Brain Data

Requires "care" version 1.1.1 (July 2011) or later

This R script reproduces the analysis of brain gene expression data from V. Zuber and K. Strimmer. 2011. *High-dimensional regression and variable selection using CAR scores.* Statist. Appl. Genet. Mol. Biol. **10**: 34 (http://dx.doi.org/10.2202/1544-6115.1730)

Load "care" package and brain gene expression data set

```
library("care")
```

```
## Loading required package: corpcor
```

Load Lu et al. (2004) data set:

```
data(lu2004)
x = lu2004$x
y = lu2004$y # age
dim(x)
```

```
## [1]  30 403
```

Regularization parameters used to fit linear model:

```
reg = slm(x,y)$regularization
```

```
## Estimating optimal shrinkage intensity lambda (correlation matrix): 0.1373
## Estimating optimal shrinkage intensity lambda.var (variance vector): 0.0246
```

```
lambda = reg[1] # correlation shrinkage
lambda.var = reg[2] # variance shrinkage
```

Compute CAR scores to rank predictors:

```
car = carscore(x, y, lambda=lambda)
ocar = order(car^2, decreasing=TRUE)
ocar[1:30]
```

```
##  [1]  73 297 262 182 336 140 362 403 389 124 239  83 244 123 148 160 212
## [18] 238 361 339  36 167 369  59  44  57 268 250 216  24
```

Fit linear models with increasing number of predictors:

```r
numpred =  c(seq(5, 45, by=10), seq(53, 403, by=25))
numpred
```

```
##  [1]   5  15  25  35  45  53  78 103 128 153 178 203 228 253 278 303 328
## [18] 353 378 403
```

```r
car.predlist = make.predlist(ocar, numpred, name="CAR")
car.models = slm.models(x, y, car.predlist, lambda=lambda, lambda.var=lambda.var)
```

```
## Determine regression coefficients for CAR.5 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.15 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.25 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.35 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.45 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.53 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.78 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.103 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.128 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.153 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.178 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
```

```
##
## Determine regression coefficients for CAR.203 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.228 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.253 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.278 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.303 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.328 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.353 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.378 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for CAR.403 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
```

For comparison, use marginal correlations to rank predictors:

```
marg = carscore(x, y, lambda=lambda, diagonal=TRUE) # shrinking not actually reqired
omarg = order(marg^2, decreasing=TRUE)
marg.predlist = make.predlist(omarg, numpred, name="MARG")
marg.models = slm.models(x, y, marg.predlist, lambda=lambda, lambda.var=lambda.var)
```

