Introduction to Learning and Intelligent Systems - Spring 2015

Martin Ivanov (ivanovma@student.ethz.ch)
Can Tuerk (can.tuerk@juniors.ethz.ch)
Jens Hauser(jhauser@student.ethz.ch)

March 14, 2015

Project 1 : Regression

Problem description

We received datafiles on measurements of passengers using services of the rail authority. The data for this regression task is, beside the response variable as passenger number, structured into seven specific explanatory variables which include a timestamp and six parameters about the weather.

Initial approach

A first intuitive approach was to concentrate only on the variables weekday and hour in the timestamp, because these variables can be seen as the most important explanation for travelling by train, due to commuting to work.

With that in mind we first started doing a research on pairs of the response and every single variable given in the dataset, but soon realised that there exists no clear linear relationship between each of these pairs. Even the predictions from a simple multiple linear regression model were far beyond the easy baseline. These results and inspections guided us to have a closer look at more complex models with different features and feature-transformations in order to be able to find out a plausible relationship in the given data set.

1st Solution: Ridge Regression

After having done our first research on the data to get a better understanding of the problem, we began setting up a ridge regression cross validation model and tried to improve our results in a random trial-and-error style by selecting features and feature-transformations. This approach left us with a best performance of about 0.6 according to the given loss function.

To do things even better and get closer to the hard baseline we decided to set up a greedy forward selection on features and feature-transformations. Therefore we created a big matrix of all possible features and feature-transformations. Technically, this approach worked quite well and the performance according to the given loss function merged to a value within the interval [0.56,0.57] as stated in

the following table and presented in the figure in the appendix. Each line in the table represents the regression loss of adding the new column 'Column-name' to our design-matrix.

VarNumber	Loss	Column-name
1	1.08883149	$hour + A$
2	0.89185466	$hour^5$
3	0.83401695	$4h$
4	0.79707187	$3h$
...		
20	0.58475916	Dec
21	0.58081406	A^3
...		
40	0.56849751	E
41	0.56826396	$A * C * F$
42	0.56813648	$D * E$
...		
49	0.56754459	Thu
50	0.56729329	Aug
...		
99	0.56525676	$hour^3$
100	0.5652391	$log(D)$

Table 1: greedy forward selection results

2nd Solution: Support Vector Regression

After having beaten the easy baseline with the previously described ridge regression approach, we were stuck at the loss somewhere between 0.56 and 0.57, so we tried to find an improved method to get close to or beat the hard baseline. We did find such a method by using the support vector regression. Thanks to the fact that a kernel transforms the data into high-dimensional space, that huge matrix from the ridge regression approach wasn't needed anymore. The best performance we got from this approach was a loss of about 0.38 on the validation set. For this result we used a gaussian kernel with the following hyperparameters found by a grid-search: 'degree' $= 1$, 'epsilon' $= 0.1$, 'C' $= 6$. The design-matrix of our best performance measure was made of the following features:

```
['A','C','E','B1','B2','B3','B4','2013','2014','2015','Jan',
'Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec',
'Mon','Tue','Wed','Thu','Fri','Sat','Sun','hour','hour+A',
'hour+E','hour+F']
```

**

Attached files:
- $greedy\_ridge.py$
- $kernelregression.py$

Appendix

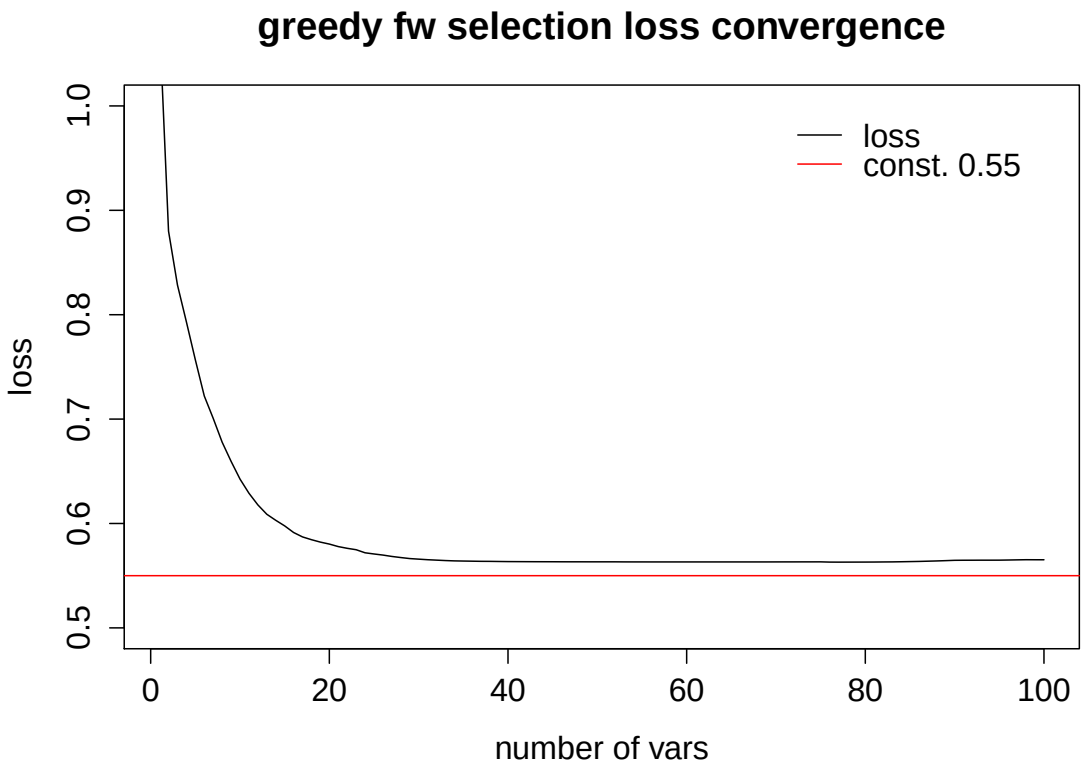

Figure 1: greedy forward selection loss convergence with ridge regression