Prediction Improvement using Optimal Scaling on Random Forest Models for Highly Categorical Data

Saurabh Mangal and Aditya Shankar
Co-Authored by Both the Authors
Institute of Systems Science, National University of Singapore
25 Heng Mui Keng Terrace, Singapore 119615

ABSTRACT
Random Forests are an effective ensemble method which is becoming increasingly popular, particularly for binary classification prediction problems. One of the most popular algorithms for implementing the Random Forest model is the Breiman and Cutler's algorithm and this forms the basis of the "randomForest" package in R. However, a Random Forest model implemented using this package has a limitation, especially in a milieu which has limited computational power, that it cannot handle highly categorical data.

In this paper, we present one of the many techniques we tried to improve the performance of a Random Forest Model using highly categorical data. The performance improvement was solely achieved using advanced pre-processing techniques like Optimal Scaling, hence the title of the paper.

Keywords
Ensemble Methods, Random Forest, Prediction with Categorical Variables, Optimal Scaling, Classification, Machine Learning, Non-Linear Categorical Prediction.

1. INTRODUCTION
The problem that this paper is based on comes from the KDD Cup 2010 competition, https://pslcdatashop.web.cmu.edu/KDDCup/ .

This competition was all about predicting a student's Correct First Attempt on a particular step of a Mathematical problem that was presented to the student. The data comes from two Cognitive Tutors which allow students to log-in and attend courses on Mathematics.

A Cognitive Tutor is a particular kind of intelligent tutoring system that utilizes a cognitive model to provide feedback to students as they are working through problems. Cognitive Tutors for mathematics are used in more than 2,500 schools and for some 500,000 students per year across the US.

All students are more or less in the same level of education and the problems are of varying degrees of toughness. The data was extremely large, of the magnitude of 3 to 5 GB worth of Training data. A part of the student's journey was captured in the training data and the remaining was presented as Test data set which allowed the authors to validate the performance of their models.

The data has the following hierarchy with respect to the problems that are presented to the students:

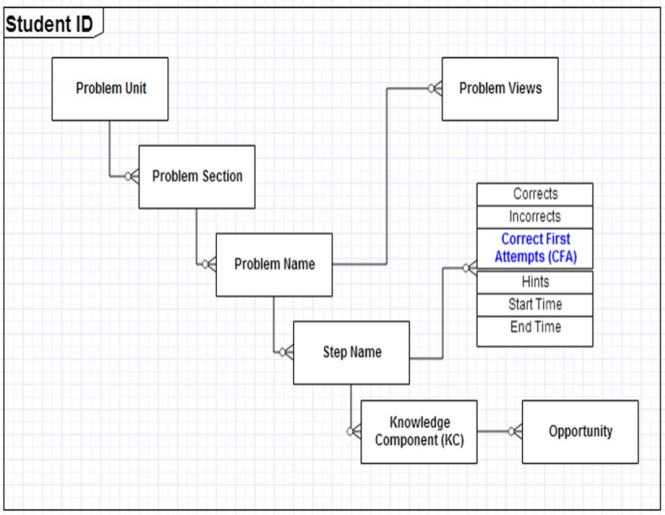

Fig.1: A Schematic Representation of the Data Hierarchy

The prediction of the Correct First Attempt (CFA) was the objective of the competition.

A brief summarization of the two datasets that was used for the results shown in this paper is as follows:

Table.1: Brief Summary – Part1

Fields	Counts
Unique Students	530
Unique Problem Hierarchy	122
Unique Problem Name	52,939
Unique Step Name	172,459
Unique KCs	840

A brief snippet of the data is provided below:

Tables 2 and 3: Data Set Snippet

Row	Student	Problem	Step
1	S01	WATERING_VEGGIES	(WATERED-AREA Q1)
2	S01	WATERING_VEGGIES	(TOTAL-GARDEN Q1)

Row	Incorrects	Hints	Error Rate	Knowledge component	Opportunity Count
1	0	0	0	Circle-Area	1
2	2	1	1	Rectangle-Area	1

As shown above, the data is highly categorical with Step Name being the most categorical variable with more than 150,000 categories.

The data has a significant temporal component to it and it can be seen that students get better with time.

