

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.

1. 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.

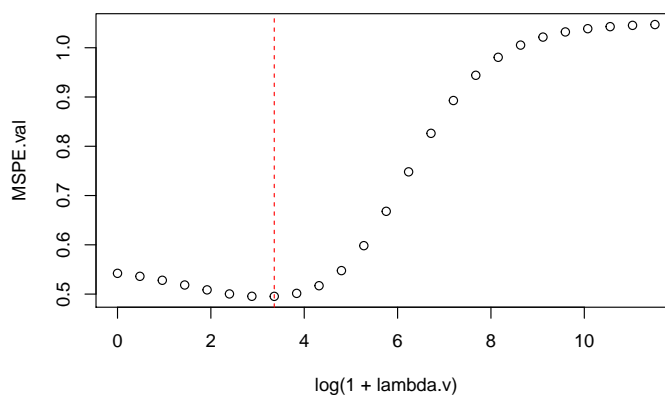
1.1. Choosing λ with MPSE and a validations set

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

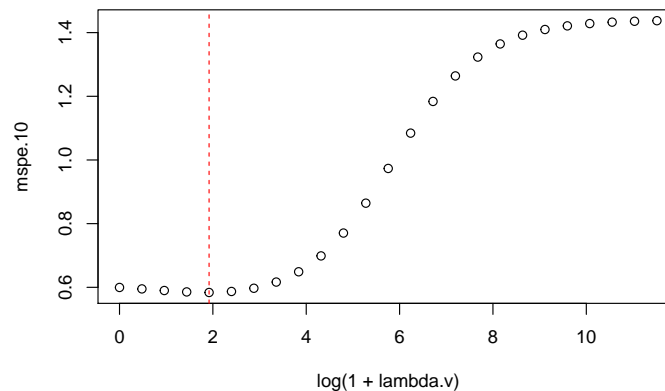
1.2. Choosing λ with MPSE and K-fold cross-validation

We have also done the implementation of k-fold cross-validation. In this case we split the dataset in K different parts, then we train the model using K - 1 parts and leave the remaining one to validate the model. In other words, calculating the MSE.

Once we have calculated the value of the errors for each λ and for each fold, we average the squared error values and return them. From these averaged values we choose the model with the lowest error globally as our MSPE.

10-Fold Cross validation

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



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

```
## [1] 5
```

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

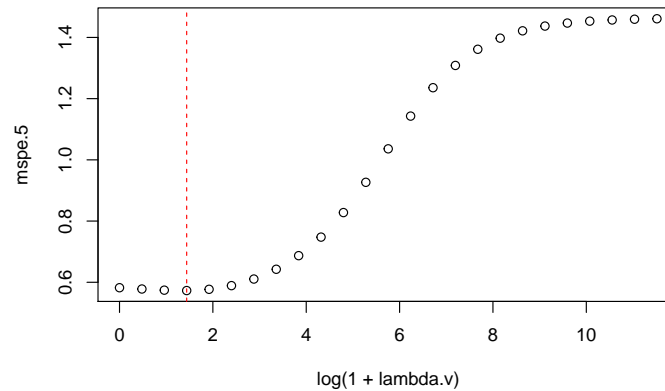
```
## [1] 5.812932
```

It seems that in this case, we are selecting the one with $\lambda_5 = 5.81$.

Which means that since λ is not that large, we are not penalizing that much.

5-Fold Cross validation

Now let's try with a lower value of k , where $k = 5$. Thus, let's try with 5-fold Cross-Validation:



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

```
## [1] 4
```

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

```
## [1] 3.21697
```

Now the selected λ with the least error is the one on the 4th position with $\lambda = 3.21$.

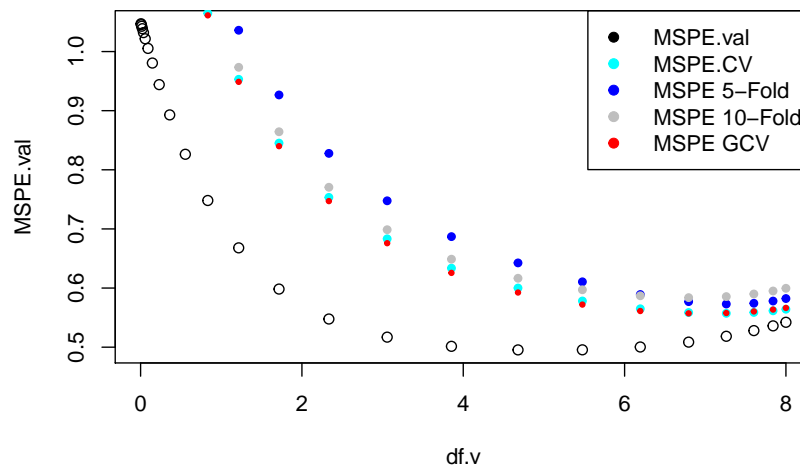
The selected lambda is much lower than that obtained with 10-fold, so it is penalizing even less than before the complexity of the model.

