ASM - Ridge Regression

Sergi Carol, Laura Cebollero, Alex Rodriguez

8th November, 2018

Some notes

All the code related to this report and delivery can be found on the files:

- `parameter_lambda.R`. Has the code for the 1st task: choosing the λ penalization parameter.
- `all_functions.R`. Contains all the abstracted functions used to compute the MSPE with different methods.
- `boston.R`. Has the code related to the 2nd task: the Boston Housing dataset Ridge Regression study. Uses the functions on `all_functions.R`.
- `ASM_report_RR.Rmd`. The RMarkdown that has produced this .pdf formatted report.

Choosing the penalization parameter

As asked, we have written two functions to choose an adequate parameter λ for the Ridge Regression.

The implementation of such functions can be found at `parameter_lambda.R` with its correspondent comments inline to understand how the code works.

We use 25 different λ that range from 0 to 100.000.

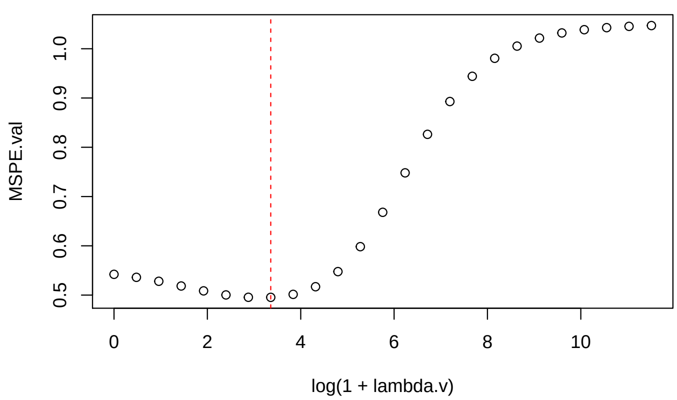
Choosing λ with MPSE and a validations set

The first function implemented is the one that allows us to choose λ using a validation set.

Such validation set has been selected by using the column `train`, which tells us which data can be used for training purposes.

This leaves us with a validation set of 30 observations, and 67 observations as the training sample.

Below is the plotting the resulting MSPE values for each lambda:




```
(minPos = which.min(MSPE.val))
```

```
## [1] 8
```

```
lambda.v[minPos]
```

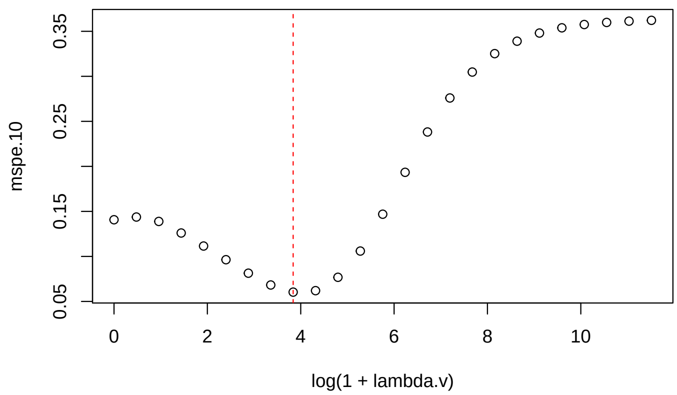
```
## [1] 27.72993
```

We can see that using this method, the λ that gives us the least error is the 8th one, with a value of 27.7.

Choosing λ with MPSE and K-fold cross-validation

We have also done the implementation of k-fold cross-validation. In this case we split the dataset in K differnet parts, then we train the model using K - 1 parts and leave the remaining one to validate the model, thats it, calculating the MSE. Once we have calculated the value of the λ for each fold, we choose the value λ from each iteration of k that gives the less error.

Using a 10-fold CV, we obtain the following plot:




```
(minPos = which.min(mspe.10))
```

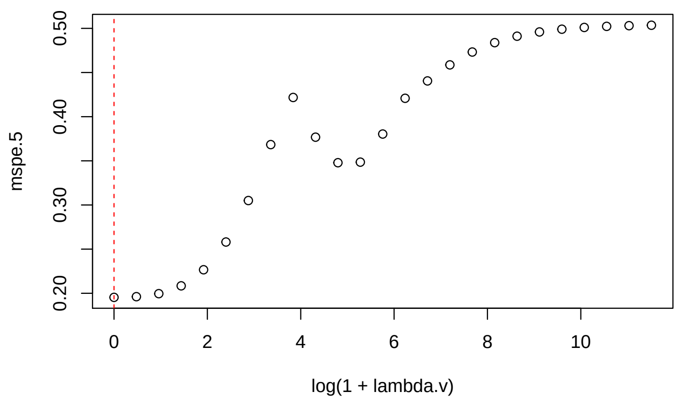
```
## [1] 9
```

```
lambda.v[minPos]
```

```
## [1] 45.41604
```

It seems that in this case, we are selecting the one with $\lambda_9 = 45.41$.

