Shao Regression Example

Jun Shao (1993), Linear Model Selection by Cross-validation, Journal of the American Statistical Association Vol. 88, Iss. 422.

Model: $y = X\beta + e$, where $e \sim \text{NID}(0, \mathcal{I}_n)$ and $y = (y_1, \ldots, y_n)'$. The design matrix $X = (x_{i,j})$ is $n \times p$. Each row of X, $x_i = (x_{i,1}, \ldots, x_{i,p})$, is normally distributed with mean vector 0 and covariance matrix, $\left(0.5^{|i-j|}\right)_{p\times p}$.

But x_i, $i = 1, \ldots, n$ are independent, so obervations y_i, $i = 1, \ldots, n$ are all independent as in the usual OLS setup. The following parameter settings were used, $p = 8$, $\beta = (3,\ 1.5,\ 0,\ 0,\ 2,\ 0,\ 0,\ 0)$, $\sigma = 1$ with sample sizes $n = 20$, 60 and 100.

The function regal() can be used for cross-validation comparisons using the following 13 regression methods.

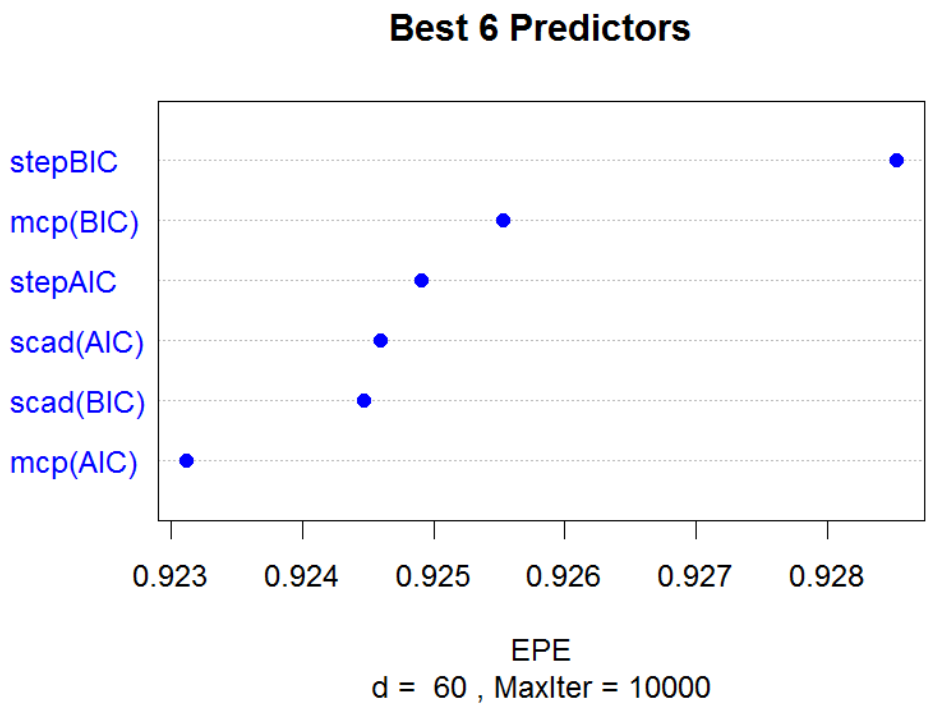
```
ABBREVIATION    PACKAGE     DESCRIPTION
lm              stats       full regression
stepAIC         stats       backward stagewise using AIC
stepBIC         stats       backward stagewise using BIC
LASSO(Cp)       lars        LASSO using Cp
mcp(AIC)        plus        MCP using AIC
mcp(BIC)        plus        MCP using BIC
scad(AIC)       plus        SCAD using AIC
scad(BIC)       plus        SCAD using BIC
H(el)           glmnet      elastic net, alpha=0.5
LASSO(el)       glmnet      elastic net, alpha=1
RR(el)          glmnet      elastic net, alpha=0.0
SVM             e1071       SVM
nn              N/A         Nearest Neighbour
```

The command to run regal() is very simple and illustrated in the script below that takes about 191 seconds on an i7 intel PC.

