energy.R

aditinabar

Wed Apr 26 23:12:55 2017

```r
setwd('/Users/aditinabar/Documents/naditi/Spring_2017/Fu/StatsLasso')

library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.2.4

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-5
```

```r
library(boot)
library(broom)
```

```
## Warning: package 'broom' was built under R version 3.2.5
```

```r
library(knitr)
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.2.5
```

```r
original <- read.csv('./energydata_complete.csv')

# # remove date
original <- original[ , -1]

train_size <- dim(original)[1]*.7
train_indices <- sample(dim(original)[1], floor(dim(original)[1]*.7), replace = FALSE)

train <- original[train_indices, ]
test <- original[-train_indices, ]

energy_matrix <- model.matrix(Appliances ~ ., data = train)

# models

ols <- lm(Appliances ~ ., data = train)
```

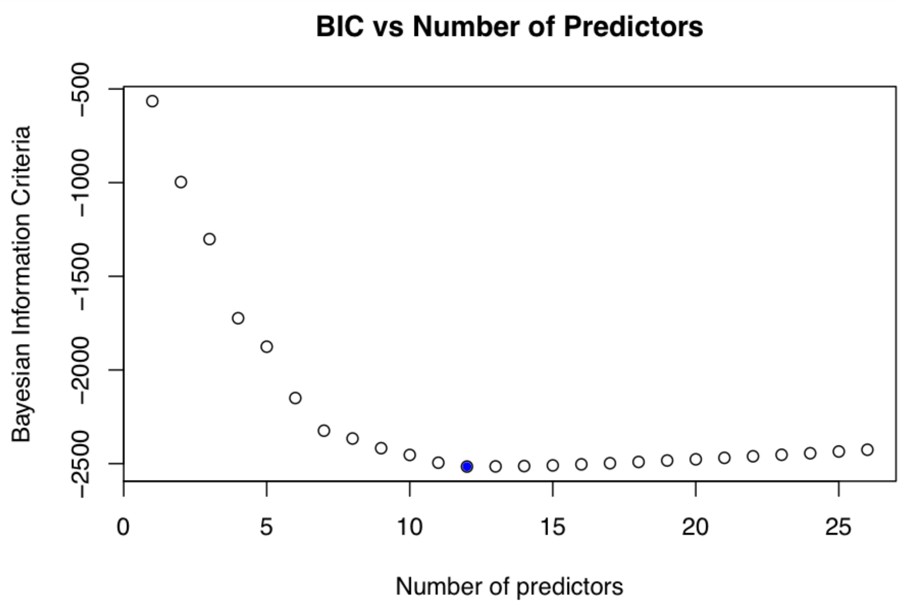
```
cv.ridge_train <- cv.glmnet(data.matrix(train[, -1]),
                            data.matrix(train[, 1]),
                            alpha=0,
                            standardize=TRUE,
                            type.measure="mse",
                            standardize.response=TRUE)

subset.Selection <- regsubsets(Appliances ~ ., data = train, method = "exhaustive", nvmax = NULL)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : nvmax reduced to 26
```

```
subset.Selection.summary <- summary(subset.Selection)
plot(subset.Selection.summary$bic,
     xlab = "Number of predictors",
     ylab = "Bayesian Information Criteria",
     main = "BIC vs Number of Predictors")
subsetMin <- which.min(subset.Selection.summary$bic)
points(subsetMin, subset.Selection.summary$bic[subsetMin], pch=20, col="blue")
```




```
coef(subset.Selection, subsetMin)
```

```
## (Intercept)      lights        RH_1          T2        RH_2          T3
##   90.005586    2.033078   16.615250  -21.318509  -15.104787   26.111951
##        RH_3          T6          T8        RH_8          T9       T_out
##    4.842659    7.040226   10.194625   -5.839190  -19.903289   -6.145558
##   Windspeed
```

```
##    2.040292
```

```
cv.lasso_train <- cv.glmnet(data.matrix(train[, -1]),
                             data.matrix(train[, 1]),
                             alpha=1,
                             standardize=TRUE,
                             type.measure="mse")

ols$coefficients
```

```
##  (Intercept)       lights           T1         RH_1           T2
##  16.87138947   2.05745256   0.83196618  16.24972594 -20.42982572
##         RH_2           T3         RH_3           T4         RH_4
## -14.82291362  27.13153372   5.06515136  -2.60776741   0.23675617
##           T5         RH_5           T6         RH_6           T7
##  -1.87931095   0.21485157   7.53784651   0.24312032   2.77022886
##         RH_7           T8         RH_8           T9         RH_9
##  -1.65917589   8.52409393  -4.91050784 -17.45432875  -0.73843240
##        T_out  Press_mm_hg       RH_out    Windspeed   Visibility
##  -9.12709368   0.14219444  -0.71595283   1.80718829   0.17465212
##    Tdewpoint          rv1          rv2
##   3.43530802  -0.07460534           NA
```

```
plot
```

