

Using Action Analysis in ActiveMath to Estimate Student Motivation

Martin Muehlenbrock & Oliver Scheuer

German Research Center for Artificial Intelligence DFKI
66123 Saarbruecken, Germany
Martin.Muehlenbrock@dfki.de, Oliver.Scheuer@dfki.de

Abstract

Recently, there is a growing interest in the automatic analysis of learner interaction data with web-based learning environments. The approach presented in this paper aims at helping to establish a basis for the automatic analysis of interaction data by developing a data logging and analysis system based on a standard data base server and standard machine learning techniques. The analysis system has been connected to a web-based interactive learning environment for mathematics teaching, but is designed to allow for interfacing also to other web based learning environments. The system has been used to correlate student logs from an educational technologies course with a motivation assessment.

1 Introduction

Recently, there is a growing interest in the automatic analysis of learner interaction data with web-based learning environments. This is largely due to the increasing availability of log data from learning environments and in particular from web-based ones. The objectives include the detection of regularities and deviations in the learners' or teachers' actions among others, and to support teachers and learners by providing them with additional information to manage their learning and teaching, respectively, and possibly suggest remedial actions. Commercial systems such as WebCT, Blackboard, and LearningSpace already give access to some information related to the activity of the learners including some statistical analyses, and provide teachers with information on course attendance and exam results. With this information already being useful, it only represents the tip of iceberg of what might be possible by using advanced technologies.

This upcoming field, i.e., addressing the automatic analysis of learner interaction data, is related to several well-established areas of research including intelligent tutoring systems, web mining, and machine learning, and can build upon results from these fields for achieving its objectives. In contrast to intelligent tutoring systems, learner interaction analysis does not rely on models of learner or domain knowledge since these are heavy to build and maintain. In this regards, learner interaction analysis is comparable to website data mining, but with a specific perspective on learning settings and with the availability of pedagogical data that usually are not available in web mining applications that

are mostly based on click through data. Click through data streams only allow for a rather shallow analysis, but with the inclusion of pedagogical data also more advanced techniques can be adopted from the field of machine learning.

Although a number of open questions have already been tackled (Arroyo et al., 2004, Heiner et al., 2004, Merceron & Yacef, 2003, Merceron et al., 2004, Mustow, J., 2004, Oliveira & Domingues, 2003, Zhang & Lu, 2001), there is not yet a systematic approach in analysis interaction data from huge learner action logs. The approach presented in this paper aims at helping to establish a basis for the automatic analysis of interaction data by developing a data logging and analysis system based on a standard data base server and standard machine learning techniques. The analysis system has been connected to a web-based interactive learning environment for mathematics teaching, but is designed to allow for interfacing also to other web based learning environments. The system has been tested with a medium scale experiment in which four classes of a secondary school participated throughout a school term of five months on a weekly basis.

2 System

ActiveMath is a web-based learning environment that dynamically generates interactive courses adapted to the student's goals, preferences, capabilities, and prior knowledge (Melis et al., 2001, Melis & Ullrich, 2003, Ullrich et al., 2004). The content is represented in a reusable XML-knowledge representation specifically designed for an educational context. ActiveMath supports individualized learning material in a user-adaptive environment, active and exploratory learning by using (mathematics) service tools and with feedback, better reusability and interoperability of the encoded content and exercises. For different purposes and for different users the learning material and its presentation can be adapted: the selection of the content, its organization, the means for supporting the user have to be different for a novice and an expert user, for an engineer and a mathematician, for different learning situations such as a quick review and a full study. Since there is no way of knowing in advance the goals, the profile, and the preferences of any user when designing the system, ActiveMath builds on adaptive course generation.

