Statistical learning

Report 3

Maciej Szczutko

2024-06-01

Properties of symmetric matrices

Now we proof that the trace of symmetric matrix X $tr(x) = \Sigma \lambda_i$. We use circular property of trace $\operatorname{tr}\left(\mathbf{A}^T \mathbf{B}\right) = \operatorname{tr}\left(\mathbf{A}\mathbf{B}^T\right)$. As the symmetric matrices X always diagonalizable in R by spectral theorem we can use spectral decomposition.

$$\operatorname{tr}(\boldsymbol{X}) = \operatorname{tr}\left(\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T\right) = \operatorname{tr}\left(\boldsymbol{\Lambda}\mathbf{P}^T\mathbf{P}\right) = \operatorname{tr}(\boldsymbol{\Lambda}) = \sum_{i=1}^{n} \lambda_i .$$

Properties of $X^T X$

Now let X be the matrix of dimension nxp $(p \leq n)$. Then $X^T X$ is always semi-positive definite. Let $z \in \mathbb{R}^n$

$$z^T \left(X^T X\right) z = \left(z^T X^T\right) X z = (Xz)^T (Xz) = \|Xz\|_2^2 \geq 0.$$

Now consider case where $p > n$. Then at least one eigen value must be equal to 0. Note the fact the $rank(X) \leq n$. The $rank(X^T X) \leq \min\{rank(X), rank(X^T)\} \leq n$. We conclude the $X^T X$ is not full rank so it must be singular. Also $det(X^T X) = 0$ implies 0 is eigen value for this matrices.

Model selection criteria

Let's remark the AIC, BIC, RIC. In general we looking for model maximazing the value of information criterium. In linear model family is equivalent to minimizing below terms:

1. AIC – $RSS + 2\sigma^2 k$
2. BIC – $RSS + \sigma^2 k \log n$
3. RIC – $RSS + \sigma^2 2k \log p$

k means number of parameter used in models.

After computing the value we can claim that the best model based on AIC is model 6. For BIC model with 10 and RIC select model with 3 variables.

Assuming the orthogonal design $(X'X = I)$ and $n = p = 10000$ calculate the expected number of false discoveries for AIC, BIC and RIC, when none of the variables is really important (i.e. $p_0 = p$)

AIC

In such setup the probability of I type error is $P\left(X_i \text{ is selected } \mid \beta_i = 0\right) = 2(1 - \Phi(\sqrt{2})) = 0.16$ Then expected number of false discovery is $10000 * 0.16 = 1600$

BIC

$P\left(X_i \text{ is selected } \mid \beta_i = 0\right) = 2(1 - \Phi(\sqrt{\log 1000})$

Then expected number of false discovery is $10000 * 0.0024 = 24.065$.

So the BIC is more restrictive and should make less false discoveries on average.

RIC

$P\left(X_i \text{ is selected } \mid \beta_i = 0\right) = 2(1 - \Phi(\sqrt{2 \log p}))$

Then expected number of false discovery is $10000 * 0.000018 = 0.177$.

As was presented on lecture RIC is developed to avoid false discoveries even in large setup and keep expected value of false discovery below 1.

When would you use AIC ? BIC ? RIC ?

The AIC is equivalent for minimizing prediction error and can be use as initial step. The RIC should be use when we handle data with a large number of variables and the cost of making a mistake is large (e.g. medical industry).

Ridge regression in orthogonal design

Ridge regression is a type of linear regression that includes a regularization term to prevent overfitting. It is achieved by modification LOSS function. Instead of minimizing L2 norm, we minize L2 + regularization term. Formally we looking for β minimanize below terms:

$$L_{\text{ridge}}\left(\hat{\beta}\right) = \|y - X\hat{\beta}\|^2 + \gamma\|\hat{\beta}\|^2.$$

for some constant value γ. γ is used to manipulate bias and variance of $\hat{\beta}$.

