A Mid Semester progress report

On

Machine Learning for Resource classification on the basis of user role

In

Bachelor of Technology

Computer Science and Engineering

By

*Harsimran Bhasin*

Enroll. No: 131100021

*Yash Phogat*

Enroll. No: 131100063

Under the guidance of

*Ms Akanksha Juneja*



National Institute of Technology Delhi

Institute of Applied manpower Research (IAMR Campus)

A-7, Institutional Area Near Satyawadi Raja Harish Chandra Hospital,

Narela, Delhi -110040

Abstract

In this project the idea is to learn and predict the resource (or files) allocation nature for different resources in a firm on the basis of an employee’s role in that firm. The role of an employee can be decided on the basis of few attributes associated with every employee (such as department, position etc.). The following document elaborates the process of digging important information from the data which apparently is highly unstable and applying Decision Tree Algorithms and Support Vector Machine (SVM) Algorithm to get a fair prediction model, which may later be improved by the ensemble of other methods.

***Keywords***: *Imbalanced classification, categorical features, Support Vector Machine, Split, Over fitting*

1. INTRODUCTION

1.1 Objective

The objective of this project is to understand the nature of the data available for every employee role and to design a prediction model that may automatically approve or reject employee’s resource application.

1.2 Contribution

So far we have been able to dig some important information from the training data which implies high order of instability in the data. The training data consists of a proportionally negligible number of rejection instances. The problem constitutes of categorical attributes with large number of categories for each attribute. Due to the large dimension of the instances of the data, the choice of using a SVM model was made. A Cross-validation framework was formulated for approximating the accuracy of the SVM model learned for different cost and gamma values. The learned model gave a ~93% accuracy, in the train set. Although the proportional inconsistency of the rejection instances in the train set led to a ~51% accuracy on the train set, which shall be handled by giving weights to the classes in the train set, so as the learnt model overcomes the data inconsistency problem. We then tried to extract important features from the data to further increase the classification accuracy. Going forward, to further increase the classifier accuracy we move from linear models to ensemble models of different classification algorithms.

2. Literature Review

The data for the project is obtained from the Kaggle [1] website, which is the historical data of Amazon Inc. from 2010 to 2011. The data consists of a training set of 32769 samples stating the employee role attributes, the resource and whether the resource was approved or rejected. The test set consists of 58922 samples without the ACTION attribute (i.e. approved: 1/rejected: 0).The following table shows the description of each feature in the data.

|  |  |  |
| --- | --- | --- |
| **Feature Name** | **Feature Meaning** | **Feature type (No. of unique categories)** |
| ACTION | “1” : Approved  “0” : Rejected | Categorical (2) |
| RESOURCE | Resource ID | Categorical (7518) |
| MGR\_ID | ID of the employee’s manager | Categorical (4243) |
| ROLE\_ROLLUP\_1 | Company role category ID1 (e.g. US Engineering) | Categorical (128) |
| ROLL\_ROLLUP\_2 | Company role category ID2 (US Retail) | Categorical (128) |
| ROLE\_DEPTNAME | Department description | Categorical (449) |
| ROLE\_TITLE | Business title description | Categorical (343) |
| ROLE\_FAMILY | Role family extended description | Categorical (67) |
| ROLE\_CODE | Unique ID for each company role (e.g. Manager) | Categorical (343) |

The data analysis also implied that the train data has 1897 samples of class 0 and 30872 samples of class 1, thereby leading to a result that on every 100 samples of class 1 there are approximately 6 samples of class 0, hence making class 0 a minority class.

While researching for classification models, our supervisor suggested an analogy of the problem to resource allocation in clouds. Clouds use SVM regression to estimate the response time in the next measurement period, and the resources are redistributed based on the current status of all virtual machine installed in physical machines [2]. This fact drove us to start by using SVM classification for our prediction model. An advantage of using SVM was that the SVM model can project a high-dimensional hyper-plane to an SVM kernel dimension.

*Cost and Gamma significance in SVM models*

In 1995, Cortes and Vapnik proposed the idea of a “soft margin” SVM that allows some of the instances that may lead to poorly fit models, to be ignored*.* ***Cost*** is a soft margin function, which controls the influence of each individual support vector. This process involves trading error penalty for stability.

***Gamma*** is used to denote the effect of biasness and variance on an SVM model. Large gamma leads to high bias and low variance models and vice-versa

3. Work Done

Initially the train data was read by the R instruction into a data frame. Now a cross validation framework was established to take 1000 random samples from the train data for testing and the rest for training the SVM. This was done to get the best cost and gamma values spread over the train-data. A random sampling was done to ensure some presence of the minority class instances in each training as well as test set.

