Towards Intelligent Adaptative E-Learning Systems – Machine Learning for Learner Activity Classification

Mirjam Köck
Johannes Kepler University Linz
A-4040, Linz, Austria
koeck@fim.uni-linz.ac.at

Abstract

As adaptivity in e-learning systems has become popular during the past years, new challenges and potentials have emerged in the field of adaptive systems. Adaptation, traditionally focused on the personalization of content, is now also required for learner communication and cooperation. With the increasing complexity of adaptation tasks, the need for automated processing of usage data, information extraction and pattern detection grows. We present learner activity mining and classification as a basis for adaptation in educational systems and discuss intelligent techniques in this context. Based on real usage data, we present the results of experiments comparing the behaviour and performance of different classification algorithms.

1 Introduction

This paper discusses how the goal of intelligent adaptive e-learning systems can be approached with the help of learner activity classification. Intelligent adaptive e-learning combines topics, issues and characteristics of various fields: e-learning, adaptivity and Computational Intelligence (CI).

E-learning environments have become fairly popular, as learning scenarios have been radically developing towards e-learning and blended learning during the past years. Almost every educational institution applies e-learning to a certain extent. Related systems usually include different kinds of facilities: tools accompanying learning content like exercises or assignments; and, communication tools like chat, fora or private messaging.

Adaptive systems offer various kinds of adaptation and personalization, most of which restricted to content (e.g. personalization of learning paths, recommendations of topics, and in some cases also a personalized view of the content). Mostly, adaptive e-learning systems have rather limited support for communication facilities and do not extend adaptation efforts to them. Recent attention to adaptive support for collaboration (see e.g. [Soller, 2007]) has been concentrating on research systems and has not been fully exposed to a large community as of yet.

Intelligent systems have been in the focus of attention for a longer period of time. Research combines principles like evolution, learning, in some sense also adaptation, fuzzy logic, etc. Intelligent systems are designed to simulate human reasoning and learning, reducing the need for human intervention in the application process. CI is promising for the further evolution of adaptive systems, especially in the context of e-learning where different learning theories [Lefrancois, 2006], [Prince and Felder, 2006], learning styles [Felder and Brent, 2005], and social processes [Vathanophas *et al.*, 2008] need to be addressed. As pointed out in [Brusilovsky and Peylo, 2003], educational systems are traditionally either intelligent or adaptive, listing prominent systems (like AHA! [De Bra and Calvi, 1998]) as adaptive but non-intelligent, and other ones as intelligent but limited regarding adaptivity.

In this paper, we focus on intelligent adaptive e-learning systems. Our approach relies on mining and processing of usage data. Usually, although activity data is monitored by the system, high levels of human intervention are required to process and use such data to achieve high-quality adaptation. We introduce an approach that is based on intelligent techniques for the classification of user activity data in e-learning environments and aims to largely supplement or even replace human efforts in this context.

The rest of this paper is structured as follows. Section 2 describes the state of the art and lists common problems in prevailing adaptive e-learning systems. Section 3 explains our classification strategy and how it can address the aforementioned issues. Section 4 compares statistical and CI-based approaches in the context of classification. Section 5 describes related experiments that were run to measure the performance of intelligent classifiers on learner activity data tasks. We summarize related work and give an outlook on future work in Sections 6 and 7.

2 The Adaptive E-Learning System - Two Pieces or One Whole?

Most e-learning systems consist of various kinds of tools which can be roughly categorized as learning facilities and facilities supporting communication/cooperation processes. In non-adaptive systems, tools are naturally independent. In adaptive ones, tools may require communication with other tools and/or a central service (e.g., an adaptation engine). In theory, this would be the basis for an integrated environment using knowledge gained in any of its facilities for system-wide adaptations.

Nevertheless, adaptive systems in the field of e-learning have been concentrating until now on some specific kinds of adaptation. In general, we can distinguish between adaptive navigation and adaptive presentation support [Brusilovsky, 1996]. These techniques are based on a user's interaction history within the system or additional information provided explicitly. They are well established, but, when it comes to e-learning, they have been primarily used to adapt content only.

Adaptation is often based on knowledge a system obtained from a user's interaction history, and that is then utilized to predict future activities which in turn become the basis for recommendations. At the moment, this information is typically not shared between different components of a system. For instance, a user's previous behaviour in the content facilities of a platform is only used to further adapt the content to the user's needs but not considered for guidance in communication tools. Therefore, adaptive e-learning systems are often not perceived as fully integrated, but rather an assembly of two independent pieces of a puzzle. A new approach would be to establish a shared pool of adaptation knowledge which is contributed to, and queried by all of a system's components.

