User Modeling and User Adapted Interaction manuscript No.
(will be inserted by the editor)

Activity Sequence Modelling and Dynamic Clustering for Personalized E-Learning

Mirjam Köck
Alexandros Paramythis

the date of receipt and acceptance should be inserted later

Abstract Monitoring and interpreting sequential learner activities has the potential to improve adaptivity and personalization within educational environments. We present an approach based on the modeling of learners' problem solving activity sequences, and on the use of the models in targeted, and ultimately automated clustering, resulting in the discovery of new, semantically meaningful information about the learners. The approach is applicable at different levels: to detect pre-defined, well-established problem solving styles, to identify problem solving styles by analyzing learner behaviour along known learning dimensions, and to semi-automatically discover learning dimensions and concrete problem solving patterns. This article describes the approach itself, demonstrates the feasibility of applying it on real-world data, and discusses aspects of the approach that can be adjusted for different learning contexts. Finally, we address the incorporation of the proposed approach in the adaptation cycle, from data acquisition to adaptive system interventions in the interaction process.

Keywords adaptivity, user modeling, e-learning, data mining, clustering, unsupervised learning

Institute for Information Processing and Microprocessor Technology
Johannes Kepler University
A-4040, Linz, Austria
{koeck,alpar}@fim.uni-linz.ac.at
http://www.fim.uni-linz.ac.at

1 Introduction

Research attention into the area of adaptive systems has been steadily increasing in the past three decades, and this is also true for the application domains in which adaptivity is used to actively support users. The field of e-learning is one such domain, and has become a focal point of adaptive systems research, as it rapidly evolves into a viable alternative to traditional learning contexts. Despite the high attention that adaptive e-learning has received, only recently has emphasis been placed on treating learning as a social process (see, e.g., [Brusilovsky et al., 2004]), or as a process consisting of interconnected activities (see, e.g., [Soller and Lesgold, 2007]), rather than a (passive or active) consumption of learning content, either at the individual, or at the group level.

The premise of the line of work reported in this paper is that the monitoring and interpretation of online learning activities can ultimately lead to enriched user- and learner- models, that, in turn, will make possible new and expanded forms of adaptive intervention in the context of e-learning. In this article we propose a new approach intended to address the extraction of information from sequential user activities, and the analysis and interpretation of such information, with the ultimate goal of deriving adaptation-oriented knowledge from naturally occurring learning behaviour.

The proposed approach covers, specifically, the modelling of sequences of user activities (as Discrete Markov Models) and the employment of the resulting models, in combination with other types of monitoring data, for the discovery of semantically meaningful information about the learner (including the discovery of new learner behaviour semantics) that can be embedded into the adaptation cycle. Discovery is driven, in both cases, by the clustering of learners' activity sequence models. The clustering can be applied at three levels, depending on the discovery goal:

- *Level I (**pattern-driven**)*, aiming at the detection of predefined behaviour patterns / styles on the part of learners that are considered to be indicative of their skills, traits, knowledge, etc.
- *Level II (**dimension-driven**)*, aiming at semi-automatically detecting concrete, but still unknown, patterns that can be related to behaviour associated with a specific learning dimension.
- *Level III (**open discovery**)*, aiming at the open-ended automatic detection of potential learning dimensions and concrete behaviour patterns, with human intervention in the process being reduced and focused on the assessment of the validity and utility of system findings.

To demonstrate the feasibility of the approach we have applied it to the area of problem solving, using real-world data. Problem solving is an essential part of the learning process in both traditional learning contexts and e-learning. Learners' problem solving styles can be described similarly to the better-known and more often considered learning styles (see, e.g., [Lefrancois, 2006]), although at a different level. Problem solving is applied for problems of different levels of granularity, from atomic to very complex. While the analysis

of users' learning styles based on activity data is mostly done using statistical knowledge, problem solving behaviour can hardly be accurately modelled without more detailed sequential information, which renders it an appropriate case study for the proposed approach.

For the purposes of the work described in this paper, we have developed custom models that capture the activity sequences of learners involved with problem solving in a particular Intelligent Tutoring System (ITS). We then use these models, in accordance with the aforementioned three levels of clustering, to: (a) detect predefined, well-established problem solving styles in students' problem solving sequences; (b) discover new problem-solving styles along predefined learning dimensions; and, (c) discover potentially interesting learning dimensions and associated problem solving styles.

The rest of this article is structured as follows. Section 2 provides an overview of related work. Section 3 presents the activity modelling part of the proposed approach, and the data used. Section 4 presents the second part of the approach, as applied to the models previously derived. Section 5 describes possible system interventions aiming at closing the adaptation cycle. And, finally, section 6 summarizes findings, discusses issues related to the application of the proposed approach to domains other than problem solving, and gives an overview of planned work.

2 Related Work

The work described herein falls within the broad field of research of Educational Data Mining (EDM) or data mining in e-learning [Romero and Ventura, 2006], which combines aspects and issues of different areas (e.g., e-learning/distance education, machine learning, adaptive systems, etc.). In [Romero and Ventura, 2010], the authors categorize work in educational data mining into *(a) statistics and visualization*, and *(b) web mining* [Srivastava et al., 2000] that can be further split into *clustering, classification and outlier detection*, *association rule mining and sequential pattern mining*, and *text mining*. Web (usage) mining can additionally be further categorized into *offline web mining* aiming at the discovery of patterns or other information to help educators to validate learning models, and *online or integrated web mining* where the patterns that are discovered are fed into an "intelligent" system that could assist learners in their online learning endeavours [Li and Zaïane, 2004].

A different viewpoint on educational data mining is provided in [Baker and Yacef, 2009] and [Baker, 2010], where the following categories are identified: *prediction*, including classification, regression and density estimation, *clustering*, *relationship mining* (including association rule mining, correlation mining, sequential pattern mining and causal data mining), *distillation of data for human judgement*, and *discovery with models*.

Besides the different ways of categorization, the process of data mining in educational settings can be split into the following phases [Romero et al.,

2007]: *data collection, data preprocessing, application of data mining,* and *interpretation, evaluation and deployment of the results.*

The work presented here spans several of the aforementioned categories, as its focus lies on *web usage mining*, especially emphasizing *clustering* and *sequential pattern mining* in the context of *student modelling*, based on the analysis of *logged user activity data*. The overarching goal of the presented work is to facilitate the (semi-)automatic discovery of patterns that occur within *activity sequences* in a learning context, so that they can be readily "recognized" and acted upon. Furthermore, although the mining activities described here are carried out offline, their results are intended to be used for online analysis of learning behaviour, potentially combined with adaptive interventions. Another important characteristic of the proposed approach is that, ultimately, human intervention is only required for assessing the results of the analysis process, but not for annotating or otherwise augmenting activity data prior to analysis - often a prerequisite for alternative approaches in the literature with comparable objectives.

The rest of this section starts with an overview of related work in the area of *activity mining and analysis in educational systems*. It then focuses more specifically on *clustering-based student modelling*, and *sequence-based clustering approaches*. A comparison of our approach to selected ones described in this section, is provided later in section 6.

2.1 User Activity Data Mining and Analysis in Educational Systems

Data acquisition and preprocessing are fundamental steps in the process of educational data mining and also constitute the first phases of the adaptation cycle. The nature of the data monitored is a decisive factor for the later stages in the cycle and further analysis. Most adaptive educational systems share a strong reliance on this early phase of the process. They might differ, however, in the way data is actually monitored, and the granularity of the data itself. For instance, systems may treat user activities as individual items (either in an aggregated or event-based way) [Amershi and Conati, 2009], [Romero et al., 2008] or consider activity sequences [Soller, 2007], [Soller and Lesgold, 2007]. A further distinction can be made by the way data is analysed later; during the past years a trend towards the combined use of data mining and machine learning techniques for the analysis of activity data can be observed [Romero and Ventura, 2010], [Romero et al., 2007], [Hämäläinen et al., 2004], [Amershi and Conati, 2009]. Systems based on individually treated user activities often aim at either the prediction of students' success (or, even more concretely, grades) [Romero et al., 2008], or future behaviour or interest [Köck, 2009], or at the extraction of individual users' and groups' characteristics [Choi and Kang, 2008].

In [Romero et al., 2007] and [Romero et al., 2008], the authors describe a data mining process driven by an extension to the Moodle Course Management System (CMS) [Moodle, 2010]. Their approach is based on aggregated log data.

The original pool of logged data contains very fine-grained activities, e.g., every single click a user makes for navigational purposes. However, data is not analysed in its original granularity but summarized and thus converted to a more aggregated format (e.g., the number of assignments done by a student, the number of quizzes failed, the number of quizzes passed, the total time spent on assignments, etc.) The ultimate goal is the evaluation of the usefulness and performance of different classification algorithms for the prediction of students' final grades.

Another perspective can be found in [Choi and Kang, 2008] where learner activity data is monitored and analysed in order to identify conflicting and facilitating factors in online collaborative learning. Conflicting factors are described as factors ultimately obstructing the achievement of learning objectives. Facilitating factors are described as elements that learners recognize as positive or supportive in attaining the learning objective. Here, the authors introduce an approach that, compared to the previously described one, relies more on semantic information behind user activities. In general, all activities are monitored; analysis, however, extracts the relevant parts and predefines common "learner behaviours" as, for instance, "summarize learning material", "outline tasks", "modify material", or "write meeting minutes".

In [Vialardi et al., 2009], the authors describe another data mining approach in the context of educational systems that aims at predicting how suitable a specific course is for a specific student (based on the system's prediction of success for the respective course) via classification, in order to provide personalized recommendations. Unfortunately the authors don't provide a detailed description of the base data records they use. From the rules generated by their classification system (including, for instance, the number of courses a student is enrolled at), however, we can conclude that they operate with accumulated data that is better comparable to what is described in [Romero et al., 2008] than to what we utilize in the work described in this paper.

A conceptually related approach is presented in [Su et al., 2011], in this issue, in which the authors describe a clustering- and decision tree- based approach to eliciting appropriate learning content (objects) to provide learners with specific requirements and learning/interaction contexts. This approach is specifically intended to match so-called "user requests" for content (which encapsulate additionally hardware capabilities, a learner's preferences, and network conditions), to content elements in a learning object repository.

In [Anaya and Boticario, 2009], the authors explore data mining in educational systems with particular focus on collaborative learning processes. The stated goals of their approach are to: reveal learners' collaboration, be domain-independent, and offer the information immediately after the process has finished. The said approach was applied with students of the National Distance Learning University in Spain (UNED), using the learning environment dotLRN [.LRN, 2010]. The participating students were provided access to discussion forums, as well as other tools such as FAQs, news, calendar, etc. Analysis was restricted to statistics of interactions in forums, thus not considering semantic information. The statistical indicators were used as a

basis for clustering through which information about learners' collaborative behaviour was extracted. The paper by the same authors in this issue [Anaya and Boticario, 2011] presents an updated and more comprehensive view of their approach, introducing metrics based on the statistical indicators, which are shown to have superior performance to clustering in characterising the collaboration behaviour of learners.

In [Beal et al., 2006] we find an approach to classification of learner engagement based on multiple data sources. It explores an integrated way of information acquisition, comprising also students' self-reported motivation profile and teachers' ratings.

2.2 Student Modelling Based on Clustering

This section explores a more concrete part of related work that describes clustering in the context of student modelling (see an even more specialized selection in section 2.3). In [Amershi and Conati, 2009] we can find a detailed description of the authors' classification and clustering approach to user modelling. Their base data originates in a learning environment more exploratory than traditional tutoring systems, with students being required to have a deeper, more structured understanding of concepts in the domain [Piaget, 1954], [Ben-Ari, 1998]. The data they use is converted to feature vectors that are later fed into the clustering phase; a feature vector represents an aggregated version of a student's activities. Thus, there is only one feature vector for each student, which results in a low overall number of vectors. The authors describe another similar approach in [Amershi and Conati, 2006] where clustering is used to automatically recognize learner groups in exploratory learning environments.

A clustering approach based on collaboration behaviour can be found in [Anaya and Boticario, 2009] where the authors describe how statistical indicators in learner activity data are used to determine cluster membership. Data was monitored for UNED students via the platform dotLRN [.LRN, 2010]. The described monitoring process started with an initial questionnaire and a mandatory individual task that had to be completed by every learner. The respective results were then used to manually group the learners into teams of 3 members each. In a later phase the teams were given additional tasks, where, for instance, every member had to solve one part of a specific problem, or, the team had to merge individual solutions. An expert observed these processes and used the findings on learner collaboration to label statistical data (i.e., an aggregated version of logged learner activities) that was then fed into a clustering algorithm with the objective of revealing relations between the statistical indicators and collaboration behaviour.

2.3 Sequence-Based Clustering Approaches

This section summarizes clustering approaches in the context of e-learning that consider sequential information in activity log data and are therefore best comparable to our work. In machine learning, the use of Markov models is prominent if the domain requires sequences to be represented or analysed as they provide a convenient way of modelling interrelated data. In the area of EDM, this advantage has recently been exploited in different pieces of research.

For example, we can find collaboration analysis based on Hidden Markov Models (HMMs) in [Soller and Lesgold, 2007] and [Soller, 2007]. In [Soller and Lesgold, 2007], the modelling process is described for the example case of knowledge sharing, defining a knowledge sharing episode as "a series of conversational contributions and actions that begins when a group member introduces new knowledge into the group conversation, and ends when discussion of the new knowledge ceases". The subsequent analysis aims at determining role distribution (knowledge sharer vs. receiver), analysing how well the knowledge sharer explained the new knowledge and evaluating how the receiver assimilated new knowledge. The communication interface via which the activity sequences are logged, includes tagging functionality that helps categorize individual activities. The tagging process is a manual one, i.e., it requires human effort. In the experiments the authors describe, trained HMMs provide very good accuracy at identifying the role of the knowledge sharer. It is, however, not entirely clear why hidden models were used, as the number of states is known in advance. In our research, as will be explained later, we use Discrete Markov Models (DMMs) with a predefined number of states (indicated by the learner actions possible on the platform).

Another pattern detection approach can be found in [Beal et al., 2007], where HMMs are used to model students' performance on problem solving. The models are fit to students' activity sequences with three hypothesized hidden states that correspond to students' "engagement levels". The resulting HMMs are later used to cluster students into groups showing specific kinds of behaviour. Furthermore, the models become a basis for prediction at a later stage of the process. In this case, HMMs are obviously a well-fitting analytic approach because they are used for explicitly modelling unobservable influences, as also indicated by the better prediction accuracy, compared to simple Markov chains.

A different sequential pattern mining approach is described in [Perera et al., 2009] where the authors exploit activity data monitored by the system to support mirroring, i.e., to extract and present patterns that characterize the behaviour of successful groups. They do not restrict the available tools or provide specific rules about how to use them, but aim at monitoring collaboration processes that are as authentic as possible, including the selection of tools and frequency of use. The main goal of this work is to "extract patterns and other information from the group logs and present it together with desired patterns to the people involved, so that they can interpret it, making use of their own knowledge of the group tasks and activities". The underlying concepts

are based on the "Big Five" theory of group work [Salas et al., 2005] that defines five key factors: leadership, mutual performance monitoring, backup behaviour (e.g., reallocating work between members), adaptability, and team orientation. After the monitoring process, the final data pool contained both the traces of user actions and the groups' progressive and final marks. Based on their performance (i.e., grades), the groups were ranked. The ranking was then used to determine what kind of behaviour distinguished the stronger from the weaker groups. This was done by a simple statistical analysis in the first phase, and by application of data mining techniques (clustering on groups and on students) in subsequent ones. Group-level clustering base data contained aggregated group activities like, for instance, the average number of events in a specific tool, student-level clustering base data contained similar information for individual students. The clusters detected during the student-level clustering phase define different types of users, e.g., "managers" or "loafers". The analysis of the activities by a pattern extraction process revealed the most important activities that were indicative of "strong" and "weak" groups. The authors point out, however, that their approach is not fully matured yet and still bears some limitations based on the data (due to limited types of events) and on the way in which output was interpreted.