For each learner, the ActiveMath environment generates an online log that lists all user actions in the learning environment in terms of general information

such as time, type of action, user name, and session number, as well as specific information including which page has been presented to the user, which item has been seen by the user, which exercise has been tackled and solved or not solved. A recent implementation of the learning environment also provides information on user actions in terms of events that other system components can subscribe to. The analysis system is comprised of three major components, i.e., the log da-

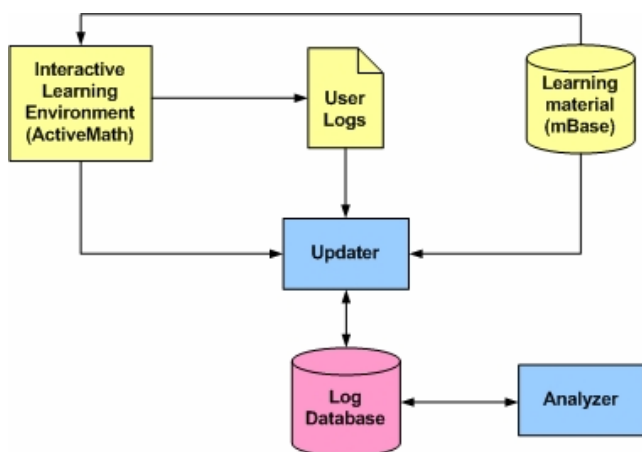


Figure 1. System architecture

tabase, the updater, and the analyzer.

- The **log database** is at the center of the action analysis system. It contains not only representations of the raw data in the user logs, but also has tables that hold the results of the analysis as well as tables for additional background knowledge concerning users or courses among others (see in Figure 1, which will be detailed below).
- The **updater** receives event information on the users' actions from the learning environment, and transforms every user event into one or more corresponding database tables. Usually, the updater receives the information online from the event queue, but it can also read in files with log data that have been generated in an offline mode. In addition to updating the event information in the database, the updater also enhances and extends the event data, as will be described below.
- The **analyzer** performs data aggregation and evaluation in terms of queries to the log database as well as incorporates a number of learning methods and takes the data from the log data base as an input. If needed, adjustments and preferences are input by the user that is running the analysis. An example will be given in section 4.

The analysis system has been implemented by using standard technology such as Java and MySQL, which are available for a number of platforms and operating systems, together with the suitable drivers for database connectivity. In addition, the analyzer is based on the Weka toolkit (Witten & Frank, 1999), which provides tools for visualizing and exploring data as well as

means for integrating machine learning functionality into applications.

The basic level of the database, which corresponds to the raw log data, is organized in tables that represent generic event information as well as event-specific data. The structure of these tables has been designed closely to the events specification, since this allows for simpler updating operations when the event subsystem is changed or replaced by another system. Table 1 lists the basic event tables together with their fields and a short description. In addition to these tables, the database includes tables that hold additional information on the users and sessions such as gender and holiday periods, respectively, as well as tables that are derived from these by means of database queries.

Table 1. Log database schema for basic level

Table	Description	Fields
event	<i>generic information for all events</i>	eventId ts source session userId type
eventUserCreated	<i>registration of a new user to the system</i>	eventId userName
eventUserLogged-In	<i>user logs into the system; start of a new session</i>	eventId ip userAgent
eventUserLogged-Out	<i>user logs out of the system; end of a session</i>	eventId
eventPage-Presented	<i>system presents a requested book page to the user</i>	eventId book page
eventExercise-Started	<i>user starts an exercise</i>	eventId exercise
eventExerciseStep	<i>user submits an input for an exercise</i>	eventId userInput
eventExercise-Finished	<i>exercise is finished</i>	eventId exercise successRate
eventMastery-Changed	<i>system changes the mastery level of an user</i>	eventId item mastery-Dimension oldValue newValue
eventItemPresented	<i>presentation of a learning item to the user</i>	eventId item itemType
eventUserProperty-Changed	<i>indicates changes of some user meta data</i>	eventId propertyName oldValue newValue

Table	Description	Fields
eventItemSeen	<i>gives a more fine-grained resolution (on item level) of the user's focus (uses information of an eye-tracker)</i>	eventId duration item
motivation-Questionnaire	<i>results of the questionnaire</i>	userId question answer
Motivation-Questionnaire-Dimensions	<i>maps question ids to the corresponding dimension</i>	question affective- Dimension

3 Experiment

The action analysis features have already been investigated in an experiment in a secondary school with about 70 students from three different classes that used the learning environment for a period of five months. The subject area was fractions and divisibility. In particular, the students' results in a written post test have been analyzed in relation to their activity in the ActiveMath system (Muehlenbrock, 2005).

