

# A Lightweight Solution to the Educational Data Mining Challenge

**Kun Liu**

**Yan Xing**

*Faculty of Automation*

*Guangdong University of Technology*

*Guangzhou, 510090, China*

CATCH0327@YAHOO.COM

YANXING@GDUT.EDU.CN

**Editor:** Unknown

## Abstract

This paper describes a lightweight solution for the education data mining challenge of KDD Cup 2010. The solution requires less computation resource, has satisfying prediction performance and produces prediction models with good capability of generalization.

**Keywords:** KDD cup 2010, imbalanced data (IDS), attribute value aggregation, ensemble learning

## 1. Introduction

The task of the KDD cup 2010 challenge is to predict student performance on mathematical problems from logs of student interaction with the intelligent tutoring systems (DataShop, 2010). There are two challenge data sets (i.e., training data sets): algebra-2008-2009 and bridge-to-algebra-2008-009. The former is smaller and contains 8,918,054 examples of 3,310 students, while the latter is larger and consists of 20,012,498 examples of 6,043 students. The computation resources we have are a laptop with a 2G processor, 2G RAM and Windows XP, and Rapid Miner community Edition 5.0 (GmbH, 2010). Looking into the data, we noticed that there are a few technical challenges:

- The volume of the training data sets is huge: since we have limited computation resources, the training data sets are too big for us.
- The data sets are imbalanced: the distribution of the student performance is highly skewed. For instance, only 14.6% of the examples in the algebra\_2008\_2009 data set are labeled with "0", while the others labeled with "1".
- Categorical attributes are massive: some of the categorical variables, such as "Problem Name", "Step Name", "KC(SubSkills)", "KC(KTracedSkills)" and "KC(Rules)" have too many value levels.

To address the above technical issues and complete the task within the time limitation, our solution is a kind of lightweight approach, where the prediction model is created on a small portion of the training data and then evaluated on the full testing data.

In the rest of the paper, we describe our solution in detail. In Section 2, we introduce the adopted preprocessing techniques. In Section 3, we discuss the classification algorithms that are used, and then we talk about our solution performance. Finally, we summarize our work.

## 2. Preprocessing

The challenge data are come from the interactions between the students and the computer-aided-tutoring systems. Just as most of the real world data, there are noises, missing values and inconsistent values. Besides that, the data are imbalanced, some categorical attributes are massive and our computation resource is limited, which have been mentioned in Section 1. Therefore preprocessing the data is necessary and critical.

### 2.1 Data Cleaning

Missing value is a serious problem in the challenge data. For an example, 50.4% of the attribute values of "KC(KTracedSkills)" is NULL. To solve this problem, a global constant is used to take the place of the missing values. Although this method is simple, it will be beneficial to the classification algorithm C4.5, which is selected as the base learning algorithm of our solution.

The main reason of the inconsistent values is the adoption of both the capital letters and the lower cases. Microsoft SQL Server 2000 is case-insensitive. Therefore it is used as our database management system (DBMS) to solve the problem.

### 2.2 Attribute Value Aggregation

In the algebra-2008-2009 data set, there are seven categorical attributes. Table 1 lists all the categorical attributes and their value levels (both original and after aggregation).

Table 1: Categorical Attributes and Their Value Levels

Attribute Name	Number of value levels	
	Original	After aggregation
Problem Name	188,368	83
KC(KTracedSkills)	921	45
KC(Rules)	2,978	94
Problem Hierarchy	165	165 (no aggregation)
Problem View	18	18 (no aggregation)
Step Name	695,674	695,674 (no aggregation)
KC(SubSkills)	1,824	1,824 (no aggregation)

From Table 1, it can be seen that some of the categorical attributes, such as "Problem Name", "Step Name", "KC(SubSkills)", "KC(KTracedSkills)" and "KC(Rules)" have a

large number of value levels. If these massively categorical attributes are used directly, there will be the following problems:

- In the process of feature selection and classification, a massive attribute with too many levels will dominate the other attributes with only a few levels. For the classification algorithms based on Decision Tree, the phenomenon is extremely serious.
- Over fitting of the prediction model may be caused, since some value levels only appear in the training data set.
- Noises usually exist in the massively categorical attributes.

Therefore value aggregation, which combines two or more value levels into a more abstract one, is necessary (Tan et al., 2005).