1.3. Comparison with other models

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)
```

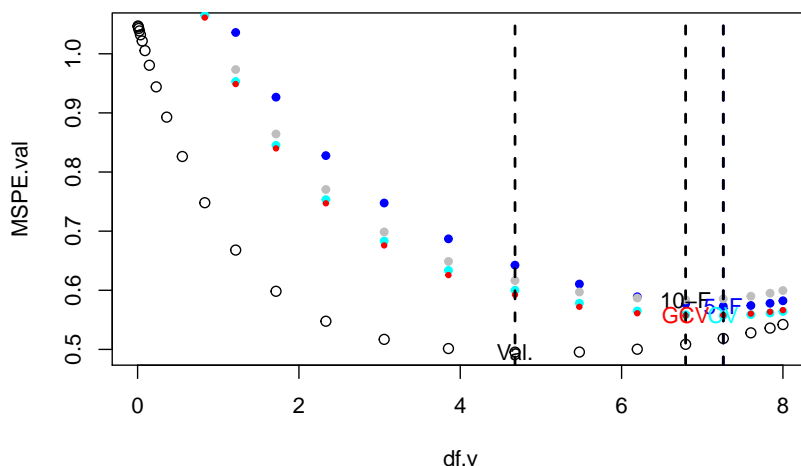
```
mspe.gcv <- MSPEgcv(X, Y, n.lambdas, lambda.v, beta.path, diag.H.lambda, n)
```



From the plot we can observe many things. First and foremost, it is clear how the method of validation is the one that gives the least error for the values of parameters in the effective degrees of freedom.

Yet the number of parameters established by the degrees of freedom is different for the validation ridge regression than for the others.

Let's replot it looking for the optimal amount of df.



Since the text overlaps and there seems to be methods where the df is the same, we are just going to print the adequate degrees of freedom for each method:

MSPE Method	df.v	MSPE
Validation	4.678791	0.4953927
Leave-one-out (CV)	7.260769	0.5572292
5-Fold val.	7.260769	0.5729769
10-Fold val.	6.793982	0.5838061
Generalized C.V.	6.793982	0.5572324

We can see that the Validation approach has the lowest optimal number of degrees of freedom. Meanwhile, Leave-one-out (CV in the plot) and 5-Fold Validation have the same df selected with slightly different obtained MSPE values.

The same happens with 10-fold validation and Generalized C.V., where the df.v chosen is the same with slightly different values for the MSPE obtained.

2. Ridge Regression for Boston Housing Area

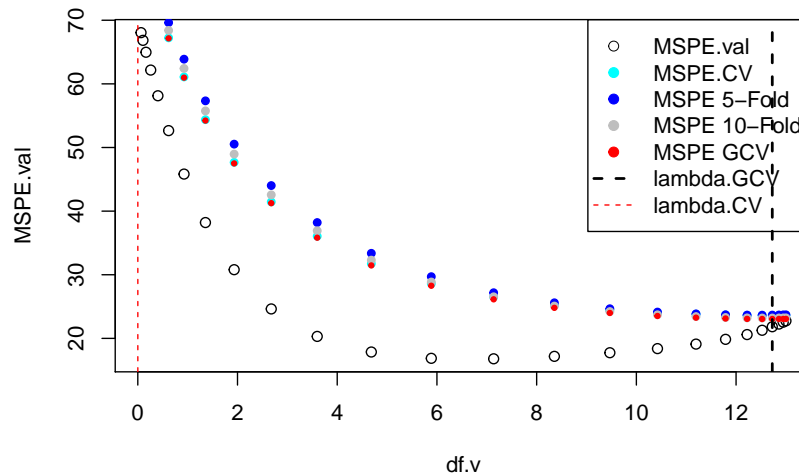
Now, we are going to proceed to apply Ridge Regression to the Boston Housing Dataset.

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 proceed as before: we first plot the comparison and then produce a table to compare results.



MSPE Method	df.v	MSPE
Validation	7.137522	16.79374
Leave-one-out (CV)	12.522692	23.61073
5-Fold val.	12.522692	23.65126
10-Fold val.	12.726176	23.20371
Generalized C.V.	12.726176	23.04939

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 $df.v > 9$. Yet the different methods provide different optimal number for optimal parameters. For all methods besides validation we can see how the optimal number of parameters is 13, while for the validation is 7.