \arg\max Q^t$$

Problem: The function Q is not tractable. ⇒

1. ~~Monte Carlo EM ?~~
   Expensive to generate a large number of samples.
2. Stochastic Approximation EM (book, Lavielle 2014)
   - One sample in each iteration;

Adapted SAEM algorithm

- *E step:*
  $Q^t = \mathbb{E}(\ell_{\text{comp}}) \quad \text{wrt} \quad \mathtt{p}(X_{\text{mis}}, \gamma, c, \theta \mid y, X_{\text{obs}}, \beta^t, \sigma^t, \mu^t, \Sigma^t).$
  - *Simulation:* draw one sample $(X^t_{\text{mis}}, \gamma^t, c^t, \theta^t)$ from
    $$\mathtt{p}(X_{\text{mis}}, \gamma, c, \theta \mid y, X_{\text{obs}}, \beta^{t-1}, \sigma^{t-1}, \mu^{t-1}, \Sigma^{t-1});$$
    [**Gibbs sampling**]
  - *Stochastic approximation:* update function Q with
    $$Q^t = Q^{t-1} + \eta_t \left(\ell_{\text{comp}} \Big|_{X^t_{\text{mis}}, \gamma^t, c^t, \theta^t} - Q^{t-1} \right).$$
- *M step:* $\beta^{t+1}, \sigma^{t+1}, \mu^{t+1}, \Sigma^{t+1} = \arg\max Q^{t+1}$.
  [**Proximal gradient descent, Shrinkage of covariance**]

Details of initialization, generating samples and optimization are in the draft (arXiv:1909.06631)

R package: ABSLOPE

Install package:

```
library(devtools)
install_github("wjiang94/ABSLOPE")
```

R package: ABSLOPE

Install package:

```
library(devtools)
install_github("wjiang94/ABSLOPE")
```

Main algorithm:

```
lambda = create_lambda_bhq(ncol(X),fdr=0.10)
list.res = ABSLOPE(X, y, lambda)
```

R package: ABSLOPE

Install package:

```
library(devtools)
install_github("wjiang94/ABSLOPE")
```

Main algorithm:

```
lambda = create_lambda_bhq(ncol(X),fdr=0.10)
list.res = ABSLOPE(X, y, lambda)
```

A fast and simplified algorithm (Rcpp):

```
list.res.slobe = SLOBE(X, y, lambda)
```

R package: ABSLOPE

Install package:

```
library(devtools)
install_github("wjiang94/ABSLOPE")
```

Main algorithm:

```
lambda = create_lambda_bhq(ncol(X),fdr=0.10)
list.res = ABSLOPE(X, y, lambda)
```

A fast and simplified algorithm (Rcpp):

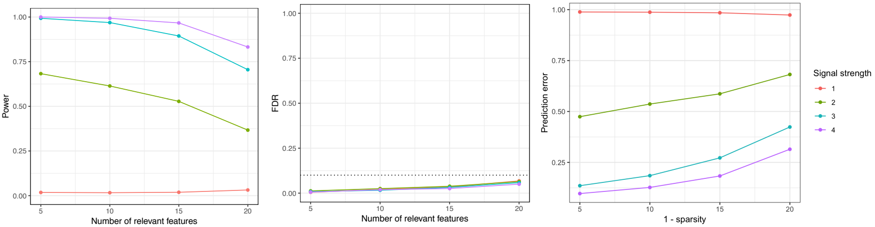
```
list.res.slobe = SLOBE(X, y, lambda)
```

Values:

```
list.res$beta
list.res$gamma
```

Simulation study (200 rep. ⇒ average)

$n = p = 100$, no correlation and 10% missingness



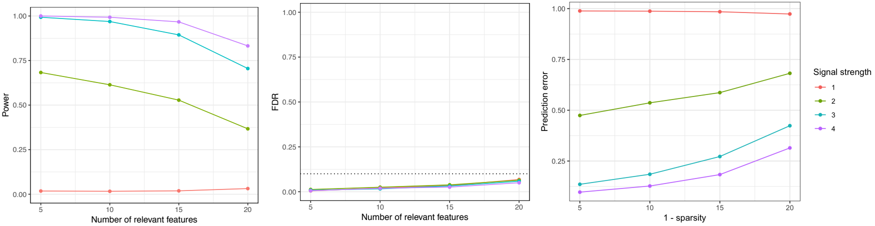
(a) Power

(b) FDR

(c) Prediction error

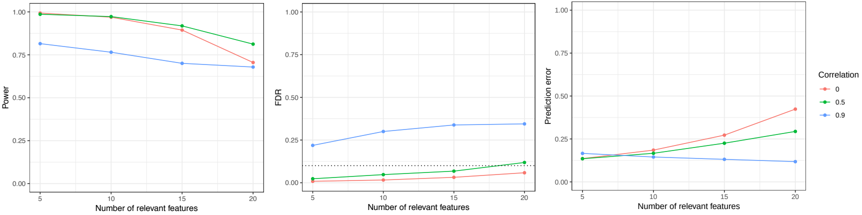
Simulation study (200 rep. $\Rightarrow$ average)

