Preprocessing Stability Experiment

L3P3

Friday, October 17, 2014

1. Introduction

In this document we will check the stability of the data preprocessing method proposed in the elastic net algorithm for the network level. To do so, we ran the model over the four available months of SCF data with 5 minutes of observation and prediction window twenty times. Checking the models' performance we will be able to check if the preprocessing phase affects the behaviour of the models. We assumed that the window span did not affect the final result in terms of stability. First we load the list containing the created models.They were created using the code found in the 'rlr-whole.R' file.

```
## Loading required package: ggplot2
```

2. Stability Check

Now we will run several tests to check the model performance and behaviour is similar in each iteration.

a) Number of Created Models

The first check will test how many models were created in each iteration.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   53.00   53.00   54.00   54.15   55.00   56.00
```

```
## [1] "Standard Deviation:" "0.93330200448673"
```

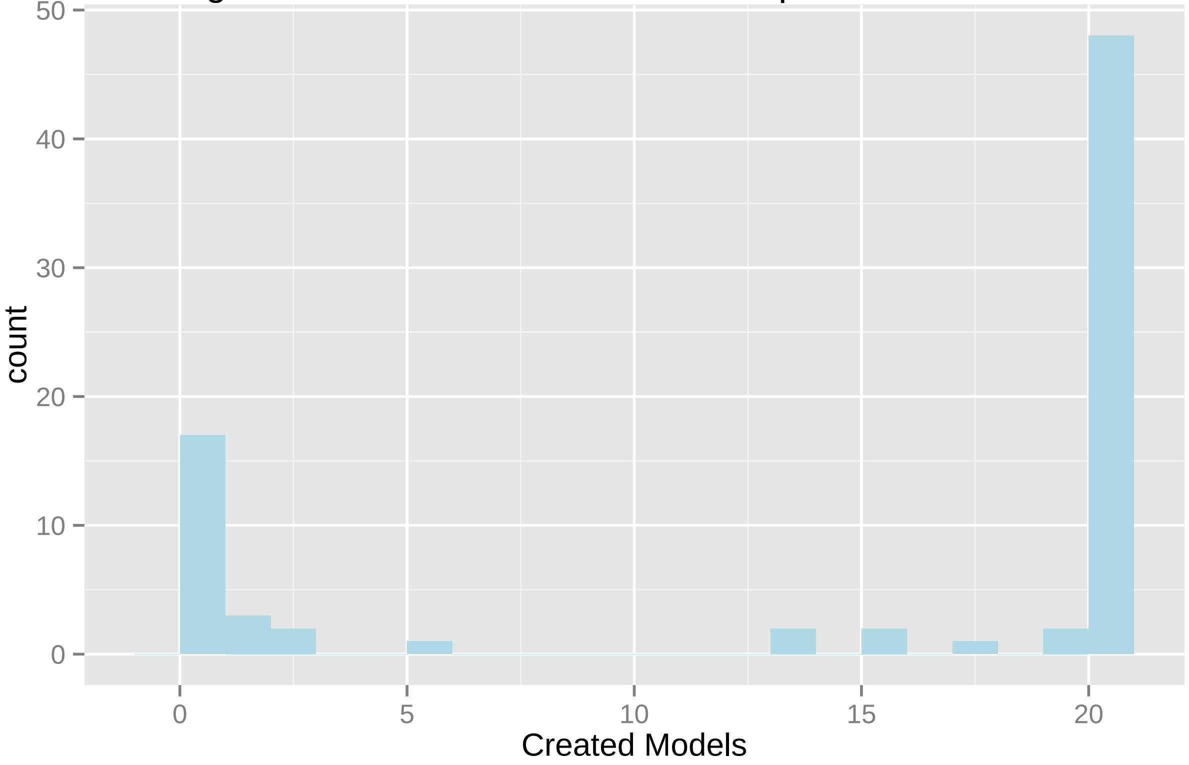
So we see how the maximum difference can be three models up between iterations, a 3.8461538% variation. This, indeed, does not seem like a large rate but it could mean not modelling a critical event. To check exactly which events are affected, on the next section we will study how each individual event was modelled.

b) Individual Events Performance

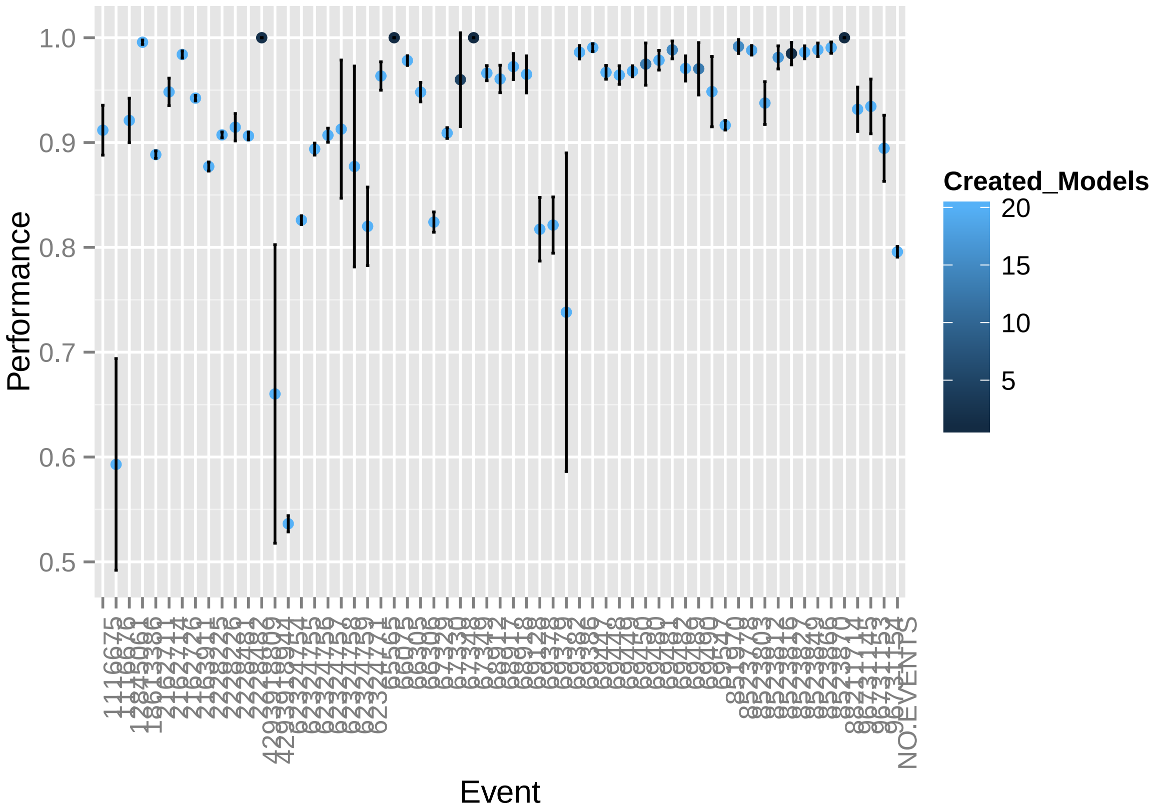
We will now create a large table with every modelled event, each iteration's performance, the average fscore and the standard deviation. This test will give large insights about the effect of the preprocessing phase.

