Ridge, LASSO and Smoothing

2023-03-02

Task 1

For this task we use the `mogavs` dataset on communities and crime:

```
library(mogavs)
data("crimeData")
```

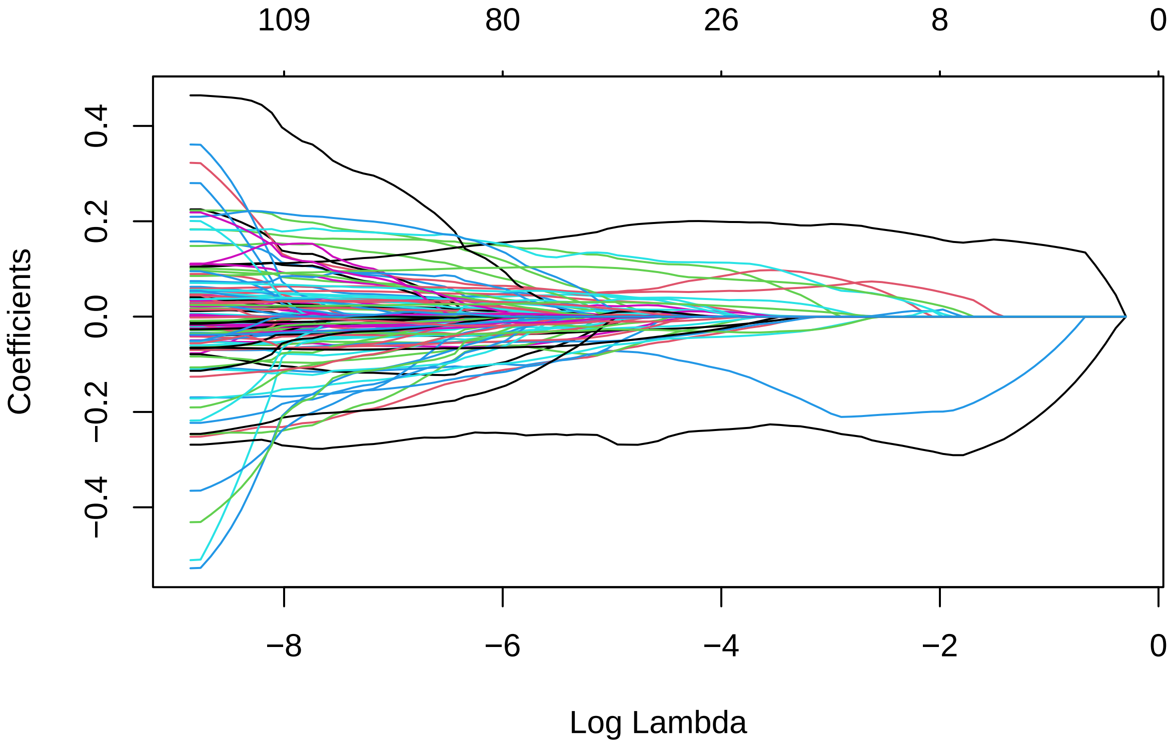
We normalise and divide our data into training and testing sets:

```
crimeData <- scale(crimeData)

train <- sample(1:nrow(crimeData), 1500)
X_train <- crimeData[train, -123]
y_train <- crimeData[train, 123]

X_test <- crimeData[-train, -123]
y_test <- crimeData[-train, 123]
```