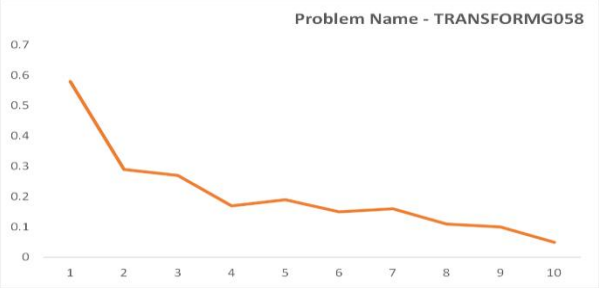

Fig.2: Students get better as they see the same problem many times.

The probability that the students get a step correct in the first attempt also increases as they encounter more and more Knowledge components.

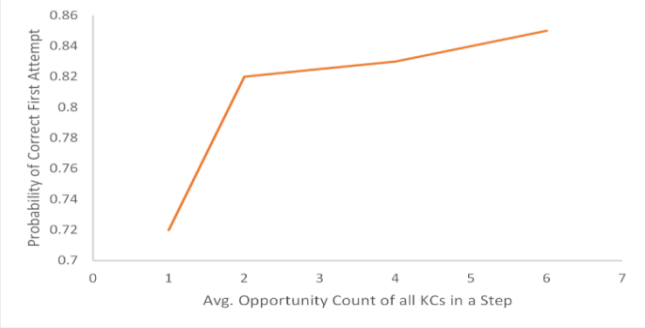

Fig.3: Students get better as they encounter a Knowledge Component many times.

The students also get more consistent as they spend more time on the Cognitive Tutor. Hence, the temporal effects are significant in the given data.

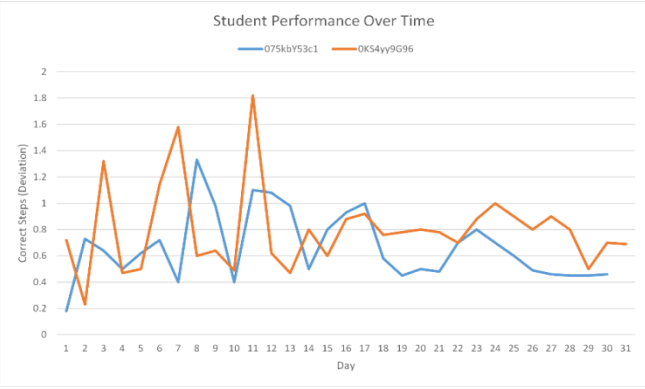

Fig.4: Student's consistency improves as they spend more time on the tutor

The figure above (4.2) shows the performance of two random students tracked over a period of 31 days spent with the tutor.

The graph is plotted with Standard Deviation of number of steps attempted correctly on the vertical axis and day number on the horizontal axis.

It can be clearly seen that as the student spends more time on the tutor, the variation in the number of correct steps performed each day reduces.

At the very beginning of his tutorial, the student appears to be very inconsistent and this inconsistency tapers off as he progresses his journey with us.

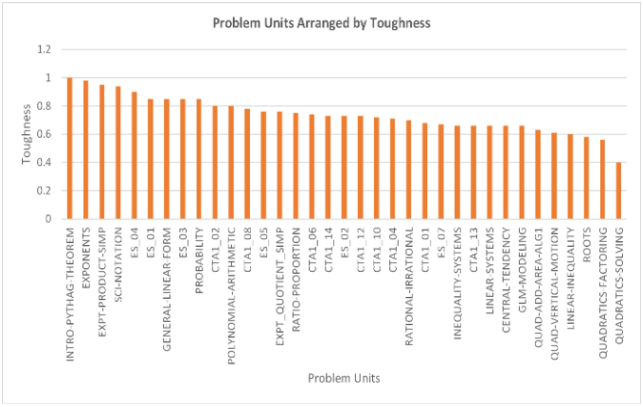

Fig.5: Problem Units – Arranged in order of toughness.

The figure above shows the problem units ranked in order of toughness. Toughness was a metric defined by the authors which takes into account the average number of Correct Steps that students attempt in that particular Problem Unit against the average total number of steps attempted in the Unit.

This metric was also one of the corner-stones of the solution presented as it helped in creating a distance matrix which was then used in Optimal Scaling.

2. MODELING PHILOSOPHY

Based on the exploratory analysis, the modelling philosophies were finalized.