A further investigation has been conducted in the context of a course on educational technologies at the Saarland University with the objective to relate students' motivation to their behavior in the ActiveMath system. The 15 students, all majoring in computer science, had been introduced to ActiveMath system as an example for an educational system, and were told to explore the system to improve their understanding. They were provided with an extensive course on mathematical calculus which was relevant and interesting to them.

The students were asked to answer a questionnaire aimed at measuring four motivational and affective dimensions, i.e., a goal orientation towards mastery, performance, performance avoidance, or utility. Mastery goal orientation is related to the desire of developing skills and improving achievement, performance goal orientation refers to the desire of outperforming others and appearing smart, performance avoidance goals is different from performance approach goals with maladaptive behaviors only being attributed to the former, and utility goal orientation refers to future goals such as career goals that can be related to achievement (Proske et al., 2003).

The questions are a subset of the Patterns of Active Learning Survey (PALS) (Midgley et al., 2000), with five questions for each of the four dimensions. The questionnaire corresponds to one that had been used in an investigation at Dresden Technical University (Proske et al., 2003), with using the questions in their original, English versions since the educational technologies course was taught in English.

The items have been statements and the students had to express their degree of agreement with these statements. For each question they selected one of six possible scores, ranging from 1 (not at all true) to 6 (very true). All of the 15 students have filled out the questionnaire. Figure 2 shows the students' average weight-

ing concerning the four different orientations. In general, it seems that the most important concern of the students is to increase their mastery level, with utility and performance-related (positive or negative) aspects being of lesser importance.

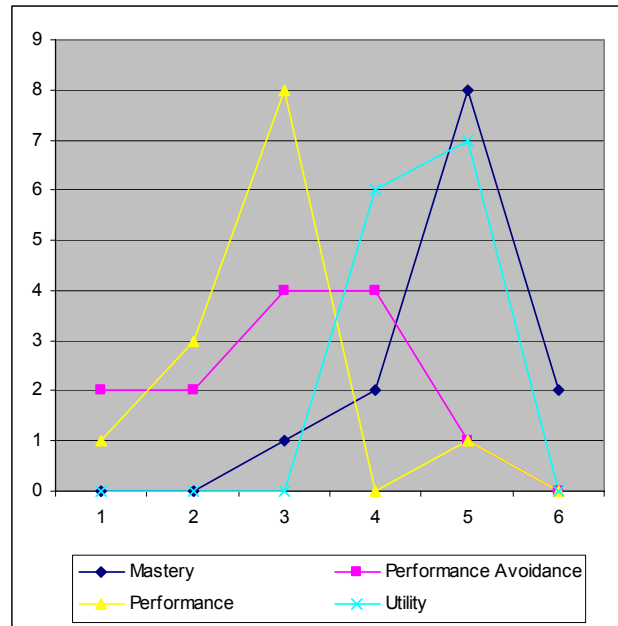


Figure 2. Results of the questionnaire

The students worked with the ActiveMath system during two weeks to increase their understanding of educational software. From the log data it can be seen that actually 13 students in total had been using the system. From the 22100 events in the log data base, 700 can be traced back to actual user actions because their types can be seen as reliable indicators (events PagePresented, ExerciseStarted, ExerciseStep, DictSearched, UserPropertyChanged, UserLoggedIn, and UserLoggedOut).

From the basic level of log data, a table with higher level information has been generated by accumulating information from several tables as well as incorporating further information that had been manually input such as gender. Table 2 shows all fields of the user information table. This table has been used for further analysis by decision tree induction as described in the following section.