In [Jeong and Biswas, 2008], another approach to behaviour modelling is presented. The authors describe a study with middle school students operating with a Teachable Agent. They again use HMMs to represent sequences of activities in order to reveal patterns that lead to learning success. The concrete goal in this case was to find out if "Learning by Teaching" provides better opportunities for learning, compared to other settings (a self-regulated- and a coaching system). Thus, the sequence models were used mainly as an aid to evaluate learning concepts in this approach.

In [Li and Yoo, 2006], the authors describe the modelling of student learning behaviour with Bayesian Markov Chains that was used in the adaptive tutoring system AtoL [Yoo et al., 2005]. Their approach presumes a specific format of tasks including specific levels of difficulty and only considers two base observations, i.e., correct and incorrect answers. Additionally, the authors assume that there are exactly three basic student models based on the three learning types they define (i.e., *reinforcement type*, *challenge type* and *regular type*). Their goal is to use clustering for modelling student behaviour and to use the resulting models to predict the learning styles of new users. The results are interesting in the context of this paper, because they show that basic sequential information can be successfully used in clustering processes and improve static models derived, for instance, through an initial survey. What is being proposed in our work, however, goes one step further and does not presume a specific number of models but rather aims to allow these to be dynamically determined, as described in more details later.

Another interesting use of Markov models for activity analysis can be found in [Martín et al., 2011], in this issue. In this case, instead of looking for similarities in the learners' behaviour directly, the authors first cluster learners on the basis of user- and context- characteristics; they then construct Markov

models of the learners' activity transitions in each cluster to derive activity sequence recommendations for future users. As such, this approach does not seek to directly analyse behaviour on the basis of the models, but rather promote activity sequences that have been shown to be of benefit to similar learners (and contexts) in the past.

3 Data Analysis and Activity Sequence Modelling

To facilitate discussion of the proposed approach, we describe here its practical employment in the modelling and analysis of real-world problem solving data monitored within the Andes ITS [VanLehn et al., 2005], and made available via the PSLC DataShop [Koedinger et al., 2008], [Koedinger et al., 2010]. Specifically, experiments were run on data from a Physics course of the US Naval Academy (USNA) and repeated with data of different academic terms (Spring 2007, 2008, and 2009).

The software we used in this study, which embodies the approach being put forward, consists of three main components: a pre-processor extracting activity sequences from raw data, a modelling unit converting the sequences to Discrete Markov Models (DMMs) (further described in section 3.2) and a clustering unit based on the clustering algorithms integrated in the Weka machine learning library [Hall et al., 2009] (further described in section 4).

3.1 Raw Data Description

In general, the Andes tutoring environment provides learners with problems to be solved, and different types of help (e.g., "explain further", "what's wrong"). The tasks themselves consist of several so-called Knowledge Components (KCs), which again consist of several so-called Unique Steps. We will later show how we model all the interactions within one KC as one problem solving sequence. A raw data set can be described as a set of activity instances, with an instance storing information about a student's attempt at a specific step, or the ITS's response. In total, the raw data contains $\sim$ 280000 activity instances produced by 73 users for 2007, $\sim$ 265000 activity instances and 97 users for 2008, and $\sim$ 115000 activity instances and 45 users for 2009. An instance is a plain line of text with comma-separated values (CSV) and holds, for instance, the student-id, session-id, time stamp and time zone, step duration, student response type (e.g., ATTEMPT, HINT_REQUEST) and subtype (e.g., Explain-Further, Whats-Wrong), tutor response type (e.g., RESULT, HINT_MSG) and subtype (e.g., Explain-Further, Whats-Wrong), problem-id, step-id, count of attempts at this step, outcome (the tutor's evaluation, e.g., CORRECT, INCORRECT, HINT), number of hints at this step, etc.

The problem solving sequences extracted from this raw data by the pre-processor are then passed on to the modelling unit which analyses and further converts them into DMMs.

3.2 Sequence Modeling

The first part of the proposed approach is concerned with modelling activity sequences in a way that allows for analysing and reasoning over sets of activities performed by different users. We have considered several alternatives to representing sequences, and concentrated on the interwoven questions of comparability and generalizability. Specifically, we sought a formalism that would enable us to compare activity sequences that may differ only little (e.g., situations where one sequence may contain more repetitions of one activity than those found in another), but also allow for comparing sequences with only small amounts of overlap. In general, the modelling of sequential data faces the challenge of not losing information about relations and dependencies between the individual items, in this case, activities.

An overview about different machine learning approaches for modelling sequences, is provided in [Dietterich, 2009]. The author lists the most important research issues in sequential supervised learning as follows: loss functions, feature selection and long-distance interactions, and computational efficiency. Although our approach described here aims at information extraction via clustering, i.e., unsupervised learning, most of these issues are relevant for us. Feature selection, for example, plays a crucial role in the process, as discussed in section 4. Too many features can inhibit the identification of the most significant properties and thus distort the picture, whereas too few features may easily cause total loss of relevant information. Computation efficiency is also a very important factor for our scenario – a sequence modelling approach suitable for our requirements should be applicable at run-time and avoid loss of information. Dietterich [Dietterich, 2009] lists several machine learning techniques suitable for modelling sequential data: the sliding window method, recurrent sliding windows, Hidden Markov Models (HMMs) and related methods, Conditional Random Fields (CRFs), and Graph Transformer Networks (GTNs).

The *sliding window method* (see different applications in, e.g. [Sejnowski and Rosenberg, 1987], [Qian and Sejnowski, 1988], or [Fawcett and Provost, 1997]), converts a sequential learning problem into a classical learning problem. The method uses a *window classifier* that is trained with input data that has been converted into windows (each representing and treated as a sequence). The sliding window method is not bound to specific algorithms but can use any machine learning technique. However, it is inflexible and does only consider specific kinds of dependencies which is why loss of information about dependencies and relations is relatively likely.

The *recurrent sliding windows* technique feeds the predicted value for a specific data instance into the system to help make the prediction for the next instance, i.e., the most recent predictions are used as inputs (the size of this "window" depends on the respective application scenario). In [Lichtenwalter et al., 2009], for example, the authors describe an approach using recurrent sliding windows for musical classification. In [Bakiri and Dietterich, 2001], the authors applied recurrent sliding windows in combination with a decision tree

algorithm to the English pronunciation problem. In their evaluation, the recurrent sliding window technique drastically improved the results of the original sliding window method. However, both the original sliding window method and the recurrent sliding window technique are mainly suitable for supervised learning not for clustering-based unsupervised approaches as in our scenario.

Markov Models (MMs) are probabilistic models similar to finite state machines consisting of a set of states $S = \{S_1, S_2, ..., S_n\}$, an NxN matrix containing state transition probabilities $A = \{a_{ij}\}$, and a vector of initial state probabilities $\pi = \{\pi_i = P(q_1 = S_i)\}$. This model is then used to compute the probabilities for specific output sequences.

HMMs (see, for instance, [Rabiner, 1990]) are special cases of MMs because the states are hidden, i.e., not observable. The hidden states form a traditional MM and can produce a set of different outputs, i.e. observable effects. HMMs are a popular way of modelling sequential data in order to be able to provide predictions for specific activity sequences, see, for example, [Beal et al., 2007], [Seymore et al., 1999], [Soller et al., 2005], [Soller, 2007], or [Soller and Lesgold, 2007]. However, [Dietterich, 2009] identifies a principle drawback of this methodology and states that the "'structure of the HMM is often a poor model of the true process producing the data"', a problem which originates in the Markov property (i.e., the probability of a system being in a particular state S_j at time t does not depend on the entire history, but only on the previous state at time $t1$); a relationship between two different y values, for example, y_1 and y_3, must be communicated via the intervening ys. An MM where the probability $P(y_t)$ only depends on y_{t-1} cannot generally capture these relationships. This problem is generally addressed by sliding window techniques. Using sliding windows for HMMs is however difficult, because a HMM generates each x_t from the corresponding y_t only. [Dietterich, 2009] argues that this problem could theoretically be overcome by replacing the output distribution $P(x_t|y_t)$ by a more complex distribution $P(x_t|y_{t-1}, y_t, y_{t+1})$, which would allow an observed value x_t to influence all three y values but is difficult to put into practice because it is not clear how to represent this complex distribution compactly.

[Dietterich, 2009] lists the following approaches to overcome these limitations: *Maximum Entropy Markov Models (MEMMs)* (see, for example, [McCallum et al., 2000]), *Input-Output HMMs (IOHMMs)* (see, for example, [Bengio and Frasconi, 1996]), and *CRFs* (see, for example, [Lafferty et al., 2001] or [Vail et al., 2007]). All of these approaches are conditional models that, unlike standard HMMs which try to explain how observed sequences are generated, represent conditional distributions of output sequences given input sequences, i.e. they try to predict output values given input values. IOHMMs and MEMMs are quite similar in the way they are trained and both suffer from the same issue called *label bias problem*, i.e., there is a bias toward states with fewer outgoing transitions (states with a single outgoing transition ignore their observations), see a more detailed description in [Lafferty et al., 2001].

CRFs, which are mostly used for labelling sequences, are an approach to overcome the label bias problem (see [Lafferty et al., 2001] or [Vail et al.,

2007]). In the CRF, the way in which adjacent y values influence each other is determined by the input features [Dietterich, 2009]. CRFs In the experiments presented by [Lafferty et al., 2001], the CRF outperforms HMMs and MEMMs regarding modelling accuracy, but it is fairly slow in comparison to the other approaches.

GTNs (see, for example, [Bottou et al., 1997] or [Bottou and LeCun, 2005]) are neural network based models that transform input graphs into output graphs [Dietterich, 2009]. For example, an input graph consisting of a sequence of inputs x_t is transformed into a graph of u_t outputs, where every x_t is a feature vector attached to an edge of the graph, and every u_t is a pair of a class label and a score. The graph of the u_t scores is then analysed with the aim of finding the path with the lowest total score. Also this methodology aims at solving complex supervised learning problems rather than unsupervised ones.

In general, most techniques for modelling sequential data as presented here, are tailored to the use in classification tasks, i.e. supervised learning. In our case, however, we want to model sequences for clustering, i.e. unsupervised learning. Thus we have to find a way to represent sequences that allows for transformation into a format processable by clustering algorithms. The reasons for the decision to use DMMs can be summarized as follows:

1. Markov models have been successfully used in the past for similar purposes in the context of modelling activities [Soller and Lesgold, 2007] [Soller, 2007] (discussed in more detail in section 2).
2. The states themselves are observable (see the description below), therefore there is no need to use hidden models.
3. Traditional statistical representations are likely to lose information bound to not the activities themselves but the relations and dependencies between them (see the example below).
4. The approach must be suitable for unsupervised learning and models must be convertible to other formats that can be fed into a clusterer, or serializable without information loss.
5. The modelling process itself should not be too expensive concerning its run-time behaviour.

To better motivate our choice of sequential representation, let us consider the concrete example of modeling problem solving sequences. In the Andes system, a problem solving sequence contains all of a user's (identified by a user id) activities related to any Unique Step associated with the problem (identified by a problem id). Thus, a solving sequence for a specific problem looks different for each user. Consider the following scenario: two hypothetical users U_1 and U_2 are working on the same problem P_1 which consists of three steps S_1, S_2 and S_3. User U_1 solves step S_1 correctly at first attempt, but fails first at S_2. Next, the user requests a hint of type H_1 that is followed by two hints of type H_2, and then solves the step correctly at the second attempt. This results in the activity sequence $I \rightarrow H_1 \rightarrow H_2 \rightarrow H_2 \rightarrow C$. A similar pattern is observed for S_3: $I \rightarrow H_1 \rightarrow H_1 \rightarrow C$. For user U_2 we observe: S_1: $I \rightarrow H_1 \rightarrow I \rightarrow H_2 \rightarrow C$, S_2: $H_3 \rightarrow C$, S_3: $H_1 \rightarrow I \rightarrow H_2 \rightarrow C$.

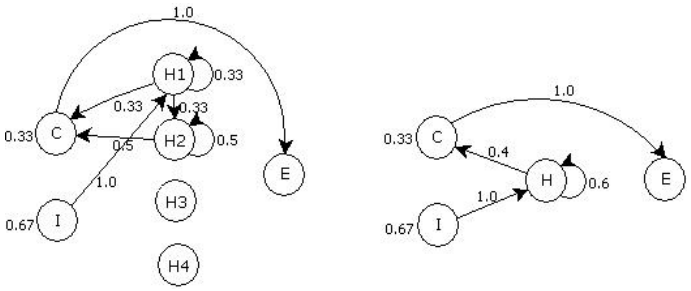


Fig. 1 This figure shows two DMMs as used for the clustering approach described later. Probabilities with 0- and default values (with which the transitions and prior probabilities are initialized) are omitted here. The numbers next to nodes denote their prior probability, the numbers next to transitions denote the transition probability.

Describing these sequences with basic statistical means one may obtain results such as these: both users have successfully completed the problem; user U_1 submitted two incorrect answers in total and requested five hints, user U_2 submitted three incorrect answers and requested five hints. A comparison of these results might lead to the conclusion that the performance of U_1 and U_2 at P_1 was similar. Even if the comparison considered the level of steps, the result for the two users at S_3 would be equal although the actual sequences were different, i.e., one dimension of the information is lost.

As already discussed, the premise of the presented work is that retaining this kind of sequential activity information in the modelling process can enhance several stages of the adaptation cycle by offering fine-grained user model input on a behavioural level.

As mentioned earlier, in our case study, the data clearly suggests a certain configuration of states, therefore there is no need to use hidden models here. Referring back to the example above, a student has basically two important possibilities of interacting with the system: submitting an answer, or requesting a hint. The system offers four different categories of hints, thus we can differentiate between four different help states in the corresponding DMM.

To be able to examine at a later point whether the distinction between hint types influences behaviour analysis, we created two DMM settings for all experiments, the first with one aggregated hint state and the second with the initial four. Figure 1 shows sample DMMs for the two settings, modelling the behaviour of user U_1 solving problem P_1 from the example above. In addition to the obvious states "correct" (C), "incorrect" (I) and "hint" (H), an artificial end state (E) is added by the modelling unit. The end state is needed in order to distinguish between the transitions within a single step and the transition to a new one. If the user starts a new step, the system inserts a transition from the current state to the end state, thus completing the step.

These DMM-based problem solving sequence models were subsequently serialized and converted to the common Attribute-Relation File Format (ARFF

Table 1 This table lists and describes the features of different data sets that are used in both text and tabular results later.

Feature name	***Feature description***
PRIOR_PROB_C	prior probability of a correct attempt
PRIOR_PROB_I	prior probability of an incorrect attempt
PRIOR_PROB_H1	prior probability of a help request of type H1
PRIOR_PROB_H2	prior probability of a help request of type H2
PRIOR_PROB_H3	prior probability of a help request of type H3
PRIOR_PROB_H4	prior probability of a help request of type H4
PRIOR_PROB_H*	prior probability of a help request of arbitrary type
PRIOR_PROB_H	prior probability of a help request (aggregated setting)
TRANS_PROB_C_C	transition from a correct attempt to a correct one
TRANS_PROB_C_I	transition from a correct attempt to an incorrect one
TRANS_PROB_C_H[*]	transition from a correct attempt to a help request
TRANS_PROB_C_E	transition from a correct attempt to end (i.e., step finish)
TRANS_PROB_I_C	transition from an incorrect attempt a correct one
TRANS_PROB_I_I	transition from an incorrect attempt to an incorrect one
TRANS_PROB_I_H[*]	transition from an incorrect attempt to a help request
TRANS_PROB_I_E	transition from an incorrect attempt to end
TRANS_PROB_H[*]_C	transition from a help request to a correct attempt
TRANS_PROB_H[*]_I	transition from a help request to an incorrect attempt
TRANS_PROB_H[*]_H[*]	transition from a help request to a help request
TRANS_PROB_H[*]_E	transition from a help request to end
TRANS_PROB_E_C	transition from end to a correct attempt
TRANS_PROB_E_I	transition from end to an incorrect attempt
TRANS_PROB_E_H[*]	transition from end to a help request
PERC_HELP_STEP	percentage of help requests in a user's activities
PERC_INCORRECT	percentage of incorrect attempts in a user's activities

[1]) which is handled by an export mechanism in the modelling unit. In addition to the aforementioned activity sequence information, we made use of basic statistical data, in order to later compare the clustering performance not only for different settings and different years but also according to different clustering aims and with different aspects of the same raw data. Using these data sources in isolation and in combination gave rise to a total of three data sets that were used in the clustering stage: SET_MARKOV, including only the information provided by the learned Markov models (i.e., prior probabilities for the states and transition probabilities between the states), $SET\_STATISTICAL$, including very basic statistical information (i.e., the percentage of incorrect attempts, the percentage of help requests and the percentage of unfinished steps), and SET_BOTH combining the previous two. Table 1 lists and explains all features that are later used in the context of the description of the experiments and results.

