Kernel Based Machine Learning and Multivariate Modeling

Kalyan S. K.[1] and Garcia-Calderon S. A.[2]

[1]DMKM, Universitat Politecnica de Catalunya, Barcelona

January 2, 2017

Abstract

In our project we explore insurance claims data. We spent most of our time working on building features for our SVR model. Our aim in this project was to explore linear dimensionality reduction techniques. Our workflow consists of stacking the outputs obtained from various transformations followed by bagging several random observations from the training data for training multiple Support Vector Regression(SVR) models.

1 Introduction

The problem approached was to focus in "Allstate Severity Claim" Kaggle competition. The objective is to build a model which is capable to predict the amount of money the insurance company spent on its client, or as it is categorized, *the loss*. For this, Allstate provided two data sets of considerable size, a training and testing set.

The Data that Allstate provided consists in a training set of 188318 individuals containing 132 attributes, and a testing set of 125546 individuals and 131 attributes since we do not have access to the testing response.

2 Computational Complexity

We already know that SVR models do not scale well with increase in number of observation. We have choose to use SVR model and explore scalable approaches. We conducted experiments to decided how many observations we need to use to train our SVR model. As seen from our plot below, we clearly see that we need to choose a sweet spot in such a way that our model generalizes well and computationally does not take a long to train.

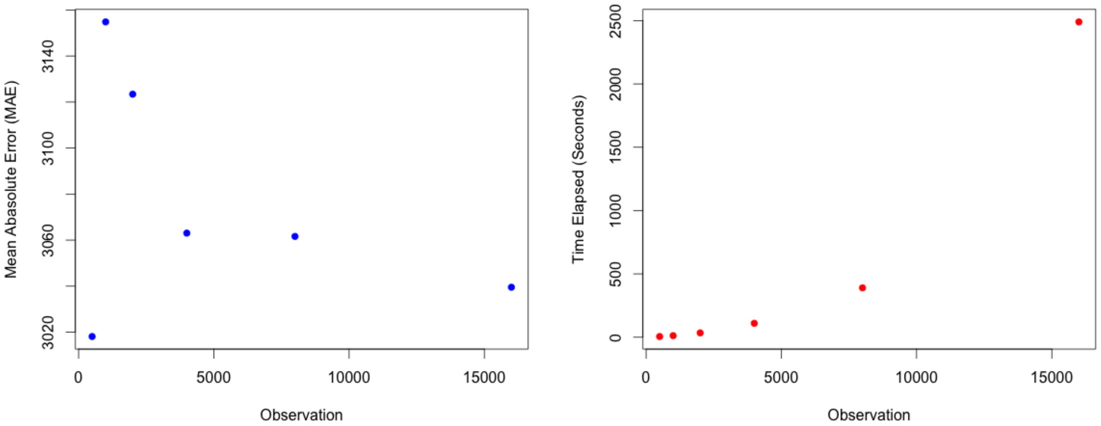

Figure 1: Mean Absolute Training Error/ Computation time vs Data Size

Based on our experiments conducted we will choose to train several SVR models with 3000 observations on only numerical features. The predictions obtained from these models were averaged. The Mean absolute error obtained were based on 5-fold cross validation technique followed.

We also read the paper by Bengio et. all on Mixture of Experts it speaks of a scalable SVM, this is something we would have liked to explore more in the future.

3 Approach

The process to approach this problem is shown in Figure 2: First we will do an Exploratory Analysis to familiarize ourselves with the data. Since we have many variables, we will compute a *Principal Component Analysis* for the continuous variables. A transformation of our target, or response, variable will be made in order to finally build a data set for training our model. Finally We will use *Random Forest* for feature selection and finally we train a *Bagged Support Vector Regression* with a "RBF" Kernel.