However, it is hard to choose which attributes to be aggregated when the data volume is huge and the related domain knowledge is lacking. From the experience of working on the development data sets, we noticed that the three attributes, "Problem Name", "KC(KTracedSkills)" and "KC(Rules)", can contribute more to the classification task than the other categorical attributes<sup>1</sup>. So they are chosen to be aggregated.

Once the attributes are chosen, the aggregation procedure is human-computer interactive and illustrated in Figure 1.

---

Algorithm Attribute-Value-Aggregation

---

For each categorical attribute to be aggregated:

- 1: Get the whole set of the value levels, which are distinct to each other.
  - 2: Cluster the value levels with the classical K-means clustering algorithm.
  - 3: Analyze the clustering result manually.
  - 4: **if** the result is unreasonable
  - 5:   return to step 2 and modify the K value
  - 6: **else**
  - 7:   assign an abstract value level to each cluster
  - 8: **end if**
  - 9: Scan the challenge data, check which cluster that the original value of each example belongs to, and then replace it with the new abstract one
- 

Figure 1: Procedure of Attribute Value Aggregation

In Figure 1, there are two key issues to be noticed. One is how to determine the value of K (step 2). Our method is trial and error. If there were sufficient domain knowledge, the value would be set more meaningfully. The other is how to judge the clustering result is reasonable (step 3). In our solution, a cluster is reasonable means that all its members

---

1. For those who are interested in the method of measuring an attribute's contribution to the classification task, please refer to Appendix A.

indicate the similar mathematical skills. For an example, for the attribute of "Problem Name", all the arithmetic expression containing addition, subtraction, multiplication and etc., should be grouped into one cluster.

### 2.3 Sampling

Because of the limitations of computation resources and time, it is not realistic for us to build prediction models on all the challenge data. Data sampling is our essential choice. A model generated on the sampled data can work almost as well as that on the entire data set if the sample is representative (Tan et al., 2005). We design an aggressive sampling scheme to obtain the sampled data. It is based on the technology of random sampling with replacement and has three extra factors to be controlled. The three factors are:

- There are about 100,000 examples in a sampled data set.
- The distribution of a sampled data set should be as similar as that of the original set, e.g., in a sampled set from algebra-2008-2009, the ratio of the positive and the negative examples should be around 85 : 15.
- In a sampled set, there is at least one example from each student.

As for the number of the sampled sets, it is set to be 14 by trial and error.

### 2.4 Feature Selection

In the process of feature selection, we directly remove the attributes, which do not appear in the challenge testing set. There are temporal information in the discarded attributes, but we do not have a good way to deal with it. Then two kinds of feature selection technology are tried: feature ranking and Wrapper with forward or backward selection.

For the technology of feature ranking, three selection criteria are adopted separately. They are information gain, information gain ratio and PCA. Unfortunately, all the final results are not satisfying.

For the technology of Wrapper with forward or backward selection, we use the algorithm of wrapper-X-Validation implemented in RapidMiner. The algorithm procedure consists of three sub-procedures: attribute ranking, model building and model evaluation. For the procedure of attribute ranking, the selection criterion is information gain ratio. The method of model building and evaluation will be described in Section 3. Since the wrapper technology uses the prediction performance of the classification models to select the subsets of the attributes, the final results are much better.

## 3. Classification

The main technology that we use to generate the classification models is ensemble learning, and the reasons are as the following:

- When the data volume is huge, the ensemble of several models usually outperforms the single global model.

- When the computation resources are limited, ensemble learning is more realistic and easier to be implemented.
- Ensemble learning is an effective strategy for handling imbalanced data. Since the base classifiers are built on various subsets of the original data, the final ensemble, which integrating the base classifiers, is expected to have a broader and better decision region for the minority class (He et al., 2009).

The ensemble learning we adopt is based on the classical bagging technology, whose procedure consists of three main steps. Firstly, the aggressive sampling techniques, which is introduced in Section 2.3, is used to create several sampled sets from the original data. Secondly, the base models are generated on the sampled sets (one base model on one sampled set). Finally, some of the based models are selected and aggregated to obtain a final ensemble model, with which a more accurate prediction can be obtained.

For the learning algorithm to build base models, algorithms of Decision Trees, Neural Networks, Support Vector Machines(SVM) and Naive Bayes are tried. Since our classification approach is a wrapping version, where the performance of the ensemble model is feedback to the procedure of feature selection, we run out of memory when neural network and SVM are the base classifiers. When Naive Bayes is the base classifiers, the final results are poor. Therefore, an algorithm of Decision Trees (C4.5 algorithm in RapidMiner) is chosen to be the base learner. For the ensemble scheme, the majority voting strategy is used.