Note that the modelling choices made here (especially the use of DMMs), as well as the the selection of features with which to populate the data sets used in the clustering stage, have been tailored to the specific needs of modelling

[1] See more information about ARFF at `http://weka.wikispaces.com/ARFF`

problem solving activity sequences. Section 6 discusses factors researchers may want to consider when applying the proposed approach to other domains of learning.

4 A Multi-targeted Clustering Approach

This section describes the clustering phase and its different directions, following the definition of the activity sequence representation and conversion of base data to the corresponding models. Figure 2 provides a concise overview about the approach as a whole.

The process starts with the pre-processing phase, including the definition of the model that will be used, based on the structure of base data and the possible user activities (here DMMs), data conversion (i.e., conversion of users' activity sequences to concrete models), and the definition of expressive metrics to evaluate the quality of clusters later. Next, in the experimental clustering phase, the number of features considered in subsequent steps is limited by identifying and evaluating their characteristics (e.g., their discriminatory capacities), and the optimal number of clusters for the respective scenario is determined. In the clustering phase, first the concrete clustering goals must be defined, i.e., it must be decided whether a predefined concrete problem-solving style should be identified in a user's activities, if a predefined problem-solving dimension should be recognized, or if the clustering process should autonomously detect potential dimensions and styles in users' behaviour. In the first case, the respective style must be defined, the most relevant features chosen and the expected values for them must be identified. Based on this information, a suitable data set is created that proceeds into the clustering process. In the second case, the respective dimension must be defined, and again the most significant features must be chosen before a data set can be created that is fed into the clustering process. In the third case, no concrete features are preselected, but constraints like the maximum number of features that can be used in a data set, are identified. Different data sets meeting these requirements are automatically created and proceed to the clustering process. The last phase, following clustering, includes cluster analysis.

4.1 The General Process

Having established a way of modelling learning activity sequences, we now turn our attention to the analysis of such sequences to discover behaviour patterns that may be characteristic of traits of the persons exhibiting the behaviour (e.g., learning- or problem solving- styles), or of the process or context within which the activities take place (e.g., progress of a collaborative learning class project). The first step in that direction is the clustering of the activity models previously derived, possibly in combination with other monitored data, usually relating to the activities themselves.

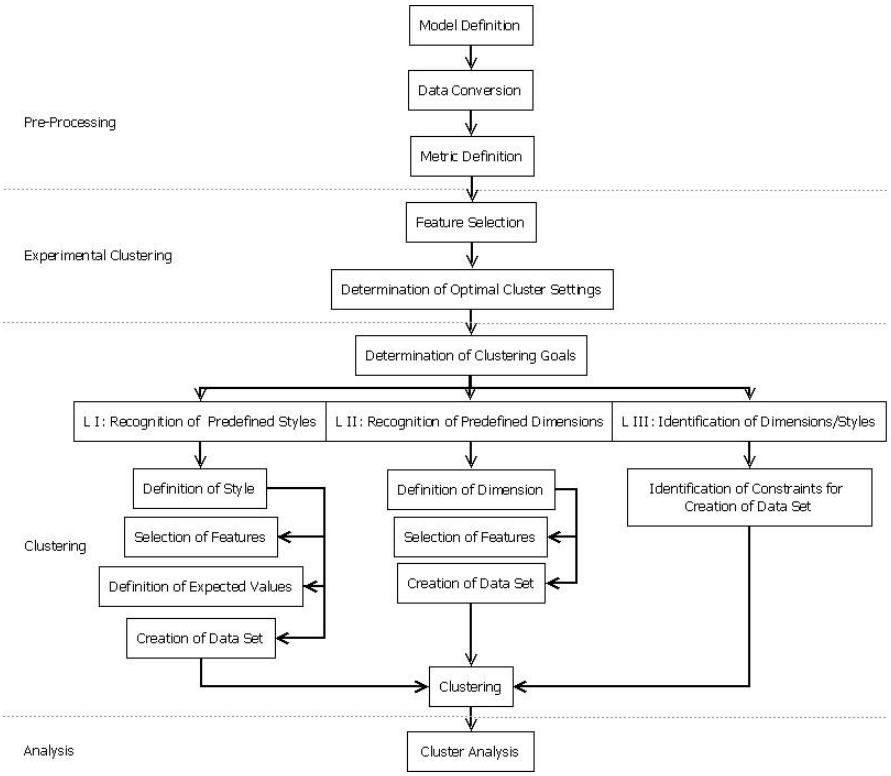


Fig. 2 Overall process of the proposed approach to sequence modelling and subsequent dynamic clustering.

A salient feature of the proposed approach is that clustering be dynamically controlled to accommodate different discovery goals in the analysis of activities. "Dynamic", in this context, is intended to convey the fact that aspects of the clustering process, such as the determination of the quality of clusters, the establishment of termination conditions for the clustering process, etc., are not constrained to the data set being clustered (as is the case with clustering algorithms that decide themselves when to "stop"), but are expanded to take into account semantic characteristics of the activities (or their results) that are not explicitly represented in the data being clustered over (e.g., how well clusters encapsulate the different strategies students exhibit when encountering a specific type of problem to solve).

The overall objective then of this part of the proposed approach is the identification of the discovery goals that guide the clustering process, as well as the establishment of metrics and criteria that can be used for its dynamic control. Whereas some of these indices may be general in nature (and, therefore, applicable to several types of activities being analysed), the requirement that they be based on activity semantics has the consequence that such indices

will often be application domain specific. The rest of this section is devoted to the application of the process in the domain of problem solving; section 6 addresses issues related to applying the approach to other domains.

4.2 Metrics and Indices for the Domain of Problem-Solving

The main goal of the clustering process in our case study is the *detection of problem solving styles*, either based on a description of pre-defined styles or with the aim to discover new ones (as further explained in section 4.4). Additionally, we use the clusters to *predict success*, i.e., whether a student will successfully complete a problem. The main task of the clustering unit is to not only pass on sets of data instances (an instance here being, e.g., the serialized, CSV-based version of a DMM for SET_MARKOV) to the k-means clustering algorithm [Jain et al., 1999], but also perform several subsequent analyses depending on what purpose we cluster for. Towards this goal, the experiments that will be subsequently described are repeated several times for n clusters, where $2 \leq n \leq 20$, assessing in each case the changes in clustering behaviour and performance and aiming at the detection of the optimal value for n, which can then be used to dynamically control the clustering process. We introduce the following metrics to evaluate a clustering result with n clusters:

1. *Average Student Entropies ($SE(C_n)$)*, providing an index for the distribution of students in clusters,
2. *Average Problem Entropies ($PE(C_n)$)*, providing an index for the distribution of problems in clusters,
3. *Average Variance ($V(C_n)$)* in the clusters, measured by the average standard deviations for the attributes, and
4. *Average Expected Prediction Error ($EPE(C_n)$)*, measuring the capability of the clusters to correctly predict success.

In an optimal cluster setting, the sequences of the same student should appear in one cluster only, presuming that the student showed consistent problem solving behaviour (which is an assumption that cannot hold in all cases in practice). Therefore, $SE(C_n)$ should remain low, which also applies for $PE(C_n)$. Both entropy indices, however, will naturally increase as the number of clusters increases, and are, therefore, not sufficient in themselves for characterizing the results of the clustering process. Furthermore, in an optimal cluster setting, $V(C_n)$ would tend to 0, meaning similar values for attributes can be found in the same clusters, and $EPE(C_n)$ would also be minimized.

Equation 1 shows how $SE(S_x)$ is computed for a student S_x. This index is an instance of the standard entropy measure $H = -K \sum_{i=1}^{n} p_i * log(p_i)$, where K is a positive constant [Shannon, 1974]. The same applies for Equation 2. S_{c_i} is the number of a specific student's problem solving sequences that can be found in cluster i. With $|S|$ being the overall number of this student's problem solving sequences, $\frac{S_{c_i}}{|S|}$ denotes the probability of a student's problem solving sequence being assigned to cluster i. $SE(C_n)$ is the respective average over all

students.

$$SE(S_x) = -\sum_{i=1}^{n} \frac{S_{c_i}}{|S|} * log_{10}(\frac{S_{c_i}}{|S|}) \quad (1)$$

Equation 2 shows how $PE(P_x)$ is computed for a problem P_x. P_{c_i} is the number of solving sequences for a specific problem that can be found in cluster i. With $|P|$ being the overall number of solving sequences for this problem, $\frac{P_{c_i}}{|P|}$ denotes the probability of a problem's solving sequence (independent of the student it was produced by) being assigned to cluster i. $PE(C_n)$ is the average over all problems.

$$PE(P_x) = -\sum_{i=1}^{n} \frac{P_{c_i}}{|P|} * log_{10}(\frac{P_{c_i}}{|P|}) \quad (2)$$

Equation 3 shows how $V(C_n)$ is computed for a cluster setting with n clusters where $\sigma^2(C_i)$ is the mean standard deviation over all attributes in cluster C_i.

$$V(C_n) = \frac{\sum_{i=1}^{n} \sigma^2(C_i)}{n-1} \quad (3)$$

Equation 4 shows how $EPE(C_n)$ is computed for a cluster setting with n clusters.

$$EPE(C_n) = \frac{\sum_{i=1}^{n} err(C_i)}{n-1} \quad (4)$$

where

$$err(C_i) = \begin{cases} \frac{co(C_i)}{tot(C_i)} & if \quad \frac{co(C_i)}{tot(C_i)} \leq 0.5 \\ 1 - \frac{co(C_i)}{tot(C_i)} & otherwise \end{cases} \quad (5)$$

with $co(C_i)$ being the number of completed steps in cluster C_i and $tot(C_i)$ being the number of total steps in cluster C_i.

For these metrics we can observe the following trends for an increasing number of clusters. $SE(C_n)$ and $PE(C_n)$ ascend logarithmically with the actual values being dependent on the base used for the logarithms. $V(C_n)$ and $EPE(C_n)$ slowly descend with $V(C_n)$ showing more fluctuations than $EPE(C_n)$ and $EPE(C_n)$ showing a slightly more significant descent. In order to use these indices to decide on an optimal number of clusters we also perform the following steps: firstly, we normalize the values to a range between 0 and 1 including the boundaries (for example, to avoid being dependent on the logarithm base), and secondly, we include weighting so that the clustering can be optimized according to a specific aim (e.g., the minimization of error). Figure 3 shows the normalized graphs for an example data configuration.

Optimization aims at finding the configuration in which the margin between the ascending and descending graphs is within a certain threshold. We define this threshold as the point of graph convergence in the normalized version and compute the optimization value for a cluster setting with n clusters as shown in equation 6.

$$Opt(n) = |\frac{\frac{no(SE(C_n))*w_s + no(PE(C_n))*w_p}{2} - \frac{no(V(C_n))*w_v + no(EPE(C_n))*w_e}{2}}{w_s + w_p + w_v + w_e}| \quad (6)$$

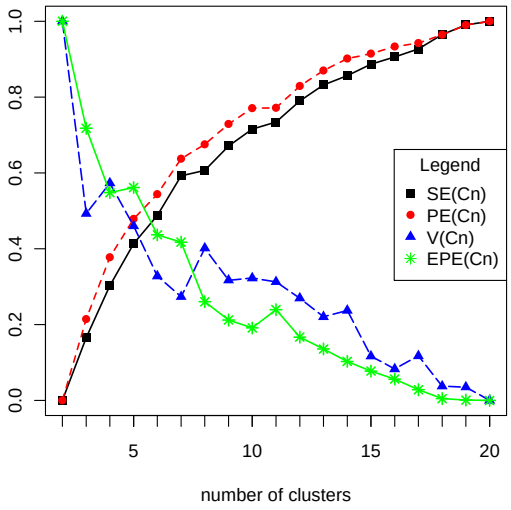


Fig. 3 This figure shows the normalized graphs for the data set SET_MARKOV in the extended hint processing on data of the year 2008.

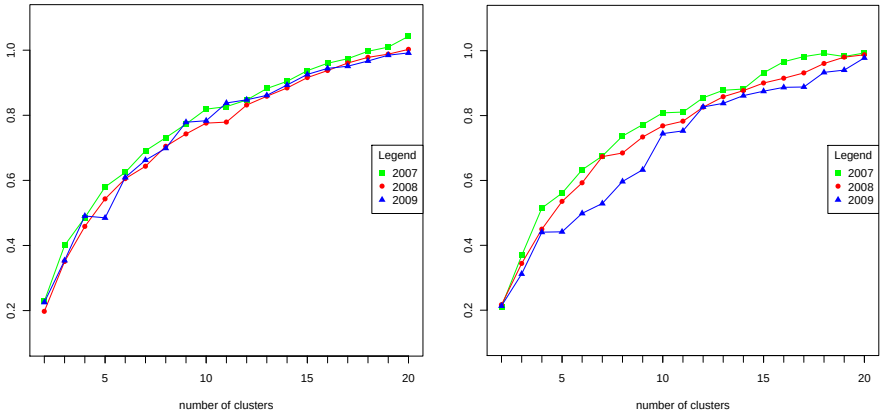
where $no(N)$ normalizes the values in N to a range between 0 and 1 including boundaries. The optimum is then the value closest to 0. For the example used in figures 3, the unweighted optimization graph would find 5 as the best number of clusters, as shown later in table 4.

4.3 Comparison of Different Data Sets and Settings

The clustering process described in section 4.2 was repeated with multiple different data sets, different settings and academic terms. The primary goal in doing so has been to determine the extent to which the proposed metrics are influenced by the specific data set(s) employed. A secondary goal was to make informed decisions with respect to the levels of aggregation that base data would be used in.

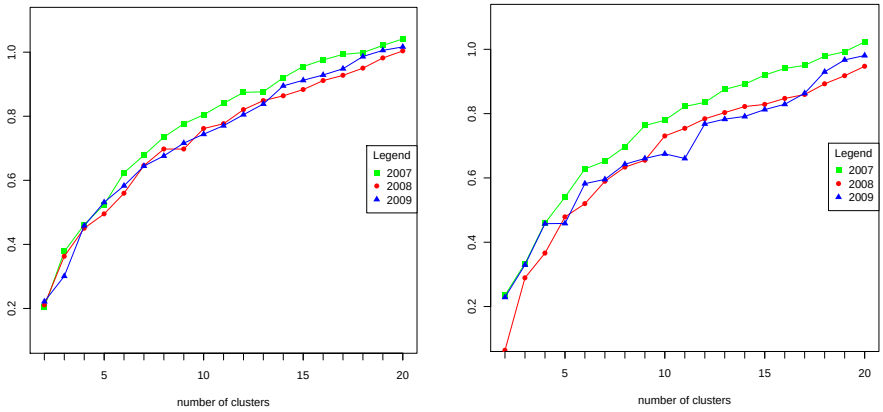
To start with, we compared for each of the four metrics $SE(C_n)$, $PE(C_n)$, $V(C_n)$ and $EPE(C_n)$ individually the clustering results in different years, settings and data sets, resulting, for instance, in a comparison of results for the data set SET_MARKOV in the aggregated help processing setting for the three academic terms of Spring 2007, 2008 and 2009. This process aims at verifying the overall mechanism that is expected to lead to similar results if a sufficiently high amount of base activity data is provided. The number of activities in the courses of 2007 and 2008 was about equal and relatively high, which leads to the assumption that at least for these two years the results will be similar. As in 2009, the number of students and activities was significantly lower, the results of this term may not be as reliable as those of the previous

two. The comparison supports this assumption, as shown in the examples in Figures 4, 5, 6, and 7. As can be seen in these figures, the results for 2007 and 2008 show very similar trends for all metrics; the results for 2009, although comparable to those of the other years, do show some small deviations.