First, we will study the deviation in created models for each event, which can be seen on Fig.(1).

Figure 1. Amount of created models per modelled event



Out of 78 different events, 65 events show a similar pattern in the 20 iterations, and 13 show some discrepancy between iterations. In terms of performance, Fig.(2) shows the average f-score obtained for each event and the standard deviation for each one, whereas the color indicates the amount of models created for each event.



The performance is all right for a lot of models, but shows large deviations for a few events. This is not good. To further understand the problem at hand, we will now plot the standard deviation of the models over the amount of training instances for each case

Again, this shows little to none relation, as further suggested by a low correlation value of -0.2191631.

3. Stability Improvements

To try to improve the obtained results, we are going to change a step in the preprocessing phase. Instead of testing the best model obtained from the validation phase against the test dataset to obtain its performance, we now take the best model, retrain it with the training and validation set together and then test it against the test dataset.The code for these models is found in the 'rlr-whole-preprocessing2.R' file.

2. Stability Check

Now we will run several tests to check the model performance and behaviour is similar in each iteration.

a) Number of Created Models

The first check will test how many models were created in each iteration.

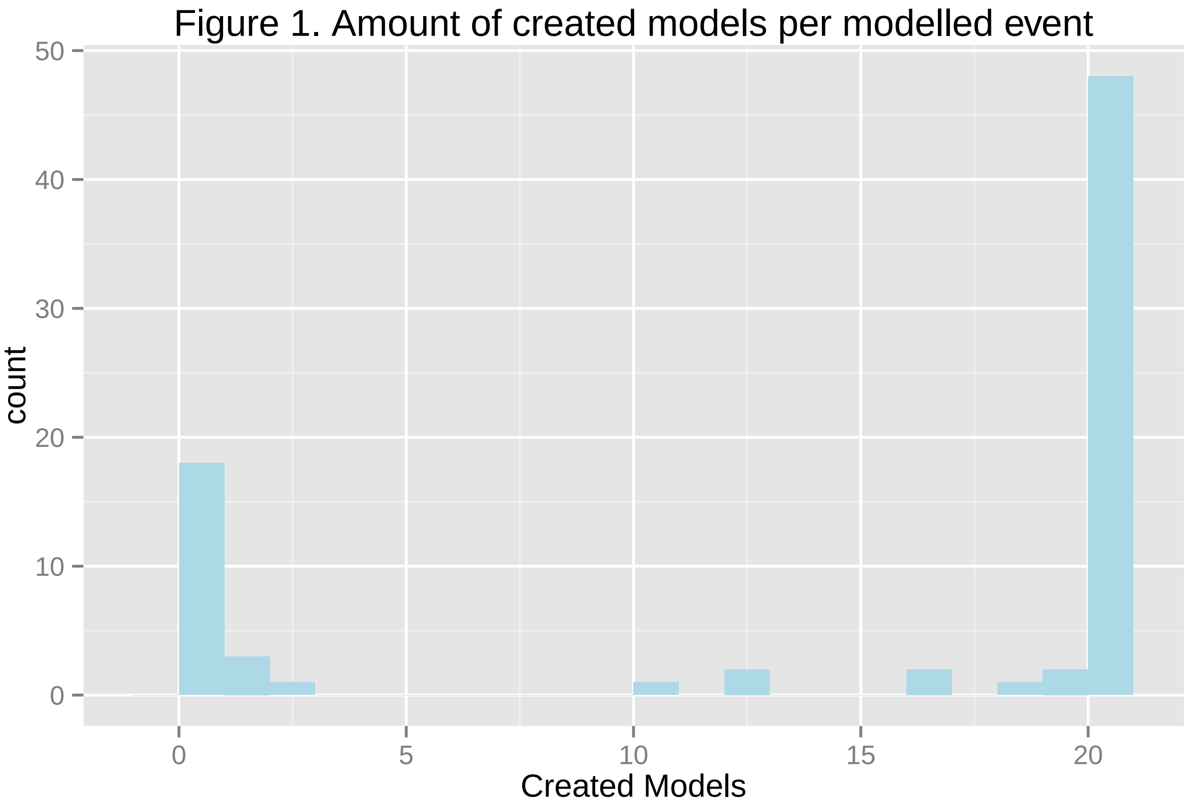
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   52.00   53.75   55.00   54.35   55.00   56.00
## [1] "Standard Deviation:" "1.08942283125661"
```

