Linear example

Contents

Deep Feature Selection

In this markdown, we will demonstrate the comparison methods that are implemented in Table 2 of our paper.

User Guide on nonlinear example

In this example, a high dimensional dataset with 1000 covariates and 500 observations is generated using the linear system:

$$y = X\beta + \epsilon$$

where $\beta \in \mathbb{R}^{1000}$, but only the first 100 elements of β is non-zero. Our task is to correctly select the important variables. Please see section 5.1 of the paper for detailed generation process.

In the markdown, the following methods will be implemented:

- LASSO
- Elastic Net
- SCAD

Data Preparation

In this section, we will read in the data that is generated using `linear_generator` from `./src/utils.py`

```
source("../../src/utils.R")
```

```
dirc = "../../data/linear/p_1000_N_1000_s_100/"
k = 0 # dataset index from 0 to 9
X <- read.table(paste(dirc, 'X_', toString(k), '.txt', sep=""))
y <- read.table(paste(dirc, 'y_', toString(k), '.txt', sep=""))
beta <- read.table(paste(dirc, "beta_", toString(k), '.txt', sep=""))
supp = which(beta!=0)
X_train = as.matrix(X[1:500,])
y_train = y[1:500,]
X_test = as.matrix(X[501:1000,])
y_test = y[501:1000,]
N = dim(X_train)[1]
p = dim(X_train)[2]
```

LASSO

In this section, we will implement LASSO for variable selections and predictive preformance. We will use R package ***glmnet***. We will use function `glmnet` with $\alpha = 1$. A sequence of λ will be tested and the best model will be selected based on EBIC.

```
library(glmnet)
LAMBDAs = exp(seq(log(0.05), log(5), length.out=100))
lasso = glmnet(as.matrix(X_train), as.matrix(y_train),
               lambda=LAMBDAs, alpha=1, seed=1)

Ss = colSums(lasso$beta!=0)
Y_Fits = predict(lasso, X_train)
Y_Preds = predict(lasso, X_test)
EBICs = EBICseq(Y_Fits, y_train, Ss, N)
best_idx = which.min(EBICs)

supp_lasso = c(1:1000)[lasso$beta[, best_idx]!=0]
train_mse_lasso = mean((Y_Fits[, best_idx]-y_train)^2)
test_mse_lasso = mean((Y_Preds[, best_idx]-y_test)^2)
fs_lasso = setdiff(supp_lasso, supp)
ns_lasso = setdiff(supp, supp_lasso)
```

The false selected variable: 106, 107, 167, 228, 276, 325, 399, 433, 453, 527, 528, 533, 546, 582, 583, 621, 625, 704, 789, 845, 870, 941, 991, 992

The negative selected variable:

The training MSE is 1.8161324, the test MSE is 2.9131399

Elastic Net

In this section, we will implement LASSO for variable selections and predictive preformance. We will use R package ***glmnet***. We will use function `glmnet` with a range of α from 0 to 0.5. A sequence of λ will be tested and the best model will be selected based on EBIC.

```
LAMBDAs = exp(seq(log(0.001), log(10), length.out=100))
ALPHAs = seq(0., 0.5, length.out=20)
EBICs_elastic = c()
for (alpha in ALPHAs) {
  elastic = glmnet(as.matrix(X_train), as.matrix(y_train),
                   alpha=alpha, lambda=LAMBDAs, seed=1)

  Ss = colSums(elastic$beta!=0)
  Y_Fits = predict(elastic, X_train)
  Y_Preds = predict(elastic, X_test)
  EBICs = EBICseq(Y_Fits, y_train, Ss, N)
  best_idx = which.min(EBICs)
  EBICs_elastic = c(EBICs_elastic, min(EBICs))
  if (min(EBICs) == min(EBICs_elastic)) {
    supp_elastic = c(1:1000)[elastic$beta[, best_idx]!=0]
    train_mse_elastic = mean((Y_Fits[, best_idx]-y_train)^2)
    test_mse_elastic = mean((Y_Preds[, best_idx]-y_test)^2)
    fs_elastic = setdiff(supp_elastic, supp)
    ns_elastic = setdiff(supp, supp_elastic)
```

```
  }
}
```

The false selected variable: 106, 107, 158, 167, 228, 276, 325, 369, 399, 432, 433, 445, 453, 527, 528, 533, 546, 582, 583, 621, 625, 704, 789, 845, 860, 870, 925, 941, 991, 992

The negative selected variable:

The training MSE is 1.9704196, the test MSE is 3.2832077

SCAD

In this section, we will implement SCAD for variable selections and predictive preformance. We will use R package ***ncvreg***. Function `ncvreg` will be used to train the model with SCAD penalty and a sequence of λ.

```
library(ncvreg)
LAMBDAs = exp(seq(log(1), log(0.01), length.out=100))
scad = ncvreg(X_train, y_train, penalty="SCAD",
              lambda = LAMBDAs, seed=1)

Ss = predict(scad, X_train, type="nvars")
Y_Fits = predict(scad, X_train)
Y_Preds = predict(scad, X_test)
EBICs = EBICseq(Y_Fits, y_train, Ss, N)
best_idx = which.min(EBICs)

supp_scad = c(1:1000)[scad$beta[, best_idx]!=0]
train_mse_scad = mean((Y_Fits[, best_idx]-y_train)^2)
test_mse_scad = mean((Y_Preds[, best_idx]-y_test)^2)
fs_scad = setdiff(supp_scad, supp)
ns_scad = setdiff(supp, supp_scad)
```

The false selected variable: 101

The negative selected variable: 61, 98

The training MSE is 0.9188832, the test MSE is 1.1346672