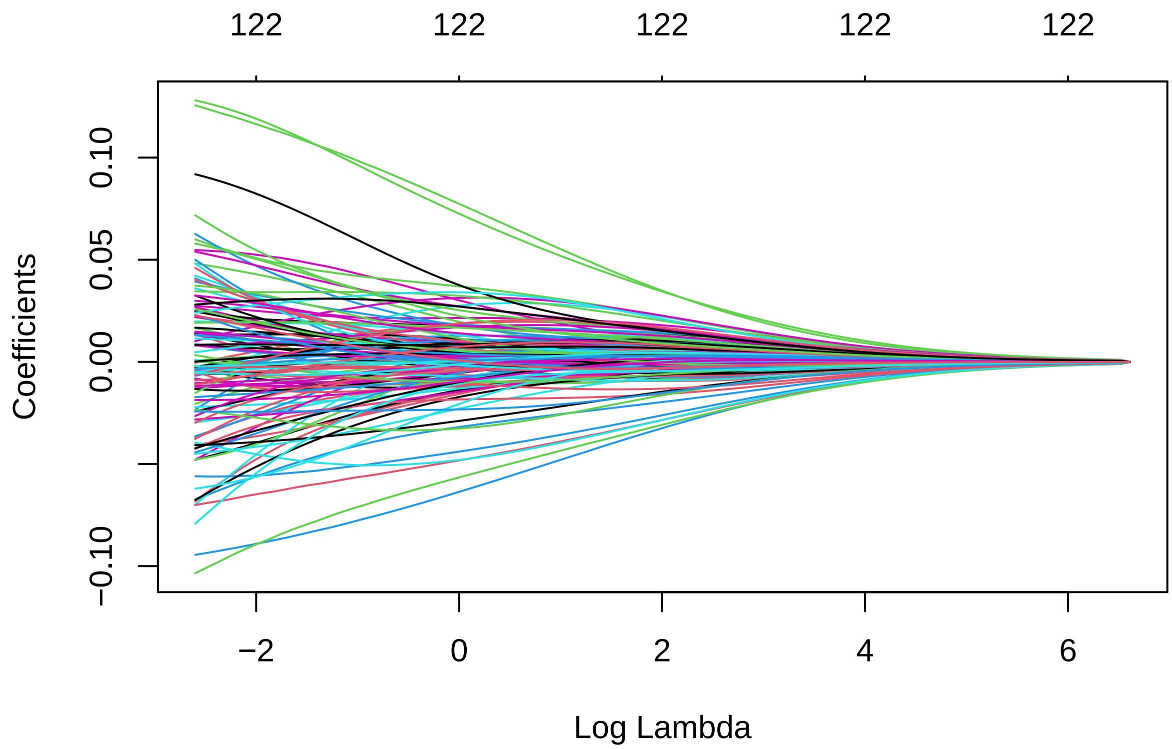
Plotting the Paths

We use the `glmnet` package to perform LASSO, setting the `alpha` parameter to be 1 and ridge regression, by using `alpha = 0`. This gives us the following path visualisations

```
library(glmnet)

lasso_fit <- glmnet(X_train, y_train, family = "gaussian", alpha = 1)
plot(lasso_fit, xvar = "lambda")
```

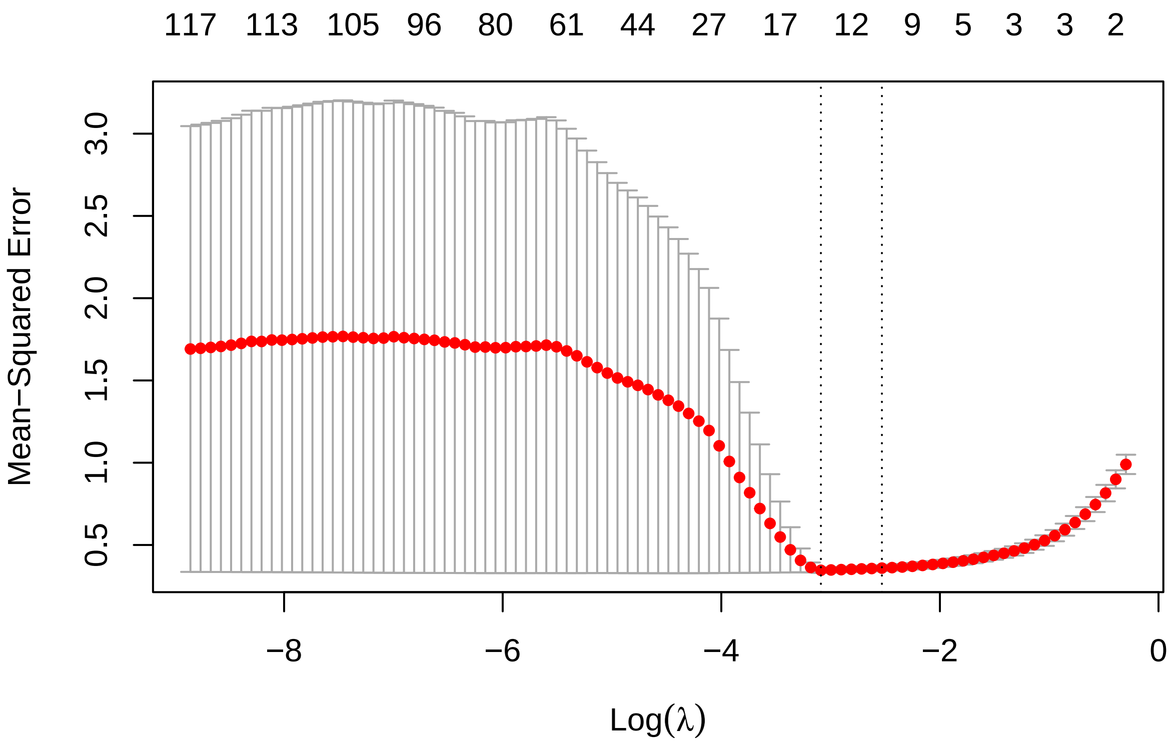



```
ridge_fit <- glmnet(X_train, y_train, family = "gaussian", alpha = 0)
plot(ridge_fit, xvar = "lambda")
```

