INCREASE PERFORMANCE BY COMBINING MODELS OF ANALYSIS ON REAL DATA

Dumitru Dan Burdescu and Marian Cristian Mihăescu
Software Engineering Department, University of Craiova, Bvd. Decebal, Nr. 107, 200440, Craiova, Dolj, Romania

Keywords: Models of analysis, machine learning, e-Learning.

Abstract: In this paper we investigate several state-of-the-art methods of combining models of analysis. Data is obtained from an e-Learning platform and is represented by user's activities like downloading course materials, taking tests and exams, communicating with professors and secretaries and other. Combining multiple models of analysis may have as result important information regarding the performance of the e-Learning platform regarding student's learning performance or capability of the platform to classify students according to accumulated knowledge. This information may be valuable in adjusting platform's structure, like number or difficulty of questions, to increase performance from presented points of view.

1 INTRODUCTION

An e-Learning platform has been developed and is currently deployed in a continuous learning program. The platform may be used by four types of users: sysadmin, secretary, professor and student. Secretaries and professors work together to manage the infrastructure the student will use. Secretaries manage the professors, disciplines and students. Professors take care of the assigned disciplines in terms of course materials, test and exam questions. Course materials are created in an e-learning format that is very attractive and asks regularly for student feedback.

The notion of "user session" was defined as being a temporally compact sequence of Web accesses by a user. A new distance measure between two Web sessions that captures the organization of a Web site was also defined. The goal of Web mining is to characterize these sessions. In this light, Web mining can be viewed as a special case of the more general problem of knowledge discovery in databases (Agrawal and Srikant, 1994), (Nasraoui et al., 1999), and (Mobasher et al., 1996).

The goal of analyzing process is to improve platform's performance from two perspectives: student's learning proficiency and platform's capability of classifying students according to their accumulated knowledge. Firstly, it wants to evaluate the learning proficiency of students which mean that they accumulated knowledge during learning process. Secondly, it wants to avoid the situation when a large number of students have only small grades or only big grades. This situation would mean that the platform is not able to classify students according to their accumulated knowledge.

The analysis process has as primary data the activity performed by users on the platform. Bagging, boosting and stacking are general techniques that can be applied to numeric prediction problems as well as classification tasks (Witten and Frank, 2000).

There are two main difficulties that may arise in analysis process. Firstly, available data should be representative in terms of quantity and quality. Secondly, the analysis process may create an over-fitted model. Over-fitting is fitting a model so well that is picking up irregularities in the data that may be unique to a particular dataset (Rud, 2001).

2 OVERVIEW OF THE E-LEARNING PLATFORM

The main goal of the application is to give students the possibility to download course materials, take tests or sustain final examinations and communicate with all involved parties. To accomplish this, four different roles were defined for the platform: sysadmin, secretary, professor and student. The main task of sysadmin users is to manage secretaries. A sysadmin user may add or delete secretaries, or change their password. He may also view the actions performed by all other users of the platform. All actions performed by users are logged.

Dan Burdescu D. and Cristian Mihăescu M. (2007).
INCREASE PERFORMANCE BY COMBINING MODELS OF ANALYSIS ON REAL DATA.
In *Proceedings of the Second International Conference on Software and Data Technologies - PL/DPS/KE/WsMUSE*, pages 255-258
DOI: 10.5220/0001331802550258

In this way the sysadmin may check the activity that takes place on the application. The logging facility has some benefits. An audit may be performed for the application with the logs as witness. Security breaches may also be discovered.

Secretary users manage sections, professors, disciplines and students. On any of these a secretary may perform actions like add, delete or update.

The main task of a professor is to manage the assigned disciplines while s discipline is made up of chapters. The professor sets up chapters by specifying the name and the course document.

The platform offers students the possibility to download course materials, take tests and exams and communicate with other involved parties like professors and secretaries. Students may download only course materials for the disciplines that belong to sections where they are enrolled. They can take tests and exams with constraints that were set up by the secretary through the year structure facility.

A history of sustained tests is kept for all students. In fact, the taken test or exam is fully saved for later use. That is why a student or a professor may view a taken test or exam as needed. For each question it is presented what the student has checked, which was the correct answer, which was the maximum points that could be obtained from that question and which was the number of obtained points. At the end it is presented the final formula used to compute the grade and the grade itself.

The logging facility that is mainly used by sysadmin is transparently implemented for all users (secretaries, professors and students). Whenever one of them performs an action (e.g. a student starts or finishes an exam) that action is recorded for later use.

After five months of deployment, the activity table contains more than 50,000 records and we suppose that until the end of the learning cycle there will be close to 100,000 records. All this logged activity may also be very helpful in an audit process of the platform. The records from the activity table represent the raw data of our analyzing process.

3 METHODS OF COMBINING MULTIPLE MODELS OF ANALYSIS

The analysis process uses activity data and employs different techniques to build classifiers. Estimating each classifier's accuracy is important in that it allows the evaluation of how accurately the classifier will label future data, that is, data on which the classifier has not been trained. Among the most used techniques for estimating classifier accuracy there are the holdout and k-fold cross-validation methods (Han, 2001).