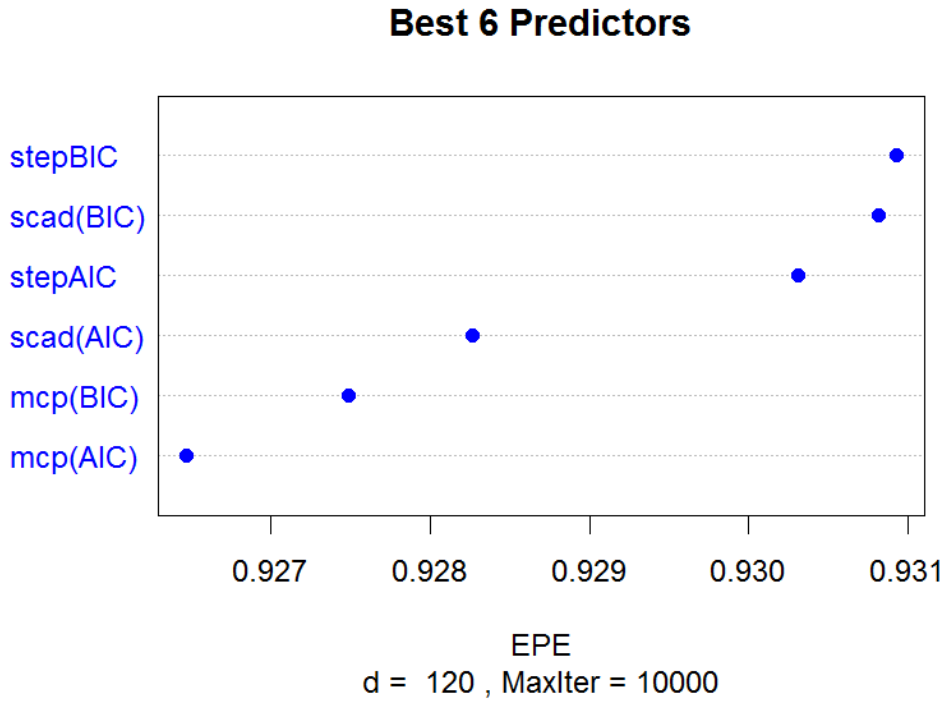
```
n <- 600
Xy <- ShaoReg(n=n)
regal(X=Xy[,1:8], y=Xy[,9], MaxIter=1000, NCores=8, d=n/2)
```