3 Learner Activity Classification

Our general idea aims at introducing new kinds of adaptation in e-learning systems, bridging the common gap between content and communication facilities. Here, we approach this aim using activity mining and classification.

3.1 Activity Data

If we want to offer recommendations related to communication and learning content, we need to infer a user's level of interest in specific topics. Therefore, we examine a user's history on the system and use previous interests to predict future ones. We can shortly outline the concept as follows. First, we collect a user's passive ("consumption") activities. We will further also refer to this kind of activities as "read activities". Reading an element (e.g. an entry in a forum or a document) denotes a user's interest. If a user was interested in one specific element, we can find similar ones and assume that these are also interesting for this user. Given this kind of "knowledge", we can try to infer user interest for as many events as possible which can then become the basis for adaptation. This general idea can be put into practice by several different implementation approaches (see also Section 4) which provide different quality and granularity of results. All of them have in common that the primary objective is to classify data continuously produced by users' activities on a platform.

3.2 Classification Levels

We distinguish between two different levels of classification: classification of individual user activities, and classification of user activities considered as an interrelated construct. The first kind, as opposed to the second one, treats activities as if they were independent. The second kind is promising for modeling dependencies between users, tools, etc. but it requires a higher amount of reference information. We concentrate on the first kind here, which can partially be done before the system has collected enough information to generate reference constructs. It does not consider the time context of, and relations between, activities but uses activity items as independent of each other. Nevertheless, in most cases (depending on the learning technique) the system must still be provided a certain amount of reference data before classification of fresh data can be performed. This means that, in this case, no long period of training is necessary as long as some representative data sets are available. Therefore, the only prerequisite for this kind of classification is a certain period of data collection (depending on users' level of activity). Classification of independent activity items can be useful at the level of both individual users and groups (see also Section 7).

3.3 Application in Adaptive E-Learning Systems

First of all, we want to provide adaptation which closes the gap between learning facilities and those supporting communication and collaboration. This can be done by extracting information of all facilities, feeding it to one shared model which is then again queried by all facilities. Regarding the first level of classifictaion, we aim at recommending both communication threads and learning content items, based on a user's previous interests. For the second level of classification, our main application idea is closely related to group work. We want to be able to determine users' collaboration behaviour and their roles in group structures in order to recommend group constellations the system predicts successful on the one hand and interesting communication partners for individual users on the other.

4 Statistical vs. Intelligent Approaches

In order to classify independent user activity items we have to find an approach that is capable of computing realistic values for every user's interest in an event. There are several ways of approaching this, basically statistical and "intelligent" ones. The main characteristic of intelligence in this context is that the respective approaches are capable of learning, which is not possible for statistical ones. The statistical approach will work for some scenarios (in [Jung *et al.*, 2005], the authors introduce a statistical model for user preferences which performs well) but it can turn out to be too inflexible in others.

The aim is to not only determine interest, but also, on a higher level, provide recommendations of specific communication threads, learning material, etc. Knowledge about users gained in any of the platform's areas (communication or learning content) should be combined for the computation of interest levels. And finally, the system should of course continuously adapt to users' behaviour, i.e. all new actions must be considered. The following sections provide an introduction to each of the two approaches.

4.1 A Statistical Approach

This approach uses a statistical formula to compute a user's interest level for an item. The formula considers past user interest (indicated e.g. by read activities) to compute statistics which then becomes the basis for further prediction. First, the distribution of a user's read activities among tools in a site has to be computed. Basically, standard statistical metrics like mean, standard deviation, and variance are used to determine probability/density distributions. Given only the mean, we would face the problem of statistical outliers distorting the overall picture. This can be partially solved by considering the standard deviation (or variance). Given standard deviation, a tool's deviation σ_{T_x} from this value can be used to identify significant (in both directions) tool results. Consider the following simple example using 5 hypothetical tools and 25 read activities produced by 1 user within our time frame, distributed among the tools as $c_1 = 10$, $c_2 = 2$, $c_3 = 4$, $c_4 = 3$, $c_5 = 6$. Consider, we want to compute this user's interest value for every tool. This would result in the following:

