Diabetes Data

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

This R script reproduces the analysis of diabetes 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 diabetes data set

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

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

Diabetes data (442 patients) from Efron et al. 2004. *Least angle regression* Ann. Statist. **32**:407-499.

```
data(efron2004)
x = efron2004$x
dim(x)
```

```
## [1] 442  10
```

```
d = ncol(x) # dimension
n = nrow(x) # samples
```

10 predictors

```
xnames = colnames(x)
xnames
```

```
##  [1] "age" "sex" "bmi" "bp"  "s1"  "s2"  "s3"  "s4"  "s5"  "s6"
```

Response

```
y = efron2004$y
length(y)
```

```
## [1] 442
```

Comparison of linear regression models

Ordering of predictors according to CAR score:

```
car = carscore(x, y, lambda=0) # no shrinkage estimation needed as n>>d
ocar = order(car^2, decreasing=TRUE)
xnames[ocar]
```

```
##  [1] "bmi" "s5"  "bp"  "s3"  "s4"  "s6"  "sex" "age" "s2"  "s1"
```

Regression coefficients for models with increasing number of predictors:

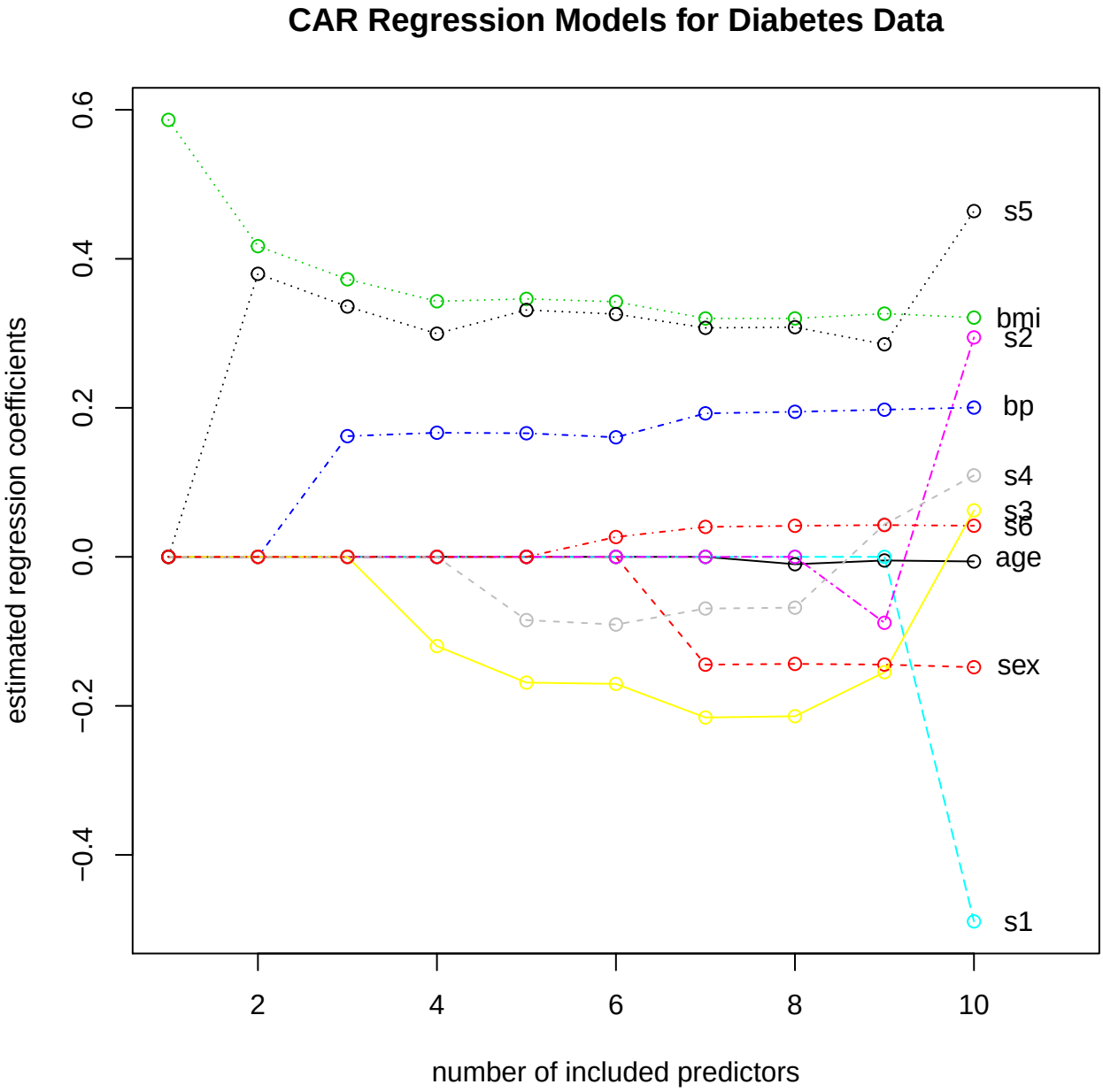
```
car.predlist = make.predlist(ocar, numpred = 1:d, name="CAR")
cm = slm.models(x, y, car.predlist, lambda=0, lambda.var=0, verbose=FALSE)
bmat= cm$coefficients[,-1]
bmat
```

```
##                 age        sex       bmi        bp         s1          s2
## CAR.1   0.000000000  0.0000000 0.5864501 0.0000000  0.0000000  0.00000000
## CAR.2   0.000000000  0.0000000 0.4169792 0.0000000  0.0000000  0.00000000
## CAR.3   0.000000000  0.0000000 0.3725089 0.1620028  0.0000000  0.00000000
## CAR.4   0.000000000  0.0000000 0.3429868 0.1665741  0.0000000  0.00000000
## CAR.5   0.000000000  0.0000000 0.3462594 0.1659129  0.0000000  0.00000000
## CAR.6   0.000000000  0.0000000 0.3423566 0.1604471  0.0000000  0.00000000
## CAR.7   0.000000000 -0.1446828 0.3199577 0.1925611  0.0000000  0.00000000
## CAR.8  -0.009891500 -0.1436770 0.3200061 0.1946733  0.0000000  0.00000000
## CAR.9  -0.004889985 -0.1446359 0.3264617 0.1975168  0.0000000 -0.08850379
## CAR.10 -0.006184366 -0.1481322 0.3210963 0.2003705 -0.4893188  0.29447786
##                 s3          s4        s5         s6
## CAR.1   0.00000000  0.00000000 0.0000000 0.00000000
## CAR.2   0.00000000  0.00000000 0.3798445 0.00000000
## CAR.3   0.00000000  0.00000000 0.3359408 0.00000000
## CAR.4  -0.11980188  0.00000000 0.2995634 0.00000000
## CAR.5  -0.16878609 -0.08493276 0.3313158 0.00000000
## CAR.6  -0.17049323 -0.09089434 0.3258394 0.02662381
## CAR.7  -0.21558569 -0.06935615 0.3073028 0.04019060
## CAR.8  -0.21392826 -0.06821105 0.3082486 0.04165875
## CAR.9  -0.15479060  0.04351624 0.2852709 0.04269798
## CAR.10  0.06241353  0.10936955 0.4640526 0.04177106
```