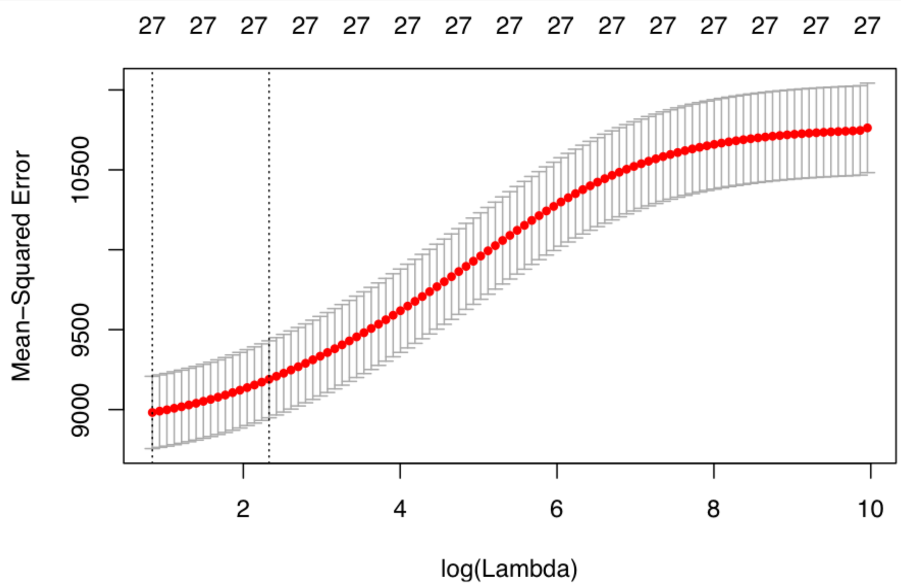
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa6d3cf7270>
## <environment: namespace:graphics>
```

```
coef(cv.ridge_train)
```

```
## 28 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) 136.256943313
## lights        2.140318281
## T1           -2.953032273
## RH_1          6.343705750
## T2            0.849504440
## RH_2         -2.825508941
## T3           12.685020683
## RH_3          3.401539496
## T4           -3.490389502
## RH_4         -0.076098498
## T5           -4.026112766
## RH_5          0.124632445
## T6            1.506803366
## RH_6         -0.007201203
## T7           -0.858021306
## RH_7         -1.962096178
## T8            2.914684603
## RH_8         -3.074003677
## T9           -5.324875515
## RH_9         -1.087593445
## T_out        -0.412042431
## Press_mm_hg  -0.126757184
```

```
## RH_out       -0.295815491
## Windspeed     1.629038425
## Visibility    0.126990744
## Tdewpoint    -0.859467933
## rv1          -0.036025471
## rv2          -0.035930010
```