$$\bar{x} = 5$$

$$\tilde{x} = 4$$

$$\sigma^2 = \frac{1}{5} * \sum_{i=1}^{5} (x_i - \bar{x})^2 = 8$$

$$\sigma \approx 2,83$$

In a next step, a tool-specific metric can be determined as

$$\sigma_{T_x} = |c_x - \bar{x}| - \sigma$$

which will mark all resulting $\sigma_{T_x} > 0$ as significant (in both directions). In our example σ_{T_1} and σ_{T_2} will be positive values, marking T_1 as significantly high (as $c_1 > \bar{x}$) and T_2 as significantly low (as $c_2 < \bar{x}$) regarding interest. This (simplified) approach can be improved, e.g. by adding weights, and in theory this improved version might be sufficient, but it still carries some non-obvious risks. For instance, a statistical formula, even if it contains variable elements, is inflexible, meaning that the core does not change for different scenarios. We have to be aware that users may behave differently in their interest across sites, tools or resources. There can be courses where communication plays a more important role than educational content, and users might differ in their communication and learning behaviour in several ways. Furthermore, we may want to weigh read activities differently based on the time when they occurred (e.g. if the timespan between the related "active" create event and the read activity is relevant).

In order to consider all of these factors, the formula might have to look different for different combinations of users, tools, courses and resources. This is hardly possible, and even if it was, it would still lack the ability to continuously and individually adapt to a user's behaviour.

4.2 A Flexible, Self-Learning Approach

In order to overcome issues and problems raised by purely statistical approaches, classification techniques of the field of machine learning can be used. These techniques do not make as many semantic assumptions as statistical approaches do, but learn from the user. Although the classifiers we used for our experiments (see a detailed description in Section 5) differ drastically in their way of model building, they have in common that their models consider all features we provide as input. In our case, 8 attributes are available, 6 of which (the anonymized user id, event id, tool id, site id, related resource and the interest class) are taken into account by the classifiers. The remaining two, index and timestamp, were removed by a filter in preprocessing because we do not consider temporal relations for this kind of classification yet. This means, all solutions we get dynamically adapt to all feature values of new input events. Thus, not only the site where the event occurred is considered, but also e.g. its creator and the tool where it originated. To further extend flexibility and personalization, the classifier then computes an event's interest value for every user individually. This implies that the approach works separately for every user. As the classifier is continuously fed with new information, it is able to learn and adapt its behaviour during the process. As each of the classifiers builds a model (e.g. a decision tree, a Bayesian Network, or a rule base) which can be queried, it is also possible to extract semantic information from it which will offer additional knowledge about users, behaviours, and the whole construct of content, courses and tools. In addition, dependencies and correlations between attributes could be found which might become important for further event design. Especially the opportunity to gain semantic information from the model built by a classifier is a significant advantage compared to a statistical approach, as the latter is limited to strict one-way information exchange, i.e. no information can be extracted from statistics in a way it can go back into and enhance the user model.

5 Experiments

This section describes experiments designed to test our classification approach on real user activity data, compare the performance of different techniques for different aims, and show how classification can improve activity-based adaptation. The experiments aim at producing a group-based interest model. In general, we can distinguish between user- and group models. A user model is created for every user individually and only fed with information about that specific user. A group model pictures group behaviour, i.e. activities of multiple users which were clustered into groups (e.g. , based on similarities, or a given course context). Our system is fed with all users' activities and tries to classify new events as interesting or non-interesting for every user individually, but uses this knowledge to build a shared model. This model will be referred to in later stages of our work to offer group-based adaptations. We can benefit from working with group models in several ways. For example, to avoid the "cold start" problem [Höök, 1997], a group model can become the default for a new course participant. The system then does not have to create new models from scratch any more but can build upon one based on the interest and activities of a group working on the same content and tasks.

5.1 Setup

Our experiments outline an extension to the behaviour of the "recent activity tool" [1] which is an add-on to the e-learning platform Sakai [Sakai, 2009]. This tool provides an overview of recent activities in various Sakai tools. It includes a personalized view marking activities as interesting for the current user, and a personalized RSS feed. The tool adapts at the user-level only at the moment. Adaptation is not done before the system has received a sufficient amount of information about the user. Recommendations are based on a statistical model similar to the one described before. Thus, the adaptive part of the recent activity tool relies on assumptions and generalizations to a certain extent. As already described, CI techniques can improve the performance, flexibility, and accuracy of adaptive components because they learn from the user instead. Our experiments use these techniques to replace the statistical model. Real usage data is provided by a monitoring extension to Sakai. The instances are independent and handled as random set elements for the first run of experiments. Yet, they contain information which can help to create relations in further post-processing.