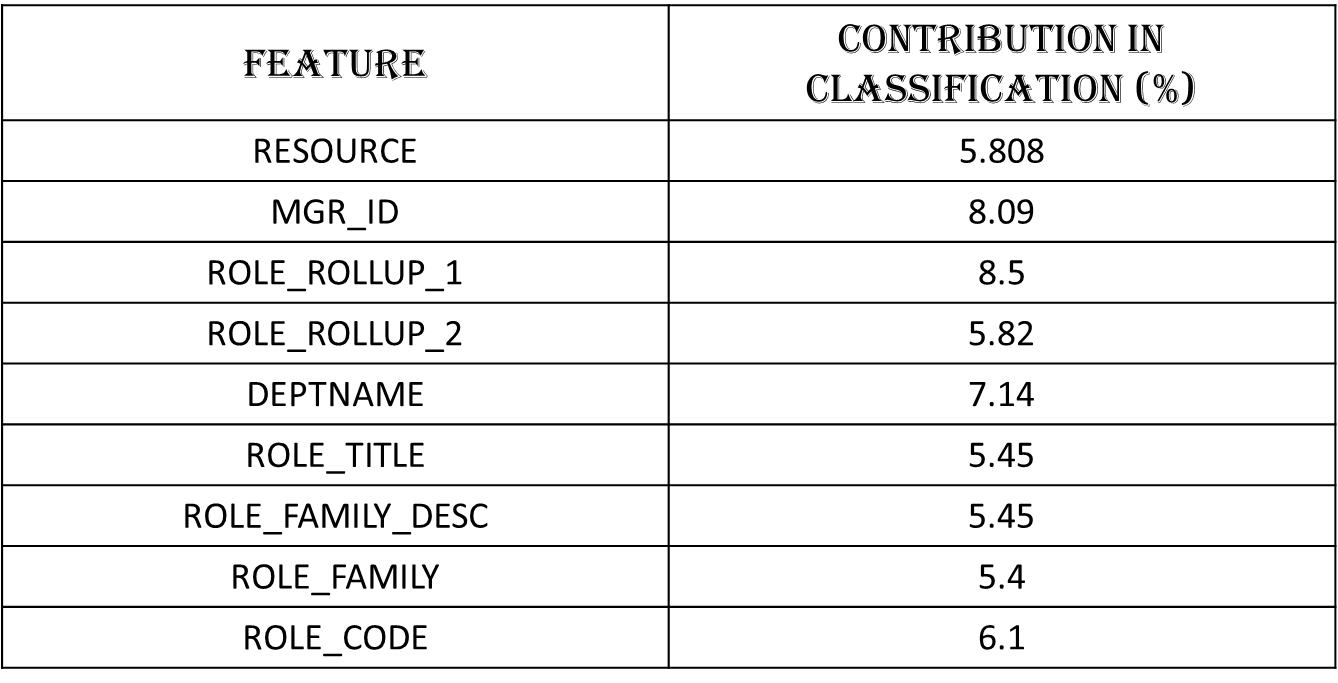
The best.tune function of the R library e1071 gave us the best cost and gamma values over the train-data. The best values of cost and gamma over linear and RBF kernel were 1 and 0.11111 respectively.

Now the task was to learn an SVM model for each resource category. This was done so that we can apply different model for different resources in test-data and therefore predict the result for each instance. There were 3766 resources with only one set of employee’s roll descriptor so their model was predicted instantaneously, but still there are nearly 50% of resources with various employee’s role. There is a need to learn SVM models for the remaining resource types which is a time consuming process. In fact an initial modelling over these resources assigned all the samples of most of the resource to class 1 (as it’s the major class in the train-set).

This model achieved a ~93% accuracy over the train-data, but the accuracy of this model was a mere ~51% over the test-set. This implies that the model over fitted and therefore there is a need to perform sampling over the data sets to include considerable amount of minority samples in every sample before modelling.

Going forward in this direction we identified the need to, weight the class labels. Due to the presence of a majority class (94%) and a minority class (06%) we had to increase the frequency/weightage of the minority class tuples. We allocated weights to our class labels (94->`0` and 06->`1`). Now, once again we run the SVM algorithm on the weighted data and observed a significant increase of 10% in the in the classification accuracy.

The next step to increase the classification accuracy is selecting the important features and rejecting the features that might be resulting in over fitting of the classification model. The individual contribution of the features, in classification are:



Since, the individual contribution of each feature in SVM classification is very small, therefore we don’t expect much improvement in the accuracy, as of now.

To complete our analysis with linear models we also did classification using Decision tree with SMOTE (Synthetic minority oversampling technique) to handle the imbalance in our class labels. The SMOTE utility introduces synthetic instances of the minority class to the sample train set. After obtaining this new sample train set we ran the decision tree algorithm for classification (rpart) and obtained a similar accuracy percentage on classifying the testset.

We now intend to start using ensemble models to increase the accuracy of our classifier.

Inspired from one of the research papers [3], we will first be using Random Forest model for classification.

Conclusion & Future Work

* So far we conclude that our train sample is inconsistent with class 0 as minority and class 1 as majority.
* We obtained an optimal value of cost=1 and gamma=0.1111 for SVM modelling.
* The SVM and Decision Tree model over fitted due to data imbalance
* Sampling was done for every resource; such that the number of minority class and majority class instances are considerable in each sample.
* Class weights were given to the class labels as Class `0`=94 and Class `1`=06.
* Normalizing the data increased the accuracy of SVM and Decision Tree classification by 10 percent to 61 percent.
* Attempts were made to select features with major contribution in SVM classification, but none of the features proved to make a major contribution individually, so we don’t extract any features for SVM classification.
* In future we try other prediction models such as Random Forest etc.

References

[1] https://www.kaggle.com/

[2] Chenn-Jung Huang, Yu-Wu Wang, Chih-Tai Guan, Heng-Ming Chen, and Jui-Jiun Jian” Applications of Machine Learning to Resource Management in Cloud Computing.” International Journal of Modeling and Optimization, Vol. 3, No. 2, April 2013

[3] Shijian Tang, Jiang Han and Yue Zhang “Amazon Employee Access Control System”.