Dotchart comparisons for the cross-validated EPE

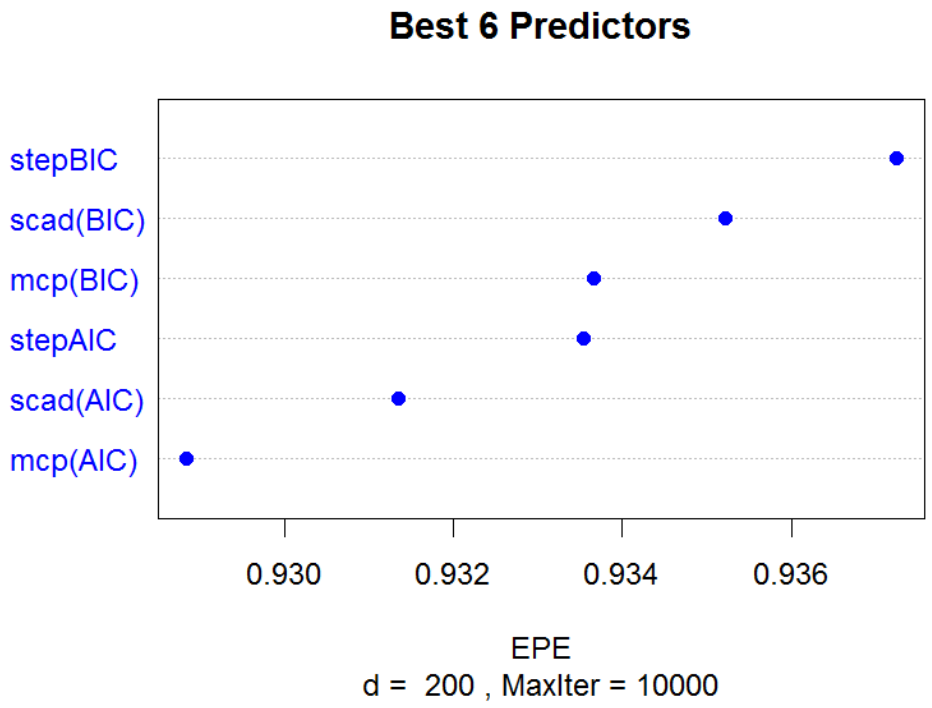
d = 60


Best 6 Predictors
stepBIC
mcp(BIC)
stepAIC
scad(AIC)
scad(BIC)
mcp(AIC)
0.923
0.924
0.925
0.926
0.927
0.928
EPE
d = 60 , MaxIter = 10000

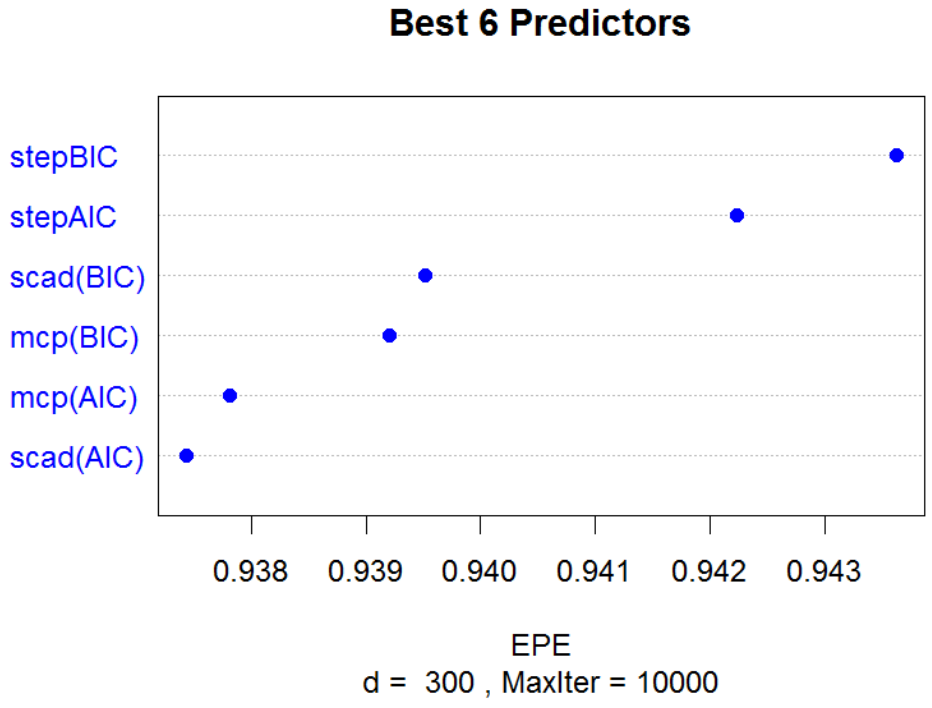
d = 120


Best 6 Predictors
stepBIC
scad(BIC)
stepAIC
scad(AIC)
mcp(BIC)
mcp(AIC)
0.927
0.928
0.929
0.930
0.931
EPE
d = 120 , MaxIter = 10000