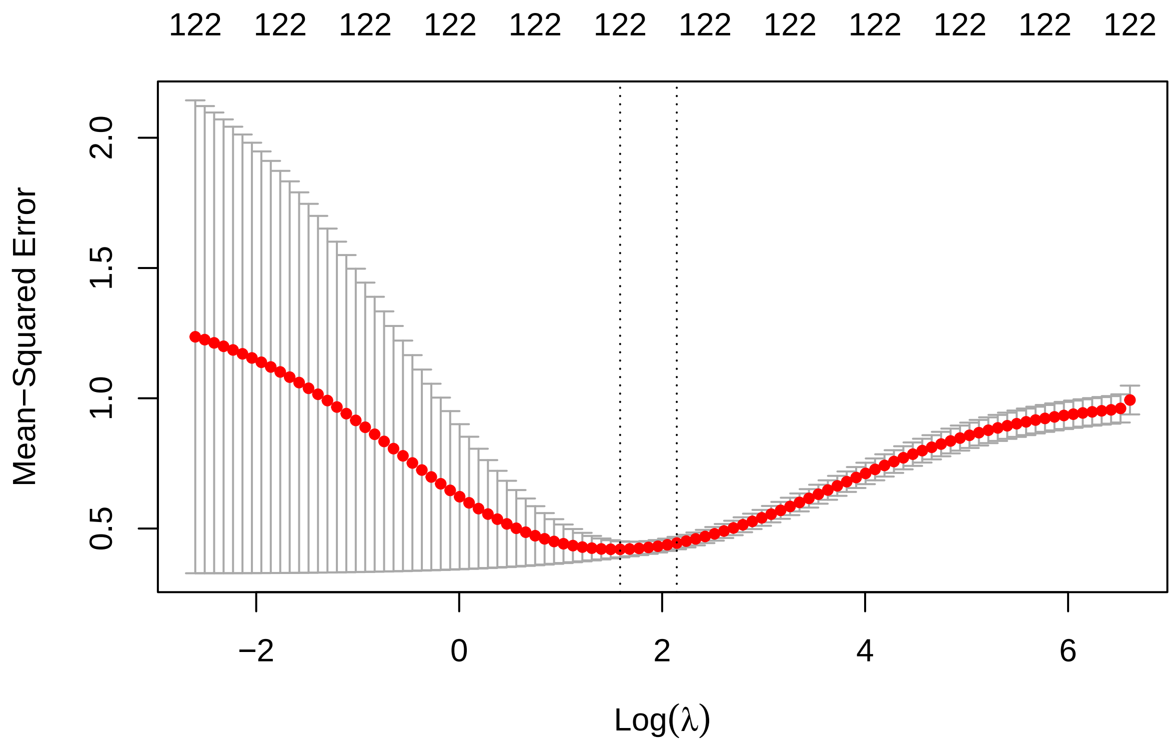


We use the `cv.glmnet` function to perform cross validation on λ. We set the measure to be `"mse"` and stay with the default number of folds which is 10:

```
lasso_cv <- cv.glmnet(X_train, y_train, type.measure = "default", alpha =1)
ridge_cv <- cv.glmnet(X_train, y_train, type.measure = "default", alpha =0)
plot(lasso_cv)
```




```
plot(ridge_cv)
```




```r
coef(lasso_cv)
```

```
## 123 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept) -0.0005964687
## x.V6         .
## x.V7         .
## x.V8         .
## x.V9        -0.2057830097
## x.V10        .
## x.V11        .
## x.V12        .
## x.V13        .
## x.V14        .
## x.V15        .
## x.V16        .
## x.V17        .
## x.V18        .
## x.V19        .
## x.V20        .
## x.V21        .
## x.V22        .
## x.V23        .
## x.V24        .
## x.V25        .
```

```
## x.V26          .
## x.V27          .
## x.V28          .
## x.V29          .
## x.V30          .
## x.V31          .
## x.V32          .
## x.V33          .
## x.V34          .
## x.V35          .
## x.V36          .
## x.V37          .
## x.V38          .
## x.V39          .
## x.V40          .
## x.V41          .
## x.V42          .
## x.V43          .
## x.V44          0.0533908716
## x.V45          .
## x.V46          0.0038587431
## x.V47          .
## x.V48          .
## x.V49          .
## x.V50         -0.2636269534
## x.V51          .
## x.V52          .
## x.V53          .
## x.V54          .
## x.V55          .
## x.V56          0.1828625478
## x.V57          .
## x.V58          .
## x.V59          .
## x.V60          .
## x.V61          .
## x.V62          .
## x.V63          .
## x.V64          .
## x.V65          .
## x.V66          .
## x.V67          .
## x.V68          .
## x.V69          .
## x.V70          .
## x.V71          .
## x.V72          .
## x.V73          .
## x.V74          0.0455770774
## x.V75          .
## x.V76          .
## x.V77          0.0723462149
## x.V78          .
## x.V79          .
```

```
