Classification of Tutor System Logs with High Categorical Features

Yasser Tabandeh yasser.tabandeh@gmail.com

Department of Computer Science and Engineering, Shiraz University, Iran

Ashkan Sami asami@ieee.org

Department of Computer Science and Engineering, Shiraz University, Iran

Abstract

In this paper we propose our method for solving KDD Cup 2010 problem. Basically we did not perform a thorough literature review and reinvent all the ideas from scratch. The problem is predicting students learning based on logs of tutor systems which includes very large number of instances. In the preprocessing stage we deleted features not present in the test dataset and created some features. Transforming categorical features into numeric ones was another pre-processing step we performed. We used very naïve sampling to deal with large number of instances. Despite of using only 3 features of 22 features and regular decision tree and regression algorithms, results are acceptable. Even though we have used so many simplifications, did not consider a lot of interrelationships among features and did not use the whole training data, our team, Y10, has reached the 4th student place and 15th rank overall.

1. Introduction

KDD Cup is one of the most challenging data mining competitions which is held annually and is based on interesting and challenging problems. This year's challenge was to predict students learning based on logs of tutor systems. Very large datasets and highly categorical features were two main aspects of this year's competition. Limitation of resources can be a challenging problem when we are dealing with a very large training datasets. Also many training algorithms such as decision trees need few numbers of distinct values for a nominal feature to expand tree, otherwise, size of tree will increase drastically. Moreover, most classifiers are not performing efficiently on large datasets with limited hardware resources. Time is another constraint when we are dealing with large datasets. KDD Cup 2010 problem is one of problems which need close attentions to these challenges.

We did not do a literature review and definitely reinvented the wheel. Simplification of problem was our main concern. Due to time and resource limitations we did not even use the state-of-the-art methods for simplifications. At preprocessing steps, we deleted the features that were not present in the test datasets. Most of the features that were missing in the test data were sufficient to solve the problem; however, all of their values were missing. We performed feature generation. Based on best of our knowledge we were not aware of this method in previous literature. The conversion algorithm converts highly categorical features to numeric ones based on their 'percentages of positive class instances'.

Due to time and hardware limitations, we sampled training datasets to reduce the size of data drastically. Very simply we deleted one-third and one-seventh of all data. Finally modeling steps to predict learning of students to solve the problem was done by C4.5 [1] and linear regression [2]. In some instances, we did not even consider the instances that had more than one knowledge component.

Irrespective of all the simplifications that we performed our results are comparable to much more sophisticated algorithms that deployed most of the information present. The rest of paper is as follows: In section 2, we describe the problem, section 3 describes our method and finally section 4 concludes the paper. As is described in the abstract we did not do a literature review. Therefore, no section is devoted to previous work which definitely devalues our work.

2. The Problem

In this section we describe datasets and main challenges with data.

2.1 Datasets

Two types of datasets exist in KDD Cup 2010 competitions which are nearly same only different number of instances:

- Algebra
  - #Features:22
  - #Train Instances:8918054
  - #Test Instances:508912
- Bridge To Algebra
  - #Features:20
  - #Train Instances:15270710
  - #Test Instances:756386

These datasets are provided to tackle the problem of predicting correct first attempt (CFA).

2.2 Challenges with Data

Data sets used in training have some challenges which must be resolved before modeling:

1. **Huge number of instances**: Datasets of the competition are in range of VLDBs which include very large number of instances for training. Enough

resources such as time and hardware are needed to model these datasets. Techniques such as sampling or instance selection should be performed to handle large size of instances.

2. **Missing values in test data sets**: Nearly big subsets of features in the test datasets are completely missing. These features are critical and important in train datasets, but are missed in test. Actually if we have had those missing features, use of regular regression could predict CFA with a very high accuracy. Handling these missed features was a big challenge in this year's competitions.
3. **Highly categorical features:** features which are most important in modeling algorithms were highly categorical. In other words, we have features that have so many distinct categorical values in them. Modeling based on such a huge number of distinct values is a big challenge in most training algorithms such as decision trees.

3. Our Method

This section includes processes deployed in modeling and reaching the final model which was submitted for the competition.

3.1 Used Tools

Most of our knowledge discovery process was done using MS SQL Server 2000. Data processing and numeric transforming of nominal features was done on it. However, WEKA [3] was used to train and create models.

3.2 Feature Selection

We first modeled training data sets without considering test datasets. Excellent results were obtained for modeling training data! Because of some features like "Incorrects" and "Correct step Duration" most algorithms predicted students learning by looking at such features, but these features were missed in entire test datasets! So we removed them from feature set. It means in the first step of feature selection these features was removed simply because of missing values in test sets:

- Step Start Time
- First Transaction Time
- Correct Transaction Time
- Step End Time
- Step Duration (sec)
- Correct Step Duration (sec)
- Error Step Duration (sec)
- Incorrects
- Hints

- Corrects

Also “problem hierarchy” was removed because of full functional dependency with “problem name” feature. Two features “Problem Name” and “Step Name” was combined into a single feature named “ProblemStep” to increase accuracy and speed in modeling.