```r
plot(cv.ridge_train)
```

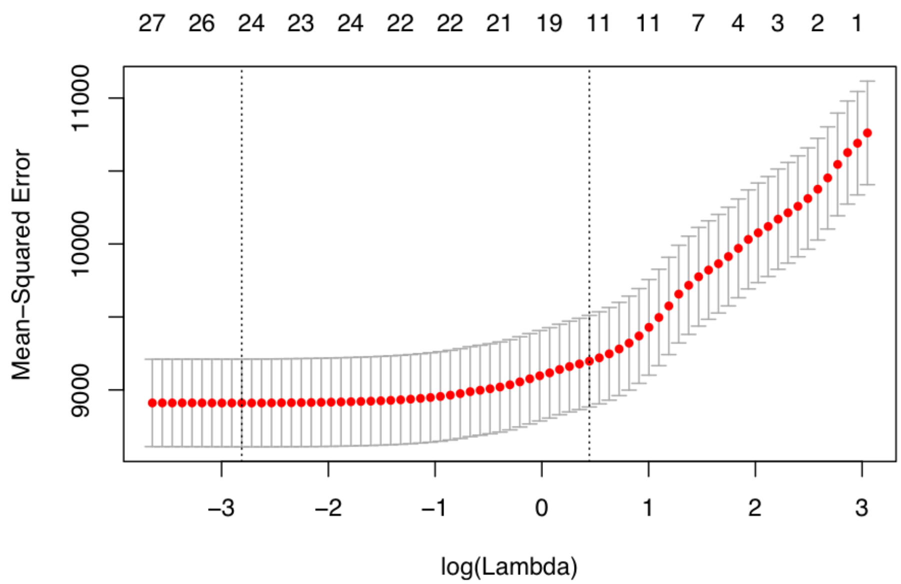



```r
coef(cv.lasso_train)
```

```
## 28 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 83.80330002
## lights       2.14502497
## T1           .
## RH_1         9.38861781
## T2           .
## RH_2        -4.47588357
## T3          13.59330939
## RH_3         0.01454651
## T4          -3.46773943
## RH_4         .
## T5          -2.40645503
## RH_5         .
## T6           .
## RH_6         .
## T7           .
## RH_7        -1.70564985
## T8           .
## RH_8        -3.59345308
```

```
## T9            -7.43839318
## RH_9           .
## T_out          .
## Press_mm_hg    .
## RH_out        -0.25979179
## Windspeed      1.36123038
## Visibility     0.01206913
## Tdewpoint      .
## rv1            .
## rv2            .
```

```r
plot(cv.lasso_train)
```




```r
# bootstrapping
n_folds = 20
```

```r
ols_model <- function(data, indices) {
  samp <- data.frame(scale(data[indices, ]))
  names(samp)
  model <- lm(Appliances ~ ., data=samp)
  return(coef(model))
}

ols_analysis <- boot(data=train , statistic=ols_model, R=n_folds)
ols_analysis
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = train, statistic = ols_model, R = n_folds)
##
##
## Bootstrap Statistics :
##           original         bias     std. error
## t1*   6.146432e-17 -1.428746e-16 1.374155e-15
## t2*   1.571041e-01  2.405450e-03 1.250562e-02
## t3*   1.294219e-02  1.180111e-02 4.405028e-02
## t4*   6.242682e-01 -3.744437e-03 4.969171e-02
## t5*  -4.337681e-01 -1.169104e-02 5.165404e-02
## t6*  -5.806006e-01 -7.542646e-03 5.081434e-02
## t7*   5.267321e-01  8.983465e-03 2.653556e-02
## t8*   1.598554e-01 -2.247377e-03 3.717232e-02
## t9*  -5.156585e-02 -1.164097e-02 2.484595e-02
## t10*  9.945737e-03  7.840863e-03 3.864670e-02
## t11* -3.364720e-02  9.585265e-03 2.925540e-02
## t12*  1.867616e-02 -9.886404e-04 1.071070e-02
## t13*  4.430040e-01  9.975283e-03 4.124250e-02
## t14*  7.324255e-02 -5.138507e-03 2.535046e-02
## t15*  5.655642e-02  3.791885e-03 3.722571e-02
## t16* -8.192102e-02  7.954075e-03 2.117045e-02
## t17*  1.613332e-01 -2.757926e-03 2.183507e-02
## t18* -2.473989e-01  4.537870e-03 2.134301e-02
## t19* -3.407820e-01 -1.108756e-02 4.519699e-02
## t20* -2.959637e-02 -6.457346e-03 1.959858e-02
## t21* -4.678224e-01 -1.517341e-02 8.411278e-02
## t22*  1.020234e-02  4.263631e-04 6.298328e-03
## t23* -1.027357e-01 -5.196764e-03 5.054534e-02
## t24*  4.265749e-02 -3.169411e-03 1.020879e-02
## t25*  1.982358e-02  2.588506e-03 7.006844e-03
## t26*  1.390444e-01  6.981334e-03 6.847148e-02
## t27* -1.045662e-02 -1.625247e-03 7.445786e-03
## WARNING: All values of t28* are NA
```