Assuming the model is in the form

$$Y = X\beta + \varepsilon$$

where X is orthonormal and the errors has 0 *mean* and $\sigma^2 = 1$. Estimator has following distribution;

$$\hat{\beta} \sim N\left(\frac{1}{1+\gamma}\beta, \frac{1}{(1+\gamma)^2}I\right).$$

so the theoretical bias $\hat{\beta}_i$ is $\frac{-\gamma}{1+\gamma}\beta_i$, variance $\frac{1}{(1+\gamma)^2}$. The MSE is

$$E\left(\hat{\beta}_i - \beta_i\right)^2 = \frac{\gamma^2}{(1+\gamma)^2}\beta_i^2 + \frac{1}{(1+\gamma)^2}$$

We want to use γ which minimize the mean square error.

$$E\|\hat{\beta}-\beta\|^2=\frac{\gamma^2}{(1+\gamma)^2}\|\beta\|^2+\frac{p\sigma^2}{(1+\gamma)^2}.$$

From this we we find $\gamma_{opt}=\frac{p}{\|\beta\|^2}$. Using γ_{opt} in MSE formula we got $MSE\left(\gamma_{opt}\right)=\frac{p\|\beta\|^2}{p+\|\beta\|^2}$.

In similar setup, if we use OLS estimator the estimator is unbiased. But the $MSE_{OLS}=p\sigma^2=p$.

Comparing prediction error (Ridge vs OLS)

Assume we have dataset with 40 variables. We build model using OLS and Ridge. RSS are equal 4.5 and 11.6 respectively. For the ridge regression the trace of $X\left(X'X+\gamma I\right)^{-1}X'$ is equal to 32.

$PE_{\mathrm{RR}}\ =11.6+2\sigma^2\cdot\operatorname{tr}(M)=11.6+64\sigma^2$

$PE_{OLS}=4.5+2\sigma^2p=4.5+80\sigma^2$

In this example PE from ridge should always be lower than OLS PE if $\sigma^2>0.44$.

LASSO

Least Absolute Shrinkage and Selection Operator is another modification of OLS. We add penalty in terms of L1 norm of β

$L_{\text{lasso}}\,(\hat{\beta})=\sum_{i=1}^n\left(y_i-x_i'\hat{\beta}\right)^2+\lambda\sum_{j=1}^p\left|\hat{\beta}_j\right|.$

The LASSO select important variables by setting some coefficients of β to 0. The solution of the problem might be express in terms of OLS estimator. Let us define shrinkage operator:

$$\eta_\lambda(x)=\operatorname{sign}x(|x|-\lambda)_+$$

The coefficients of LASSO estimator are given by $\hat{\beta}_{LASSO}=\eta_\lambda\left(\hat{\beta}_{OLS}\right)$.

Recall that OLS estimator is unbiased. So if true $\beta_i=0$ then $\hat{\beta}_i^{OLS}\sim N\left(0,\sigma^2\right)$. Now we will calculate the probability of single false discovery in β vector.

$P\text{ (I type error)}=P\left(\left|\hat{\beta}_i^{OLS}\right|>\lambda|\beta_i=0\right)=P\left(\frac{\left|\hat{\beta}_i^{OLS}\right|}{\sigma}>\frac{\lambda}{\sigma}|\beta_i=0\right)=2\left(1-\Phi\left(\frac{\lambda}{\sigma}\right)\right).$

Assuming the number of non zero coefficients within β is p_0 the expected number of false discovery is $2p_0\left(1-\Phi\left(\frac{\lambda}{\sigma}\right)\right)$ and depend on choice of the λ.

Adaptive LASSO

Adaptive LASSO is extension where instead of use L1 norm we us L1-weighted norm.

$L_{\text{lasso}}\,(\hat{\beta})=\sum_{i=1}^n\left(y_i-x_i'\hat{\beta}\right)^2+\lambda\sum_{j=1}^p w_i\left|\hat{\beta}_j\right|.$

The weights can be set to $w_i=\frac{1}{|\hat{\beta}_i^{RIDGE}|}$. This can be interpret as follow: the ridge shrink the coefficients. If the estimated coefficients from Ridge is close to 0 then the penalty become large and likely aLASSO zero this coefficients. Roughly speaking RIDGE detect initial importance of features.

This give direct instruction how to calculate aLASSO with **glmnet**.

```
#Calculate RIDGE estimator
lambda_MSE_min <- cv.glmnet(X,y, alpha = 0)$lambda.min
ridge <- glmnet(X, y, alpha = 0, lambda = lambda_MSE_min, intercept = F)
beta_ridge <- coef(ridge)[-c(1)] # (-1 to exclude Intercept)
adaptive_lasso <- glmnet(X,y, alpha = 1, penalty.factor =  1/abs(beta_ridge))
```

Using previous notation we can write explicitly the coeficients $\hat{\beta}_{LASSO} = \frac{1}{w}\eta_\lambda\left(w\hat{\beta}_{OLS}\right)$ where $w = (w_1, w_2, \ldots, w_p)$.