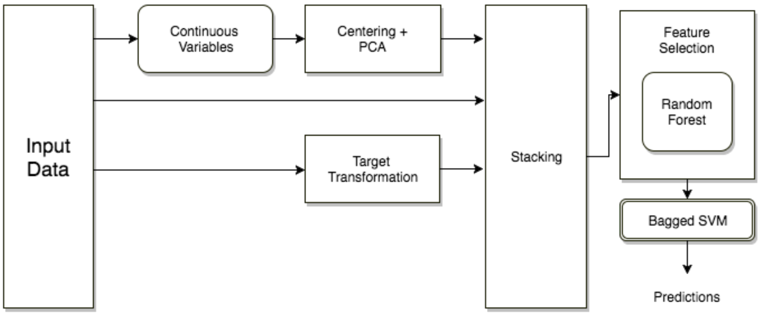

Figure 2: Data Science Workflow.

4 About the Data

Important thing to mention is that we are dealing with a dataset which we are not familiar with. *Allstate Insurance Co.* provided a competition problem and a dataset with 132 attributes, or explanatory variables. These variables are decomposed as shown in the Table 2. Only the company, or owners of the data, are aware of what each variable means, and we are left with the challenge of figuring out how to use them (Anonymous features).

Id	Numerical Columns	Categorical Columns	Response
1	14	116	1

Table 1: The Data

Total modalities observed in our data set for our categorical columns 1102.

4.1 Correlation

We started by computing correlations for continuous variables. As shown in Figure 3, it can be observed that some of our continuous variables are correlated. This means it makes sense to use *PCA* in order to reduce dimensionality and learn latent representations.

4.2 Target Transformation

We plot our response variable, *loss* in order to see which distribution it follows. As it is shown in Figure 4, our response variable seems to be skewed. By applying a Log-Transformation, we are turning our response variable into a normal distribution, which becomes more interpretable and turns out to be helpful at the time of building a model for predictions.

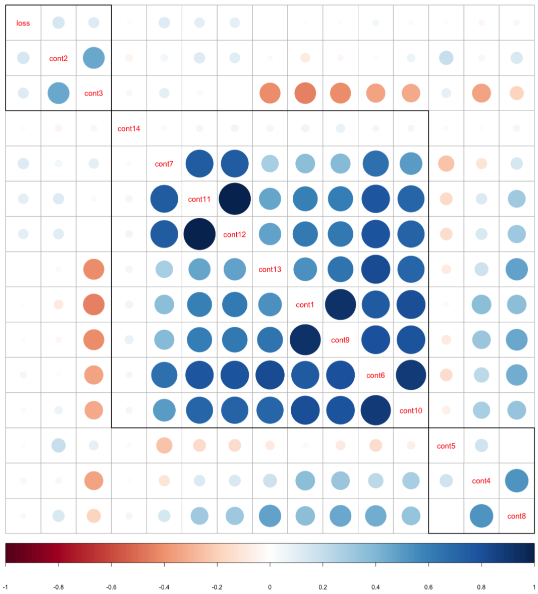

Figure 3: Categorical Correlation with Loss.

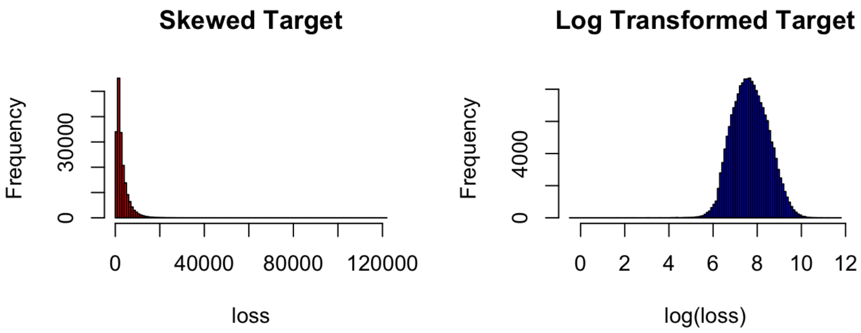

Figure 4: Logarithmic Transformation of our response

4.3 Categorical Variables

We analyzed the categorical variables, we observed that we have more than 1100 modalities, in only 116 attributes. By further exploring, we found that 10 attributes have more than 17 modalities. We plotted this predictors in Figure 5, in order to see the distribution of the modalities. It is important to know if the data follows a distribution or some variables might be not significant by mostly containing values on 1 modality out of the total amount.