## x.V80         .
## x.V81         .
## x.V82         .
## x.V83         .
## x.V84         .
## x.V85         .
## x.V86         .
## x.V87         .
## x.V88         .
## x.V89         .
## x.V90         .
## x.V91         .
## x.V92         .
## x.V93         .
## x.V94         .
## x.V95         .
## x.V96         0.0450718488
## x.V97         .
## x.V98         .
## x.V99         .
## x.V100        .
## x.V101        .
## x.V102        .
## x.V103        .
## x.V104        .
## x.V105        .
## x.V106        .
## x.V107        .
## x.V108        .
## x.V109        .
## x.V110        .
## x.V111        .
## x.V112        .
## x.V113        .
## x.V114        .
## x.V115        .
## x.V116        .
## x.V117        .
## x.V118        .
## x.V119        .
## x.V120        .
## x.V121        .
## x.V122        .
## x.V123        .
## x.V124        .
## x.V125        .
## x.V126        .
## x.V127        .
```

```
coef(ridge_cv)
```

```
## 123 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -4.665502e-03
```

```
## x.V6          1.097762e-02
## x.V7         -1.724367e-03
## x.V8          3.226239e-02
## x.V9         -3.037919e-02
## x.V10         1.199880e-03
## x.V11         7.354398e-03
## x.V12        -3.713125e-03
## x.V13        -9.969908e-04
## x.V14        -2.410645e-03
## x.V15         2.723730e-03
## x.V16         1.145061e-02
## x.V17         9.365107e-03
## x.V18        -8.327985e-03
## x.V19        -8.551916e-03
## x.V20        -8.002734e-03
## x.V21        -1.880041e-02
## x.V22         2.937739e-03
## x.V23         1.732528e-02
## x.V24        -4.053631e-03
## x.V25        -9.384827e-03
## x.V26        -5.266277e-03
## x.V27         1.478325e-03
## x.V28        -6.557919e-03
## x.V29        -4.793018e-04
## x.V30         3.021269e-04
## x.V31        -2.034591e-03
## x.V32         5.772151e-05
## x.V33         1.236608e-02
## x.V34         1.286359e-02
## x.V35         9.066156e-03
## x.V36         1.319794e-02
## x.V37        -7.054643e-03
## x.V38         1.209547e-02
## x.V39        -7.461867e-03
## x.V40        -6.267937e-03
## x.V41        -3.861715e-03
## x.V42         3.810360e-03
## x.V43        -6.878622e-03
## x.V44         2.102164e-02
## x.V45         9.483465e-03
## x.V46         2.117715e-02
## x.V47         2.144981e-02
## x.V48         4.599732e-03
## x.V49        -2.691873e-02
## x.V50        -2.902845e-02
## x.V51        -2.471635e-02
## x.V52        -2.680581e-02
## x.V53        -9.255791e-04
## x.V54        -5.914669e-03
## x.V55         1.563141e-02
## x.V56         3.217789e-02
## x.V57         6.730869e-03
## x.V58         3.541310e-03
## x.V59         4.157258e-03
```