d = 200



d = 300



Complete output from script

```
> cnames <- ceiling(c(n/10,n/5,n/3,n/2))
> names(out) <- cnames
```

```
> out
$`60`
                epe      95%MOE     pcorr    cpu rank
mcp(AIC)  0.9231157 0.003138304 0.9791204  34.20    1
scad(BIC) 0.9244729 0.003163675 0.9790893  42.76    2
scad(AIC) 0.9245924 0.003167385 0.9790866  42.46    3
stepAIC   0.9249067 0.003180808 0.9790794 126.03    4
mcp(BIC)  0.9255312 0.003137149 0.9790651  76.54    5
stepBIC   0.9285235 0.003168719 0.9789967 108.77    6
LASSO(Cp) 0.9307448 0.003163040 0.9789460  18.92    7
lm        0.9323714 0.003180384 0.9789088  16.72    8
H(el)     0.9829861 0.003326783 0.9777508 257.64    9
LASSO(el) 0.9834622 0.003289858 0.9777399 248.75   10
RR(el)    0.9876071 0.003440972 0.9776450 288.17   11
SVM       1.8240708 0.010296910 0.9583034 156.42   12
nn        5.2245576 0.017749256 0.8752842  24.82   13

$`120`
                epe      95%MOE     pcorr    cpu rank
mcp(AIC)  0.9264737 0.002123108 0.9790436  32.81    1
mcp(BIC)  0.9274894 0.002092152 0.9790204  74.76    2
scad(AIC) 0.9282679 0.002139620 0.9790026  41.70    3
stepAIC   0.9303114 0.002125657 0.9789559 121.64    4
scad(BIC) 0.9308115 0.002092328 0.9789444  41.60    5
stepBIC   0.9309227 0.002095361 0.9789419 107.79    6
LASSO(Cp) 0.9339196 0.002108536 0.9788734  18.58    7
lm        0.9373145 0.002155947 0.9787957  16.61    8
H(el)     0.9861349 0.002273696 0.9776787 253.55    9
LASSO(el) 0.9873869 0.002204088 0.9776500 251.47   10
RR(el)    0.9919783 0.002332514 0.9775449 283.61   11
SVM       1.9220552 0.008179312 0.9560121 132.78   12
nn        5.3998603 0.012495376 0.8707899  36.00   13

$`200`
                epe      95%MOE     pcorr    cpu rank
mcp(AIC)  0.9288405 0.001502143 0.9789895  32.72    1
scad(AIC) 0.9313441 0.001513751 0.9789322  41.14    2
stepAIC   0.9335472 0.001540574 0.9788819 117.34    3
mcp(BIC)  0.9336586 0.001463613 0.9788793  73.62    4
scad(BIC) 0.9352261 0.001466498 0.9788435  40.88    5
stepBIC   0.9372407 0.001480435 0.9787974 107.46    6
LASSO(Cp) 0.9379624 0.001492894 0.9787809  18.21    7
lm        0.9405785 0.001527525 0.9787211  16.87    8
LASSO(el) 0.9948059 0.001636829 0.9774802 246.89    9
H(el)     0.9968481 0.001678348 0.9774334 247.69   10
RR(el)    1.0032891 0.001728852 0.9772859 272.39   11
SVM       2.0829172 0.007602364 0.9522385 102.55   12
nn        5.6646141 0.010093149 0.8639580  46.76   13

$`300`
                epe      95%MOE     pcorr    cpu rank
scad(AIC) 0.9374327 0.001063763 0.9787930  40.48    1
mcp(AIC)  0.9378175 0.001095432 0.9787842  32.06    2
mcp(BIC)  0.9392076 0.001042578 0.9787524  72.70    3
```

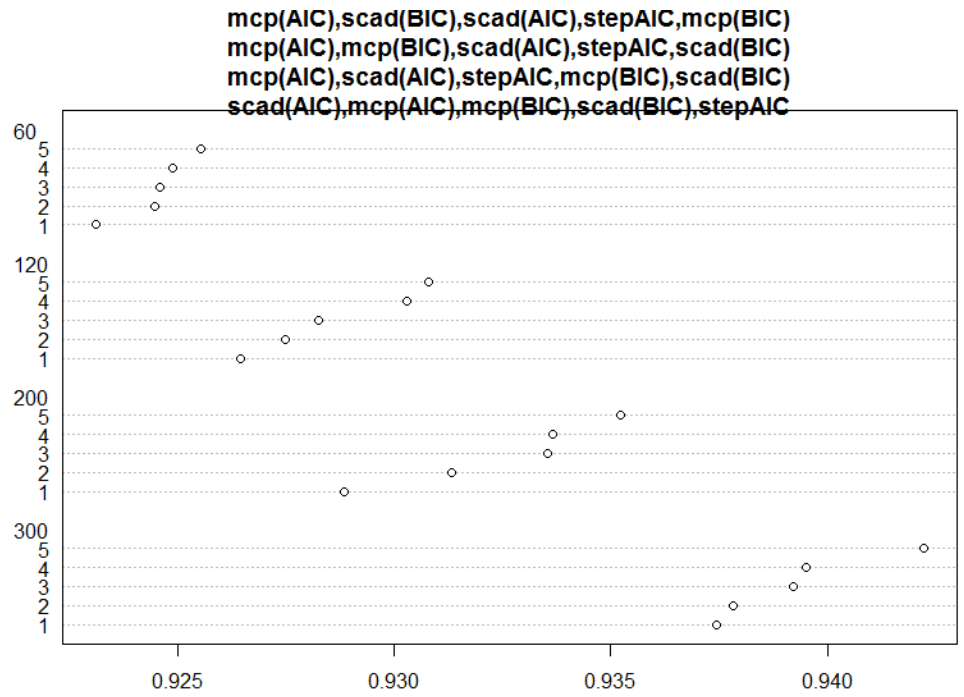
```
scad(BIC) 0.9395212 0.001055631 0.9787453  40.19    4
stepAIC   0.9422359 0.001096816 0.9786832 111.47    5
stepBIC   0.9436203 0.001055387 0.9786515 104.87    6
LASSO(Cp) 0.9438638 0.001089538 0.9786460  18.10    7
lm        0.9494607 0.001097444 0.9785179  16.50    8
LASSO(el) 1.0074498 0.001308288 0.9771906 240.91    9
H(el)     1.0105333 0.001359568 0.9771200 245.36   10
RR(el)    1.0185110 0.001403163 0.9769372 264.06   11
SVM       2.3666648 0.007891031 0.9455454  70.22   12
nn        6.0857429 0.008976182 0.8529780  54.84   13
```