```r
ridge_model <- function(data, indices) {
  samp <- data[indices, ]
  model <- cv.glmnet(data.matrix(samp[, -1]), data.matrix(samp[, 1]), alpha=0, type.measure="mse")
  return(as.double(coef(model)))
}

ridge_analysis <- boot(data=train, statistic=ridge_model, R=n_folds)
ridge_analysis
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = train, statistic = ridge_model, R = n_folds)
##
##
## Bootstrap Statistics :
##          original        bias    std. error
## t1*  204.75448708 -47.180166206 75.93584826
## t2*    2.00544642   0.066346392  0.16231954
## t3*   -1.46188684  -0.517810286  0.99522952
## t4*    4.48749227   0.770108766  1.11916135
## t5*    1.75863563  -0.202520561  0.64865648
## t6*   -1.55438265  -0.450945925  0.81887569
## t7*    7.99743029   1.827681799  2.42578419
## t8*    2.63780976   0.132505761  0.44744479
## t9*   -2.42979659  -0.448454152  0.74898616
## t10*   0.15373980  -0.102291067  0.24272977
## t11*  -3.05829001  -0.392968349  0.78573423
## t12*   0.09078259   0.042260439  0.08881592
## t13*   1.02885036   0.184962805  0.27182980
## t14*  -0.03590807   0.015312243  0.03340812
## t15*  -1.02419997   0.114502533  0.51567159
## t16*  -1.65842615  -0.123084969  0.22677353
## t17*   1.60150785   0.339658242  0.97196635
## t18*  -2.37614580  -0.307911909  0.41042068
## t19*  -3.36652802  -0.605469111  1.05642595
## t20*  -0.99761106   0.009058519  0.24727054
## t21*  -0.05891120  -0.158433378  0.17577776
## t22*  -0.19264949   0.049319494  0.08914405
## t23*  -0.37706342   0.038889459  0.07320379
## t24*   1.54077634   0.032411725  0.33897571
## t25*   0.10365075   0.015860008  0.05549716
## t26*  -0.74223433  -0.067179145  0.13379192
## t27*  -0.03469326  -0.002831935  0.02192534
## t28*  -0.03464462  -0.002797371  0.02192072
```

```r
lasso_model <- function(data, indices) {
  samp <- data[indices, ]
  model <- cv.glmnet(data.matrix(samp[, -1]), data.matrix(samp[, 1]), alpha=1, type.measure="mse")
  return(as.double(coef(model)))
}