```
## x.V60        5.817128e-03
## x.V61        7.856269e-03
## x.V62        6.002165e-03
## x.V63        6.740407e-03
## x.V64        7.482298e-03
## x.V65        8.189002e-03
## x.V66       -5.818047e-03
## x.V67        6.710048e-03
## x.V68        1.257180e-02
## x.V69        9.445274e-03
## x.V70       -5.330157e-04
## x.V71       -4.394542e-03
## x.V72        7.774767e-03
## x.V73       -1.532677e-02
## x.V74        1.440931e-02
## x.V75        1.375085e-02
## x.V76       -9.084995e-03
## x.V77        1.554929e-02
## x.V78       -1.362537e-02
## x.V79       -1.285207e-02
## x.V80        1.905128e-02
## x.V81       -6.051694e-04
## x.V82        2.083504e-04
## x.V83        1.461343e-02
## x.V84        1.011105e-02
## x.V85       -2.930713e-03
## x.V86       -1.730061e-03
## x.V87       -6.587337e-04
## x.V88       -4.579885e-03
## x.V89       -2.728560e-03
## x.V90       -2.069518e-03
## x.V91       -1.893974e-03
## x.V92        1.078908e-02
## x.V93        6.552006e-03
## x.V94       -3.983191e-04
## x.V95        1.222652e-02
## x.V96        1.447437e-02
## x.V97        6.869012e-03
## x.V98       -5.616945e-03
## x.V99       -3.112732e-03
## x.V100       3.846757e-03
## x.V101      -1.088403e-03
## x.V102       1.697002e-03
## x.V103      -2.916569e-03
## x.V104      -7.868916e-04
## x.V105      -2.542820e-03
## x.V106       5.136841e-03
## x.V107       1.581455e-03
## x.V108       6.387748e-03
## x.V109      -2.917573e-03
## x.V110      -1.510554e-02
## x.V111      -1.428430e-02
## x.V112       1.962641e-02
## x.V113       4.671136e-03
```

```
## x.V114       3.654369e-03
## x.V115       1.605888e-02
## x.V116       4.038269e-03
## x.V117       2.662755e-03
## x.V118       4.408756e-03
## x.V119       8.134426e-03
## x.V120       8.929409e-03
## x.V121       7.916969e-03
## x.V122       8.558489e-03
## x.V123       3.073143e-03
## x.V124      -4.399602e-03
## x.V125       5.519635e-05
## x.V126       1.435120e-02
## x.V127      -2.893454e-03
```

We now use the out-of-sample error obtained from the `predict()` function to compare the ridge regression and LASSO regression we obtained.

```
lasso_residuals <- y_test - predict(lasso_cv, X_test)
ridge_residuals <- y_test - predict(ridge_cv, X_test)

c(sum(lasso_residuals^2), sum(ridge_residuals^2))
```

```
## [1] 207.9464 246.7404
```