Penalized regression inference regarding variable selection in high dimensions: presentation of selected methods implemented in R

Marta Karas

Contact: http://statsox.github.io

Association between an outcome variable and predictors. To assess the association between an outcome $y \in \mathbb{R}^n$ and a set of predictors $x_j \in \mathbb{R}^n$, $j = 1, ..., p$, one might consider the model:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon,$$

where $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\beta \in \mathbb{R}^p$ is vector of coefficients, and $\epsilon \in \mathbb{R}^n$ is a vector of errors with mean zero and constant variance. If the number of variables p is much smaller than n, we could perform a formal statistical test for whether an element of β is zero using classical methods, such as likelihood ratio or Wald test. However, in the high-dimensional setting, when the number of variables p is large, these tests have low power, or are undefined.

Penalized regression techniques. In the case where p is large, penalized regression techniques such as Ridge and Lasso can be employed to obtain β estimates:

$$\widehat{\beta}_\lambda = \underset{b \in \mathbb{R}^p}{arg\ min} \left\{ \frac{1}{2n} ||\mathbf{y} - \mathbf{X}\mathbf{b}||_2^2 + \lambda J(b) \right\},$$

where $J(b) = \frac{1}{2}||b||_2$ for Ridge and $J(b) = ||b||_1$ for Lasso. However, Lasso and Ridge yield biased estimators of β, thus these procedures do not provide p-values or confidence intervals.

Methods. Penalized regression inference. Here, we present examples of usage of a few selected methods available in R:

- `lassoscore {lassoscore}`: **Score test based on penalized regression**. Performs penalized regression of an outcome on all but a single feature, and test for correlation of the residuals with the held-out feature; applied on each feature in turn.
- `hdi {hdi}`: **Multi sample-splitting**. Splits the sample into two equal halves, I_1 and I_2. First half I_1 is used for variable selection (with the use of Lasso) and the second half I_2, with the reduced set of selected variables (from I_1), is used for "classical" statistical inference in terms of p-values. Repeats the splitting procedure B times and aggregates obtained p-values.
- `grace.test {Grace}`: **Grace test**. Proposes how to overcome that Ridge is a biased estimator of β and its estimation bias is negligible only if the Ridge tuning parameter λ is close to zero. To construct a test statistic for the null hypothesis $H_0 : \ \beta_j^* = 0$ for some $j \in \{1, ..., p\}$, it adjusts for the potential estimation bias by using a stochastic bound derived from an initial estimator. Since with this adjustment the tuning parameter λ needs not be very small, coefficient estimation and corresponding p-values for penalized regression might be obtained.

Methods. Assessing the inference results. In regression settings, False Discovery Proportion (FDP) is often used to describe the proportion of false "discoveries" (whose coefficients in the true *full model* are zero). However, in settings with the presence of correlated predictors, more than one variable is likely to be capturing the same underlying signal. Then, "classical" FDP suffers from unintuitive and potentially undesirable behavior.

- Here, we use **False Variable Proportion (FVP)** measure ([x]), which considers a variable to be an interesting selection if it captures signal that has not been explained by any other variable in the selected model. Mathematically, for a selected variables set $A \subseteq \{1, ..., p\}$, we project the mean $\mathbf{X}\beta$ from the *full model* onto subset of predictors $\mathbf{X}_A$ to obtain a projected mean $\mathbf{X}\beta^{(A)}$. We define a selected variable to be a false selection if it has a zero coefficient in this projected mean vector:

$$FVP = |\{j \in A : \beta_j^{(A)} = 0\}| / |A|.$$

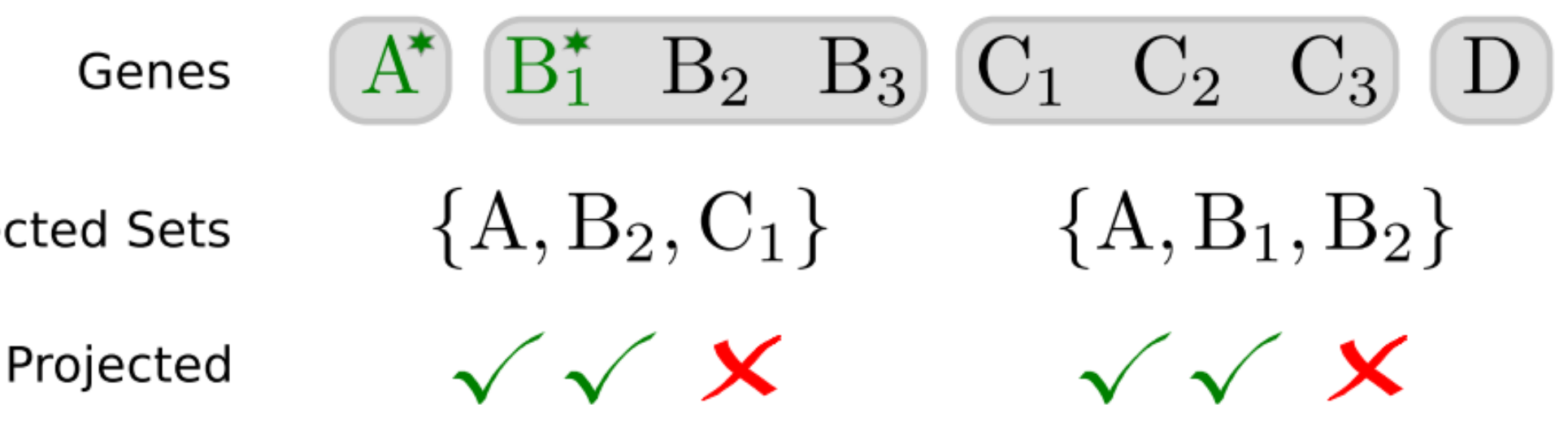
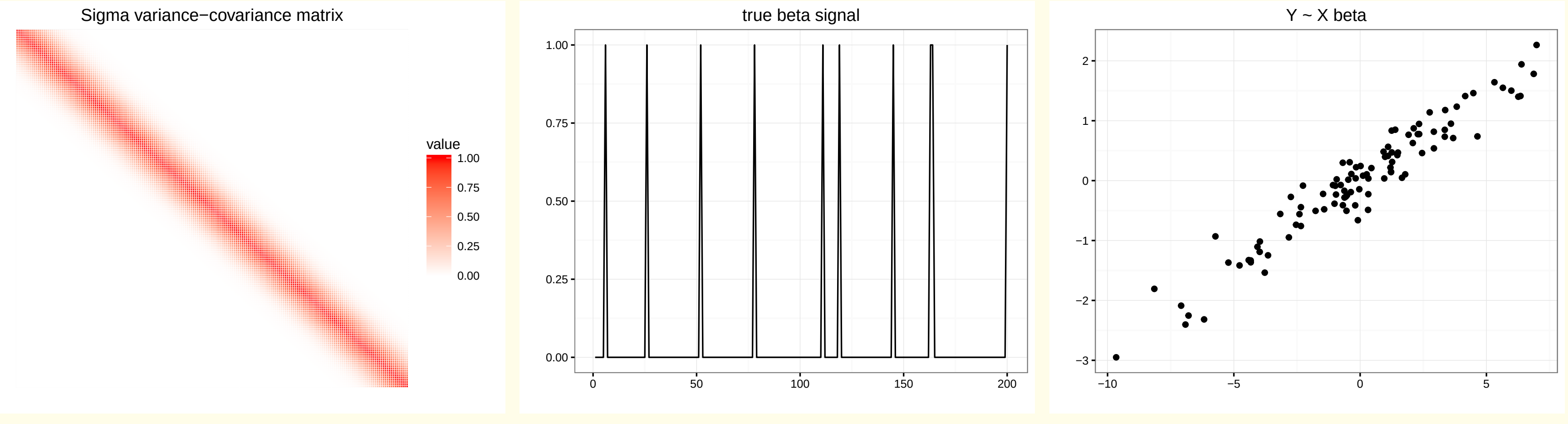


Figure 1. **False Variable Proportion (FVP)** illustration. Variables are denoted as correct selections if they are capturing unique signal among the selected variables. Thus B_2 is correctly selected in the first set. However, B_2 is considered a false selection in the second set because it adds no information beyond B_1. Figure & caption source: X.

False Variable Proportion application. In non-orthogonal settings, one is not likely to obtain exact zeros in a projected mean $\mathbf{X}\beta^{(A)}$ vector and heuristics is used to define threshold below which a variable is considered to be a false selection (here: 0.1).

R **usage example.** Assume we are given data matrix $X_{100 \times 300} \sim N(0, \Sigma)$, true signal β and observed response variable $Y \sim N(X\beta, 1^2)$.