Note: w_i are positive.

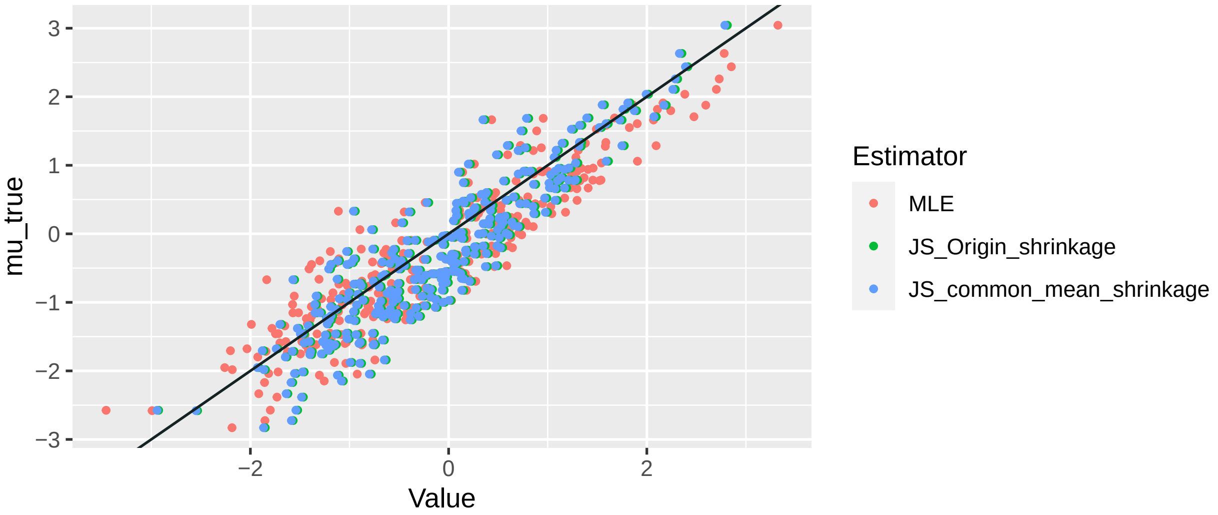
The ordinary least squares estimator of β_1 underr the orthognal design ($\mathrm{X}'\mathrm{X} = \mathrm{I}$) is equal to 3 and the LASSO estimator of this parameter is equal to 2. What is the value of the adaptive LASSO estimator of β_1 if we use the same value of λ and the weight for X_1 is $w_1 = 1/4$.

Then we find $\lambda = 1$ and using above formula we calculate $4\eta_{\lambda=1}\left(\frac{1}{4} * 3\right) = 0$.

James-Stein estimator and Prediction Error in Multiple Regression

We analyze data with observation of 300 genes for 210 patients. We standarize data and later add previous mean vector. Here we will use two estimator JS estimator shrinking toward zero and shrinking to common mean. First estimator is $\hat{\mu}_c = c_{JS}\hat{\mu}_{MLE}$ where $c_{JS} = 1 - \frac{(p-2)\sigma^2}{\|\hat{\mu}_{MLE}\|^2}$.

The second one is $\hat{\mu}_d = (1-d)\hat{\mu}_{MLE} + d\mu_{\bar{M}LE}$ where $d_{JS} = \frac{p-3}{p-1}\frac{\sigma^2}{\mathrm{Var}(\hat{\mu}_{MLE})}$.



I can't infer anything from such a chart. I don't know what it would help me with in practice.

$\hat{\mu}$	MSE
MLE	79.685
JS_Origin_Shrinkage	74.089
JS_Common_Mean_Shrinkage	71.633

From above table we see the estimator works as were designed. We achieved lower MSE for both JS estimator.

Note σ^2 used for JS constant computation is 0.2 because sample comes from distribution $N(0, \sqrt{\frac{1}{5}})$.

Prediction error estimation in linear model under orthogonal design.

We consider linear model $Y = \beta X + \epsilon$, where X is a $n \times p$ plan matrix, $\epsilon \sim N\left(0_n, \sigma^2 I_{n\times n}\right)$ is a vector representing random noise and $\beta \in R^n$ is a vector of parameters. Let $\hat{\beta}$ be the estimate of β based on Y and some subset $X_{\tilde{p}}$ of X columns. We discuss some criteria of $X_{\tilde{p}}$ selection.