The overall data set contains 4967 instances with 6 features as described in Section 4.2. String attributes were normalized to nominal ones, meaning that before data went into the classifiers, a filter collected all possible values for a feature (for instance, all tools where activity was monitored). The resulting finite set of values then allows for better computation of probabilities.

The experiments were run on data of one specific course about the Unified Modeling Language, with 31 participants in total. Data was collected over a period of several months and went through some preprocessing during which irrelevant or pseudo-data (e.g. produced by test users) was removed. During these steps anonymization was also performed by encoding user IDs with a one-way hashing algorithm.

[1] The recent activity tool was developed in the context of the Adaptive Learning Spaces (ALS) project. For further information, please refer to http://www.als-project.org

5.2 Process and Technologies

Experiments were carried out iteratively with training, testing and evaluation steps repeated for different classification algorithms. Validation was performed in two different ways – by 10-fold cross-validation, and by specifically split training- and test sets. A comparison of the algorithms' results concludes the experiments and becomes the basis for classifier rating and final selection. We used the Weka [Witten and Eibe, 2005] machine learning software to run the experiments. For a more detailed description of the algorithms please refer to Weka documentation and tutorials. The following paragraphs describe the configuration of the classification algorithms which were used.

Naïve Bayes: The naïve Bayesian approach builds a simple network with one parent node (the class label, in our case the interest value). There are no important additional configuration alternatives.

Bayesian Network: The network applied for the experiments uses the SimpleEstimator approach for finding the conditional probability tables of the net. The TAN algorithm (determining the maximum weight spanning tree and returning a Bayesian Network augmented with a tree) is applied for searching network structures.

SMO (Sequential Minimal Optimization): SMO is used to train a support vector machine. We used standard settings with relatively low complexity (the higher, the fewer wrong classifications are accepted) and a polynomial kernel $K(x, y) =< x, y >^p$ with exponent $p = 2$.

Multilayer Perceptron (Backpropagation Neural Network, later referred to as *NN*)**:** We used a network with $a = attributes + classes$ hidden layers of sigmoid nodes, a learning rate of 0.7, momentum of 0.2 and 500 learning cycles. Please note that run-time filters like nominal to binary slow down the process significantly.

IBk (Nearest Neighbour): We used $k = 10$ and the LinearNearestNeighbourSearch (brute force) algorithm for nearest neighbour search and cross-validation.

JRip (Rule-based): This algorithm implements a propositional rule learner and provides a set of rules which are then used as a basis for classification decisions. Our experiments use 10 folds (for pruning and growing rules) and 6 optimization runs.

J48 (Tree-based): This algorithm, building a decision tree, uses a confidence factor (small values mean more pruning) of 0.25 and reduced error pruning here.

RandomTree: This algorithm, building a decision tree, uses a KValue (i.e. the number of randomly chosen attributes) of 1 and an unlimited tree depth.

5.3 Results

The results of the described base experiment are listed in Table 1 and Figure 1. The base experiment uses 10-fold cross-validation to get a first impression of the classifiers' performance. Subsequently, more specific experiments were conducted in order to find out how their performance changes over time. The experiments were conducted on a 2,98 GHz dual-core machine with 4 GB RAM, running 64-bit Windows XP. As a first experimental step, we compared the performance of different classification techniques to the performance of a statistical approach as described in Section 4.1. The percentage of correctly classified instances ranges from 96.63 (Naïve Bayes) to 98.41 (SMO) for the machine learning techniques. The statistical model obtains a result of 68.94%. In the following, we do a more detailed comparison of the classifiers listed in Section 5.2. The percentage of correctly classified instances from now on refers to "positive" instances (i.e. the instances with an interest value of 1) only. The overall results, containing "negative" instances also are less expressive, as the number of these instances is higher and their classification much easier (for the CI techniques only). This leads to a very similar overall performance of the classifiers and subsequently to a misleading picture and potentially wrong conclusions. The results show that the classification task itself can be handled relatively well by different classification techniques. As there is only little discrepancy regarding the number of correctly classified instances, process time becomes an even more important criterion. After running experiments with cross-validation, which provided a first general impression on the performance of machine learning techniques on our data, we modeled a similar experimental setup in order to measure how fast the classifiers learn from given input data. The experiments were run on split data (training set and test set). As we are building a group model, the percentage split

Table 1: This table lists classification results of various algorithms on the overall data set (10-fold cross-validation). The NN classifier is listed twice, once with filters. The table further displays the percentage of correctly classified positive instances, the True Positive rate for class 1, the Root Mean Squared Error, the time taken to build the model T_m, and the time taken for the overall process T_o.