Table 2. Table for user information to be used for further analysis

Table	Attribute	Description
userInfo	dictSearch	# dictionary look ups
	exerStarted	# exercise started
	exerFinished	# exercise finished
	exerCorrectRatio	sum exercise success / num exercise finished
	pagePresented	# pages presented

Table	Attribute	Description
	pageRepetitionSquare	sum over squared number of repetitions of all pages
	avgExertime	average exercise duration
	gender	student's gender
	totalEventsFired	# events
	motivationAbsolute	assigned motivational category

4 Analysis

The analyzer component incorporates a number of machine learning methods for automatically analyzing the data in the log data base. In addition to getting a better insight into the underlying relationships in the data, this also allows for prediction and classification of future sessions. Many machine learning methods provide their output in an intelligible, human readable form. For instance, methods for generating decision trees from data, such as C4.5 (Quinlan, 1993), allow for a tree-shaped representation of the learning results.

The aim of using the decision tree method was to characterize the students' motivation in terms of the other attributes that are automatically generated from the log data, in order to provide an abstracted view on the underlying data as well as to allow for predicting motivational aspects in other students.

In order to be used for the decision tree learning, the data from the questionnaire was prepared by assigning to each student the dominant goal orientation, which is one of mastery, performance, performance avoidance, and utility (see also last row in Table 2). In case of ambiguities concerning this absolute dominant motivation for a given student, i.e., two or more dimensions have the same, highest value, the one was chosen for which the relative dominant motivation as compared to the average among all students for the same dimension was higher. This had to be done in one case only. Not surprisingly, it turned out that most student were mastery oriented. Furthermore, some were utility oriented, one was performance oriented, and none was performance avoidance oriented. Overall this means 9 mastery, 3 utility, and 1 performance orientations.

Figure 3 shows the decision tree that was generated by the system for characterizing the dominant motivation (attribute motivationAbsolute). According to this decision tree, the attributes exercise correct ratio is the most important and second most important features to classify the motivation, with students having less than 25% of exercises correct being classified as performance oriented. Otherwise, the decision tree classifies students with a small number of pages read or, otherwise, with no exercise finished, as utility oriented. All other students, would be classified as mastery oriented, in particular those with more than 25% correct answers and having read more than eight pages.

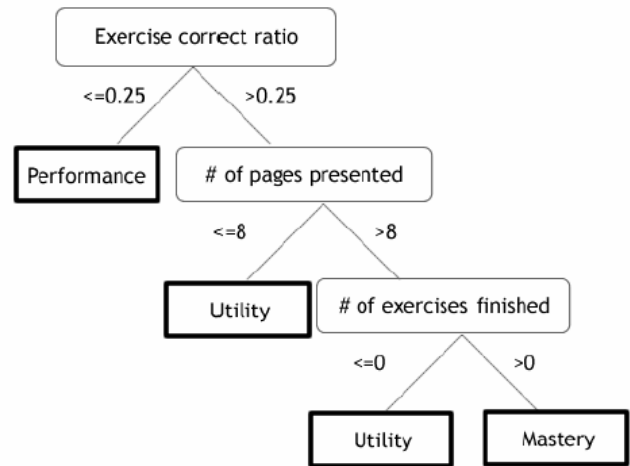


Figure 3. Decision tree for characterizing student motivation

The quality of the decision tree is presented in Table 3 as a confusion matrix. A confusion matrix displays the result of testing the decision tree with data as a two-dimensional matrix with a row and a column for each class. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column, which has been produced by using stratified fourfold cross-validation. The confusion matrix indicates quite a good quality of the decision tree by having most examples classified correctly as indicated in the diagonal (70%), and only some incorrectly classified examples (30%).

Table 3. Confusion matrix with fourfold cross validation for the decision tree

	Mastery	Performance Avoid	Utility	Performance
Mastery	8	0	1	0
Performance-Avoidance	0	0	0	0
Utility	2	0	1	0
Performance	1	0	0	0

How are the results to be interpreted? Since the decision tree was derived from a small set of examples only, they have to be taken with a pinch of salt. In addition, there were derived from students exploring the system features, and not actually learning the material that was presented. However, as indicated by the decision tree, those students that would have a reasonable number of exercises correct and have looked at considerable number of pages are said to be mastery oriented. In the context of exploring system features, this could mean that they are not just playing around with the system but are actually trying to understand how it works, thereby improving the mastery and skills. In contrast, the performance oriented students are not in a situation where they can compare their results to those of other students, hence they tend to invest less effort in solving the exercises and getting only a small number correct.