Least squares estimator $\hat{\beta}_{LS}$ minimizes the error in the training sample (a.k.a. residual sum of squares) $RSS = \|Y - \hat{Y}\|^2$, where $\hat{Y} = X\hat{\beta}$ and Y is the response used to fit the model. It might seem to be a good idea to pick $X_{\tilde{p}}$ resulting in smallest value of RSS. Indeed it doesn't make sense when comparing models with different number of columns (which we want to do), because RSS never increases when we add more variables. The thing we should minimize instead is the prediction error defined as

$$PE = E\left(\hat{Y} - Y^*\right)^2,$$

where $Y^* = X\beta + \epsilon^*$ and ϵ^* is a new noise, random and independent of that in training sample. The expression can be rewritten as follows:

$$PE = E\left\|X\hat{\beta} + \epsilon^* - X\beta\right\|^2 = E\sum_{i=1}^{n}\left(X(\beta - \hat{\beta}) + \epsilon^*\right)^2 = E\sum_{i=1}^{n}\left[(X(\beta - \hat{\beta}))^2 + 2X(\beta - \hat{\beta})\epsilon^* + (\epsilon^*)^2\right] =$$

$$E\sum_{i=1}^{n}\left[(X(\beta - \hat{\beta}))^2 + 2\sum_{i=1}^{n}E[X(\beta - \hat{\beta})]E\left[\epsilon^*\right] + \sum_{i=1}^{n}E\left(\epsilon^*\right)^2\right] = E\sum_{i=1}^{n}(X(\beta - \hat{\beta}))^2 + 0 + n\sigma^2 = E\|X(\beta - \hat{\beta})\|^2 + n\sigma^2$$

Stein's identity allows us to replace the first term with something called Stein's Unbiased Risk Estimator (exact form depends on the used estimator). If we use least squares estimator for parameters $\hat{\beta} = \left(X^T X\right)^{-1} X^T Y$, then $\hat{Y} = X\hat{\beta} = X\left(X^T X\right)^{-1} X^T Y = MY$ and by Stein's identity

$$E\|X(\beta - \hat{\beta})\| = RSS + \mathrm{tr}(M)\sigma^2 - n\sigma^2 \Longrightarrow PE = RSS + \mathrm{tr}(M)\sigma^2.$$

Trace of M is p if it's full rank. If σ^2 is unknown it should be replaced with its unbiased estimator $s^2 = \frac{RSS}{n-p}$. Another way of estimating the prediction error is by this formula which makes it easy to compute result of leave-one-out cross validation:

$$\hat{P}E = \sum_{i=1}^{n}\left(\frac{Y_i - \hat{Y}_i}{1 - M_{i,i}}\right)^2,$$

where $M = X\left(XX^T\right)^{-1} X^T$ is a matrix of projection onto $\mathrm{Lin}(X)$ (X denotes here $X_{\tilde{p}}$, a subset of X).

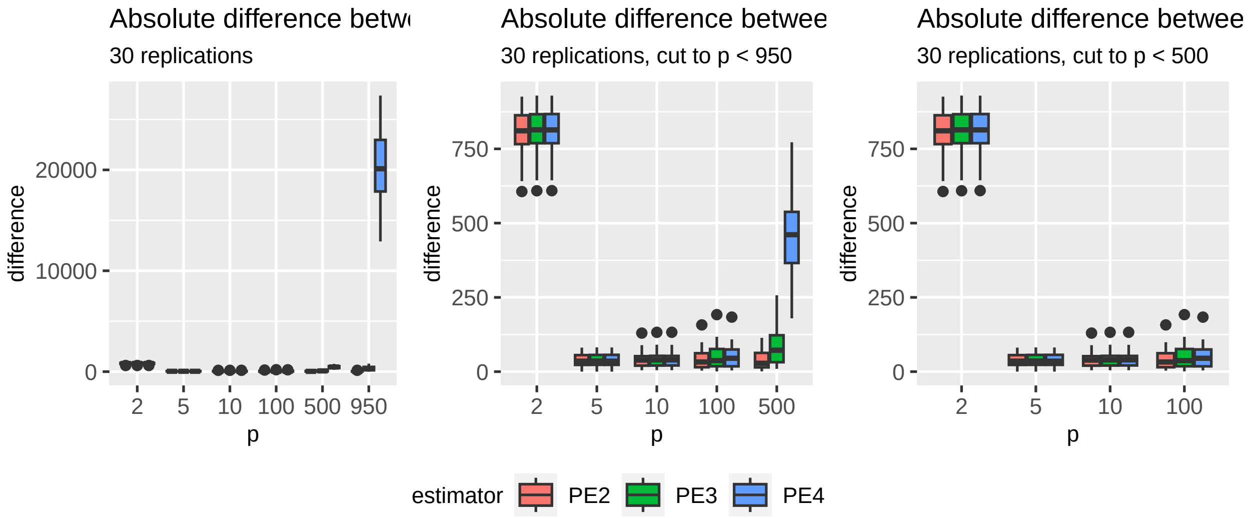
The table below show numeric results of experiment replicated 30 times, where PE1, PE2, PE3, PE4 are respectively: true prediction error, SURE with known σ^2, SURE with unknown σ^2 and LOO cross-validation estimator.