The main purpose of the analysis process is to obtain a classifier with great accuracy. This makes sure that obtained knowledge is sound and may be used for improving the performance of the e-Learning platform. Performance is seen from two perspectives. One regards the learning proficiency of students and the other the capability of the platform to classify students.

Combining the output of multiple models is a good method for making decisions more reliable. The most prominent methods for combining models generated by machine learning are called bagging, boosting, and stacking. They can all, more often than not, increase predictive performance over a single model. However, the combined models share the disadvantage of being rather hard to analyze: it is not easy to understand in intuitive terms what factors are contributing to the improved decisions. Bagging, boosting and stacking are general techniques that can be applied to numeric prediction problems as well as classification tasks. Bagging and boosting both uses the same method of aggregating different models together (Witten and Frank, 2000).

In Figure 1 there are presented the bagging and boosting general techniques for improving classifier accuracy. Each combines a series of T learned classifiers, C1, C2, …, CT, with the aim of creating an improved classifier, C*(Han, 2001).

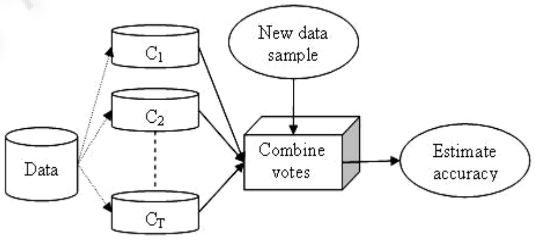

Figure 1: Increasing classifier accuracy: Bagging and boosting each generate a set of classifiers, C1, C2,…,CT. Voting strategies are used to combine the class predictions for a given unknown sample.

In boosting, weights are assigned to each training sample. A series of classifiers is learned. After a classifier Ct is learned, the weights are updated to allow the subsequent classifier, Ct+1 , to "pay more attention" to the misclassification errors made by Ct. The final boosted classifier, C*, combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy. The boosting algorithm can be extended for the prediction of continuous values (Witten and Frank, 2000).

Stacked generalization (Witten and Frank, 2000), or stacking for short, is a different way of combining multiple models. It is widely used than bagging and boosting, partially because it is difficult to analyze theoretically, and partially because there is no general accepted best way of doing it. Stacking is not used to combine models of the same type (e.g. a set of decision trees).

One of the important questions is: what algorithms are suitable for level-1 inducer? In principle, any learning scheme maybe applied. However, since most of the work is already done by the level-0 learners, the level-1 classifier is basically just an arbiter, and it makes sense to choose a rather simple algorithm for this purpose. Simple linear models have turned out best in practical situations (Wolpert, 1992).

4 EXPERIMENTAL RESULTS

Activity data obtained while running the platform represents raw data. The analysis process is conducted by running algorithms implemented in Weka workbench (www.cs.waikato.ac.nz). This workbench accepts data that has a specific format called arff. That is why we developed an of-line application that gets data from the platform's database and creates a file called activity.arff. This file is used as input in our analyzing process.

The first method of combining models is bagging. J48 decision trees are used as learning algorithm. We used three decision trees (C1, C2 and C3) on a training set of 375 instances. For each of these trees sampling with replacement was used. C* learner uses the votes from C1, C2 and C3 learners. We choose three voters (three iterations) such that for each instance C* learner should not have any problems in setting up the class. Time taken to build the model by bagging algorithm was 0.02 seconds. Table 1 presents the results of bagging.

The second method of combining models is boosting. J48 decision trees (C1, C2 and C3) are also used on the same training set of 375 instances. As in bagging, it was used sampling with replacement and we used three learners (three iterations) for the same reason. Time to build the model by boosting was 0.06 seconds. Table 2 presents the results of boosting.

The third method of combining models is stacking. As level-0 learner we have chosen a J48 decision tree, a Naïve Bayes (Cestnik, 1990) learner and a LMT(Logistic Model Tree) learner. As level-1 learner (or meta classifier) we used a J48 decision tree learner. Time taken to build the model by stacking algorithm was 38.08 seconds. Table 3 presents the results of stacking.

Table 1: Results of bagging.

Classifier	Algorithm	No. of leafs	Accuracy
C_1	J48	13	88.8%
C_2	J48	12	86.3%
C_3	J48	13	87.7%
C*	J48	12	89.5%

Table 2: Results of boosting.

Classifier	Algorithm	No. of leafs	Accuracy
C_1	J48	14	90.8%
C_2	J48	12	89.3%
C_3	J48	13	91.7%
C*	J48	13	92.5%

Table 3. Results of stacking.

