Lasso and Ridge First look

Daniel Martin-Williams

2023-12-28

Load Preliminaries, standardise and format data

```
library(tictoc) # Load tictoc to enable timing runtime
library(boot) # Load boot to get cv.glm
library(glmnet)

bbdat <- read.csv('bigbikedat.csv')

#Standardize the regressors
bbdat[,2:ncol(bbdat)] <- scale(bbdat[,2:ncol(bbdat)])

#Turn bbdat into matrix for glmnet function
bbdat.mat <- as.matrix(bbdat)

#Define X and Y variables
y <- log(bbdat.mat[,1])
x <- bbdat.mat[,2:ncol(bbdat.mat)]
```

Run regressions

Full, lasso and ridge regressions all run for comparison.

```
#Run full regression for comparison
fullbbdat <- glm(log(cnt) ~ ., data = bbdat)
cv.fullbbdat <- cv.glm(bbdat, fullbbdat, K = 10)
```

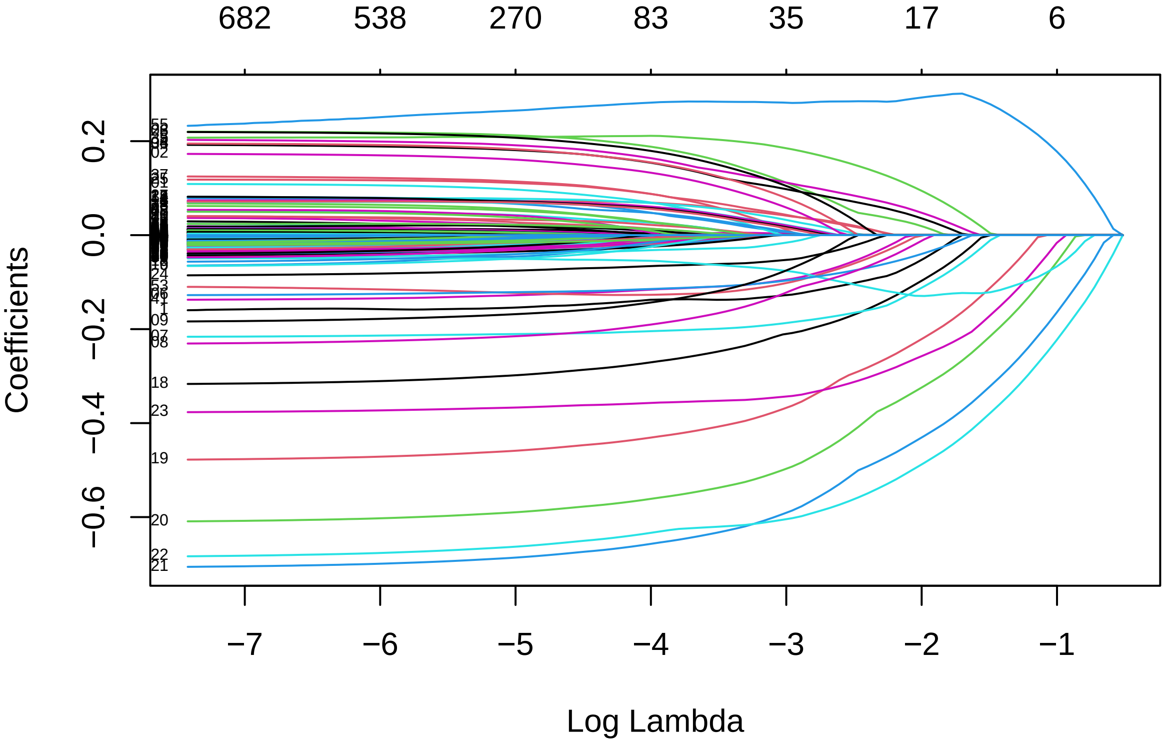
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
bbdatlasso.path <- glmnet(x, y, alpha = 1, lambda.min.ratio = 1/1000, nlambda = 100)
bbdatridge.path <- glmnet(x, y, alpha = 0, lambda.min.ratio = 1/1000, nlambda = 100)
```

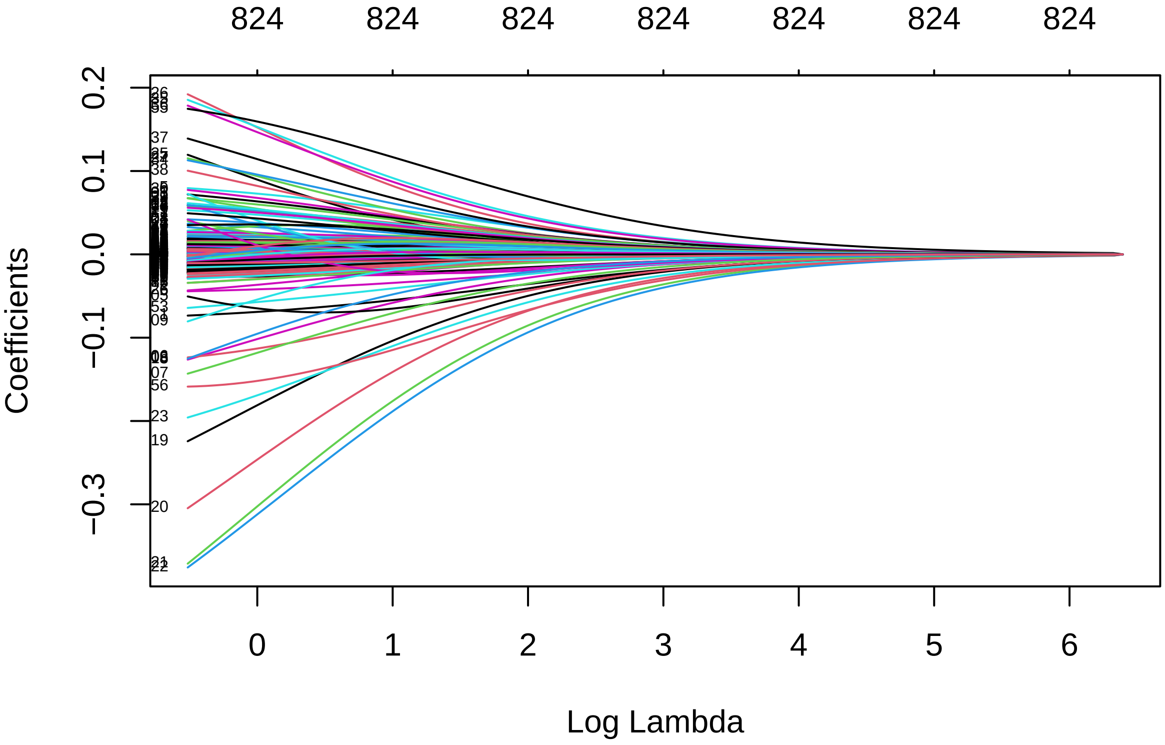
Create lasso regularization path for log(cnt). In order to chose the penalty weights, we decide on a ratio, lambdamin/lambdamax to input into our function. Here lambdamax is the value of lambda at which our function gives us all coefficients to be = 0 for lasso (close to 0 for ridge), and our lambdamin is the value of lambda that gives us the desired ratio, lambdamin/lambdamax. while glmnet does this for us generally, here we get better results chosing manually

Plot regularisation paths