lasso_analysis <- boot(data=train, statistic=lasso_model, R=n_folds)
lasso_analysis
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = train, statistic = lasso_model, R = n_folds)
##
##
## Bootstrap Statistics :
##          original        bias     std. error
## t1*   83.80330002 28.0616978636 67.340143157
## t2*    2.14502497  0.0197990987  0.117636622
## t3*    0.00000000 -0.7082977670  1.236191525
## t4*    9.38861781 -0.0931275269  1.420435348
## t5*    0.00000000 -0.6979957138  1.440431183
## t6*   -4.47588357 -0.3089570793  1.487173192
## t7*   13.59330939 -0.1283383128  3.943544543
## t8*    0.01454651  0.7937874378  0.855251571
## t9*   -3.46773943  0.4641599507  1.391760684
## t10*   0.00000000  0.0000000000  0.000000000
## t11*  -2.40645503  0.2750883949  1.395653849
## t12*   0.00000000  0.0258442112  0.048595789
## t13*   0.00000000  0.1440851803  0.298893923
## t14*   0.00000000  0.0000000000  0.000000000
## t15*   0.00000000  0.0000000000  0.000000000
## t16*  -1.70564985 -0.1116354257  0.457206384
## t17*   0.00000000  0.5796927620  1.429581144
## t18*  -3.59345308 -0.0026728810  0.533804361
## t19*  -7.43839318 -0.1933105411  2.759302439
## t20*   0.00000000 -0.0600351094  0.185753342
## t21*   0.00000000  0.0000000000  0.000000000
## t22*   0.00000000 -0.0435239935  0.080812928
## t23*  -0.25979179  0.0262505668  0.129615191
## t24*   1.36123038 -0.0397017613  0.414056121
## t25*   0.01206913  0.0220949003  0.051667048
## t26*   0.00000000 -0.0143630650  0.064211476
## t27*   0.00000000 -0.0121930142  0.027229034
## t28*   0.00000000 -0.0002690298  0.001127571
```

```r
# subset_model <- function(dat, indices) {
#   samp <- dat[indices, ]
#   tmp_model <- regsubsets(Appliances ~ ., data = samp, method = "exhaustive", nvmax = NULL)
#   model_summary <- summary(tmp_model)
#   return(tidy(coef(tmp_model, which.min(model_summary$bic))))
# }
```

```r
# subset_analysis <- boot(data=train, statistic=subset_model, R=n_folds)
# subset_analysis

comparison <- cbind(tidy(ols_analysis$t0)[2], tidy(ridge_analysis$t0), tidy(lasso_analysis$t0))
names(comparison) <- c("OLS", "Ridge", "Lasso")
comparison
```

```
##              OLS        Ridge       Lasso
## 1   6.146432e-17 204.75448708 83.80330002
## 2   1.571041e-01   2.00544642  2.14502497
## 3   1.294219e-02  -1.46188684  0.00000000
## 4   6.242682e-01   4.48749227  9.38861781
## 5  -4.337681e-01   1.75863563  0.00000000
## 6  -5.806006e-01  -1.55438265 -4.47588357
## 7   5.267321e-01   7.99743029 13.59330939
## 8   1.598554e-01   2.63780976  0.01454651
## 9  -5.156585e-02  -2.42979659 -3.46773943
## 10  9.945737e-03   0.15373980  0.00000000
## 11 -3.364720e-02  -3.05829001 -2.40645503
## 12  1.867616e-02   0.09078259  0.00000000
## 13  4.430040e-01   1.02885036  0.00000000
## 14  7.324255e-02  -0.03590807  0.00000000
## 15  5.655642e-02  -1.02419997  0.00000000
## 16 -8.192102e-02  -1.65842615 -1.70564985
## 17  1.613332e-01   1.60150785  0.00000000
## 18 -2.473989e-01  -2.37614580 -3.59345308
## 19 -3.407820e-01  -3.36652802 -7.43839318
## 20 -2.959637e-02  -0.99761106  0.00000000
## 21 -4.678224e-01  -0.05891120  0.00000000
## 22  1.020234e-02  -0.19264949  0.00000000
## 23 -1.027357e-01  -0.37706342 -0.25979179
## 24  4.265749e-02   1.54077634  1.36123038
## 25  1.982358e-02   0.10365075  0.01206913
## 26  1.390444e-01  -0.74223433  0.00000000
## 27 -1.045662e-02  -0.03469326  0.00000000
## 28            NA  -0.03464462  0.00000000
```