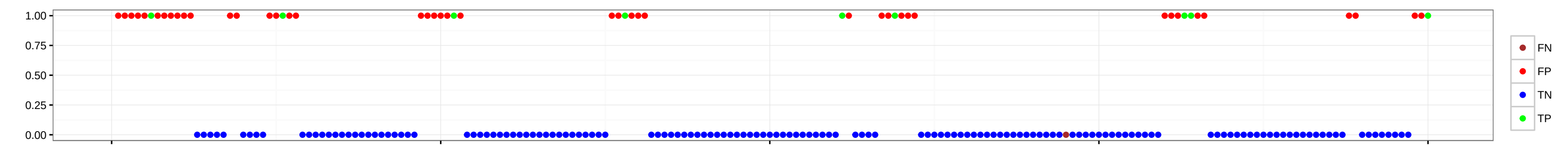


R: **Projected mean & FVP.** Values "close" to 0 indicate false discovery.

```
source("https://raw.githubusercontent.com/statsox/Penalized-Regression-Inference/master/R/false_variable_fw.R")

# True beta indices
(beta.sel.idx <- which(beta != 0))

##  [1]   6  26  52  78 111 119 145 163 164 200

round(FVP.proj.mean(beta.true = beta, beta.sel.idx = beta.sel.idx, Sigma = Sigma), 2)

##  [1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1

# Make change in selected beta indices vector we check for "false discovery"
beta.sel.idx <- c(c(7), beta.sel.idx[-1])
round(FVP.proj.mean(beta.true = beta, beta.sel.idx = beta.sel.idx, Sigma = Sigma), 2)

##  [1] -0.77 -0.85 -0.85 -0.85 -0.85 -0.85 -0.85 -0.85 -0.85 -0.85  0.85
```

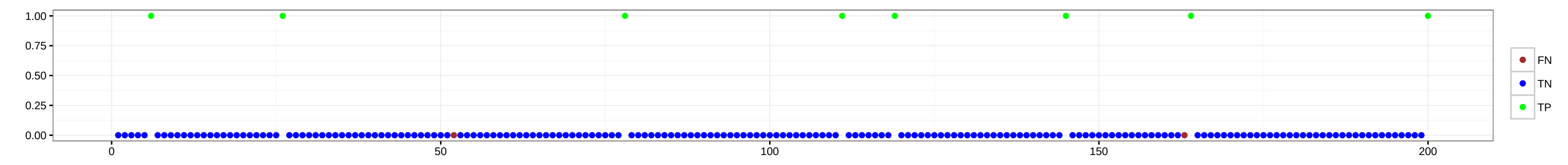
R: *p*-values & FVP: **Score test based on penalized regression.**

```
cv.res <-  cv.glmnet(X, Y) # Run cv.glmnet to choose *exemplary* lambda for which we compute lassoscore
res.lassoscore <- lassoscore(Y, X, lambda =  cv.res$lambda.1se)
beta.selected.idx <- which(res.lassoscore$p.model < 0.1) # subset of p.values < 0.1
```



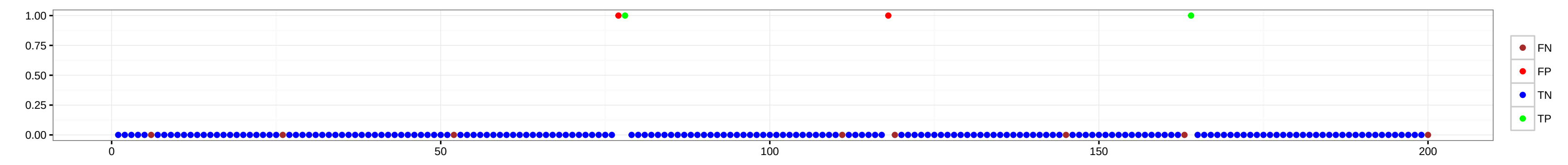
R: *p*-values & FVP: **Multi sample-splitting**

```
res.hdi.multi <- hdi(X, Y, method = "multi.split", B = 50, model.selector = lasso.cv,
                     args.model.selector = list(nfolds = 10))
beta.selected.idx <- which(res.hdi.multi$pval.corr < 0.1)
```



R: *p*-values & FVP: **Grace test**

```
lambda.2.seq <- exp(seq(-6, 10, length.out = 100))
res.grace <- grace.test(Y, X, L = matrix(0, p.tmp, p.tmp), lambda.L = 0, lambda.2 = lambda.2.seq)
beta.selected.idx <- which(res.grace$pvalue < 0.1)
```


FN
FP
TN
TP