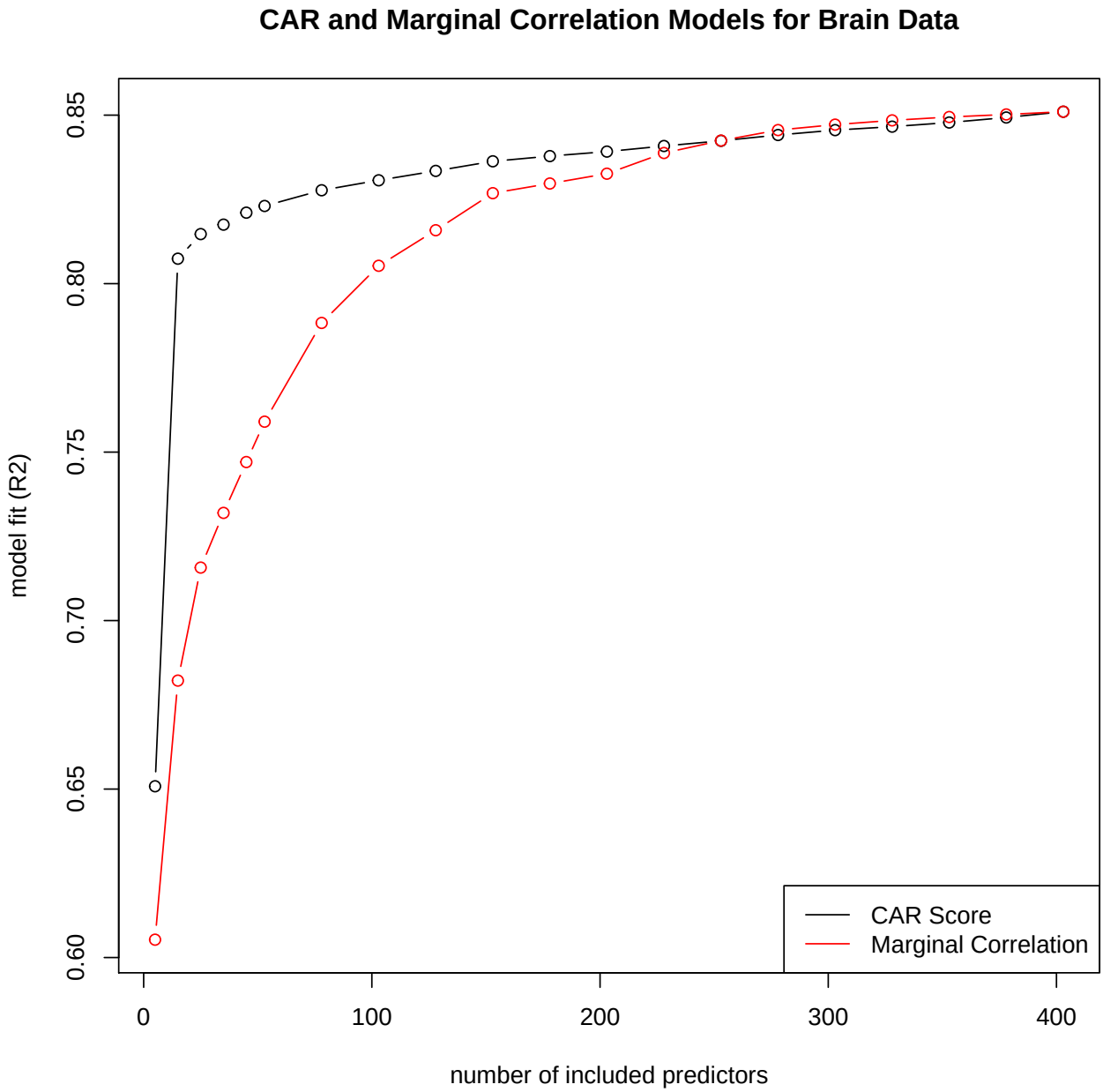
```
## Determine regression coefficients for MARG.5 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.15 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.25 model
```

```
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.35 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.45 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.53 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.78 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.103 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.128 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.153 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.178 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.203 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.228 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.253 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.278 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.303 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
```

```
##
## Determine regression coefficients for MARG.328 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.353 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.378 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
##
## Determine regression coefficients for MARG.403 model
## Specified shrinkage intensity lambda (correlation matrix):  0.1373
## Specified shrinkage intensity lambda.var (variance vector): 0.0246
```

Plot model fitted R^2 for all models comparing marginal correlation with CAR scores:

```
ylim = range( c(marg.models$R2, car.models$R2) )
plot(car.models$numpred, car.models$R2, type="b", ylim=ylim,
  xlab="number of included predictors",
  ylab="model fit (R2)",
  main="CAR and Marginal Correlation Models for Brain Data")
points(marg.models$numpred, marg.models$R2, col=2, type="b")
legend("bottomright", c("CAR Score", "Marginal Correlation"), col=c(1,2), lty=c(1,1) )
```



Estimate prediction error by crossvalidation

```
library("crossval")
```

Standardize data following Zuber and Strimmer (2011):

```
xs = scale(x)
ys = scale(y)

K=5  # number of folds
B=100 # number of repetitions
```

Rank by CAR scores, fit and predict using a specified number of predictors:

```
predfun = function(Xtrain, Ytrain, Xtest, Ytest, numVars)
{
  # rank the variables according to squared CAR scores
  car = carscore(Xtrain, Ytrain, verbose=FALSE)
  ocar = order(car^2, decreasing=TRUE)
  selVars = ocar[1:numVars]

  # fit and predict
  slm.fit = slm(Xtrain[, selVars, drop=FALSE], Ytrain, verbose=FALSE)
  Ynew = predict(slm.fit, Xtest[, selVars, drop=FALSE], verbose=FALSE)

  # compute squared error risk
  mse = mean( (Ynew - Ytest)^2)

  return(mse)
}
```

Compute results from Table 9 in Zuber and Strimmer (2011):

```
set.seed(12345)
cvp = crossval(predfun, xs, ys, K=K, B=B, numVars = 36, verbose=FALSE)
c(cvp$stat, cvp$stat.se) # 0.3441316  0.007449175
```

```
## [1] 0.344131632 0.007449175
```

```
set.seed(12345)
cvp = crossval(predfun, xs, ys, K=K, B=B, numVars = 60, verbose=FALSE)
c(cvp$stat, cvp$stat.se) # 0.3085344  0.006405896
```

```
## [1] 0.312875469 0.006584369
```

```
set.seed(12345)
cvp = crossval(predfun, xs, ys, K=K, B=B, numVars = 85, verbose=FALSE)
c(cvp$stat, cvp$stat.se) # 0.297824   0.006178467
```