Class.	***Corr.***	***TP***	***RMSE***	T_m ***(s)***	T_o ***(h,m,s)***
NB	47.6%	76.1%	0.1550	$< 0.01s$	$< 1s$
BN	70.0%	72.3%	0.1112	$< 0.01s$	$< 1s$
SMO	70.4%	84.5%	0.1261	$105.39s$	$16m54s$
NN(f)	70.3%	12.3%	0.1668	$17.22s$	$21h16m8s$
NN	56.0%	60.6%	0.1366	$17.16s$	$2m51s$
IBk	70.4%	84.5%	0.1092	$< 0.01s$	$5s$
JRip	68.0%	85.2%	0.1143	$0.63s$	$7s$
J48	70.7%	74.8%	0.1137	$< 0.01s$	$< 1s$
RT	70.4%	84.5%	0.1092	$0.13s$	$< 1s$

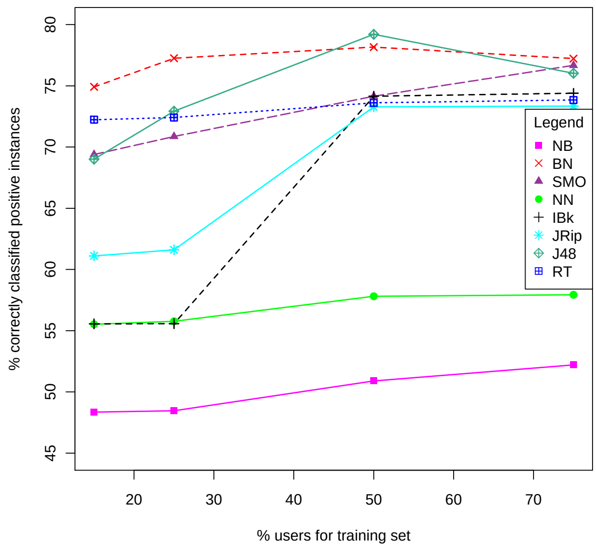

Figure 1: This plots show how the performance of classifiers increases with an increasing amount of training data.

for the data set is based on users, not resources. This means that the training set does not contain a certain percentage of the data but all data of a certain percentage of users. We ran the experiments several times with the events for 15%, 25%, 50%, and 75% of the users as training and remainder as test set. As depicted in the plots (Figure 1), the results show three different trends. Bayesian Network, RandomTree and J48 (the last two both tree-based approaches) show good classification performance right from the beginning and relatively steady behaviour. SMO could also be added to this "cluster" of algorithms, regarding its effectiveness. Next, we can see a second cluster containing IBk and JRip. These algorithms show good results but not right from the beginning. However, their plateau is at about the same place as the first cluster. The third trend can be seen in MultilayerPerceptron (NN) and Naïve Bayes which are steady in their performance but don't provide promising results. This means that for subsequent work we will not consider the classifiers of the third cluster. IBk and JRip will be further explored, but the most likely candidates are those in the first cluster, where the favourites are Bayesian Network and the tree-based classifiers. SMO, compared to the other classifiers, is relatively slow, with the time needed to build a model increasing at least linearly as the training set grows. In general, a linear algorithm is reasonable for run-time employment. In our case, comparing SMO to the faster classifiers, the discrepancy in computation complexity (< 0.01 seconds as opposed to 1.15 seconds for building the model for the smallest training set) is significant enough to be an exclusion criterion. SMO will be kept for further observation, but does not remain a first choice candidate.

Another important criterion for the selection of a classifier is in our case the possibility of information extraction, given a model. Descriptive classifiers like Bayesian Networks, rule- or tree-based approaches enable very simple extraction of semantic information, whereas function-based ones like neural networks or support vector machines tend to behave like blackboxes. Generally we can conclude that learning classifiers perform well on our data. Therefore, also considering the issues and potential problems and limitations of statistical approaches (see Section 4), these techniques are highly promising for our scenario and all subsequent ones operating on data of a similar structure.