We have also observed that few modalities that we observed in the training set were not present in the test set and vice-versa. For example below we see that we have more modalities in our test set compared to our training set. This becomes a problem at the moment of making predictions. In order to be able to properly compute predictions of a model with a new data set, it needs to have the same format. We had to get the total number of categories for each variable from either the training or testing set and assign them to the other one so both data sets have the same type of attributes. In this specific example, we have to add the missing level to the training data set.

```
> levels(train[,'cat92'])
[1] "A" "B" "C" "D" "F" "H" "I"
> levels(test[,'cat92'])
[1] "A" "B" "C" "D" "E" "G" "H" "I"
```

We also dummy coded the categorical variables to using the model matrix function provided by R libraries. This was done so that we can learn numerical representations of our categorical data.

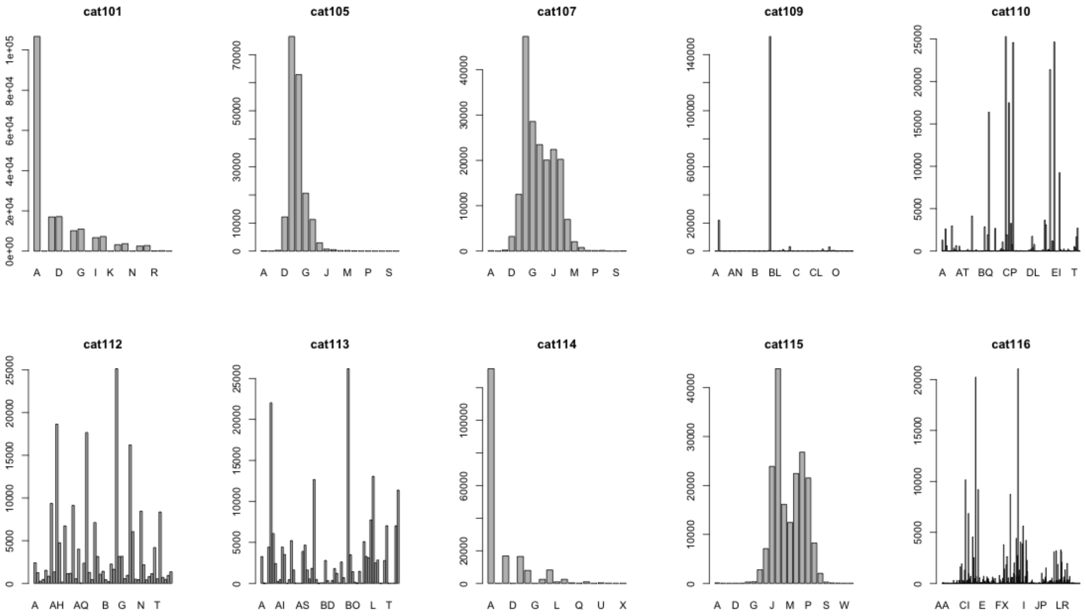

Figure 5: Modalities of Categorical Variables

5 Building the Model

Our aim is to reduce mean absolute error (MAE) between the predicted loss and the actual loss. This type of error is usually used when we want our error to be sensitive to outliers.

5.1 PCA

We noticed some of correlations with our continuous variables in our dataset. We capture latent information in our data by applying PCA. We observed that around 58 percent of the variance was explained by our first 2 principle components.