Plot regression coefficients:

```
plot(1:d, bmat[,1], type="l",
  ylab="estimated regression coefficients",
  xlab="number of included predictors",
  main="CAR Regression Models for Diabetes Data",
  xlim=c(1,d+1), ylim=c(min(bmat), max(bmat)))

for (i in 2:d) lines(1:d, bmat[,i], col=i, lty=i)
for (i in 1:d) points(1:d, bmat[,i], col=i)
for (i in 1:d) text(d+0.5, bmat[d,i], xnames[i])
```



Estimate prediction errors by crossvalidation

```
library("crossval")

K=10 # number of folds
B=50 # number of repetitions
```

Prediction function used in crossvalidation: Rank by CAR scores, then fit and predict using a specified number of predictors (note this takes into account the uncertainty in selection and ordering of the predictors)

```
predfun = function(Xtrain, Ytrain, Xtest, Ytest, numVars)
{
  # rank the variables according to squared CAR scores
  car = carscore(Xtrain, Ytrain, verbose=FALSE, lambda=0)
  ocar = order(car^2, decreasing=TRUE)
  selVars = ocar[1:numVars]

  # fit and predict
  slm.fit = slm(Xtrain[, selVars, drop=FALSE], Ytrain, verbose=FALSE,
       lambda=0, lambda.var=0)
  Ynew = predict(slm.fit, Xtest[, selVars, drop=FALSE], verbose=FALSE)

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

  return(mse)
}
```