If we use a 5-fold CV:




```
(minPos = which.min(mspe.5))
```

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

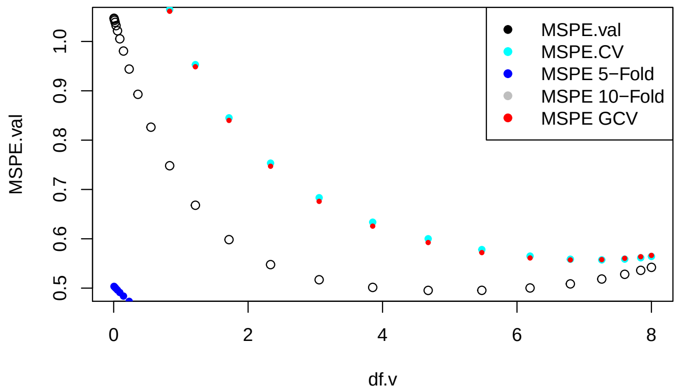
```
lambda.v[minPos]
```

```
## [1] 0
```

Now the selected λ with the least error is the one on the 9th position with $\lambda = 45.41$.

Let's now proceed onto compare those results with those obtained from using the methods of leave-one-out cross-validation and Generalized Cross-Validation.

```
mspe.cv <- MSPEcv(X, Y, n.lambdas, lambda.v, n)
mspe.gcv <- MSPEgcv(X, Y, n.lambdas, lambda.v, beta.path, diag.H.lambda, n)
```



TODO: Interpret plot

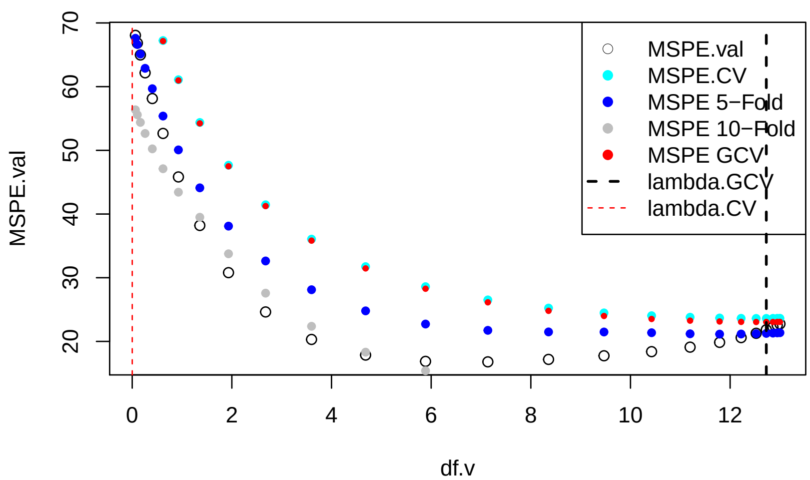
Ridge Regression for Boston Housing Area

Now, we are going to proceed to apply Ridge Regression to the Boston Housing Dataset.

Please note that we have not found uploaded on `El Racó` the corrected dataset. Thus, we have **worked using the dataset provided by the `MASS` library**.

To apply the Ridge Regression method onto this dataset, we have created a matrix X which contains all the 13 explanatory variables. This matrix has been centered and scaled.

We have then created a vector Y which is the response variable. It has been centered but not scaled.



We can see that we get higher values on the MSPE 5-Fold, the lowest ones obtained from MSPE val (that using, using an 80% of the data as training and 20% of the original dataset as a validation set).

In this case, the 10-fold method does not prove as a good method for it declines until reaching 0.

The generalized Cross-validation method as well as the leave one out (MSPE.CV) follow close the MSPE 5-fold evolution and in fact have a greater value when df.v > 9.