Combining Different Classifiers in Educational Data Mining

He Chuan, Li Ruifan, and Zhong Yixin

School of Computer Science,
Beijing University of Posts and Telecommunications,
Beijing, China
hcl258@yeah.net

Abstract. Educational data mining is a crucial application of machine learning.The KDD Cup 2010 Challenge is a supervised learning problem on educational data from computer-aided tutoring. The task is to learn a model from students' historical behavior and then predict their future performance. This paper describes our solution to this problem. We use different classification algorithms, such as KNN, SVD and logistic regression for all the data to generate different results, and then combine these to obtainthe final result. It is shown that our resultsarecomparable to the top-ranked ones in leader board of KDD Cup 2010.

Keywords: data mining, logistic regression, k-nearest neighbor, singular value decomposition, classifiers combination.

1 Introduction

In KDD Cup 2010, the task is to predictstudent algebraic problem performance giveninformation regarding pasperformance. This prediction task presents not only technical challenges for researchers,but is also of practical importance, as accurate predictions can be used, for instance, tobetter understand and ultimately optimize the student learning process.Specifically, participants were provided with summaries of the logs of student interactionwith intelligent tutoring systems. Two data sets are available: algebra 2008-2009 andbridge to algebra 2008-2009. In the rest of this paper, we refer to them as A89 and B89, respectively. Each data set contains logs for a large number of interaction steps. Some interaction log fields areincluded in both training and testing sets, such as student ID, problem hierarchy includingstep name, problem name, unit name, section name, as well as knowledge components (KC)used in the problem and the number of times a problem has been viewed. However, somelog fields are only available in the training set: whether the student was correct on the firstattempt for this step (CFA), number of hints requested (hint) and step duration information. The details are list in Table 1.

J. Zhang (Ed.): ICAIC 2011, Part V, CCIS 228, pp. 467–473, 2011.

Table 1. Dataset statistics

Datasets	Algebra 2008-2009	Bridge to Algebra 2008-2009
Lines (train)	8,918,054	20,012,498
Students (train)	3,310	6,043
Steps (train)	1,357,180	603,176
Problems (train)	211,529	63,200
Section (train)	165	186
Units (train)	42	50
KC (train)	2,097	1,699
Steps (new on test)	4,390	9,807

The competition regards CFA, which could be 0 (i.e., incorrect on the first attempt) or 1, as the label in the classification task. For each data set, a training set with knownCFA is available to participants, but a testing set of unknown CFA is left for evaluation. The evaluationcriterion used is the root mean squared error (RMSE). In the competition, participantssubmitted prediction results on the testing set to a web server, where the RMSE generatedbased on a small subset of the testing data is publicly shown. This web page of displayingparticipants' results is called the "leader board."

Facing such a complicated problem, we use different classification algorithms such as KNN, SVD [2, 3] and logistic regression [2] for all the data to generate different results, and then combine these to get final result. In particular, logistic regression needs many proper features to work well, so feature engineering is a necessary step. KNN and SVD, which are transferred from collaborative filtering community, will exploit the basic information in the given data. In the following sections, we make the arrangement: Section 2 shows our method in details, including some preprocessing, data grouping, feature engineering, logistic regression and trust region optimization, and combining methods. Section 3 gives the final result and the discussion for our method.

2 Our Method

This section is the main part of our paper. It explains the whole procedure of our method.

A. Validation Set Generation

Because we do not have all the ground truth labels for test data, we have to generate validation sets by ourselves. Table 2 shows the number of samples in validation set of Algebra 2008-2009 and Bridge to algebra 2008-2009.

Table 2. The number of samples in validation and training set of Algebra2008-2009 and Bridge to algebra 2008-2009

Datasets	algebra 2008-2009	bridge to algebra 2008-2009
Training(V)	8,407,752	19,264,097
Validation()	510,303	748,402
Training(T)	8,918,055	20,012,499
Training()	508,913	756,387

B. Feature Engineering

Basic features: Some basic yet important features considered in our early experiments can be categorizedinto two types: student name, unit name, section name, problem name, step name and KC arecategorical features, while problem view and opportunity are numerical features.