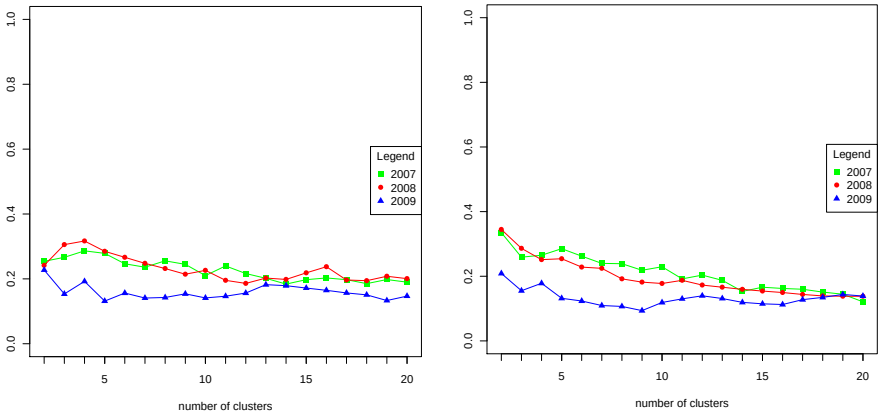
(a) Aggregated help states (b) Extended help states

Fig. 4 This figure shows the $SE(C_n)$ results for the data set SET_MARKOV in aggregated (a) and extended (b) help processing settings for all three academic terms.



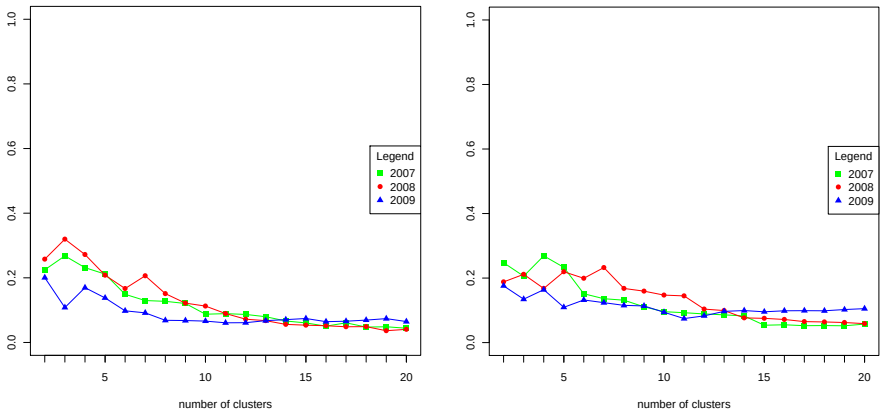
(a) Aggregated help states (b) Extended help states

Fig. 5 This figure shows the $SE(C_n)$ results for the data set $SET\_STATISICAL$ in aggregated (a) and extended (b) help processing settings for all three academic terms.



(a) Aggregated help states (b) Extended help states

Fig. 6 This figure shows the $EPE(C_n)$ results for the data set SET_MARKOV in aggregated (a) and extended (b) help processing settings for all three academic terms.

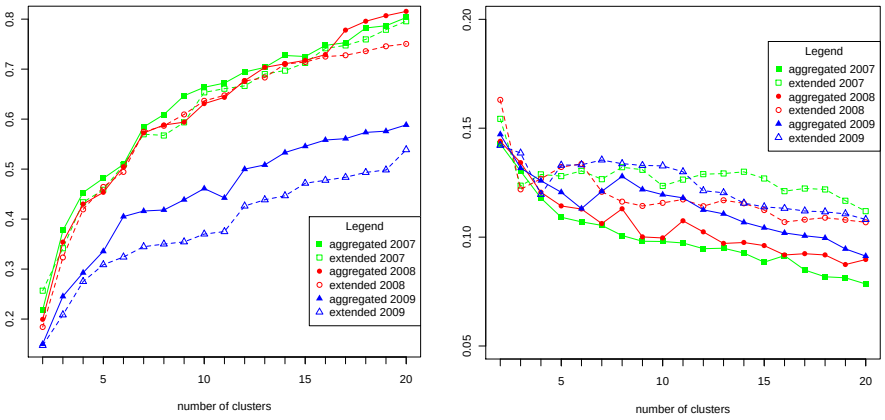


(a) Aggregated help states (b) Extended help states

Fig. 7 This figure shows the $EPE(C_n)$ results for the data set $SET\_STATISICAL$ in aggregated (a) and extended (b) help processing settings for all three academic terms.

The decision whether to use aggregated or extended help states depends on the clustering purpose and can best be determined by running initial clustering experiments on both settings, using data sets specifically tailored to the respective purpose. For instance, we may be interested in analysing the use of help and create data sets that specifically contain help-related features (see also section 4.4). Figure 8 shows the trends of $PE(C_n)$ and $V(C_n)$ as examples; each trend line represents the results for a specific year and setting (aggregated or extended). Figure 8(a) demonstrates that the results for the

aggregated and extended help processing settings are very similar in the cases of the 2007 and 2008 data sets; for 2009, the extended help processing setting shows the better results. Figure 8(b) shows better results for the aggregated help processing setting for all years.



(a) $PE(C_n)$ (b) $V(C_n)$

Fig. 8 This figure shows the $PE(C_n)$ (a) and $V(C_n)$ (b) clustering results for a data set containing help-related features in aggregated and extended help processing settings for all three academic terms.

As the more reliable data from 2007 and 2008 all showed either about equal results for the two settings or better results for aggregated help processing, we decided to use this setting in situations where either of the two would be potentially applicable. We conclude that for tasks not especially aimed at analysing styles based on different types of help, it is better to decrease the number of attributes. Thus, the results presented later use aggregated help processing. However, the models with extended help processing can be of use for specialized tasks.

Comparing the different data sets, $SET\_STATISTICAL$ provides the best models for predicting the value for "completed" attribute (i.e., $EPE(C_n)$ is low). For the detection of patterns in learner activities however, the models based on SET_MARKOV perform best. We therefore use a combined version as a basis for specifically tailored, dynamically created new data sets for further purposes, as described in the following sections.

4.4 Three-Level Clustering and Cluster Analysis

This section describes a multi-level approach for clustering and cluster analysis, based on the metrics and processes explained in section 4.2. The results presented here are based on the Andes Physics course data for the Spring 2008

term. The data of the other terms were used for rerunning the same experiments in identical settings, which confirmed the results presented later in this section.

4.4.1 Level I (Pattern-Driven)

Clustering at this level aims at the detection of predefined behaviour patterns on the part of learners that are considered to be indicative of their skills, traits, knowledge, etc. For the purposes of our case study this means that clustering is performed in order to detect predefined, well-established problem solving styles in students' problem solving sequences. To demonstrate the process at this level, we chose the well-known problem solving style *Trial and Error* [Jarvis, 2005], [Thorndike, 1903] (also referred to as *Trial and Success*), describing behaviour that is based on chance at the beginning and on learning by making mistakes until the problem is solved later. Although it is hard to find a recent psychological description of the *Trial and Error* style, there seems to be common agreement on its definition and use (see mentioned in, for instance, [Kanninen, 2008], [Brown et al., 2007], [Butler and Pinto-Zipp, 2006], [Cassidy, 2004], [Kolb, 1984], [Schaller et al., 2007], [Liu and Dean, 1999], [Dewar and Whittington, 2000], [Terrell, 2005], [Ballone and Czerniak, 2001], [Felder and Silverman, 1988], [Simon, 2000], or [Richmond and Cummings, 2005]).

We evaluated the available attributes from the data set SET_BOTH with regards to their potential to contribute to the identification of *Trial and Error* problem solving behaviour. Specifically, we expected a person with this behaviour, based on how the style is described in the relevant literature, to have high prior probabilities for incorrect attempts and low-to-medium prior probabilities for correct attempts. Note, that the probability of guessing a correct answer is much lower than the probability of getting it wrong by chance. This is due to the fact that, usually, problems only have one correct answer as opposed to several incorrect possibilities. Furthermore, we expected such a person to generally have a low hint request rate, low transition probabilities from incorrect to hints, and a relatively high rate of incorrect attempts. The corresponding attributes were selected, stored in a new data set SET_TRIAL_ERROR, and clustering was performed on this set. The results of a cluster configuration with 8 clusters (as listed in table 2) show that two clusters (1 and 6) provide a clear identification of the *Trial and Error* problem solving style. Note in both cases the high prior probabilities for incorrect attempts, the low help state rate, the high percentage of incorrect attempts and the fact that the prior probabilities for correct and incorrect attempts sum up to ~ 1, indicating that the prior probabilities for the help states are ~ 0. The procedure described above can be applied for every other pre-defined problem solving style, thus opening up the possibility of clustering activity sequences with the explicit goal of detecting expected behaviour, so that predetermined interventions can be applied in response to it.

It is important to note at this point that, with a preselected set of patterns to identify, and a set of hypotheses of what features might be relevant to

Table 2 This table shows the clustering results on SET_TRIAL_ERROR with an 8 cluster configuration. The numbers in parentheses after the cluster id show the number of problem solving instances in the respective cluster.

Attribute	C_0(1506)	C_1(456)	C_2(2503)	C_3(489)	C_4(1770)	C_5(1606)	C_6(464)	C_7(671)
PRIOR_PROB_C	0.9923	**0.1219**	0.239	0.4187	0.7405	0.6201	**0.2347**	0.3079
PRIOR_PROB_I	0	**0.7896**	0.1408	0.2045	0.2324	0.304	**0.7422**	0.0045
TRANS_PROB_I_I	0.1429	**0.5568**	0.2377	0.0772	0.0348	0.5452	**0.0991**	0.1429
TRANS_PROB_I_H1	0.1429	**0**	0	0	0	0	**0**	0.1429
TRANS_PROB_I_H2	0.1429	**0.0082**	0.0951	0.0336	0.0251	0.0301	**0.0087**	0.1429
TRANS_PROB_I_H3	0.1429	**0.032**	0.0767	0.6779	0.0262	0.052	**0.0231**	0.1429
TRANS_PROB_I_H4	0.1429	**0.0044**	0.0399	0.0124	0.0163	0.011	**0.0056**	0.1429
PERC_HELP_STEP	0.0146	**0.0918**	0.658	0.5533	0.0652	0.1432	**0.0465**	0.728
PERC_INCORRECT	0	**0.6345**	0.1081	0.1257	0.1989	0.3877	**0.4659**	0

these patterns, one could also follow a supervised learning approach to derive the results attained at this level. This would involve, for example, labelling a number of training examples, and using decision trees or similar "transparent" models to confirm or reject the hypotheses on relevancy of the various features. However, this approach would require human intervention (for the labelling) and would not be readily transferable to other patterns. More importantly, this approach cannot scale to the subsequent levels of discovery described below - where the patterns are a *derivative* rather than a given of the analysis process.

4.4.2 Level II (Dimension-driven)

At this level we are concerned not with the "recognition" of expected behaviour when it occurs, but rather with establishing whether it is possible to identify distinct behavioural patterns in relation to specific semantic dimensions of the activities being analysed. For our case study this translates into performing clustering along known learning dimensions, in order to identify concrete problem solving styles the learners may exhibit. In more detail, in this case we do not want to cluster for a specific problem-solving style directly, but rather for a kind of general learning behaviour, which may involve several different styles itself. We have chosen *Help-Seeking* behaviour [Nelson-Le Gall, 1985], [Aleven et al., 2003] as a well-known learning dimension, and identified the behaviour elements that we expected to be defining in this case.

Help-Seeking can generally be defined as "[...] the ability to utilize adults and peers appropriately as resources to cope with difficulties encountered in learning situations [...]" [Nelson-Le Gall, 1985], based on, among others, [Anderson and Messick, 1974] or [Nelson-Le Gall, 1981]. In [Aleven et al., 2003], the authors discuss a framework to understand help-seeking that was originally presented by [Nelson-Le Gall, 1981] and later elaborated by [Newman, 1994] and [Ryan et al., 2001]. The framework contains a task analysis of the help-seeking process and includes the following steps:

1. Become aware of need of help.
2. Decide to seek help.

3. Identify potential helper(s).
4. Use strategies to elicit help.

This process is to a great extent transferable to the scenario described here. Students need to be aware of the need of help first, before they actually decide to utilize the help functionality. They need to decide for a type of help, i.e., identify the kind of help potentially best suitable.

Taking the above into considerations, we identified the following elements as the ones we expected to be defining for this kind of behaviour: the rate at which learners request help, the transition probabilities from an incorrect attempt to a hint request, the prior probability for hint requests, and the help-internal transition probabilities (e.g., transitions caused by the learners' asking for more detailed help hints).

The selected help-related attributes were stored in a new data set $SET\_HELP\_SEEKING$, and clustering was performed on this set. Again, the results show clear variations of the examined behaviour. This indicates that the attributes selected formed a coherent whole, capable of exposing the elements of variability in the learners' behaviour along the, thus, successfully "recognized" learning dimension. The next step was to analyse different concrete styles within this learning dimension, which successfully leads to the detection of the help-related problem solving styles as described in Table 3. The four *Help-Seeking* styles identified here can be explained as follows. A problem solver of type H_1 shows *Trial and Error* behaviour and tends to request hints in sequences, whereas a problem solver of type H_2 makes sure not to submit wrong answers but requests a lot of help, even before having tried. This might lead to the assumption that this problem solver uses the help functionality instead of sufficient preparation. A problem solver of type H_3 does not request help right at the beginning and does not request help too often; when help is requested though, this is done in sequences. This may be indicative, for instance, of a learner that is interested in really understanding a problem before continuing. The problem solver of type H_4 is very similar to H_2, and in settings with a lower number of clusters these styles might have been combined. The number of clusters selected depends on the aspired level of granularity. If it is the aim in the actual setting to find rough types the students can be assigned to, one would chose a lower number of clusters and would get a combined type for $H2/H4$. If it is the aim in a setting to distinguish between subtypes of the same category, one would choose the result with a higher number cluster setting and get separate $H2$ and $H4$ types.

We can compare the results at this level to the help-seeking model discussed in [Aleven et al., 2006]. The authors introduce a taxonomy of "help-seeking bugs" in students' behaviour and list the following categories: *Help Abuse, Help Avoidance, Try-Step Abuse and Miscellaneous Bugs.* The type *Help Abuse* comprises behaviour like clicking through hints or asking for hints even if it would not be necessary because the student would be skilled enough to solve the task without help. The H_2 and H_4 types identified by our system partly correspond to this *Help Abuse* type in that a H_2/H_4 problem solver may also

show the behaviour of clicking through hints instead of spending more time on understanding the content before. Our model can, however, additionally identify if the student uses help before or after trying to submit an answer first. This undesired kind of behaviour can also be compared to *gaming the system* as explained in [Baker et al., 2006]; in fact, in another paper in this issue [Muldner et al., 2011], the authors explicitly define such behaviour as indicative of gaming, and use it to detect occurrences of abuse in the same ITS as used here. The *Try-Step Abuse* can be compared to the *Trial and Error* behaviour as shown by type H_1 who also tends to solve a problem too early even if not sufficiently skilled yet. Furthermore, some parallels can be drawn between the H_3 type and the *Help Avoidance* style described in [Aleven et al., 2006] concerning the general tendency to keep the amount of requested help low. However, the *Help Avoidance* type mainly considers trying unfamiliar steps without help and could thus also be described as a subcategory of the *Try-Step Abuse* type. In our case, the H_3 and H_1 are clearly distinct as the behaviour of a problem solver of type H_3 can also be described by the desire to avoid the submission of incorrect answers. The level II clustering experiments have thus not only confirmed the taxonomy of "help-seeking bugs" described in [Aleven et al., 2006] but also added some distinct aspects to it.