The full linear regression was fit and used for the predictions. In each case it ranked 8/13. Consistenly the glmnet regressions (LASSO, RR and H) were next, followed by SVM and lastly by nn.

```
> ranklm <- numeric(4)
> names(ranklm) <- cnames
> for (i in 1:4)
+   ranklm[i] <- out[[i]]["lm",5]
> ranklm
 60 120 200 300
  8   8   8   8
```

```
> EPElm <- numeric(4)
> names(EPElm) <- cnames
> for (i in 1:4)
+   EPElm[i] <- out[[i]]["lm",1]
> EPElm
       60       120       200       300
0.9323714 0.9373145 0.9405785 0.9494607
```

W




```
#Source: ShaoReg-example.R
#
MaxIter <- 10^4 #about 1.5 hours
n <- 600
Xy <- ShaoReg(n=n)
epe <- (1+ncol(Xy)/n)
totTime <- proc.time()[3]
out <- NULL
system.time(
  out[[1]] <- regal(X=Xy[,1:8], y=Xy[,9], MaxIter=MaxIter, NCores=8)
)
system.time(
  out[[2]] <- regal(X=Xy[,1:8], y=Xy[,9], MaxIter=MaxIter, NCores=8, d=n/5)
)
system.time(
  out[[3]] <- regal(X=Xy[,1:8], y=Xy[,9], MaxIter=MaxIter, NCores=8, d=n/3)
)
system.time(
  out[[4]] <- regal(X=Xy[,1:8], y=Xy[,9], MaxIter=MaxIter, NCores=8, d=n/2)
)
(totTime <- proc.time()[3]-totTime)
paste(round(totTime/3600,2), "hours")
#
cnames <- ceiling(c(n/10,n/5,n/3,n/2))
names(out) <- cnames
out
#
EPElm <- numeric(4)
names(EPElm) <- cnames
for (i in 1:4)
```

```
  EPElm[i] <- out[[i]]["lm",1]
EPElm
#
ranklm <- numeric(4)
names(ranklm) <- cnames
for (i in 1:4)
  ranklm[i] <- out[[i]]["lm",5]
ranklm
#
EPE <- matrix(numeric(0), ncol=4, nrow=13)
for (i in 1:4)
  EPE[,i] <- out[[i]][,1]
EPE <- EPE[-c(6:13),]
colnames(EPE) <- names(EPElm)
ti <- paste(
  paste(rownames(out[[1]])[1:5],collapse=","),
  paste(rownames(out[[2]])[1:5],collapse=","),
  paste(rownames(out[[3]])[1:5],collapse=","),
  paste(rownames(out[[4]])[1:5],collapse=","),sep="\n")
dotchart(EPE, main=ti)
```