As described in Section 2.3, there are 14 subsets of the original data. In ensemble learning, 13 of them are used for training and the rest for validation. Therefore there are 13 base classifiers and each of them are created by ten-fold cross-validation. After all the base classifiers are created, seven of them are chosen for ensemble using a greedy algorithm with backward elimination <sup>2</sup> (Han and Kamber, 2006).

The final prediction performance of our solution is listed in Table 2, where the performance of the champion’s solution is also listed for comparison.

Table 2: Prediction Performance of Our Solution

Solution	Algebra-2008-2009	Bridge-to-2008-2009	Total Score
Our solution	0.332801	0.577638	0.455219
The champion’s solution	0.274568	0.271337	0.272952

From Table 2, it can be seen that for the algebra-2008-2009 data set, the performance of our solution is only around 6% worse than that of the champion’s solution. However, for the bridge-to-2008-2009 data set, the performance of our solution is much worse. The reason is that we actually do not generate prediction models for the bridge-to-2008-2009 data set due to the time limitation. The score is obtained by using the prediction models generated on the algebra-2008-2009 data set. If we have enough time, we believe that the real performance of our solution on the bridge-to-2008-2009 data set is similar to that on the algebra-2008-2009 data set.

---

2. For those who are interested in the algorithm of selecting base classifiers, please refer to Appendix B.

## 4. Conclusion

To complete the data mining task of KDD Cup 2010, we design and implement a solution based on the limited computation resources that we have. There are several advantage of our solution:

- The generalization capability of the prediction model is good. The prediction model is built on a small part of the training data and tested on all the testing data.
- The performance of the prediction model is satisfying.
- It requires few computation resources.

The main limitation of our solution is that the temporal information in the challenge data sets is not considered. This may be the main factor to produce negative effect on the prediction performance of our solution.

## Appendix A.

In the procedure of measuring a massively categorical attribute’s contribution to the classification task, there are two main steps: 1)calculating the information gain ratio of the attribute, 2)calculating the correlation between the attribute and the others. Table 3 lists the information gain ratio of all the attributes, and Table 4 lists the correlations between every two of the attributes.

Table 3: Information Gain Ratio of The Massive Attributes

Attribute name	Problem Name	Step Name	KC(SubSkills)	KC(KTracedSkills)	KC(Rules)
Gain Ratio	0.08	0.10	0.09	0.08	0.12

Table 4: Correlation Matrix of The Massive Categorical Attributes

Attributes	Problem Name	Step Name	KC(SubSkills)	KC(KTracedSkills)	KC(Rules)
Problem Name	1	0.641	0.048	0.021	0.142
Step Name	0.641	1	-0.026	-0.073	-0.042
KC(SubSkills)	0.048	-0.026	1	0.845	0.590
KC(KTracedSkills)	0.021	-0.073	0.845	1	0.538
KC(Rules)	0.142	-0.042	0.590	0.538	1

From Table 3, it can be seen that the information gain ratio of all the attributes are similar. However, Table 4 indicates that the correlations of some attribute-pairs, such as "Problem Name" and "Step Name", "KC(SubSkills)" and "KC(KTracedSkills)",are high. Thus only three of the attributes, "Problem Name", "KC(KTracedSkills)" and "KC(Rules)",are chosen to be aggregated.

## Appendix B.

In our ensemble learning approach, the algorithm procedure of selecting base classifiers is illustrated in Figure 2.

---

Algorithm: Base classifiers selection by stepwise backward elimination

---

```
1: for k = 1 to the size of the base classifier set
2:   Randomly remove a classifier from the classifier set.
3:   Evaluate the performance of combining the remaining classifiers.
4:   if the performance is better than the previous
5:     jump to step 8
6:   else
7:     add the classifier removed to the classifier set
8:   end if
9: end for
```

---

Figure 2: Algorithm for base classifiers selection

## References

- PSLC DataShop. Kdd cup 2010: Educational data mining challenge. Available electronically via <https://pslccdatashop.web.cmu.edu/KDDCup/>, 2010.
- Rapid-I GmbH. Rapidminer community edition 5.0. Available electronically via <http://rapid-i.com/>, 2010.
- Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Elsevier Pte Ltd, Maryland Heights, MO 63043, USA, 2006.
- Haibo He, Member, IEEE, and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.
- Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Morgan Kaufmann, San Fransisco, CA 94104, USA, 2005.