After various tests and checks, the authors came to the conclusion that a student's entire journey with the tutor has to be learnt by the model, before it can start predicting whether a student can get a particular step correct in the first attempt.

The idea was to allow the model to "learn" from a student's entire record about the student's learning history.

3. DATA PRE-PROCESSING

3.1 Student Competence and Learning Rate

A part of the Data Pre-Processing included creation of new variables.

Two new variables were created:

1. Student Competency, using the value of hints asked, incorrect first attempts and correct first attempts. The weighted average of corrects, incorrect and hints was used to suitably calculate competency of every student.

The value of competency was then used in the models and appeared significant with all the models that we applied including Logistics Regression and Random forest.

2. Learning rate variable as:-

Total no of correct first Attempts per Problem Unit and Problem Section / Total no of Attempts per Problem Unit and Problem Section

By this variable we can take into account the variation in difficulty across units and sections.

Inclusion of variables boosted the accuracy and the consistency of the ensemble Random Forest models built.

Please refer to the Model Evaluation section for a detailed explanation of the improvements over time.

3.2 Feature Selection Algorithms

To identify which were the most important features in the data we used feature selection algorithms such as: Information Gain (Gini), Gain Ratio, Symmetrical Uncertainty, Entropy Gain. Although the findings were in line to what the authors had concluded from the exploratory analysis, two additional features that were important were obtained and the order of importance of the features that affected the outcome variable CFA was consolidated.

The Features that were concluded to be most important were:

- Step Duration (seconds)
- Total number of Corrects
- Total number of Incorrects
- Hints asked for
- Correct Step Duration
- Error Step Duration
- Step Name
- Number of times that Problem was Viewed
- Opportunity provided to learn a particular Knowledge Component
- Learning Rate (as defined earlier)
- Competence (as defined earlier)

The above features are arranged in order of their importance starting from the most important.

3.3 Data Balancing

CFA was not evenly distributed in the dataset with 1s (corrects) occurring far less than 0s (Incorrects) hence there was a scope to improve our models by training them on a balanced train set. SMoTE in the DMwR package in R was used to randomly oversample the minority class and undersample the minority class. Although class imbalance does not systematically hinder the performance of learning systems, the data overlapping among classes is another factor that lead to decrease in the performance of learning algorithm. Using SMOTE in DMwR resulted in improvements in RMSE after adopting this method.

3.4 Optimal Scaling

The need to introduce optimal scaling was because of highly categorical nature of the data.

The highly categorical data occurred at all levels of the Problem Hierarchy, especially at the lowest level, the Step which had more than 150,000 levels.

Hence, it was imperative to create a distance matrix of the complete Problem Hierarchy.

The idea was to use the distance matrix to then score the different problems and steps numerically based on their toughness. Hence, two Steps of comparable toughness will be ranked closer to each other.

The score of these steps were then used rather than the steps themselves for model prediction. Since, these scores are now continuous and not factor, the highly categorical nature of the data will have been effectively removed.

The matrix that was used to create these scores was Toughness, based on the below formula devised by the authors:

$$Toughness = \left\{\frac{Average\ \ Total\ Number\ \ of\ Correct\ \ Steps}{(Average\ \ Total\ Number\ \ of\ Attempts\ )}\right\} * \left\{\frac{Average\ \ Total\ Opportunity}{(Average\ \ Total\ Number\ \ of\ Attempts\ )}\right\}$$

The Problem Hierarchy was then scored in the following order:

Problem Unit -> Problem Section -> Problem Name -> Step Name

Hence, Steps, which were deemed to be very important according to the Feature Selection algorithms were de-categorized and optimally scaled.

3.5 Duplicate Removal

There were a lot of steps which were essentially the same steps but were named differently. This was earlier a big issue and was causing a lot of inaccuracies in the model. An example of a pair of such steps would be:

$$3x+5=10 \text{ And } 6x+10=20$$

Correcting this ambiguity was causing a lot of concerns. However, due to the Optimal Scaling, this problem was un-intentionally resolved.

The algorithm rounded off the scores to two decimal places, intrinsically stating that any pair of steps with a difference in their scores of less than 1% will be assigned the same score.