So we see how the maximum difference can be three models up between iterations, a 3.8461538% variation. This, indeed, does not seem like a large rate but it could mean not modelling a critical event. To check exactly which events are affected, on the next section we will study how each individual event was modelled.

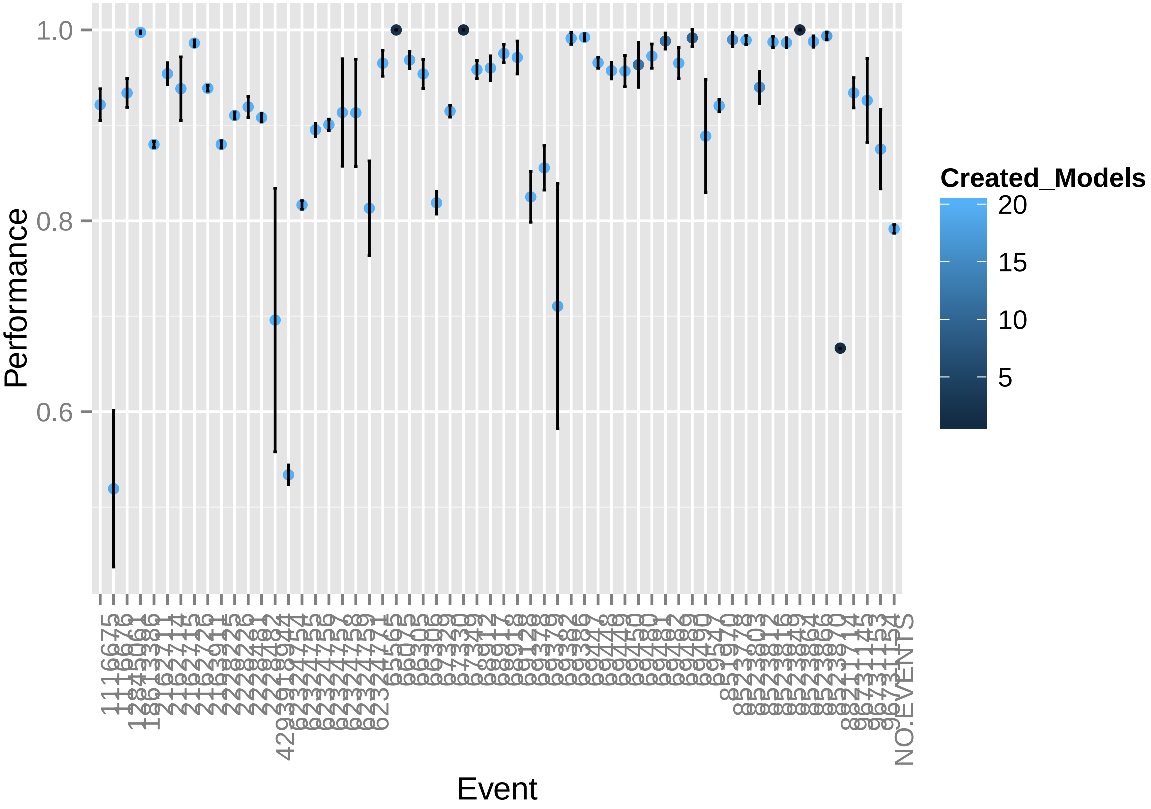
b) Individual Events Performance

We will now create a large table with every modelled event, each iteration's performance, the average fscore and the standard deviation. This test will give large insights about the effect of the preprocessing phase.

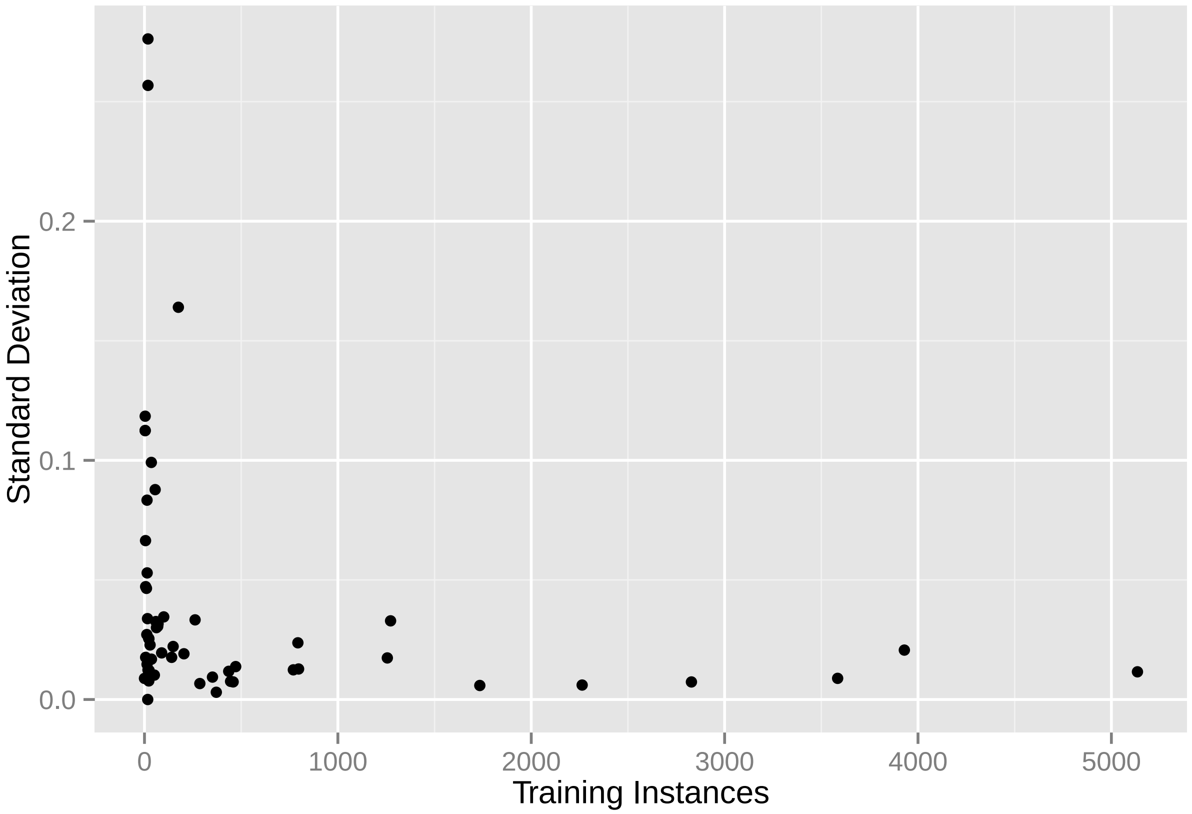
First, we will study the deviation in created models for each event, which can be seen on Fig.(1).



Out of 78 different events, 66 events show a similar pattern in the 20 iterations, and 12 show some discrepancy between iterations. In terms of performance, Fig.(2) shows the average f-score obtained for each event and the standard deviation for each one, whereas the color indicates the amount of models created for each event.



The performance is all right for a lot of models, but shows large deviations for a few events. This is not good. To further understand the problem at hand, we will now plot the standard deviation of the models over the amount of training instances for each case



Again, this shows little to none relation, as further suggested by a low correlation value of -0.2225926.