$n = p = 100$, no correlation and 10% missingness



(g) Power (h) FDR (i) Prediction error

$n = p = 100$, with 10% missingness and strong signal



(j) Power (k) FDR (l) Prediction error

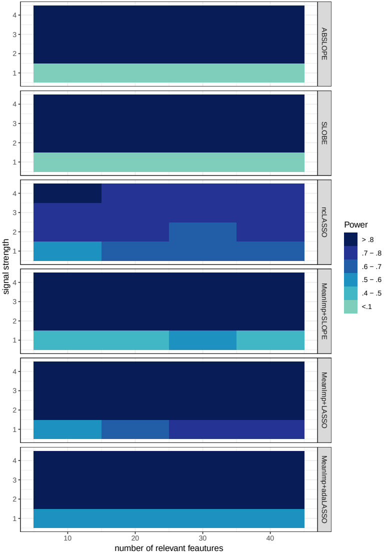
Method comparison

- **ABSLOPE** and **SLOBE**
- **ncLASSO:** non convex LASSO (Loh and Wainwright, 2012)
- **MeanImp + SLOPE:** Mean imputation followed by SLOPE with known σ
- **MeanImp + LASSO:** Mean imputation followed by LASSO, with λ tuned by cross validation
- **MeanImp + adaLASSO:** Mean imputation followed by adaptive LASSO (Zou, 2006)

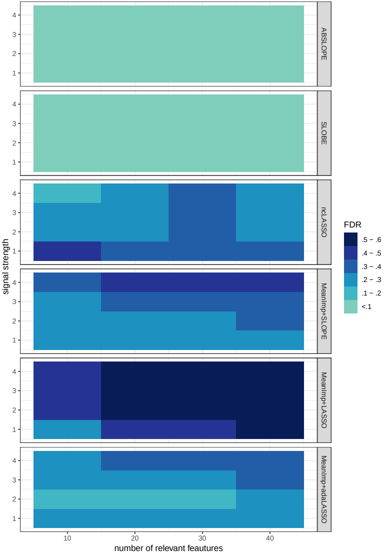
In the SLOPE type methods, λ = BH sequence which controls the FDR at level **0.1**

Method comparison (200 rep. ⇒ average)

500×500 dataset, 10% missingness, with correlation



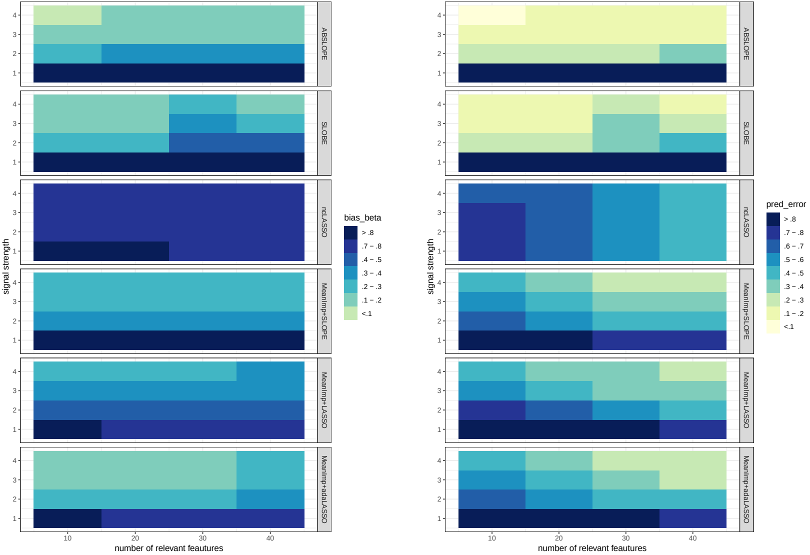
(m) Power



(n) FDR

Method comparison (200 rep. ⇒ average)

500×500 dataset, 10% missingness, with correlation



(a) Bias of β

(b) Prediction error

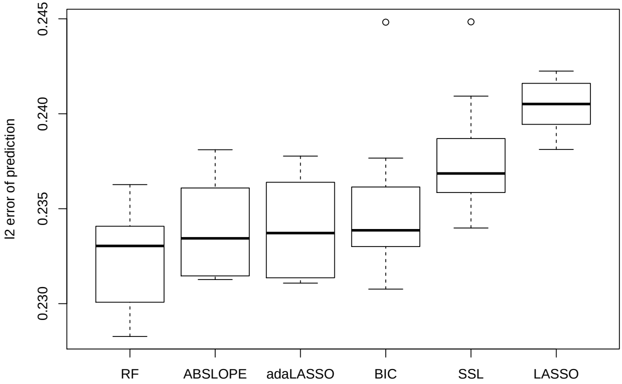
Computational cost

Execution time (seconds) for one simulation	$n = p = 100$			$n = p = 500$		
	min	mean	max	min	mean	max
ABSLOPE	12.83	14.33	20.98	646.53	696.09	975.73
SLOBE	0.31	0.34	0.66	14.23	15.07	29.52
ncLASSO	16.38	20.89	51.35	91.90	100.71	171.00
MeanImp + SLOPE	0.01	0.02	0.09	0.24	0.28	0.53
MeanImp + LASSO	0.10	0.14	0.32	1.75	1.85	3.06

[**Fast implementation:** Parallel computing + Rcpp (C++)]

More on the real data...

TraumaBase: Measurements $\xrightarrow{\text{Predict}}$ Platelet
Cross-validation: random splits to training and test sets × 10



- Comparable to random forest
- Interpretable model selection and estimation results

Conclusion & Future research

Conclusion:

- ABSLOPE penalizes larger coefficients more stringently to **control FDR**, meanwhile it applies a weighting matrix to **improve the estimation**;
- Modeling in a Bayesian framework gives detailed structure of predictors as **sparsity** and **signal strength**;
- Simulation study shows that ABSLOPE is competitive to other methods in terms of power, FDR and prediction error.

Future research:

- Consider categorical or mixed data
- Deal with other missing mechanisms
- Application on genetic dataset

Thank you!

Dziękuję

Merci