Features used in second step were:

- Anon Student Id
- ProblemStep
- Problem View
- KC (SubSkills)
- Opportunity (SubSkills)
- KC (KTracedSkills)
- Opportunity (KTracedSkills)
- KC (Rules)
- Opportunity (Rules)
- Correct First Attempt

3.3 First Training Models

For the first tries on modeling, we tested naïve Bayes, Bayesian network [4] and KNN with K=10, but best results on leader board using these methods had RMSE about 0.365 using bagging + Bayesian network. Other good algorithms such as decision trees and logistic regression were impossible to use because of highly categorical features.

3.4 Second Feature Selection step

A semi wrapper method was used in second step of feature selection to select best features. Backward elimination of features and using Bayesian Network as classifier was used for this goal. As a result, set of selected features in second step was:

- Anon Student Id
- ProblemStep
- Problem View
- KC (Rules)
- Opportunity (Rules)
- Correct First Attempt

Using these features and using bagging + Bayesian network RMSE on leader board decreased to 0.325.

3.5 Feature Transforming

Many features in training step were nominal features with huge number of distinct values such as “Anon Student Id”, “ProblemStep”, “KC (Rules)”. With limited time and hardware resources running a typical decision tree algorithm on these data was impossible. Also regression algorithms work better with numeric features. So a need to convert nominal and categorical features into numeric features existed. A simple method that replaced percentage of positive instances of that distinct value was used to do the transformation as is describe in Figure 1.

```
For each categorical feature Fc
        Add a new numeric feature to feature set: Fn
        For each distinct value v in Fc
                N=Number of instances which contain v
                Np=Number of instances which contain v and are in positive class
                A=Np/N (percentage of positive instances of v)
                Fill Fn with A
        Remove Fc from feature set
```

Figure1.transforming nominal features into numeric features

Three new numeric features were created using this method:

- StudentChance: transformed from “Anon Student Id” (ability of a student to solving problems)
- PSChance: transformed from “ProblemStep” (easiness of a step of a problem to be solved)
- RuleChance: transformed from “KC (Rules)” (usefulness of using a rule)

3.6 Final Modeling

For final training we used samples of datasets instead of full training sets. 1/3 of Algebra and 1/7 of Bridge to Algebra were used for training. Again we did not deploy state of the art instance selection or sampling methods. Simply we deleted instances based on a simple counting scheme. Feature normalization was done before training. Modeling was done using 10-fold cross validation on train datasets. Logistic regression and decision tree were used to predict labels in train datasets which both had nearly same results.

- **Logistic Regression**

  By running logistic regression algorithm on train dataset, target labels were predicted using this formula:

  $$Target=7.7719 - 3.991\times StudentChance - 5.3247\times PSChance - 2.7282\times RLChance$$

  Using this method resulted in RMSE 0.302 on leader board.

- **C4.5**

  As a powerful decision tree, C4.5 was used to create the final model. See details on this model in appendix A. RMSE reached 0.301 deploying C4.5. Results of this model were the final submission for competition.

4. Conclusion

We invented simple transformation of highly categorical features, used one third and one seventh of the training samples did not use the interrelationship among features and did not deploy highly sophisticated and state-of-the-art modeling techniques. However our method reached the 4th student teams and 15th overall rank. Considering the fact the only three features were used, we have achieved exceptional results. Definitely using more features and more sophisticated classification and/or prediction models, even instance based selection techniques can result in much more improvements. Lack of literature survey is another arena that can improve our method drastically.

5. References

[1] J. R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[2] Yule, G. Udny, "On the Theory of Correlation". *J. Royal Statist. Soc.* (Blackwell Publishing) 60 (4): 812–54, 1895

[3] Ian H. Witten; Eibe Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco. 2005. http://www.cs.waikato.ac.nz/~ml/weka/book.html.

[4] J. Pearl, "Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning" (UCLA Technical Report CSD-850017). Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA. pp. 329–334, 1985 http://ftp.cs.ucla.edu/tech-report/198_-reports/850017.pdf. Retrieved 2010-05-01.