6 Related Work

Our general approach is based on a combination of the fields of adaptive systems, e-learning and CI. Thus, we do not only have to consider challenges of the particular areas but also the potentials lying in the aggregation. This is not the first attempt pointing in that direction. For instance, our work relates to recent research issues in the field of adaptive collaboration support as described in [Paramythis, 2008]. In general, the matter of distributed collaboration entails some challenges. Their specific effects on the development process regarding adaptive support was e.g. elaborated in [Soller, 2007] where the author also describes relevant social processes. Additionally, personalization in distributed environments is further discussed in [Dolog *et al.*, 2004], where the authors introduce recent projects and also address personalization on the Semantic Web.

Regarding Machine Learning (ML), we can refer to research described in [Webb *et al.*, 2001], where the authors particularly treat student modeling and explicate specific requirements of this area. A concise overview on data mining techniques from the perspective of adaptive systems is given in [Voges and Pope, 2000].

Regarding the context of data mining in education, we find particularly interesting results in [Romero *et al.*, 2008], where the authors compare different algorithms to classify students. They also describe experiments aiming at predicting students' final grades based on usage data. The selected set of algorithms is partly congruent to ours, but operates at the user level instead of the activity level as in our approach, i.e. their data set contains items already aggregating information about user activities. This approach seems perfectly sound at the first glance, but it is less flexible as only a specific number of information elements can be considered which makes it hard to add further semantics later if necessary. As both approaches use semantically similar data but for different objectives and on a different level, it is very interesting to compare the results. Some trends can be found in both reports, whereas in other areas there is relatively high discrepancy. For example, the authors argue against e.g. Neural Network and Nearest Neighbour classifiers in their scenario in particular and data mining in general, due to the lack of comprehensibility. As these classifiers are in cluster two and three in our evaluation, we agree with them here, although the Neural Network achieved a better classification result on their data. Our second classifier in cluster three, Naïve Bayes was not included in their study. Tree-based classifiers performed very well in both cases. Unfortunately, the performance of Bayesian Networks cannot be compared because it was not included in the evaluation of Romero et al. However, they did an additional step of comparing the classifiers' performances on "plain" data to those after preprocessing and identified what classifiers can actually be improved by preprocessing, which we will consider during our next steps.

Further related work can be found in [Oakley *et al.*, 2004], where the author describes data-driven modeling of students' interactions, aiming at predicting students' ability to correctly answer a question and whether a student's interaction is beneficial in terms of learning. The experiments focus on Bayesian Network models. Additionally, we can also find interesting information in research on statistical approaches in machine learning, which is relevant for our approach because it identifies scenarios where statistical approaches work particularly well. Find a description of a statistical rule learning approach in [Rückert and Kramer, 2006]. In [Jung *et al.*, 2005], a detailed comparison of statistical and non-statistical approaches is given.

7 Conclusions and Future Work

As the experiments have shown, flexible classification approaches perform well on user activity data as produced on a learning platform like Sakai. The results can potentially be improved by combining (complementary) classifiers (using ensemble methods like bagging, boosting or stacking). The solution is not restricted to Sakai or even the recent activity tool, as data of any learning environment can easily be converted to a similar format. The intelligent classification approach is extendable in several ways. First, it can be applied on different levels, building models for individual users, groups or other clusters (e.g. any specifically interesting combination of features). Second, as described in Section 3, classification is not restricted to individually handled events, it can also be applied at the level of activity paths. These paths, representing a sequence of (related) instances, are a way of modeling relations between activities or any of their features. As a next step we will concentrate on modeling users and their collaboration behaviour with this approach. Several issues have to be considered:

Some factors in the path building process are strongly dependent on specific features. For instance, the timespan between the occurrence of subsequent events must be handled differently for various tools. In a synchronous communication tool, like a chat, an event which occurs hours after another one is more likely to be independent than in an asynchronous communication tool, like a forum, where the context is more important that time. In order to avoid wrong conclusions due to similar conditions, we have to set up a knowledge base containing factors and their respective values which may vary for different tools, etc.

What are the concrete questions we want to be able to answer given the path model? Before any design-specific decisions can be made, we have to define what should be modeled, like e.g. the level of communication between users or the context-based relations between communication and learning content tools.