Combining features: Because all feature meanings are available, we are able to manuallyidentify some useful pairs offeatures. For example, hierarchical information can be modeledby indicating the occurrences of the following pairs: (student name, unit name), (unit name,section name), (section name,problem name) and (problem name, step name). In addition to two-feature combinations, we have also explored combinations of higher-order features.

Temporal features: Because learning is a process of skill-improving over time, temporal information shouldbe taken into consideration. There are some well-established techniques, utilizing a quitedifferent data model than traditional classification problems, to model student latent attributessuch as knowledge tracing and performance factor analysis. We considered a simple and commonapproach to embed temporal information into featurevectors. For each step, step name and KC values from the previous few steps were added asfeatures.

Other features: KCs are obtained by splitting the KC string with"~~"following the suggestion at the "Data Format" page on the competition website.Then binaryfeatures are used to indicate the KCs associated with each step. However, thissetting results in many similar KCs. To remedy this problem, we tokenized KC strings and used each token as a binary feature.Our experiences indicated thatthis method for generating KC features is very useful for the data set A89. We also usedtechniques such as regular expression matching to parse knowledge components, but didnot reach a setting clearly better than the one used for the baseline.For problem name and step name, we tried to group similar names together via clusteringtechniques. This approach effectivelyreduced the number of features without deteriorating the performance. We have also tried to model the learning experience ofstudents. A student's performance may depend on whether that student had previouslyencountered the same step or problem. We added features to indicate such information.Results showed that thisinformation slightly improves the performance.Earlier problem view was considered as a numerical feature. In some situations, wetreated it as a categorical one and expanded it to several binary features. This setting ispossible because there are not too many problem view values in training and testing sets.One student sub-team reported that this modification slightly improved RMSE.

C. Logistic Regression

After feature generation, we are building a classifier with feature vectors as input. We here use logistic regression as the classification algorithm, which is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. In particular, the logistic regression model in [2] is:

$$P(y|x) = \frac{1}{1+\exp(-yw^T x)} \tag{1}$$

where x is feature vector of a sample and y is the label of the sample (here y is CFA). Then a regularized logistic regression model is:

$$\min_w \frac{1}{2} w^T w + C\sum_{i=1}^{l} \log(1+e-y_i w^T x_i) \tag{2}$$

where w is the weight vector to be obtained through training, and the first term is the l2-regularization and C is a penalty parameter. In our experiments, we use liblinear toolkit [1] to build the models and test the data. Liblinear is a linear classifier which can deal with data with millions of instances and features. It supports to solve seven optimization problems which of course includes *l2*-regularized logistic regression, and is very efficient for training large-scale problems even much faster than libsvm. Liblinear adapts trust region method to optimize *l2*-regularized logistic regression problem.

D. KNN and SVD

In this section, we explain two classification methods which have been working in collaborative filtering (CF) for a long time. In the educational data mining problem, we consider the students are users in CF and the steps are items such as books or movies in CF. Except for these two fields in given data, we basically ignore other fields which can provide a bunch of information though.

In machine learning, the k-nearest neighbor algorithm (k-NN) is a method for classifying samples based on closest training examples in the feature space. *K*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of its nearest neighbor.

The singular value decomposition is well known in linear algebra and states that an arbitrarymatrix *C* with real or complex entries can be decomposed into a product of three matrices. In the collaborative filtering context the matrix*C* is sparse, meaning that most of the elements are unknown. The basic idea of SVD todecompose the matrix *C* can be still applied. During the Netflix competition, this methodbecame very popular because of two reasons. First it delivers good results out of the box andsecond it is easy to tune and customize. Over the last years there were lots of publicationsdescribing extensions to the basic SVD. In collaborative filtering, SVD in [3] is equivalent to non-negative matrix factorization, that is

$$\min_{A,B} \left\| C - AB^T \right\|_C^2 \tag{3}$$

To keep it from over-fitting, the formula can be written as:

$$E(A,B) = \left\| C - AB^T \right\|_C^2 + \lambda(\left\| A \right\|_F^2 + \left\| B \right\|_F^2) \quad (4)$$

With stochastic gradient descent algorithm, it can be solved efficiently.