Table 2: Task 1 - numeric results averaged over 30 reps (seed = 42)

p	RSS	PE1	PE2	PE3	PE4
2	1799	1002	1803	1807	1807
5	1001	1005	1011	1011	1011
10	988	1010	1008	1008	1008
100	894	1101	1094	1093	1104
500	490	1504	1490	1469	1963
950	53	1945	1953	2078	22125

From table above we can see why we can't use RSS as model selection criterion. Even if we adding dummy column we can fit better to data, but we don't bring any useful information for model.

Boxplots (Prediction Error)



Model selection and regularization in multiple regression models.

We will use same setup as before. The only change is the β vector which is now $\beta_1 = \ldots = \beta_k = 6, \beta_{k+1} = \ldots = \beta_p = 0$ with $k = 20$. Note the fact this time we make assumption we do not know nothing about the data. We will perform selection and regularization methods feeding the model with full matrix X (950 predictors!). Following method will be used:

mBIC2

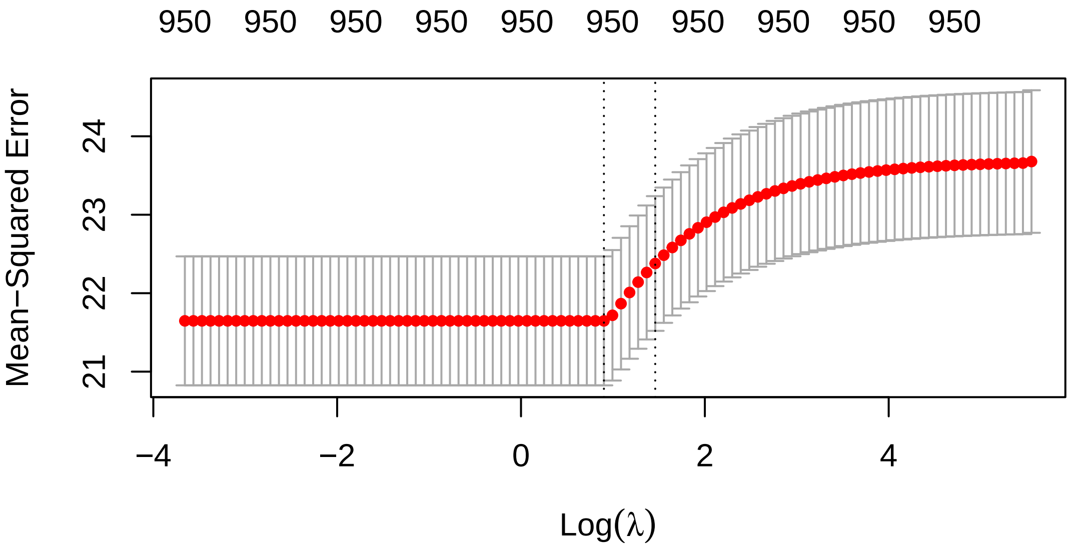
mBIC is designed to handle sligthly different case (when the number of parameters is greater than observation), but mBIC2 should adopt well to data with unknown sparisty.

We will use implementation from **bigstep** combined with stepwise selection method.

The method choose 20 variables with indexes 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20. $\|\hat{\beta} - \beta\|^2 = 0.581209786517244$, $\|X(\hat{\beta} - \beta)\|^2 = 18.8108317024012$. FDP 0 and power 1.

Ridge with the tunning parameter selected by cross validation

First we need to estimate the λ by CV. We will use k-folds provided in **glmnet** library.