What kind of data representation is best suitable for the model? Given information about semantics of the model and requirements for the information which should be extracted from it, adequate representation must be chosen. There are several ways of modeling entities, relations and weights, such as graphs. Implementing and evaluating an approach based on a combination of matrices, graphs and a set of new metrics (measuring e.g. the degree of so-called parental relationships between users or other features) will be the next step in the process. The aims include modeling collaboration, defining metrics indicating "success", classifying the outcome as successful or not and, during the process, finding out what leads to successful collaboration and what has adverse effects.

In synthesis, we can state that our CI-based classification techniques are promising in several ways. Not only are they capable of replacing strongly assumption-based approaches and thus improve the performance and flexibility of adaptive features; in addition, we can potentially overcome the problem of a gap between different kinds of facilities in adaptive e-learning systems as introduced in the first sections. We consider data produced in practically any different kinds of tools, and, once the model is integrated in the learning environment, it can also be queried from all the system's components. Thus, knowledge about a student gained in one area can be used for adaptations in others. Moreover, using our extended classification approach operating on interrelated data, we can easily model relations between tools and other features which can become the basis for new kinds of adaptation and recommendations.

Acknowledgements

The work reported in this paper is funded by the "Adaptive Support for Collaborative E-Learning" (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15).

References

[Brusilovsky and Peylo, 2003] P. Brusilovsky and C. Peylo. Adaptive and Intelligent Web-Based Educational Systems. *Int. Journal of Artificial Intelligence in Education*, 2003.

[Brusilovsky, 1996] P. Brusilovsky. Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2–3):87–129, 1996.

[De Bra and Calvi, 1998] Paul De Bra and Licia Calvi. AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*, 4, 1998.

[Dolog *et al.*, 2004] Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. Personalization in Distributed E-Learning Environments. In *WWW (Alternate Track Papers & Posters)*, pages 170–179, 2004.

[Felder and Brent, 2005] Richard M. Felder and Rebecca Brent. Understanding Student Differences. *Journal of Engineering Education*, 2005.

[Höök, 1997] Kristina Höök. Evaluating the Utitlity and Usability of an Adaptive Hypermedia System. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 179–186, 1997.

[Jung *et al.*, 2005] Sung Young Jung, Jeong-Hee Hong, and Taek-Soo Kim. A Statistical Model for User Preference. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 2005.

[Lefrancois, 2006] Guy R. Lefrancois. *Psychologie des Lernens*, volume 4. Springer, Heidelberg, 2006.

[Oakley *et al.*, 2004] Barbara Oakley, Rebecca Brent, Richard M. Felder, and Imad Elhajj. Turning Student Groups into Effective Teams. *Journal of Student Centered Learning*, 2(1), 2004.

[Paramythis, 2008] Alexandros Paramythis. Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet. In *Proceedings of the Workshop on Adaptive Collaboration Support*, 2008.

[Prince and Felder, 2006] Michael J. Prince and Richard M. Felder. Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases. *Journal of Engineering Education*, 2006.

[Romero *et al.*, 2008] Cristóbal Romero, Sebastián Ventura, Pedro G. Espejo, and César Hervás. Data Mining Algorithms to Classify Students. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM08)*, 2008.

[Rückert and Kramer, 2006] Ulrich Rückert and Stefan Kramer. A Statistical Approach to Rule Learning. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, 2006.

[Sakai, 2009] Sakai. http://www.sakaiproject.org, 2009.

[Soller, 2007] A. Soller. Adaptive Support for Distributed Collaboration. In *The Adaptive Web*. 2007.

[Vathanophas *et al.*, 2008] Vichita Vathanophas, Kingkarn Suensilpong, and Tullawat Pacharapha. Task-Related Information Sharing in Group Support Systems (GSS). In *Proceedings of the European and Mediterranean Conference on Information Systems 2008*, 2008.

[Voges and Pope, 2000] K.E. Voges and N.K. Pope. An Overview of Data Mining Techniques from an Adaptive Systems Perspective. In *Proceesings of ANZMAC 2000: Australian and New Zealand Marketing Academy Conference*, Gold Coast, Australia, November 2000.

[Webb *et al.*, 2001] G.I. Webb, M.J. Pazzani, and D. Billsus. Machine Learning for User Modeling. *User Modeling and User-Adapted Interaction*, 2001.

[Witten and Eibe, 2005] Ian H. Witten and Frank Eibe. *Data Mining*. Morgan Kaufmann, 2005.