Classifier	Algorithm	No. of leafs	Accuracy
C_1	J48	13	90.7%
C_2	NB	12	89.5%
C_3	LMT	14	91.8%
C*	J48	14	92.8%

The effectiveness of stacked generalization for combining three different types of learning algorithms was demonstrated in (Ting and Witten, 1997) and used by (Breiman, 1996), (LeBlanc and Tibshirani, 1993).

5 CONCLUSIONS AND FUTURE WORK

We have designed and implemented the e-Learning platform. The design of the platform is based on MVC model that ensures the independence between the model (represented by MySQL database), the controller (represented by the business logic of the platform implemented in Java) and the view. The platform is currently deployed (stat257.central.ucv.ro) and used by 400 students and 15 professors.

There was implemented an embedded mechanism within the platform that monitors and records all user's activity. Data obtained in this manner represents the raw material for our analysis.

All data is preprocessed by an off-line application that transforms it into a structured format, called arff. Once we have obtained the arff file we may start the analysis.

There are many machine learning algorithms that may be used. In this paper we focused on techniques that may be applied in order to improve the accuracy of models obtained by using one algorithm. We used state-of-the-art methods of combining models of analysis like bagging, boosting and stacking.

Our dataset consisted of 375 instances represented by students that were registered as students within e-Learning platform. For each student the activity was represented in terms of four parameters: the number of loggings, the number of taken tests, the average of taken tests and the number of sent messages.

Bagging, boosting and stacking techniques were used with the aim of creating models of data representation with greater accuracy.

Bagging exploited the instability that is inherent in learning systems. Combining multiple models helps when these models are significantly different from one another and each one treats a reasonable percentage of the data correctly. Ideally the models complement one another, each being a specialist in a part of the domain where the other models don't perform very well.

Boosting produced a classifier that was more accurate than one generated by bagging. However, unlike bagging, boosting sometimes generates a classifier that was significantly less accurate than a single classifier built from the same data. This indicates that the combined classifier overfit the data. The time sent for building the model is greater for boosting that for bagging although the algorithm complexity is the same. The difference comes from computational complexity which is greater in boosting due to the weight introduced as parameter for each instance.

The best performance regarding accuracy was obtained by using stacked generalization. Combination of three different types of learning algorithms proved to achieve better classification accuracy than both previous ways of combining models (bagging and boosting) that used only one type of learner. The obtained performance was obtained with a high cost regarding computational time.

The obtained accuracy is the guarantee that obtained knowledge from the analysis process is valid and may be used together with domain knowledge to improve the performance of the e-Learning platform.

In future, we plan using the same platform for other students. It is our primary concern to create analysis models that are able to classify students according to their accumulated knowledge. The platform, represented by the entire infrastructure (disciplines, course materials, and test and exam questions) represents an invariant. On the same platform setup, different methods of analyzing student's activity may be employed. Future work will take into consideration other ways of combining different models of analysis that have good accuracy and produce knowledge that may be used to improve our e-Learning system. Changes that are made at platform's infrastructure should be noted very carefully and analysis process should be repeated in order to look for correlations. An interesting thing would be to evaluate the analysis process on data from other e-Learning systems.

REFERENCES

Olivia Parr Rud, "Data Mining Cookbook – Modeling Data for Marketing, Risk, and Customer Relationship Management", Wiley Computer Publishing, 2001.

Jiawei Han, Micheline Kamber "Data Mining – Concepts and Techniques" Morgan Kaufmann Publishers, 2001.

Ian H. Witten, Eibe Frank "Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations" Morgan Kaufmann Publishers, 2000.

http://www.cs.waikato.ac.nz/ml/weka

R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," Proc. of the 20th VLDB Conference, pp. 487-499, Santiago, Chile, 1994.

Nasraoui O., Joshi A., and Krishnapuram R., "Relational Clustering Based on a New Robust Estimator with Application to Web Mining," Proc. Intl. Conf. North American Fuzzy Info. Proc. Society (NAFIPS 99), New York, June 1999.

B. Mobasher, N. Jain, E-H. Han, and J. Srivastava "Web mining: Pattern discovery from World Wide Web transactions," Technical Report 96-050, Universityof Minnesota, Sep, 1996.

R. Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

Schapire, R.E., Y. Freund, P. Bartlet, and W.S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods", Proc. Fourteenth International Conference on Machine Learning, Nashville, San Francisco, 1997.

Wolpert, D.H., "Stacked generalization", Neural Networks, 1992.

http://stat257.central.ucv.ro/

B. Cestnik, "Estimating probabilities: A Crucial Task in Machine Learning", Proc. of European Conference on Artificial Inteligence, 1990.

Ting, K.M. and Witten, I.H., "Stacked generalization: when does it work?" Proc International Joint Conference on Artificial Intelligence, pp. 866-871, Japan, August, 1997.

Leo Breiman, "Stacked regression", Machine Learning, Vol. 24, pp. 49-64, 1996.

Michael LeBlanc, Robert Tibshirani, "Combining Estimates in Regression and Classification", Technical Report 9318, Department of Statistics, University of Toronto, Canada, 1993.