Table 3 This table shows four problem solving styles in the *Help-Seeking* dimension discovered by the clustering process (again with the example configuration of 8 clusters). The remaining clusters not shown here contain non-*Help-Seeking* behaviour. The syntax is to be read as follows: the percentage results have been abstracted to the five categories *very low, low, medium, high, very high*, which are represented by the more easy to read identifiers $--$, $-$, o, $+$, $++$.

	Size	$PRIOR\_I$	$PRIOR\_H^*$	$TRANS\_I\_H^*$	$TRANS\_H^*\_H^*$	$PERC\_I$	$PERC\_H^*$
H_1	- -	o	o	o	+	o	o
H_2	+	- -	+	-	++	- -	+
H_3	o	-	-	-	+	-	o
H_4	o	- -	+	-	++	- -	+

Table 4 shows the comparison of optimization results with different weight configurations for the data sets discussed so far, for the year 2008. It can clearly be observed how the optimum number of clusters changes with changing weights, i.e., a changing clustering purpose. In row 1, the default results without optimization towards a specific focus is shown. The results for the optimum number of clusters lie between 5 and 7 for the different data sets. Rows 2 to 5 show the optimum number of clusters, emphasizing one specific criterion and equally considering the others with low priority. Here, we can observe that the criteria $SE(C_n)$ and $PE(C_n)$ both suggest a lower number of clusters if weighed high, whereas the criteria $V(C_n)$ and $EPE(C_n)$ suggest a higher number of clusters, compared to the default results. Thus, we can conclude, for instance, that if we want to minimize the prediction error, we

should use a number of clusters $n > default$. Rows 6 to 11 show the results for a process that optimizes for combinations of two criteria.

Table 4 This table compares the optimization results for different data sets Markov, Statistical, Both, Trial&Error (TE) and Help-Seeking (HS) with different weight configurations for the year 2008. The numbers in the data sets' rows show the optimum number of clusters found for the respective data set and weight configuration.

Weights (w_s, w_p, w_v, w_e)	***Markov***	***Statistical***	***Both***	***TE***	***HS***
1 — 1 — 1 — 1	5	6	7	6	5
3 — 1 — 1 — 1	4	4	4	4	3
1 — 3 — 1 — 1	4	3	4	4	3
1 — 1 — 3 — 1	7	7	11	9	7
1 — 1 — 1 — 3	8	6	9	10	9
1 — 1 — 2 — 2	8	9	10	10	8
1 — 2 — 1 — 2	5	5	6	6	5
1 — 2 — 2 — 1	5	5	7	6	5
2 — 2 — 1 — 1	4	4	4	4	3
2 — 1 — 1 — 2	5	6	6	6	5
2 — 1 — 2 — 1	5	5	7	6	5

4.4.3 Level III (Open discovery)

This level, as its name suggests, goes one step further than its predecessor and is intended to perform open-ended analysis with the goal of identifying, firstly, potential new dimensions of learning behaviour, and, secondly, concrete patterns within each dimension. For our case study, this process has been apparently targeted towards the identification of concrete types of problem solving behaviour.

This level is controlled by the system (excluding, of course, the assessment and interpretation of results that need to be performed by a human operator) whose task it is to: (a) automatically select feature combinations with high discriminatory capacity, (b) create new data sets containing attributes of one feature combination each, (c) perform clustering on each of the new data sets, and (d) analyse the resulting clusters for significant trends in order to autonomously detect problem solving styles. For level III, we implemented an additional feature selection unit performing the necessary steps as described in detail below.

Automatic Feature Selection and Combination: the feature selection unit receives a basic data set, extracts the attributes and starts the feature combination process. The results presented here are based on the data of SET_BOTH from spring 2008 with aggregated help states. The initial data set contains 20 features. Both learning dimensions and concrete styles can be described and defined by a much smaller number of attributes. The process of selecting subsets of the initial feature set, clustering on them and determining the quality of the set according to, e.g., its discriminatory capacity, is a

relatively simple one. However, it cannot be done by humans because of the high number of calculations required. The problem of finding combinations is of exponential complexity, hence we do not compute all possible combinations (i.e., the power set) but limit the number of features in the resulting combination sets. As the number of highly relevant features describing a dimension or concrete type in the aggregated help setting turned out to be relatively low (indicated by manual analysis of previously defined dimensions and styles) for the data and settings used here, we decided to set a limit of 7 for the automated feature combination process. The system then computes all combinations with n elements, where $1 \leq n \leq 7$. This limit is not a universal suggestion for similar approaches but must be individually determined for different scenarios. A possible direction of future work includes the automatic detection of the potentially best-fitting limit for the number of features (see Section 6 for a more in-depth discussion of this limit).

Creation of New Data Sets: using the feature combinations computed before, the process is continued by creating a "copy" of the original data set containing only the selected features and values for these features. This results in a high amount of data, all depicting different aspects of the same activities produced by the users.

Clustering on the New Data Sets and Cluster Analysis: all data sets are passed through to the clustering process. The clustering results are then stored and compared according to a specific algorithm measuring average cluster quality $Q(FS_i)$ for a feature set i (see Equation 7). The algorithm we used is based on Linear Discriminant Analysis (LDA) as described in [Martínez and Kak, 2001], maximizing the distance between cluster centroids and minimizing the average distance between the elements within the clusters (we used the Euclidean Distance [Black, 2004] in all cases). A different approach would be to use an algorithm that does not keep the original features in order to select subsets, but creates new ones based on combinations of the original ones. In [Martínez and Kak, 2001], one such approach, Principal Component Analysis (PCA), is described and compared to LDA. In our case, we chose LDA in order to preserve information about the significance of the original features.

$$Q(FS_i) = \frac{D_b * w_b}{D_w * w_w} \tag{7}$$

where D_b is the average distance between the cluster centroids, D_w is the average distance between the elements within a cluster, averaged again over the clusters, w_b and w_w are weights (with $0 < w_* \leq 1$). Here we used the default equal weight configuration with $w_* = 1$. The results for $Q(FS_i)$ for all feature sets are the basis for a feature set ranking. The top ranked feature sets become then the system's recommendation as potentially meaningful dimensions. These recommended feature sets are finally analysed by a human investigator who makes a final decision about what set to pass back to Level II clustering in order to detect concrete learning/problem solving types. Tables 5 and 6 show the top ranked feature sets for the base data set SET_BOTH from spring 2008 with aggregated help states.

Table 5 This table shows the feature combinations ranked 1 to 10 and their respective $Q(FS_i)$ results.

Rank	$Q(FS_i)$	***Features***
1	5.8086	TRANS_PROB_H_H
2	4.0364	TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
3	3.9017	TRANS_PROB_H_H TRANS_PROB_H_E
4	3.7714	TRANS_PROB_C_H TRANS_PROB_H_E PERC_HELP_STEP
5	3.7149	TRANS_PROB_H_I TRANS_PROB_H_H
6	3.5501	TRANS_PROB_H_E PERC_HELP_STEP
7	3.5124	TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
8	3.4429	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H PERC_HELP_STEP
9	3.4331	TRANS_PROB_C_H PERC_HELP_STEP
10	3.4076	PRIOR_PROB_H TRANS_PROB_C_H PERC_HELP_STEP
11	3.3662	TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP

Table 6 This table shows the feature combinations ranked 11 to 20 and their respective $Q(FS_i)$ results.

Rank	$Q(FS_i)$	***Features***
12	3.3302	TRANS_PROB_H_H TRANS_PROB_E_H
13	3.2656	PRIOR_PROB_H TRANS_PROB_H_H TRANS_PROB_H_E
14	3.2652	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_E_H PERC_HELP_STEP
15	3.2270	PRIOR_PROB_C
16	3.2250	TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
17	3.2092	TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP
18	3.2090	TRANS_PROB_C_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
19	3.2026	TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H PERC_HELP_STEP
20	3.1810	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP

The first 20 feature combinations shown in Table 5 and 6 do not include sets with 6 or 7 features. The first 6-feature combination is ranked 39th and contains $PRIOR\_PROB\_H$, $TRANS\_PROB\_C\_H$, $TRANS\_PROB\_H\_I$, $TRANS\_PROB\_H\_H$, $TRANS\_PROB\_E\_H$ and $PERC\_HELP\_STEP$. The first 7-feature combination is ranked 79th and contains $PRIOR\_PROB\_H$, $TRANS\_PROB\_C\_H$, $TRANS\_PROB\_H\_I$, $TRANS\_PROB\_H\_H$, $TRANS\_PROB\_H\_E$, $TRANS\_PROB\_E\_H$, and $PERC\_HELP\_STEP$. Given the total number of $\sim$ 140000 ranks, these sets can still be expected to be potentially interesting. Already at a first glance, the top ranked results suggest variations of a *Help-Seeking* dimension similar to the one we manually defined for Level II, based on descriptions in related literature.

For a more detailed analysis, the system provides a partitioned ranking that assigns the feature sets to groups based on their number of contained features and compares the sets within each group as shown in Table 7. The table's rows show groups of feature sets with the same number of features in them. The individual feature sets in a row appear in descending order of their rank within the group. Each feature set is described in terms of the actual

features it contains, its overall ranking (not group-related), and the $Q(FS_i)$ results.

The top ranked feature sets listed in Table 7 were used to carry out experimental clustering, the results of which are shown in Table 8. The number of clusters used in this case (5) was selected to ensure the preservation of similar types of behaviour. The results listed in this table are an example of what a human observer would see when applying level II clustering on the dimensions suggested by level III (note that several feature sets other than the top ranked ones would normally also be included).

The rest of this section provides a detailed analysis of the results depicted in the aforementioned two tables for the top-ranked feature sets in each group. Each feature set is treated as a potential dimension of problem solving behaviour and different types of behaviour are identified for each such dimension. Table 9 provides a collective overview and additional explanations about the types identified. These different types along each dimension can be directly used for determining appropriate adaptive system interventions, in a manner similar to the one exemplified in Section 5 for the previously identified types of Help-Seeking behaviour.

On the basis of the above, the results obtained can be analysed as follows.

$Rank_G = 1, n = 1$ This dimension, defined by one single feature, models the users' tendency to request help in sequences. The clusters show a clear distinction between different types of behaviour (e.g., cluster 2 vs. cluster 4). The concrete types we could identify along this dimension are $T_{1.1}$ showing a strong tendency to request help in sequences (see clusters 1, 3, 4), $T_{1.2}$, not requesting help in sequences (see cluster 2), and $T_{1.3}$, occasionally requesting help in sequences (see cluster 0).

$Rank_G = 1, n = 2$ This dimension is defined by two features, modelling again the tendency to request help in sequences, and, additionally, to end a problem solving sequence with a hint request (i.e., in most cases, without having submitted a final solution). Three clusters (1, 2, 4) show similar results and can thus be summarized as type $T_{2.1}$, tending to request help in sequences and not to conclude a problem with a hint request. The second type identified here, $T_{2.2}$ is described in clusters 0 and 3, where users request help in sequences occasionally and also occasionally end a problem solving sequence with a hint.

$Rank_G = 1, n = 3$ This dimension is defined by three features, adding the percentage of help requests to the two attributes already explained for $Rank_G = 1, n = 2$. Here, we could identify significant types as follows. Learners of $T_{3.1}$ do not request help in sequences, do not end a problem solving sequence with help requests, and in general request only little help (as seen in cluster 2). Learners of type $T_{3.2}$ tend to request help in sequences but do not end problems with help requests and in general tend to request a lot of help (see clusters 1 and 3). In cluster 4 we can find the behaviour of type $T_{3.3}$, not requesting too much help and when so, not in sequences, but showing a strong tendency to end

Table 7 This table shows the feature combinations ranking split into groups containing sets with equal number of elements n. The $Gr.$-column shows the number of features in the respective group, $Rank_G$ is the rank within a group. For every $RANK_G$ in a specific group, the following information is provided: the features in the set, the overall ranking (not group-related), and the $Q(FS_i)$ results.

Gr.	$Rank_G = 1$	$Rank_G = 2$	$Rank_G = 3$	$Rank_G = 4$
$n = 1$	TRANS_PROB_H_H	PRIOR_PROB_C	TRANS_PROB_E_C	TRANS_PROB_I_C
	5.0806	3.2270	2.9853	2.6309
	1	15	37	134
$n = 2$	TRANS_PROB_H_H TRANS_PROB_H_E	TRANS_PROB_H_I TRANS_PROB_H_H	TRANS_PROB_E_H PERC_HELP_STEP	TRANS_PROB_C_H PERC_HELP_STEP
	3.9017	3.7149	3.5501	3.4331
	3	5	6	9
$n = 3$	TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	TRANS_PROB_C_H TRANS_PROB_H_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_H_H TRANS_PROB_H_E
	4.0364	3.7714	3.4076	3.2656
	2	4	10	13
$n = 4$	TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H PERC_HELP_STEP	TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_E_H PERC_HELP_STEP
	3.5124	3.4429	3.3662	3.2651
	7	8	11	14
$n = 5$	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP	TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H PERC_HELP_STEP
	3.1810	3.1348	3.1138	3.0901
	20	23	25	26
$n = 6$	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_I TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
	2.9672	2.9378	2.8925	2.8755
	39	44	52	59
$n = 7$	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_I TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_I TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_C TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP
	2.7744	2.6684	2.6400	2.6299
	79	116	128	136

Table 8 This table shows experimental clustering results based on the top ranked feature sets listed in Table 7. In order not to neglect variations of similar types of behaviour, we used the relatively high number of 5 clusters here. The results listed here are an example of what a human observer would see when applying level II clustering on the dimensions suggested by level III. Of course, a human observer would be provided not only one top ranked feature sets but several. The values for the clusters denote the mean for the respective feature in this cluster.

Feature Set	***Features***	$C0$	$C1$	$C2$	$C3$	$C4$
$Rank_G = 1, n = 1$	TRANS_PROB_H_H	0.25	0.69	0.03	0.52	0.77
$Rank_G = 1, n = 2$	TRANS_PROB_H_H	0.25	0.69	0.75	0.15	0.51
	TRANS_PROB_H_E	0.25	0.06	0.02	0.27	0.01
$Rank_G = 1, n = 3$	TRANS_PROB_H_H	0.25	0.74	0.01	0.60	0.05
	TRANS_PROB_H_E	0.25	0.03	0.00	0.06	0.88
	PERC_HELP_STEP	0.00	0.67	0.10	0.34	0.28
$Rank_G = 1, n = 4$	TRANS_PROB_H_I	0.25	0.03	0.08	0.00	0.23
	TRANS_PROB_H_H	0.25	0.74	0.69	0.00	0.34
	TRANS_PROB_H_E	0.25	0.04	0.26	1.00	0.08
	PERC_HELP_STEP	0.00	0.71	0.48	0.24	0.23
$Rank_G = 1, n = 5$	PRIOR_PROB_H	0.00	0.76	0.45	0.11	0.09
	TRANS_PROB_C_H	0.00	0.01	0.04	0.01	0.02
	TRANS_PROB_H_H	0.24	0.72	0.70	0.66	0.18
	TRANS_PROB_E_H	0.30	0.12	0.14	0.09	0.18
	PERC_HELP_STEP	0.00	0.72	0.59	0.34	0.09
$Rank_G = 1, n = 6$	PRIOR_PROB_H	0.75	0.00	0.45	0.11	0.07
	TRANS_PROB_C_H	0.00	0.00	0.05	0.01	0.01
	TRANS_PROB_H_I	0.04	0.25	0.07	0.10	0.59
	TRANS_PROB_H_H	0.73	0.25	0.67	0.52	0.01
	TRANS_PROB_E_H	0.71	0.03	0.42	0.10	0.09
	PERC_HELP_STEP	0.72	0.00	0.58	0.15	0.07
$Rank_G = 1, n = 7$	PRIOR_PROB_H	0.00	0.43	0.74	0.31	0.09
	TRANS_PROB_C_H	0.00	0.04	0.04	0.05	0.01
	TRANS_PROB_H_I	0.25	0.07	0.04	0.01	0.18
	TRANS_PROB_H_H	0.25	0.71	0.73	0.13	0.48
	TRANS_PROB_H_E	0.25	0.04	0.04	0.83	0.03
	TRANS_PROB_E_H	0.03	0.41	0.71	0.32	0.08
	PERC_HELP_STEP	0.00	0.57	0.72	0.28	0.26

Table 9 This table provides a concise overview about the types discovered in level III clustering, including a detailed description.