E. Combination

Combination is the last not the least step in our method.The individual predictions stemmingfrom the methods described above are combined. This is a typical approach to improve the prediction accuracy, which wasshown in [5][6].It is widely known that an ensemble of predictions performsbetter than the best individual result.

3 Experiment Results and Discussion

A. Results

To compare the performance of all the classifiers, we experiment on KNN, SVD, logistic regression and their combination. Table 3 to 4 show the results from all the classifiers over the given two datasets.

Table 3 is the results from KNN algorithm, where K is the number of nearest neighbors.

Table 3. Results from KNN algorithm

Dataset	RMSE	Meta parameters
Algebra 2008-2009	0.3257	$K=41, \alpha=12.9, \beta=1.5, \delta=6.2, \gamma=-1.9$
Bridge to Algebra 2008-2009	0.3049	$K=41, \alpha=12.9, \beta=1.5, \delta=6.2, \gamma=-1.9$

The following table is the results from SVD, where N is the latent factor number, it is the learning rate, and lambda is the L_2 regularization coefficient.

Table 4. Results from SVD

Dataset	RMSE	Meta parameters
Algebra 2008-2009	0.446277	$N=10, \eta=0.002, \lambda=0.02$
Bridge to Algebra 2008-2009	0.3049	$N=10, \eta=0.002, \lambda=0.02$
Bridge to Algebra 2008-2009	0.3159	$N=20, \eta=0.002, \lambda=0.01$
Bridge to Algebra 2008-2009	0.3178	$N=20, \eta=0.002, \lambda=0.03$

Table 5 is the result from logistic regression:

Table 5. Result from logistic regression

Dataset	RMSE
Algebra 2008-2009	0.2895
Bridge to Algebra 2008-2009	0.2985

Table 6 is the result after combination

Table 6. Result after combination

Dataset	RMSE
Algebra 2008-2009	0.2820
Bridge to Algebra 2008-2009	0.2850

B. Discussion

From the four tables above, we can see the combination method works better than any of the single classifier. Logistic regression has the best performance among the three methods; due to not only it exploits almost all the information in the given data but also it uses very fine feature vectors as input. Even so, the combined classifier can reach new lower RMSE value than it over the two dataset.

As for CF-like methods, we can also try to make use of other information such as KC components, problems, unit, which is different from the features used in logistic regression, but can provide more discriminative information. Besides the combination method, there are some other ways which are considered to be interesting such as dynamic Bayesian network. HMM in [4] is one of DBNs considering the changing math ability of students. In a word, there would be a great number of ways to be explored.

Acknowledgement. This paper is supported by Project 60873001 of National Natural Science Foundation of China and by Project 2009RC0212 of Chinese Universities Scientific Fund.

References

1. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: A libraryfor large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
2. Yu, H.-F.: Feature engineering and classifier ensembling for KDD Cup 2010. Journal of Machine Learning Research, Workshop and Conference Proceedings (2010)
3. Töscher, A.: BigChaos, Collaborative Filtering Applied to Educational Data Mining. Journal of Machine Learning Research, Workshop and Conference Proceedings (2010)

4. Pardos, Z.A.: Using HMMs and bagged decision trees to leverage rich features of user and skill. Journal of Machine Learning Research, Workshop and Conference Proceedings (2010)
5. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and anapplication to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
6. Jahrer, M., Töscher, A., Legenstein, R.: Combining predictions for accuraterecommender systems. In: KDD: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)