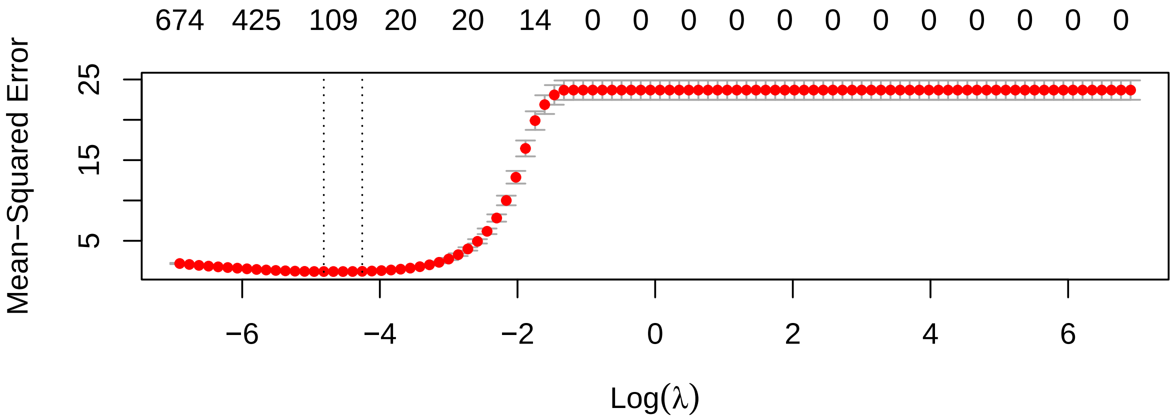


The λ estimated through k-folds CV is 2.4643. From plot we can see that we might as well use some other values like $exp(0)$ or lower without incorporating model performance. Note we analyse here just one particular design matrix.

$\|\hat{\beta} - \beta\|^2 = 643.2104$, $\|X(\hat{\beta} - \beta)\|^2 = 18591.78$.

LASSO with the tuning parameter selected by cross-validation

For LASSO we will use to λ computed via cv.glmnet. Lambda.min : λ of minimum mean cross-validated error. lambda.1se : largest value of λ such that error is within 1 standard error of the crossvalidated errors for lambda.min.



For LASSO model with lambda.min $\|\hat{\beta} - \beta\|^2 = 3.765$, $\|X(\hat{\beta} - \beta)\|^2 = 117.002$. For LASSO model with lambda.1se $\|\hat{\beta} - \beta\|^2 = 5.638$, $\|X(\hat{\beta} - \beta)\|^2 = 171.764$.

For this case we can see the 1se version has worse score in case of square errors but lower the false discovery proportion. Also power is preserved.

LASSO with the tuning parameter $\lambda = \Phi^{-1}\left(1 - \frac{0.1}{2p}\right)$

For this setup the $\lambda = 0.001$. The value looks like special value for Bonferroni correction. Indeed if we recall formula for flase discovery probability from exercise ($2\left(1 - \Phi\left(\frac{\lambda}{\sigma}\right)\right)$) and use given λ we got $\frac{\alpha}{p}$. So this value is use to force LASSO to use Bonferroni correction.

SLOPE with the BH sequence of the tuning parameters $\lambda_i = \Phi^{-1}\left(1 - \frac{0.1i}{2p}\right)$

No we will change canonical loss function. Instead using L2 penalty we will try to minimazie L1 sorted norm.

$$\hat{\beta} = \text{argmin}_{b \in \mathbb{R}^p} \frac{1}{2}\|y - Xb\|_{\ell_2}^2 + \sum_{i=1}^{p} \lambda_i |b|_{(i)}.$$

The estimator can be compute with SLOPE package.

	$\|\hat{\beta} - \beta\|^2$	$\|X(\hat{\beta} - \beta)\|^2$	FDP	TPP
mBIC2	0.581	18.811	0	1
Ridge	643.21	18591.78	-	-
LASSO	3.765	117.002	0.863	1
LASSO_1se	5.638	171.764	0.429	1
LASSO_custom	40.431	636.747	0.971	1
SLOPE	756.289	1010.749	0.979	1

The mBIC2 is the best among all tested method. It was able to correctly detect all true signals from data and make no false discovery. Ridge regression fails in such high dimensional setup. The difference between LASSO and LASSO_1se is as mentioned before. The second version lower the FDP.

I don't know what is natural interpretation of SLOPE - mainly how to interpret the sorted L1 norm. I try to get some intuition from orignal paper but Im fail. I know it has good properties like convexity but in practice I don't know when I would use this method. Also the yielded results seems to be very strange.

Note Without replication.