Appendix A. Final Model of C4.5 Tree

```
PSChance <= 0.831
|  RLChance <= 0.422
|  |  RLChance <= 0.19
|  |  |  PSChance <= 0.773: 0 (1225.0/57.0)
|  |  |  PSChance > 0.773
|  |  |  |  PSChance <= 0.777
|  |  |  |  |  StudentChance <= 0.685157: 0 (3.0)
|  |  |  |  |  StudentChance > 0.685157: 1 (12.0/3.0)
|  |  |  |  PSChance > 0.777
```

```
| | | | | PSChance <= 0.794: 0 (47.0)
| | | | | PSChance > 0.794
| | | | | | RLChance <= 0
| | | | | | | PSChance <= 0.8
| | | | | | | | StudentChance <= 0.812594: 0 (16.0/4.0)
| | | | | | | | StudentChance > 0.812594: 1 (11.0/2.0)
| | | | | | | PSChance > 0.8
| | | | | | | | PSChance <= 0.806: 0 (15.0)
| | | | | | | | PSChance > 0.806
| | | | | | | | | PSChance <= 0.809
| | | | | | | | | | PSChance <= 0.808: 0 (9.0/1.0)
| | | | | | | | | | PSChance > 0.808: 1 (10.0/2.0)
| | | | | | | | | PSChance > 0.809: 0 (62.0/9.0)
| | | | | | RLChance > 0: 0 (9.0)
| | RLChance > 0.19
| | | RLChance <= 0.311: 0 (236.0/51.0)
| | | RLChance > 0.311
| | | | PSChance <= 0.641: 0 (453.0/153.0)
| | | | PSChance > 0.641: 1 (73.0/32.0)
| RLChance > 0.422
| | PSChance <= 0.691
| | | PSChance <= 0.487
| | | | PSChance <= 0.206: 0 (107.0/12.0)
| | | | PSChance > 0.206
| | | | | StudentChance <= 0.724138: 0 (715.0/234.0)
| | | | | StudentChance > 0.724138
| | | | | | StudentChance <= 0.856072
| | | | | | | PSChance <= 0.421: 0 (792.0/316.0)
| | | | | | | PSChance > 0.421: 1 (776.0/376.0)
| | | | | | StudentChance > 0.856072: 1 (396.0/156.0)
| | | PSChance > 0.487
| | | | StudentChance <= 0.764618
| | | | | PSChance <= 0.566: 0 (1046.0/493.0)
| | | | | PSChance > 0.566
| | | | | | StudentChance <= 0.668666
| | | | | | | RLChance <= 0.648
| | | | | | | | RLChance <= 0.64
| | | | | | | | | RLChance <= 0.606: 0 (91.0/33.0)
| | | | | | | | | RLChance > 0.606: 1 (21.0/6.0)
| | | | | | | | RLChance > 0.64: 0 (50.0/12.0)
| | | | | | | RLChance > 0.648
| | | | | | | | PSChance <= 0.638: 0 (475.0/232.0)
| | | | | | | | PSChance > 0.638: 1 (708.0/296.0)
| | | | | | StudentChance > 0.668666: 1 (2010.0/751.0)
| | | | StudentChance > 0.764618: 1 (6133.0/1844.0)
| | PSChance > 0.691
| | | StudentChance <= 0.757121
| | | | StudentChance <= 0.574213
| | | | | StudentChance <= 0.337331: 0 (104.0/41.0)
```

```
| | | | | StudentChance > 0.337331: 1 (871.0/342.0)
| | | | StudentChance > 0.574213
| | | | | PSChance <= 0.783
| | | | | | RLChance <= 0.953
| | | | | | | PSChance <= 0.723
| | | | | | | | RLChance <= 0.891: 1 (1187.0/391.0)
| | | | | | | | RLChance > 0.891: 0 (67.0/32.0)
| | | | | | | PSChance > 0.723: 1 (2492.0/722.0)
| | | | | | RLChance > 0.953: 1 (25.0/1.0)
| | | | | PSChance > 0.783
| | | | | | RLChance <= 0.6: 0 (26.0/9.0)
| | | | | | RLChance > 0.6: 1 (3034.0/680.0)
| | | StudentChance > 0.757121: 1 (12671.0/2240.0)
PSChance > 0.831
| PSChance <= 0.952
| | RLChance <= 0.523
| | | StudentChance <= 0.790105
| | | | RLChance <= 0
| | | | | PSChance <= 0.884: 0 (116.0/32.0)
| | | | | PSChance > 0.884
| | | | | | StudentChance <= 0.704648: 0 (77.0/24.0)
| | | | | | StudentChance > 0.704648
| | | | | | | PSChance <= 0.949
| | | | | | | | PSChance <= 0.944: 1 (106.0/49.0)
| | | | | | | | PSChance > 0.944: 0 (9.0)
| | | | | | | PSChance > 0.949: 1 (10.0)
| | | | RLChance > 0: 0 (44.0/8.0)
| | | StudentChance > 0.790105
| | | | RLChance <= 0: 1 (251.0/86.0)
| | | | RLChance > 0: 0 (29.0/8.0)
| | RLChance > 0.523: 1 (35513.0/3297.0)
| PSChance > 0.952
| | PSChance <= 0.988
| | | RLChance <= 0.276
| | | | StudentChance <= 0.7991
| | | | | StudentChance <= 0.686657: 0 (14.0/2.0)
| | | | | StudentChance > 0.686657: 1 (62.0/23.0)
| | | | StudentChance > 0.7991: 1 (51.0/1.0)
| | | RLChance > 0.276
| | | | StudentChance <= 0.725637
| | | | | StudentChance <= 0.367316
| | | | | | PSChance <= 0.968: 0 (14.0/5.0)
| | | | | | PSChance > 0.968: 1 (11.0/2.0)
| | | | | StudentChance > 0.367316: 1 (2808.0/121.0)
| | | | StudentChance > 0.725637: 1 (10981.0/205.0)
| | PSChance > 0.988: 1 (13926.0/11.0)

Number of Leaves  :     53
Size of the tree :      105
```

Figure2. C4.5 tree model