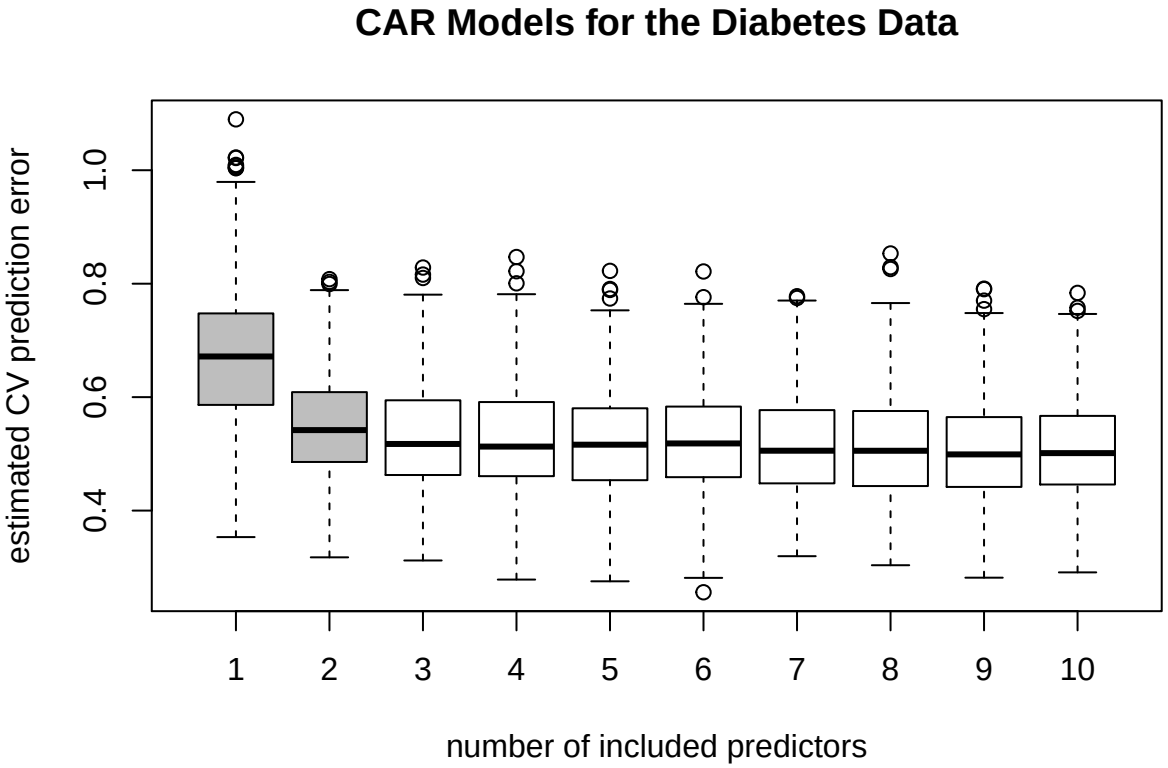
Perform crossvalidation:

```
numpred = 1:10 # number of predictors
set.seed(12345)
cvsim = lapply(numpred,
  function(i)
  {
    cat("Number of predictors:", i, "\n")
    cvp = crossval(predfun, x, y, K=K, B=B, numVars = i, verbose=FALSE)
    return( cvp$stat.cv )
  }
)
```

```
## Number of predictors: 1
## Number of predictors: 2
## Number of predictors: 3
## Number of predictors: 4
## Number of predictors: 5
## Number of predictors: 6
## Number of predictors: 7
## Number of predictors: 8
## Number of predictors: 9
## Number of predictors: 10
```

Plot results:

```
boxplot(cvsim, names=numpred,
col=c(rep("grey", 2), rep("white", 8)),
 main="CAR Models for the Diabetes Data", xlab="number of included predictors",
       ylab="estimated CV prediction error")
```



Conclusion: after including the three top ranked predictors ("bmi", "s5", "bp") no further reduction of MSE is seen.