Type	***Description***
$T_{1.1}$	Requests help in sequences.
$T_{1.2}$	Does not request help in sequences.
$T_{1.3}$	Occasionally requests help in sequences.
$T_{2.1}$	Tends to request help in sequences and does not conclude problem solving sequences with help requests.
$T_{2.2}$	Occasionally requests help in sequences and occasionally concludes problems with help requests.
$T_{3.1}$	Does not request help in sequences, does not end a problem solving sequence with help requests, requests only little help.
$T_{3.2}$	Tends to request help in sequences, does not end problems with help requests, tends to request a lot of help.
$T_{3.3}$	Does not request much help and when so, not in sequences, shows a strong tendency to end problem solving sequences with hints.
$T_{3.4}$	Does not use help at all.
$T_{4.1}$	Stops problem solving sequences with help requests in 100% of the cases.
$T_{4.2}$	Shows a very high help request rate, a strong tendency to request help in sequences and a very low rate of incorrect submissions or quits after a hint.
$T_{4.3}$	Behaves in a similar way as $T_{4.2}$, shows a slightly lower rate of help requests and a medium rate of quits after a hint.
$T_{4.4}$	Shows a medium rate of help requests, a medium rate of incorrect attempts or further help requests after a help request, and a low rate of quits after a help request.
$T_{5.1}$	Shows a high help rate, a high prior probability for the use of help and a tendency to request help in sequences.
$T_{5.2}$	Shows a high help sequence rate, a medium overall help request rate and a relatively low prior probability for help.
$T_{5.3}$	Is similar to $T_{5.2}$ but shows a lower rate of help sequences and a lower overall help rate.
$T_{6.1}$	Shows a relatively high prior probability for help requests, a high general help rate and a tendency to help request sequences.
$T_{6.2}$	Like $T_{6.1}$, and shows a high help sequence rate.
$T_{6.3}$	Like $T_{6.1}$, and shows a very low help sequence rate and a relatively high percentage of incorrect attempts after a hint.
$T_{7.1}$	shows a strong tendency to close a problem solving sequence with a help request.

problem solving sequences with hints (i.e., not completing them). In cluster 0 we can see that this dimension is more expressive than the previous ones. Here, the percentage of requested help steps is low enough to round down to 0.00. Thus, the values indicating tendencies of requesting help in sequences and of ending a problem solving sequence with a hint are not relevant, as they are based on a very small set of samples. At this point, we may conclude that the types discovered before are useful, but only in combination with a basic statistical indicator on the general use of help. We define type $T_{3.4}$ behaviour as tending to not use help at all.

$Rank_G = 1, n = 4$ This dimension adds to the previously described features the probability of submitting a wrong answer directly after a hint request. Cluster 0 behaves in the same way as for $Rank_G = 1, n = 3$, therefore, we can not define a new type. Cluster 3 identifies a type of behaviour $T_{4.1}$ that has not been detected by the previously analysed dimensions; learners of this type stop their problem solving sequence with a hint request in 100% of the cases while not showing a generally very low help request rate. This type of behaviour is rare and in this case only affects 1% of the problem solving sequences monitored. In the clusters 1, 2, and 4 we identify the types $T_{4.2}$, $T_{4.3}$, and $T_{4.4}$. $T_{4.2}$ shows a very high help request rate, a strong tendency to request help in sequences and a very low rate of incorrect submissions or quits after a hint. $T_{4.3}$ differs from $T_{4.2}$ only in a slightly lower rate of help requests and a medium rate of quits after a hint, and $T_{4.4}$ is defined by a medium rate of help requests in general, a medium rate of incorrect attempts or further help requests after a help request, and a low rate of quits after a help request.

$Rank_G = 1, n = 5$ This dimension comprises, in addition to the already discussed general help rate and the tendency to request help in sequences, the prior probability for help, i.e. when users request help as a first step, before having tried to submit a solution, and the rate of requested hints directly following a correct attempt. The latter is not sufficiently discriminatory however and is therefore not a decisive factor for the identification of types. Again, cluster 0 does not suggest a new type but can be described by $T_{3.4}$. Clusters 1 and 2 define $T_{5.1}$ by a high help rate and a high prior probability for the use of help. Furthermore, this type includes a tendency to request help in sequences. $T_{5.2}$ is derived from cluster 3, defined by a high help sequence rate, a medium overall help request rate and a relatively low prior probability for help. Type $T_{5.3}$ is similar to $T_{5.2}$ regarding the prior probability for help requests but differs in the other aspects and includes a lower rate of help sequences and in general a lower help rate.

$Rank_G = 1, n = 6$ This dimension adds to the features in $Rank_G = 1, n = 5$ the probability of a wrong answer submission after a hint request. Cluster 1 again corresponds to and can be sufficiently well described by $T_{3.4}$. Clusters 0 and 2 identify type $T_{6.1}$ and show a relatively high prior probability for help requests, a high general help rate and a tendency to help request sequences.

Clusters 3 and 4 both show a low help rate, a low prior probability for help requests and a very low probability of help requests after a correct submission. Nevertheless we identify two different types $T_{6.2}$ and $T_{6.3}$. The first is additionally characterized by a high help sequence rate, whereas the second does, on the contrary, show a very low help sequence rate but a relatively high percentage of incorrect attempts after a hint.

$Rank_G = 1, n = 7$ This dimension again adds to the features in $Rank_G = 1, n = 6$ the probability of a help request being the last activity in a problem solving sequence. We can again identify $T_{3.4}$ in this dimension. In addition to the types already described earlier, cluster 3 shows a new style strongly dependent on the new feature. Learners of this type $T_{7.1}$ show a strong tendency to close a problem solving sequence with help, which in most cases is indicative for "giving up" before the problem was solved.

We conclude that dimensions with only one or very few features can be indicative of problem solving types but results are prone to being distorted. A very high number of features may however not allow for the identification of the most significant types but rather suggest a range of "subtypes" many of which could be combined. In order not to fall prey to either of these potential threats, we suggest a medium number of features for the purpose of dimension detection that lies between a fourth and a third of the overall count. The results presented here, show that the *Help-Seeking* dimension dominates. However, already if we consider the 5 top-ranked results of every group, we discover a different dimension including the features $PRIOR\_PROB\_0$, $PRIOR\_PROB\_H$, $TRANS\_PROB\_C\_H$, $TRANS\_PROB\_H\_I$, $TRANS\_PROB\_H\_H$, $TRANS\_PROB\_E\_H$, and $PERC\_HELP\_STEP$, including the rate of initial correct attempts (i.e., without having requested help or submitted an incorrect answer) and the rate of incorrect attempts directly following a help request. Clustering along this dimension, we get, among others, the following resulting types: (a) including very high probability for correct answers at first attempt, extremely low help request rate, low help sequence rate, and (b) including a medium rate of correct answers at first attempt, a low rate of initial help requests and a relatively low overall help rate. Type (b) is very similar to the *Trial and Error* type discussed in section 4.4.1 where we did not operate at the level of dimensions yet but considered a predefined concrete style.

5 Closing the Circle – Potential System Interventions

This section discusses ways in which the results and findings of the clustering approach described here can be fed back into the adaptation cycle. In general terms, this can be done by integrating the novel information into the user models and by providing additional adaptive system interventions based on them.

The way in which the derived information can be integrated into a user- or learner- model very much depends on the modelling approach used in the adaptive e-learning system. For systems utilizing simple, vector-based models, one could, for instance, introduce the identified dimensions as behavioural attributes of the learner, and use the associated patterns and types as the value space of the said attributes. A more elaborate modelling scheme could maintain different behaviour models for different types of problems, to account for the fact that learners may employ different strategies in each case (a subject we return to in Section 6). Another improvement over the basic scheme described before would be to maintain a set of possible behavioural patterns per attribute, indicating the likelihood that they be exhibited through probabilities. The addition of semantic / causal relationships to the later could also give rise to (or be used as the basis for) a Bayesian learner model. These are, of course, only very few of the feasible approaches one might employ for modelling dimensions and patterns in an adaptive system, and a full enumeration of possibilities is beyond the scope of this paper. Nevertheless, some of the examples presented in the following subsections do also incorporate a discussion of the concrete models that could underlie specific intervention approaches.

In the rest of this section we suggest possible system interventions based on selected problem solving types discussed in section 4, namely the *Trial and Error* type and the different *Help-Seeking* types H_1, H_2/H_4 and H_3. System interventions can be grouped into means of individual user support (see, for instance, [Koedinger and Aleven, 2007]) and means of collaboration support (see, for instance, [Soller et al., 2005] or [Walker et al., 2009]), yet based on individual users' model information. It should be noted that the example interventions proposed in this section are not being suggested as the best possible approaches in the respective cases (something that would definitely also depend on the didactic approach employed). They are rather meant to demonstrate how the adaptation cycle can be completed, with the participation of the adaptation / interaction designer, on the basis of user information discovered through the proposed approach.

5.1 Supporting Individual Users

Focusing on the provision of help, we have to deal with the trade-offs between giving and withholding information or assistance, a problem defined as *assistance dilemma* in [Koedinger and Aleven, 2007]. The interactive learning environment has to decide when and to what degree a student should be given information or provided with assistance (e.g., discussed in [Rummel and Krämer, 2010], [Koedinger and Aleven, 2007], or [Borek et al., 2009]). In some cases, the system might even decide not to provide help at all or, to the contrary, provide a full solution to a problem [Razzaq and Heffernan, 2009]. Furthermore, the system should be able to offer assistance in a way that suppresses undesired student behaviour like *gaming the system* [Baker et al., 2006], [Baker et al., 2008]. Usually, intelligent tutors have a *production rule model*

that represents the target competence that the tutor is meant to help students acquire [Koedinger and Aleven, 2007]. This model thus enables the tutor to solve the same class of problems the students have to solve. In [Koedinger and Aleven, 2007], the authors describe two basic algorithms, *model tracing*, using the model to interpret each student action, and *knowledge tracing*, estimating how well a student has mastered each key production rule. The results of the *model tracing* steps are then used to provide students with feedback and to individualize instructional advice. Therefore, the *model tracing* process is a potentially very important point of contact for our approach in that the new model information can be used to determine the kind and amount of feedback and individualized support to be offered. As the Andes tutoring system [VanLehn et al., 2005] (which generated the data used in this case study) uses a variant of the *model tracing* algorithm, the results presented here could later be fed back into the tutoring system, thus enhancing and refining the help provision strategies. Different types of system decisions regarding interactivity are listed in [Koedinger and Aleven, 2007], including feedback content, hint content and timing. These principles are of high relevance to the results reported herein, as the process of help provision is essential for the idea of supporting individual users based on their *help-seeking* styles.

In our concrete case, starting with learners in the *Trial and Error* style, a possible intervention might have the goal of preventing the student from making uneducated guesses and encouraging the use of help. Therefore, the system might decide to offer initial hints directly after the description of the problem. This kind of hints would probably not exhaust the level of detail available, but bring the student on the right track and engage interest.

In the case of *Help-Seeking*, the kind of interventions possible would strongly depend on the actual concrete type of behaviour. For example, an H_1 type student might invoke similar system actions as a *Trial and Error* type student, whereas H_2/H_4 and H_3 would probably receive different treatment. H_2/H_4 types seem to show a natural aversion for submitting incorrect answers but use a disproportionate amount of help. In order to encourage a more independent problem solving approach, the system might limit the available hints for these students.

In the case of H_3 type students, the system might try to assist them in finding a more balanced use of help. These students might generally have a high inhibition threshold regarding the request of help. This is affirmed by the fact that they do not request help often, but if they do, they request it in long sequences, which could mean that they only ask for help if they are extremely insecure about a problem. The system could, in this case, actively offer additional help during the problem solving process, based for example on the time already spent on the problem.

5.2 Supporting User Groups and Collaboration

As explained in [Dillenbourg et al., 1996], collaborative learning can be viewed as either comprising independent cognitive systems which exchange messages, or as a single cognitive system with its own properties. For the first understanding, the unit of analysis is the individual, whereas for the second understanding, it is the group. The proposed approach can, in general, be applied from both perspectives. Since, in our case study, the phase of analysis treated users as individuals, we aim at using the information in individual users' models that might be suggestive of traits that influence collaboration behaviour. Adaptive collaboration support can be split into the two phases *adaptive support for the establishment of collaboration* and *adaptive support during the collaboration process* [Paramythis, 2008]. Again, the general approach is applicable in both cases; yet, the nature of the information analysed here is better suited for collaboration establishment support. This kind of support is usually based on learners' personal- and learning characteristics and preferences, either explicitly stated by the users, or observed or inferred by the system during the interaction process [Paramythis, 2008], [Carro et al., 2003a], [Quignard and Baker, 1999].

[Paramythis, 2008] identifies a set of high level requirements as prerequisites for adaptive collaboration support: (a) capability to automatically collect/infer user- and learner profile data of individual learners, (b) capability to collect/infer and model collaboration activity data for individual learners, (c) capability to represent and employ algorithms/strategies that govern how learner information is used to identify appropriate collaboration partners, and (d) the opportunity to allow for alternative policies for, and approaches to, group initiation, with the latter being listed as optional supplement. We have already shown that the approach is clearly capable of (a) and (b). It is our ultimate goal to utilize the new model information to provide adaptive collaboration establishment support, including (c), keeping the implementation sufficiently generic to allow for (d) (also see section 6).

As already mentioned, the actual way of adaptively supporting collaboration establishment will be strongly dependent on the respective learning scenario and underlying teaching concepts and learning theories. The model is not restricted to supporting a specific theory or setting, but instead provides information that can be queried by arbitrary algorithms used for the identification of suitable collaboration settings. Adaptive collaboration establishment support includes encouraging students to cooperate with others, or recommendations of tools to use for collaboration, or partners to collaborate with [Carro et al., 2003b]. Group synthesis recommendations are based on specific rules may consider, for instance, users' preferences, backgrounds, interaction behaviour, etc.

In general, it may be desirable for the system to group students that could potentially benefit from cooperation in a joint session, taking into account criteria like complementarity or competitiveness [Alfonseca et al., 2006]. In [Alfonseca et al., 2006] and [Liu et al., 2008], for instance, the authors anal-

yse and model the student learning style based on the Felder and Silverman model [Felder and Silverman, 1988], [Felder and Brent, 2005] which categorizes learning styles along five dimensions (*active/reflective, sensing/intuitive, visual/verbal, sequential/global, inductive/deductive*), and conclude, among others, that (a) learning styles affect the performance of students when working together, (b) for the dimensions *active/reflective* and *sensing/intuitive*, the mixed pairs tend to work better, (c) heterogeneous groups in general get better results, and (d) students themselves tend to group randomly without respect to their learning styles. Their findings show that it is a worthy goal to utilize the models as a basis for group synthesis recommendations, and that learning styles are a potentially relevant criterion to base the later applied grouping algorithm on.

6 Discussion

In this article, we described a novel approach involving modeling of, and multi-level clustering based on, sequential learning activity data. We demonstrated its feasibility by applying it on real-world problem-solving data and running a variety of experiments to be able to compare and verify the results for different settings, data sets, students and academic terms. We demonstrated how the different clustering levels can detect

1. predefined problem-solving styles (level I),
2. problem-solving styles along predefined learning dimensions (level II), and
3. learning dimensions (level III) that again can comprise multiple problem solving styles.