```r
plot(bbdatlasso.path, xvar = "lambda", label = TRUE)
```



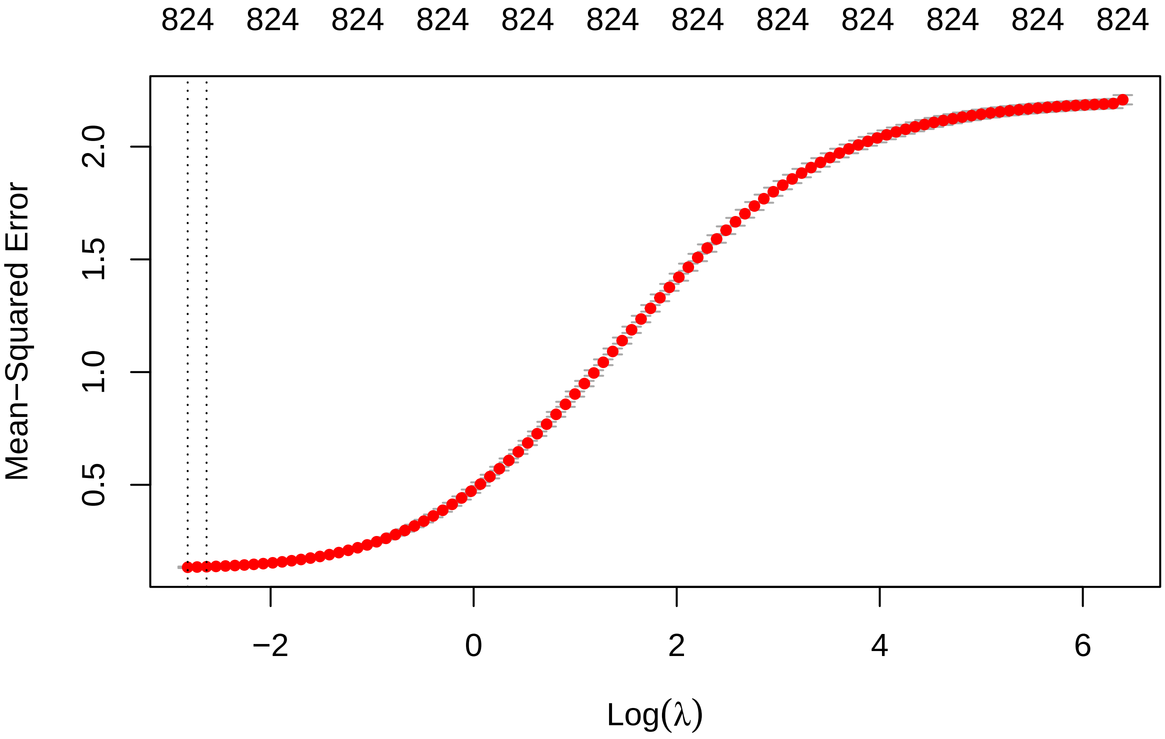
```r
plot(bbdatridge.path, xvar = "lambda", label = TRUE)
```



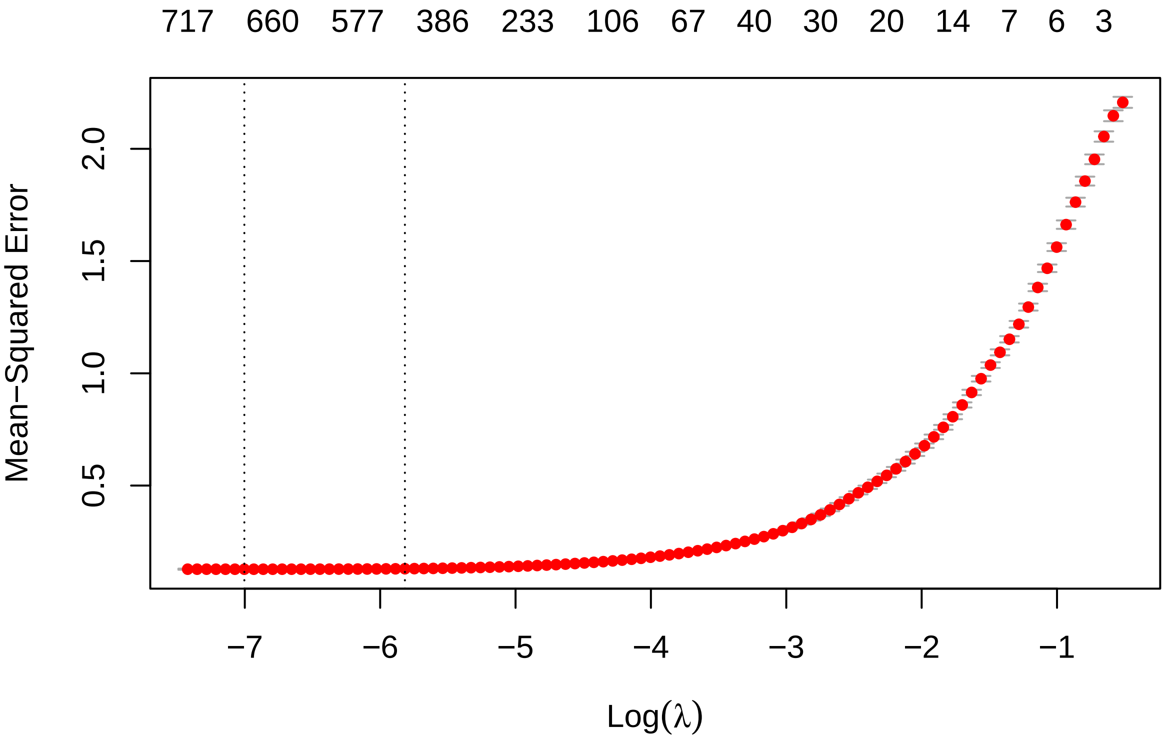
Lasso and ridge both shrink coefficients as penalty weight increases, as expected lasso auto selects relevant coefficients, ridge doesnt actually shrink any coefs to 0, number of coefs doesnt reduced. (due to cost function in each model,|B| compared to Bˆ2)

#Plot cross validated regularisation paths