4. CONCLUSION

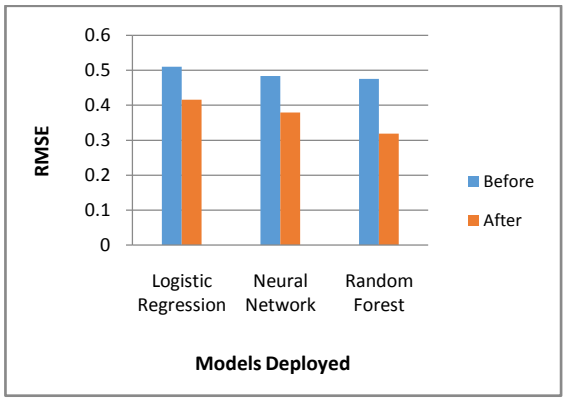

Fig.6: RMSE Comparison –Proposed Techniques result in significant reduction in error.

From the chart above we can see the improvements in RMSE after we applied the optimal scaling with all three models.

Highly categorical data as encountered in this problem is extremely tough to deal with, especially if the problem at hand is highly non-linear and requires the employment of advanced

Machine Learning and Ensemble Techniques to achieve accurate predictions.

Below (Tables 3 & 4) is a comparison of Coincidence matrix and accuracy on test set before and after applying Optimal Scaling for Random Forest.

Tables 3 and 4: Coincidence Matrix before and after

Coincidence Matrix for $Random Forest-CFA (**Before**)		
Actual ↓	(Pred) 0.0	(Pred) 1.0
0.0	8077	1582
1.0	5873	16203

Accuracy: 76.50%

Coincidence Matrix for $Random Forest-CFA (**After**)		
Actual ↓	(Pred) 0.0	(Pred) 1.0
0.0	7546	155
1.0	1949	22085

Accuracy: 93.37%

Optimal Scaling is an effective technique to address this concern as the categories can be converted to continuous values and scored while conserving the distance between these categories.

The key is to create the best metric to score the different levels of the categories.

Once, this metric has been created and the categories scored, the categorical data will have been effectively converted to continuous data while preserving the implied distance of the categories.

Advanced Ensemble Techniques like Random Forest can then be smoothly applied for prediction on this converted data, improving prediction and performance of these models.

We, hence, conclude that effective data pre-processing is often sufficient to improve model performance. However, this data pre-processing has to be done in the context of the data provided and the problem that is being solved.

5. REFERENCES

[1] Greer, J.E. and G. McCalla, 1994. Evaluating a Simulated Student using Real Students Data for Training and Testing.

[2] Anderson, J.R. 1995, Cognitive tutors: Lessons learned, Carnegie Mellon University.

[3] Noboru Matsuda1, William W. Cohen 2010, Tuning Cognitive Tutors into a Platform for Learning by-Teaching with SimStudent Technology Carnegie Mellon University.

[4] Noboru Matsuda, Applying Machine Learning to Cognitive Modelling for Cognitive Tutors 2006, in Machine Learning Department Technical Report (CMU ML).

[5] Muggleton, S. and L. de Raedt 1994, Inductive Logic Programming: Theory and methods

[6] Lau, T.A. and D.S. Weld, 1998 an inductive learning formulation.

[7] Johnson, W.L. 1998, Integrating pedagogical agents into virtual environments.

[8] Baffes, P. and R. Mooney, 1996, Refinement-Based Student Modelling and Automated Bug Library Construction.

[9] Merceron, A and K. Yacef, A web-based tutoring tool with mining facilities to improve learning and teaching, 2003.

[10] Mertz, J.S. 1997, Using Simulated Student for Instructional Design.

[11] Koedinger, K.R. and A. Corbett, 2006, Cognitive Tutors: Technology Bringing Learning Sciences to the Classroom, in The Cambridge Handbook of the Learning Sciences.

[12] Matsuda, N., W.W. Cohen, and K.R. Koedinger 2005, Applying Programming by Demonstration in an Intelligent Authoring Tool for Cognitive Tutors.

6. AUTHOR'S PROFILE

Saurabh Mangal – Master of Technology in Enterprise Business Analytics student at National University of Singapore.

Aditya Shankar – Master of Technology in Enterprise Business Analytics student at National University of Singapore

Guided by: **Ding Yi** – Lecturer at National University of Singapore.

Catherine Khaw – Lecturer and Senior Member at Institute of System Sciences at National University of Singapore.