For level I, we showed how our approach identifies the well-known problem solving style *Trial and Error* based on students' activity sequences. For level II, we chose to demonstrate the detection of problem-solving styles within predefined dimensions at the example of *Help-Seeking* behaviour. The process did not only successfully cluster for the required dimension but also identified concrete styles within that dimension. The results at this level also confirmed different models on help-seeking behaviour described in the recent literature [Aleven et al., 2006], [Baker et al., 2006]. For level III, we described a system-driven clustering approach aiming at the automatic detection of learning dimensions. The results did not only show that the process was in general able to autonomously identify dimensions, but also confirmed the styles and dimension we used in a pre-defined way for levels I and II.

A comparison of our approach to the most relevant efforts described in detail in Section 2 can be summarized as follows. Although the base data used here is comparable to what is described in [Romero and Ventura, 2010] and [Romero et al., 2008], the further processing and the ultimate goals are quite different. In these publications, the authors report the use of classification algorithms in order to predict students' final grades (i.e., supervised learning). In our case we concentrate on clustering (i.e., unsupervised learning). This

difference is also reflected in the selected way of data processing. Their system does not operate on the activity sequences directly, but first aggregates them, which leads to an abstracted representation of the data. This is what is avoided in our case, in order to not lose a dimension of information (relations and dependencies within the sequences) that is essential for pattern detection.

Compared to [Beal et al., 2006], which uses not only activity data but also students' self-reported motivation profile and teachers' ratings, our approach (using learner activity data only), does not rely on human effort during the monitoring process. The nature of the learners' activity data is similar to to what was described in this paper - correct and incorrect attempts and help requests were monitored by the ITS they used (Wayang Outpost [Wayang Outpost, 2010]). The main objective and further process, however, differ from ours: the aim in that case was to classify students in terms of the constellation of beliefs that they bring to the learning scenario, and, to show that multiple data sources can be used in order to reach this aim in an integrated way.

The approach described in [Amershi and Conati, 2009] is similar to ours in that it delays the necessity for human intervention until the end of the process, i.e., until after behavioural patterns have been automatically detected. It is different in that their system does not provide a clear notion of "correct" or "incorrect" behaviour, i.e., students don't receive feedback and help based on the correctness of their answers. In contrast to their approach, which uses only one feature vector per student (representing an aggregated version of this student's activities), in our approach, each of a student's problem solving sequences is converted to a feature vector, thus resulting in a much higher number of vectors and more fine-grained information represented. Yet, the long-term goals (individual adaptations and guidance based on the knowledge retrieved from the models) of the two approaches are similar, but are applied at different levels in different environments (exploratory systems vs. intelligent tutoring systems).

Comparing our approach to what is described in [Anaya and Boticario, 2009], we can observe that, although both approaches are based on clustering, the overall process discussed in [Anaya and Boticario, 2009] significantly differs from our ideas in several ways. Firstly, their approach requires a considerable amount of human intervention and effort, which is, arguably, not realistic to expect in real-world settings. Secondly, activity data is aggregated, thus losing sequential information. And, thirdly, the ultimate goals are again different. In our approach we use statistical information as an optional supplement to sequential data in order to detect specific types of behaviour, whereas in their approach, it is a main goal to reveal relations between statistical indicators and collaboration behaviour.

The HMM-based pattern detection approach of [Beal et al., 2007] is also different to ours in the modelling aims. Moreover, in our work, DMMs were chosen over HMMs as the sequences relevant for us exclusively consist of observable actions, thus predefining a certain state configuration. The clustering and prediction results of [Beal et al., 2007] are however highly relevant for us,

as they indicate Markov-based models to be an ideal way of modelling and analysing interrelated student activity data.

Finally, an interesting link can be established between our approach and the work of [Li and Yoo, 2006]. The authors, working with Bayesian Markov Chains in order to ultimately support adaptive tutoring, use models that are structurally comparable to ours. However, as in [Li and Yoo, 2006], the number of possible models is limited to exactly three (based on three predefined learning types), and the granularity of the outcome is also limited (and restricted to a very specific kind of information). In our approach, the models get dynamically created for the students' problem-solving sequences individually, which allows for a more fine-grained analysis and therefore also bears potential for much higher information gain.

The preceding sections, and discussion thus far, have concentrated on the argued benefits of the proposed approach. In the rest of this section we will turn our attention to challenges, implications and potential limitations of the approach itself, and of the case study described in this paper. Particular emphasis is placed on what would be required to transfer the approach to different (and potentially more challenging) settings and application domains.

To start with, due to the nature of the base data that was employed in this study, there exists a dimension of student behaviour that we did not need to explicitly address. Specifically, the problems that were presented to students were relatively uniform, which led to comparatively homogeneous / consistent problem-solving strategies being employed by individuals across the entire series of problems. However, if the problems to be solved showed greater variability (or, even, originated from different knowledge domains), students might employ a variety of strategies in addressing each kind / category of problems. This, in turn, might result in a high distribution of student behaviours among the identified clusters, making it difficult to identify "overall" behaviour models for individuals. Adjusting the proposed approach to cater for this dimension would involve: (a) segmenting the analysis to mirror the categorization of problems categories / domains, and (b) analyzing student entropies in comparison to problem entropies to determine whether any observed high distribution is attributable solely to students' intentionally employing alternative strategies on the same type of problems, or it is also (or predominantly) due to problem variability. One remedy that one could employ when multiple problem categories are evidently present would be to segment the user model explicitly distinguishing between behavioural patterns employed in the different settings. An alternative remedy would be to apply low weights for student entropies in a second clustering phase, so that clusters are formed based on aspects other than student entropies (thus accounting for the "instability" in students behaviour). The employed solution would also impact the adaptive interventions possible. Ideally, a system that is cognizant of the different types of problems that is presents students with, would be able to employ different "sub-models" of the student, each encapsulating the behaviour patterns / strategies that the student employs for each type of problem. These sub-models could be created through the grouping of patterns that typically occur

for a student when encountering problems of a specific type, and could thus be created and tested (for example in terms of their predictive capacity in anticipating exhibited behaviour) dynamically. Interventions, in this case, could still be associated with (families of) patterns, albeit on a per-problem-type basis.

Applying the proposed approach to learning domains other than problem solving would involve adjustments at different levels. Firstly, when applying this approach one has to decide how activities will actually be modelled in the Markov models. Our work has shown that DMMs are sufficient when all related activities and resulting states are observable. Nevertheless, the literature provides evidence that HMMs may also be applicable in such settings. Further comparative work would be required to establish the relative merits of each type of model when applied for the modelling of activities. Until such results are available, it may be advisable that researchers base their selection on the intrinsic characteristics of the modelled activities and states.

Once a specific type of Markov model has been selected, the next step is to decide the states that will be represented in the model. Even when modelling the activities of individuals, the findings in the reported work indicate that alternative aggregation levels may offer different advantages. An arguably even less straightforward task is deciding how to model the activities of groups. Two alternative approaches that may be considered include: modelling the activities of all members of the group as if the group were a single entity (the model states would then represent the collective "status" of the group after each activity has taken place); modelling the group activities from the perspective of the participating individuals, but introducing into the model also the activities to which the learner's own activities may be a "response" (including joint activities, such as online conferencing sessions). Naturally, there exist numerous variations that can be used based on the above themes, and the selection of a specific one should be based on the clustering goals.

Another decision that needs to be made in terms of the modelling of activities concerns the delineation of "episodes", i.e., of sets of activities that are semantically related and distinct from other such sets. One criterion that naturally lends itself for this type of delineation is the temporal dimension of activities; this, however, highly depends on the nature of activities, and the communication and collaboration tools that learners employ. Another possibility would be to use any structure that is intrinsic in the modalities and tools used for joint work (e.g., discussions that took place under a single topic within a discussion forum). It is anticipated that, often, it may be necessary to apply multiple criteria to establish episode boundaries.

A valid concern that may be voiced in relation to the proposed approach is the number of user activity categories that are possible (and monitored) in the context of the data used in the case study, and the relatively limited complexity of the resulting models (as expressed in the respective DMMs). As far as the first aspect of this observation is concerned, one could counter that the activities learners performed may have been limited, but their semantics were heavily dependent on the sequences in which they were performed (e.g.,

the number of times a student requested a specific type of help hint, and what else the student may have tried -or not- in the meantime). As the emphasis of our work lies precisely on the development of techniques that allow the analysis and subsequent detection of behavioural patterns in e-learning activities, we would argue that the presented case study goes a long way towards demonstrating that substantial, non-obvious results can be derived for such activity sequences, *although* the variability of activities is constrained. In fact, we expect that the proposed approach would be able to offer an even more diverse set of insights should more types of activities be available for analysis. From a different perspective, even if this approach were limited to only small sets of possible activity types, one could argue that richer sets of activities might also be possible to aggregate (e.g., by putting together all synchronous communication actions of learners within a group), thus arriving at a set of types and a level of granularity at which the proposed process would yield the best results. Despite the preceding argumentation, we would readily agree that the validation of the proposed approach in situations where more diverse activities are monitorable is a step that would further ascertain its more general applicability, and, indeed, an important component of future work.

The second stage of the proposed approach would also require adjustments when applied in different domains. The first step would involve the establishment of the metrics to be used as the basis on which to dynamically guide the clustering process. It is argued that the indices presented in section 4.2 are of a sufficiently general nature to be used as a starting point towards this end. Specifically, of the four indices, the Average Expected Prediction Error is domain-specific. However, this is a metric related to the "success" of an activity sequence, and can be replaced by any measure (or, depending on the circumstances, combination of measures) that sufficiently captures the semantics of desirable and undesirable effects of activity sequences in the target domain. From the rest of the indices, Average Problem Entropies is also largely domain specific, as it captures the context within which activities are carried. Furthermore, it represents the context within activities occur, and is therefore directly related to the "episodes" discussed above. Also important to note is that the two metrics thus far discussed influence each other as one is an indicator of success that is bounded by the activities contained in the other one's instances. A last decision that needs to be made in this context is the empirical establishment of the weights in equation 6, to match the researcher's objectives in deriving the appropriate number of clusters for different clustering goals.

Once the metrics have been established, one has to apply them in the different levels of the clustering process itself. This entails the determination of the data sets that will be actually fed into the process. Behavioural pattern-oriented clustering (level I) is probably the most straightforward, and involves the selection of those features in the models that convey states that result from activities that are included in the pattern being examined. Dimension-driven clustering (level II) is potentially more challenging, as it requires that all behaviour that may be related to a given dimension be included. This would be possible for learning dimensions that are well defined in the literature,

but possibly more difficult if a dimension is ill-defined, or defined in non-behavioural terms.

For "open discovery" clustering (level III), the main challenge facing researchers is the establishment of an upper limit for the feature combinations that will be used in the clustering process. In our case, the upper limit was established empirically on the basis of results from the previous levels. This may not be a feasible approach in all circumstances though (e.g., if this is the only level of the approach one applies). In these cases it may be feasible to establish an initial upper limit as a reasonable percentage of the features in the complete data set, or as an approximation based on the largest set of semantically coupled features in the data set. Another possibility -which, however, would require additional ground work to fully explore- would be the incorporation in the system of meta-information about the features and their relations, which could then be used to decide autonomously what attributes to use in what combinations (e.g., ensuring that semantically related attributes are not used disjointly). An alternative approach one might consider applying towards the same goal would involve the application of a "dimension reduction" method (such as Principal Component Analysis), with the aim of identifying a small number of "primary" features that would provide a sufficient characterization of the complete data set. Such approaches may bear promise at first sight, but have an intrinsic characteristic that renders them largely inappropriate for the the purposes discussed here: reduction in dimensionality for a data set is typically achieved through the combination of features. Due to this fact, the resulting features, however fewer in number and more "descriptive" they might be, they may have lost their original behavioural semantics, making the interpretation of the results of clustering based on these features an even more challenging endeavour than normal.

This brings us to a related limitation of our approach, as presented in this paper, which is that the qualitative analysis of the results of the third level of clustering may be challenging to carry out for complex behavioural models; put simply, especially when exploring novel domains of online learning activity, dimensions and related patterns may be difficult to recognise for the human observer. A potentially promising approach in partially facilitating this task might reside with the aforementioned introduction of semantic meta-information in the data, so that when candidate dimensions and patterns are decided upon by the system, their "interpretability" could also be systematically established. If this proved to be a too demanding extension to substantiate and implement, one could also consider providing appropriate visualizations of the semantic relations underlying the system's propositions for the human observers to more readily judge their connotations and significance. Implied in these suggestions is the requirement for tools capable of constructing visual representations of the dimensions and patterns harnessed from the clustering process; such tools could be, apparently, of great assistance even when having to deal with less complex behavioural models, as in our case study.

In closing, ongoing and future work concentrates on the implementation of what is described in section 5. We plan to apply the proposed approach on both short-term and long-term collaboration activity data monitored during a game-based team-work scenario and during Computer Science courses of the winter term 2010 in order to demonstrate that it is independent from specific activity types. At a higher level, we additionally want to show that the approach is applicable to domains other than learning. Our ultimate goal is to integrate our approach into a learning management system and run the dynamic clustering, analysis and intervention process during another Computer Science course. This would demonstrate our approach as a holistic concept including all stages of the adaptation cycle, from data acquisition to adaptive system interventions.

Acknowledgements

The work reported in this article was funded by the "Adaptive Support for Collaborative E-Learning" (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15). Data used in this research was provided by the Pittsburgh Science of Learning Center DataShop, funded by the National Science Foundation award No. SBE-0354420. We would like to thank the anonymous reviewers and the special issue editors who have helped us to substantially improve the quality of the paper.

References

Vincent Aleven, Bruce McLaren, Ido Roll, and Kenneth Koedinger. Toward Meta-Cognitive Tutoring: A Model of Help Seeking With a Cognitive Tutor. *International Journal of Artificial Intelligence and Education*, 16(2):101–128, 2006.

Vincent Aleven, Elmar Stahl, Silke Schworm, Frank Fischer, and Raven Wallace. Help Seeking and Help Design in Interactive Learning Environments. *Review of Educational Research*, 73(3):277–320, 2003.

Enrique Alfonseca, Rosa M. Carro, Estefanía Martín, Alvaro Ortigosa, and Pedro Paredes. The Impact of Learning Styles on Student Grouping for Collaborative Learning: A Case Study. *User Modeling and User-Adapted Interaction*, 16(3–4):377–401, 2006.

Saleema Amershi and Christina Conati. Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments. *Journal of Educational Data Mining*, 1(1):18–71, 2009.

Saleema Amershi and Cristina Conati. Automatic recognition of learner groups in exploratory learning environments. In Mitsuru Ikeda, Kevin Ashley, and Tak-Wai Chan, editors, *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 463–472. Springer Berlin / Heidelberg, 2006.

Antonio R. Anaya and Jesús G. Boticario. Clustering Learners according to their Collaboration. In *Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design*, pages 540–545, 2009.

Antonio R. Anaya and Jesús G. Boticario. A Data Mining Approach to Reveal Representative Collaboration Indicators in Open Collaboration Frameworks. In *Proceedings of the 2nd International Conference on Educational Data Mining (EDM09)*, pages 210–219, 2009.

Antonio R. Anaya and Jesús G. Boticario. Content-Free Collaborative Learning Modeling Using Data Mining. *International Journal on User Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems (this issue)*, 2011.

S. Anderson and S. Messick. Social Competency in Young Children. *Development Psychology*, 10(2):282–293, 1974.

Ryan S. Baker, Albert T. Corbett, Ido Roll, and Kenneth R. Koedinger. Developing a Generalizable Detector of When Students Game the System. *User Modeling and User-Adapted Interaction*, 18(3):287–314, 2008.

Ryan S.J.D. Baker. Data Mining for Education. In Barry McGaw, Eva Baker, and Penelope Peterson, editors, *International Encyclopedia of Education*, volume 7, pages 112–118. Elsevier, Oxford, UK, 3 edition, 2010.

Ryan S.J.D. Baker and Kalina Yacef. The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining*, 1(1):3–17, 2009.