```
set.seed(1)
cv.bbdatlasso.path <- cv.glmnet(x, y, alpha = 1, lambda.min.ratio = 10^(-3))
cv.bbdatridge.path <- cv.glmnet(x, y, alpha = 0)
plot(cv.bbdatridge.path)
```



```
plot(cv.bbdatlasso.path)
```



Compare SE and optimal lambdas of each model

```
OOSMSE <- matrix(,2,5)
colnames(OOSMSE) <- c("Full","Lasso, best lambda","Lasso, cons lambda", "ridge,
                      best lambda", "ridge, cons lambda")
rownames(OOSMSE) <- c("Oos MSE", "lambda")
OOSMSE[1,1] <- cv.fullbbdat$delta[1]
OOSMSE[1,2] <- cv.bbdatlasso.path$cvm[cv.bbdatlasso.path$lambda == cv.bbdatlasso.path$lambda.min]
OOSMSE[1,3] <- cv.bbdatlasso.path$cvm[cv.bbdatlasso.path$lambda == cv.bbdatlasso.path$lambda.1se]
OOSMSE[1,4] <- cv.bbdatridge.path$cvm[cv.bbdatridge.path$lambda == cv.bbdatridge.path$lambda.min]
OOSMSE[1,5] <- cv.bbdatridge.path$cvm[cv.bbdatridge.path$lambda == cv.bbdatridge.path$lambda.min]
OOSMSE[2,2] <- cv.bbdatlasso.path$lambda.min
OOSMSE
```

```
##              Full Lasso, best lambda Lasso, cons lambda
## Oos MSE 0.1282191       0.1273722606          0.1294926
## lambda         NA       0.0009088529                 NA
##         ridge,\n                      best lambda ridge, cons lambda
## Oos MSE                                 0.1342817          0.1342817
## lambda                                         NA                 NA
```