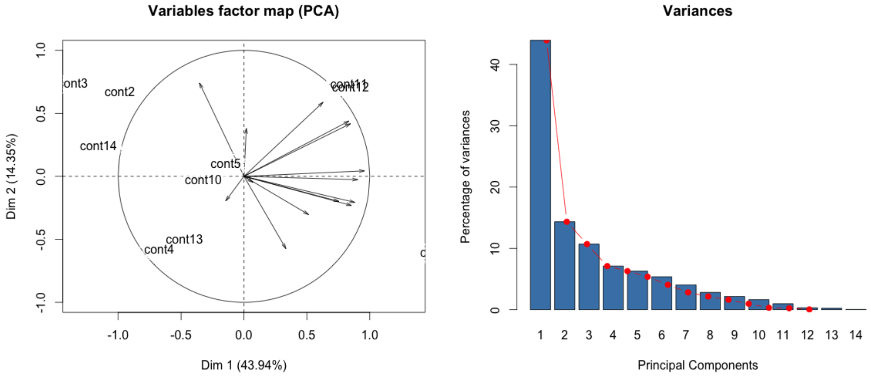

Figure 6: Variable Plot and Scree Plot

5.2 MCA

We have more than 1100 modalities observed in 116 attributes, on exploring further we observed that 10 attributes have more than 17 modalities. We chose top '100' components generated from our MCA model. Could have been more scientific while choosing the number of components. We gave up this approach as we had memory and computational issues. It was also difficult to decide the number of components to retain based the total variance explained. Another issue we had was that the package had a bug with the predict function, hence we could not obtain projections on the test data set. FactoMineR predict issue.

5.3 Feature Selection

Using random forest we were able to extract all the important features. We used the random forest package in the H2O library to learn important features. We had to dummy code the categorical data and stack it with the intermediate results obtained from the PCA analysis. Top 50 features explain 83 percent of the variance within our data. We started with 144 features after dummy coding we had 1051 to choose from and we choose top 50 features for our SVR model.

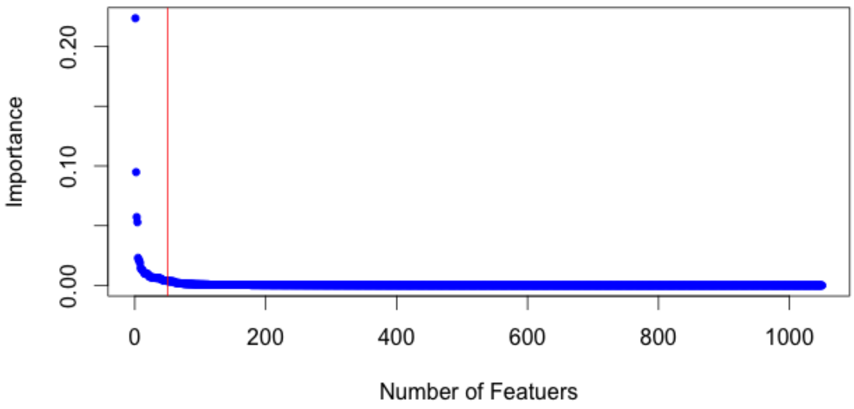

Figure 7: Variable Plot and Scree Plot

Some of the top features learned by our random forest model given below.

Id	Feature	Importance
1	cat80D	0.22
2	cat80B	0.09
3	cat12B	0.05
4	cat79D	0.05
5	cat1B	0.02

Table 2: Feature Importance

We choose H20s random forest library instead of using the regular R packages library is because the default implementation can only handle less than 30 modalities in categorizes where are H2os random forest library can handle more than 100 modalities.

5.4 Bagged SVM

We chose 3000 random observations for training followed by averaging the predictions made from several SVRs based on the number of bags. We observed that our predictions were best on averaging 10 SVR models with 3000 observations. We also observed that after a certain point ($>$ 10) increasing the number of bags did not improve on our error. We choose the number of samples based on our experiments conducted earlier.

6 Results

So far we have completed half of our project. We also have our intermediate results generated from PCA and MCA. Now we need to conduct experiments (Feature Selection) after stacking our data. would like to have your opinion on how to go about choosing the right tuning parameters and data size for our SVM model. Currently we are finding it extremely hard to choose the data size. Below we have some experiments we conducted on raw training data with 5 fold cross validation.

The experiments we conducted can be found by following the link Evaluation Score.

7 Conclusion

We observed that our bagged SVR model built from top 50 features improved from the SVR model created only from numerical features. However based on experiments we conducted we could not beat the random forest model. Most of the participants in this competition mostly used mixture of Neural network based model and XGBoost. I am sure our accuracy would have improved if we had used mixture of SVM experts based on the paper by bengio et all which was cited above. Some things that we tried and have not mentioned was clustering and training SVMs based on the the cluster centers produced.