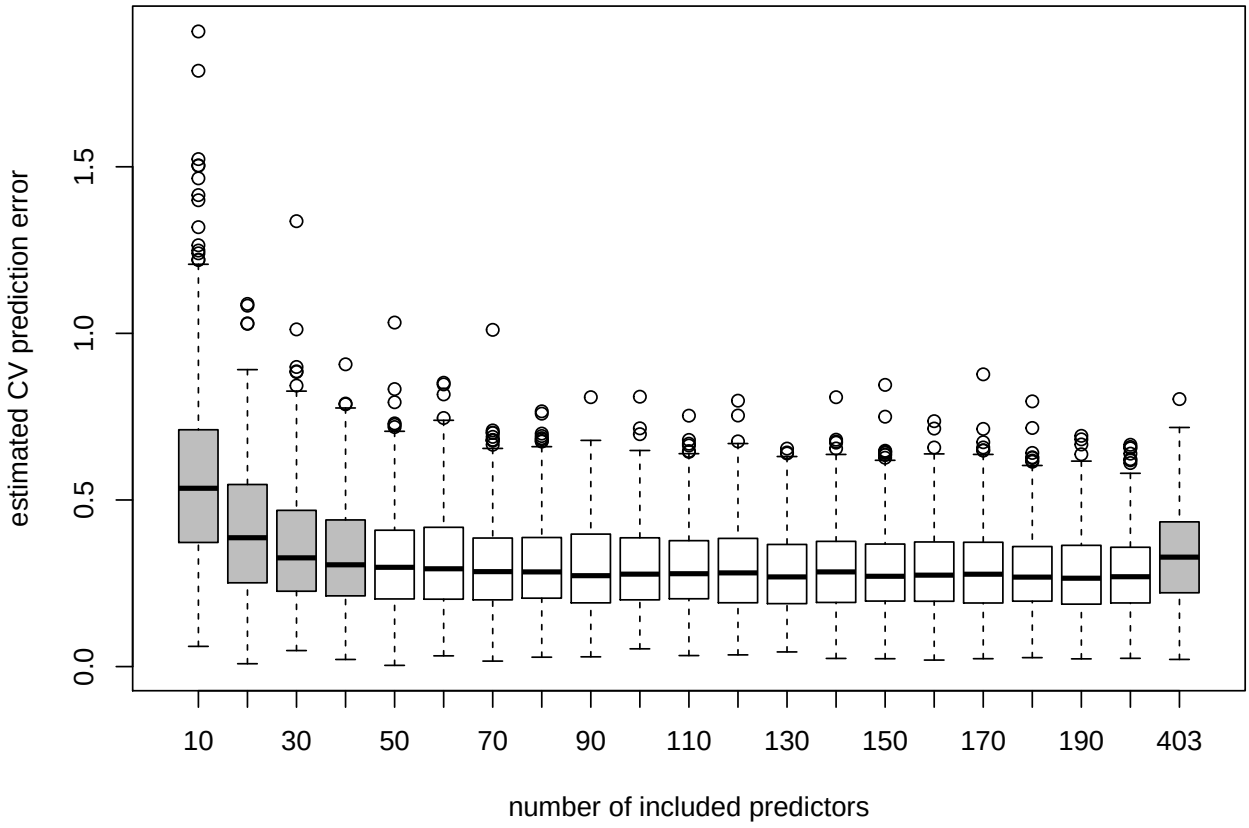
```
## [1] 0.297823963 0.006178467
```

This produces Figure 3 in Zuber and Strimmer (2011)

```
numpred = c(seq(10, 200, 10), 403) # number of predictors
set.seed(12345)
cvsim = lapply(numpred,
  function(i)
  {
    cat("Number of predictors:", i, "\n")
    cvp = crossval(predfun, xs, ys, K=K, B=B, numVars = i, verbose=FALSE)
    return( cvp$stat.cv )
  }
)
```

```
## Number of predictors: 10
## Number of predictors: 20
## Number of predictors: 30
## Number of predictors: 40
## Number of predictors: 50
## Number of predictors: 60
## Number of predictors: 70
## Number of predictors: 80
## Number of predictors: 90
## Number of predictors: 100
## Number of predictors: 110
## Number of predictors: 120
## Number of predictors: 130
## Number of predictors: 140
## Number of predictors: 150
## Number of predictors: 160
## Number of predictors: 170
## Number of predictors: 180
## Number of predictors: 190
## Number of predictors: 200
## Number of predictors: 403
```

```
boxplot(cvsim, names=numpred,
col=c(rep("grey", 4), rep("white", 16), "grey"),
 main="CAR Models for the Gene Expression Data", xlab="number of included predictors",
       ylab="estimated CV prediction error")
```


CAR Models for the Gene Expression Data
estimated CV prediction error
0.0
0.5
1.0
1.5
10
30
50
70
90
110
130
150
170
190
403
number of included predictors