Ryan S.J.D. Baker, Albert T. Corbett, Kenneth R. Koedinger, Shelley Evenson, Ido Roll, Angela Z. Wagner, Meghan Naim, Jay Raspat, Daniel J. Baker, and Joseph E. Beck. Adapting to When Students Game an Intelligent Tutoring System. In *Intelligent Tutoring Systems*, volume 4953/2006 of *Lecture Notes in Computer Science*, pages 392–401. 2006.

Ghulum Bakiri and Thomas G. Dietterich. Constructing high-accuracy letter-to-phoneme rules with machine learning. In R.I. Damper, editor, *Data-Driven Techniques in Speech Synthesis*, pages 27–44. Kluwer, Boston, MA, 2001.

Lena M. Ballone and Charlene M. Czerniak. Teachers' Beliefs About Accommodating Students' Learning Styles In Science Classes. *Electronic Journal of Science Education*, 6(2):1–40, 2001.

Carole Beal, Lei Qu, and Hyokyeong Lee. Classifying learner engagement through integration of multiple data sources. In *Proceedings of the 21st International Conference on Artificial Intelligence*, pages 151–156, 2006.

Carole Beal, Sinjini Mitra, and Paul Cohen. Modeling Learning Patterns of Students With a Tutoring System Using Hidden Markov Models. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED)*, pages 238–245, 2007.

Mordechai Ben-Ari. Constructivism in Computer Science Education. *ACM SIGCSE Bulletin*, 30(1):257–261, 1998.

Yoshua Bengio and Paolo Frasconi. Input-Output HMMs for Sequence Processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, 1996.

Paul E. Black. "Euclidean Distance" in Dictionary of Algorithms and Data Structures, U.S. National Institute of Standards and Technology, 2004. Available from http://www.itl.nist.gov/div897/sqg/dads/HTML/euclidndstnc.html. Accessed February 2010.

Alexander Borek, Bruce M. McLaren, Michael Karabinos, and David Yaron. How Much Assistance Is Helpful to Students in Discovery Learning? In *Learning in the Synergy of Multiple Disciplines*, volume 5794/2009 of *Lecture Notes in Computer Science*, pages 391–404. 2009.

Léon Bottou and Yann LeCun. Graph transformer networks for image recognition. *Bulletin of the 55th Biennial Session of the International Statistical Institute (ISI)*, 2005. URL `http://leon.bottou.org/papers/bottou-lecun-2005`.

Léon Bottou, Yoshua Bengio, and Yann Le Cun. Global Training of Document Processing Systems Using Graph Transformer Networks. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 489–494, 1997.

Elizabeth Brown, Tony Fisher, and Tim Brailsford. Real Users, Real Results: Examining the Limitations of Learning Styles Within AEH. In *Proceedings of the 18th Conference on Hypertext and Hypermedia*, pages 57–66, 2007.

Peter Brusilovsky, G. Chavan, and Rosta Farzan. Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. In *Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 24–33, 2004.

Thomas J. Butler and Genevieve Pinto-Zipp. Students' Learning Styles and Their Preferences for Online Instructional Methods. *Journal of Educational Technology Systems*, 34(2):199–221, 2006.

Rosa Carro, Alvaro Ortigosa, and Johann Schlichter. *Adaptive Collaborative Web-Based Courses*, volume 2722 of *Lecture Notes in Computer Science*, pages 130–133. Springer Berlin / Heidelberg, 2003a.

Rosa M. Carro, Alvaro Ortigosa, Estefana Martn, and Johann Schlichter. *Dynamic Generation of Adaptive Web-Based Collaborative Courses*, volume 2806 of *Lecture Notes in Computer Science*, pages 191–198. Springer Berlin / Heidelberg, 2003b.

Simon Cassidy. Learning Styles: An Overview of Theories, Models and Measures. *Educational Psychology*, 24(4):419–444, 2004.

Hyungshin Choi and Myunghee Kang. Analyzing Learner Behaviours, Conflicting and Facilitating Factors of Online Collaborative Learning using Activity System. In *Proceedings of the 2008 International Conference on Learning Sciences*, 2008. URL `http://www.fi.uu.nl/en/icls2008/318/paper318.pdf`.

Tammy Dewar and Dave Whittington. Online Learners and Their Learning Strategies. *Journal of Educational Computing Research*, 23(4):385–403, 2000.

Thomas G. Dietterich. Machine Learning for Sequential Data: A Review. In *Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in

Computer Science, pages 227–246. 2009.

Pierre Dillenbourg, M. Baker, A. Blaye, and C. O'Malley. The Evolution of Research on Collaborative Learning. In E. Spada and P. Reiman, editors, *Learning in Humans and Machine: Towards an Interdisciplinary Learning Science*, pages 189–211. Oxford:Elsevier, 1996.

Tom Fawcett and Foster Provost. Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

Richard M. Felder and Rebecca Brent. Understanding student differences. *Journal of Engineering Education*, 94(1):57–72, 2005.

Richard M. Felder and Linda K. Silverman. Learning and Teaching Styles in Engineering Education. *Journal of Engineering Education*, 78(7):674–681, 1988.

Mark Hall, Frank Eibe, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18, 2009.

W. Hämäläinen, J. Suhonen, E. Sutinen, and H. Toivonen. Data Mining in Personalizing Distance Education Courses. In *Proceedings of the 21st ICDE World Conference on Open Learning and Distance Education*, pages 18–21, 2004.

A.K. Jain, M.N. Murty, and P.J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

Matthew Jarvis. *The Psychology of Effective Learning and Teaching.* Nelson Thornes, 2005.

H. Jeong and G. Biswas. Mining Student Behaviour Models in Learning-by-Teaching Environments. In *Proceedings of the 1st International Conference on Educational Data Mining*, pages 127–136, 2008.

Essi Kanninen. Learning Styles and E-Learning. Master's thesis, Tampere University of Technology, 2008.

Mirjam Köck. Towards Intelligent Adaptive E-Learning Systems – Machine Learning for Learner Activity Classification. In *Proceedings of the 17th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 09)*, pages 26–31, 2009.

K. Koedinger, K. Cunningham, A. Skogsholm, and B. Leber. An Open Repository and Analysis Tool for Fine-Grained, Longitudinal Learner Data. In *Proceedings of Educational Data Mining 2008: 1st International Conference on Educational Data Mining*, pages 157–166, 2008.

Kenneth R. Koedinger and Vincent Aleven. Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. *Educational Psychology Review*, 19 (3):239–264, 2007.

K.R. Koedinger, R.S.J.d. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A Data Repository for the EDM Community: The PSLC DataShop. In Cristobal Romero, Sebastian Ventura, Mykola Pechenizkiy, and Ryan S.J.d. Baker, editors, *Handbook of Educational Data Mining.* CRC Press, Boca Raton, 2010.

David A. Kolb. *Experiential Learning.* Prentice Hall, 1984.

John D. Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

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

Cen Li and Jungsoon Yoo. Modeling Student Online Learning Using Clustering. In *ACM-SE 44: Proceedings of the 44th Annual Southeast Regional Conference*, pages 186–191. ACM, 2006.

Jia Li and Osmar R. Zaïane. Combining Usage, Content, and Structure Data to Improve Web Site Recommendation. In *Proceedings of the 5th International Conference on Electronic Commerce and Web Technologies (EC-Web)*, pages 305–315, 2004.

Ryan Lichtenwalter, Katerina Lichtenwalter, and Nitesh V. Chawla. Applying Learning Algorithms to Music Generation. In *Proceedings of the 4th Indian International Conference on Artificial Intelligence (IICAI 2009)*, pages 483–502, 2009.

Shuangyan Liu, Mike Joy, and Nathan Griffiths. Incorporating Learning Styles in a Computer-Supported Collaborative Learning Model. In *Proceedings of the International Workshop on Cognitive Aspects in Intelligent Adaptive Web-Based Education Systems (CIAWES 2008)*, pages 3–10, 2008.

Yuliang Liu and Ginther Dean. Cognitive Styles and Distance Education. *Online Journal of Distance Learning Administration*, 2(3), 1999. URL `http://www.westga.edu/~distance/liu23.html`.

.LRN. .LRN, 2010. www.dotlrn.org.

Estefanía Martín, Pablo A. Haya, Alan Dix, and Rosa M. Carro. Improving E-learning Recommendation Systems by using Markov Models in Adaptive Hypermedia. *International Journal on User Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems (this issue)*, 2011.

Aleix M. Martínez and Avinash C. Kak. PCA versus LDA. *IEEE Transactions on Knowledge and Data Engineering*, 23(2):228–233, 2001.

Andrew McCallum, Dayne Freitag, and Fernando C.N. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598, 2000.

Moodle. Moodle.org: Open-Source Community-Based Tools for Learning, 2010. www.moodle.org.

Kasia Muldner, Winslow Burleson, Brett Van e Sande, and Kurt VanLehn. An Analysis of Students Gaming Behaviors in an Intelligent Tutoring System: Predictors and Impacts. *International Journal on User Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems (this issue)*, 2011.

Sharon Nelson-Le Gall. Help-Seeking: An Understudied Problem-Solving Skill in Children. *Development Review*, 1(3):224–246, 1981.

Sharon Nelson-Le Gall. Help Seeking Behaviour in Learning. *Review of Research in Education*, 12(1):55–90, 1985.

R.S. Newman. Adaptive Help Seeking: A Strategy of Self-Regulated Learning. In D.H. Schunk and B.J. Zimmermann, editors, *Self-Regulation of Learning and Performance: Issues and Educational Applications*, pages 283–301. Hillsdale, NY:Erlbaum, 1994.

Alexandros Paramythis. Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet? In *Proceedings of the Workshop on Adaptive Collaboration Support, in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'08)*, pages 17–29, 2008.

Dilhan Perera, Judy Kay, Irena Koprinska, Kalina Yacef, and Osmar R. Zaïne. Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(6):759–772, 2009.

Jean Piaget. *The Construction of Reality in the Child*. New York: Basic Books, 1954.

Ning Qian and Terrence J. Sejnowski. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202(4):865–884, 1988.

Matthieu Quignard and Michael Baker. Favouring Modellable Computer-Mediated Argumentative Dialogue in Collaborative Problem-Solving Situtations. In *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, pages 129–136, 1999.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in speech recognition*, pages 267–296, San Francisco, CA, 1990. Morgan Kaufmann Publishers Inc.

Leena M. Razzaq and Neil T. Heffernan. To Tutor or Not to Tutor: That is the Question. In *Proceedings of the 2009 Artificial Intelligence in Education Conference*, pages 457–464, 2009.

Aaron S. Richmond and Rhoda Cummings. Implementing Kolb's Learning Styles into Online Distance Education. *International Journal of Technology in Teaching and Learning*, 1(1):45–54, 2005.

C. Romero and S. Ventura, editors. *Data Mining in E-Learning*, volume 4 of *Advances in Management Information*. WITPress.com, 2006.

Cristóbal Romero and Sebastián Ventura. Educational Data Mining: A Review of the State-of-the-Art. *IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews.*, 40(6):601–618, 2010.

Cristóbal Romero, Sebastián Ventura, and Enrique García. Data Mining in Course Management Systems: Moodle Case Study and Tutorial. *Computers & Education*, 51(1):368–384, 2007.

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)*, pages 8–17, 2008.

Nikol Rummel and Nicole Krämer. Computer-supported instructional communication: A multidisciplinary account of relevant factors. *Educational Psychology Review*, 22(1):1–7, 2010.

A.M. Ryan, P.R. Pintrich, and C. Midgley. Avoiding Seeking Help in the Classroom: Who and Why? *Educational Psychology Review*, 13(2):93–114, 2001.

Eduardo Salas, Dana E. Sims, and C.Shawn Burke. Is There a "Big Five" in Teamwork? *Small Group Research*, 36(5):555–599., 2005.

David T. Schaller, Minda Borun, Steven Allison-Bunnell, and Margaret Chambers. One Size Does Not Fit All: Learning Style, Play, and On-line Interactives. In *Proceedings of Museums and the Web 2007 - the International Conference for Culture and Heritage On-Line*, 2007. URL `http://www.archimuse.com/mw2007/papers/schaller/schaller.html`.

Terrence J. Sejnowski and Charles R. Rosenberg. Parallel Networks That Learn to Pronounce English Text. *Journal of Complex Systems*, 1(1):145–168, 1987.

Kristie Seymore, Andrew McCallum, and Ronald Rosenfeld. Learning Hidden Markov Model Structure for Information Extraction. In *Proceedings of the AAAI 99 Workshop on Machine Learning for Information Extraction*, pages 37–42, 1999.

Claude E. Shannon. A Mathematical Theory of Communication. In D. Slepian, editor, *Key Papers in the Development of Information Theory*. IEEE Press, New York, 1974. URL `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`.

Steven John Simon. The Relationship of Learning Style and Training Method to End-User Computer Satisfaction and Computer Use: A Structural Equation Model. *Information Technology, Learning, and Performance Journal*, 18(1), 2000. URL `http://www.osra.org/itlpj/simon.PDF`.

Amy Soller. Adaptive support for distributed collaboration. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 573–595. Springer Berlin / Heidelberg, 2007.

Amy Soller and Alan Lesgold. Modeling the process of collaborative learning. In Pierre Dillenbourg, H. Hoppe, Hiroaki Ogata, and Amy Soller, editors, *The Role of Technology in CSCL*, volume 9 of *Computer-Supported Collaborative Learning*, pages 63–86. Springer US, 2007.

Amy Soller, Alejandra Martínez, Patrick Jermann, and Martin Muehlenbrock. From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *International Journal of Artificial Intelligence in Education*, 15(4):261–290, 2005.

Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2):12–23, 2000.

Jun-Ming Su, Shian-Shyong Tseng, Chun-Han Chen, and Huan-Yu Lin. A Personalized Learning Content Adaption Mechanism to Meet Diverse User Needs in Mobile Learning Environments. *International Journal on User*

Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems (this issue), 2011.

Steven R. Terrell. Supporting Different Learning Styles in an Online Learning Environment: Does it Really Matter in the Long Run? *Online Journal of Distance Learning Administration*, 8(2), 2005. URL `http://www.westga.edu/~distance/ojdla/summer82/terrell82.htm`.

Edward Lee Thorndike. *Educational Psychology*. Kessinger Publishing , 1903.

Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. Conditional Random Fields for Activity Recognition. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8, 2007.

Kurt VanLehn, Collin Lynch, Kay Schulze, Joel A. Shapiro, Robert H. Shelby, Linwood Taylor, Don Treacy, Anders Weinstein, and Mary Wintersgill. The Andes Physics Tutoring System: Lessons Learned. *International Journal of Artificial Intelligence in Education*, 15(3):147–204, 2005.

César Vialardi, Javier Bravo, Leila Shafti, and Álvaro Ortigosa. Recommendations in Higher Education Using Data Mining Techniques. In *Proceedings of the 2nd International Conference on Educational Data Mining (EDM09)*, pages 190–199, 2009.

Erin Walker, Nikol Rummel, and Kenneth R. Koedinger. CTRL: A Research Framework for Providing Adaptive Collaborative Learning Support. *User Modeling and User-Adapted Interaction*, 19(5):387–431, 2009.

Wayang Outpost. Wayang Outpost, 2010. www.wayangoutpost.com.

Jungsoon Yoo, Cen Li, and Chrisila Pettey. Adaptive Teaching Strategy for Online Learning. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*, pages 266–268. ACM, 2005.

Author Biographies

Mirjam Köck is a Ph.D. candidate in Computer Science at the Johannes Kepler University (Linz, Austria). She received her degree from the Upper Austria University of Applied Sciences (Hagenberg, Austria) where she now holds a position as an assistant professor. Her primary research interests lie in the areas of adaptive systems, machine learning and educational data mining, collaborative e-learning and human-computer interaction. The paper summarizes the current state of her thesis work on unsupervised learning in the context of adaptive e-learning settings.

Dr. Alexandros Paramythis received his Ph.D. in the area of adaptive systems from the Johannes Kepler University (Linz, Austria) where he is currently employed as a researcher. He has long-standing experience in the design and development of adaptive systems, gained through participation in several research projects in the field. His research interests lie in the areas of adaptive support for personalized and collaborative learning, evaluation of adaptive systems, and meta-adaptivity.