5 Summary and further work

In this paper a system is presented for the automatic analysis of user actions in web-based learning environments. It has been used in an investigation with students from an educational technologies course at the Saarland University to explore user behavior in ActiveMath in relation to motivational aspects. A decision tree has been generated that reveals some interesting underlying aspects in the log data, which can be used for prediction with new users of the system.

Further work will include a large scale investigation based on log data from about 250 students attending a course on mathematics for computer scientists, with a pilot study in August and the main study starting in September.

Acknowledgements

The students Jakub Petrykowski, Jan Otop, and Radoslaw Kotowicz contributed in generating the decision tree. The work presented in this paper is partially supported by European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD - project iClass contract IST-507922.

References

- Arroyo, I., Murray, T., Woolf, B. P. (2004). Inferring unobservable learning variables from students' help seeking behavior. In *Proceedings of the workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, pages 29-38, Maceio, Brasil, August.
- Heiner, C., Beck, J., Mostow, J. (2004). Lessons on Using ITS Data to Answer Educational Research Questions. In *Proceedings of the workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes at ITS 2004*, pages 1-9, Maceio, Brasil, Aug.
- Melis, E., Büdenbender, J., Andres, E., Frischauf, A., Gogvadze, G., Libbrecht, P., Pollet, M., and Ullrich, C (2001). ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, 12(4): 385-407.
- Melis, E. and Ullrich, C. (2003). How to teach it - Polya-inspired scenarios in ActiveMath. In *Proceedings of the International Conference on AI in Education AIED-2003*, pages 141-147, IOS Press.
- Merceron, A. and Yacef, K. (2003). A Web-Based Tutoring Tool with Mining Facilities to Improve Learning and Teaching. In *Proc. of the 11th International Conference on Artificial Intelligence in Education AIED 2003*, pages 201-208, IOS Press.
- Merceron, A., Oliveira, C., Scholl, M., & Ullrich, C. (2004). Mining for Content Re-Use and Exchange - Solutions and Problems. In *Poster Proceedings of the 3rd International Semantic Web Conference, ISWC2004*, pages 39 - 40, Nov.
- Midgley, C., Maehr, M. L., Hruda, L. Z., Anderman, E., Anderman, L., Freeman, K. E. et al. (2000). Manual for the Patterns of Adaptive Learning Scales. Ann Arbor, MI: The University of Michigan
- Mostow, J. (2004). Some Useful Design Tactics for Mining ITS Data. In *Proceedings of the workshop Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes at ITS 2004*, pages 20-28, Maceio, Brasil, Aug.
- Muehlenbrock, M. (2005). Automatic action analysis in an interactive learning environment. In *Proceedings of the workshop on Usage Analysis in Learning Systems at the 12th International Conference on Artificial Intelligence in Education AIED-2005*. Amsterdam, The Netherlands, July.
- Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA.
- Ullrich, C., Libbrecht, P., Winterstein, S., and Muehlenbrock, M. (2004). A flexible and efficient presentation-architecture for adaptive hypermedia: Description and technical evaluation. In *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies ICALT 2004*, pages 21-25, IEEE Computer Society.
- Oliveira, C. and Domingues, M. (2003). Data Warehouse for Strategic Management of an eLearning System, In *Proceedings of: the 2nd International Conference on Multimedia and ICT in Education m-ICTE 2003*, pages 627-631. Badajoz, Spain, Dec.
- Proske A., Narciss S. and Körndle H. (2003). Relationships among student's motivation, learning behaviour and performance in Web-based learning. In R. Martens, editor, When does e-learning work and when does it fail? Searching for critical factors. Paper presented in a symposium conducted at the 10th EARLI Conference, Padova, Italy.
- Witten, I. H., Frank, E. (1999). *Data Mining*. Morgan Kaufmann, San Francisco.
- Zhang, J. and Lu, J. (2001). An educational data mining prototype. In *Proceedings of the 10th International Conference on Artificial Intelligence in Education AIED 2001*, pages 616-618, IOS Press.