



HAL
open science

Insights into Learning Competence Through Probabilistic Graphical Models

Anna Saranti, Behnam Taraghi, Martin Ebner, Andreas Holzinger

► **To cite this version:**

Anna Saranti, Behnam Taraghi, Martin Ebner, Andreas Holzinger. Insights into Learning Competence Through Probabilistic Graphical Models. 3rd International Cross-Domain Conference for Machine Learning and Knowledge Extraction (CD-MAKE), Aug 2019, Canterbury, United Kingdom. pp.250-271, 10.1007/978-3-030-29726-8_16 . hal-02520054

HAL Id: hal-02520054

<https://inria.hal.science/hal-02520054>

Submitted on 26 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Insights into Learning Competence through Probabilistic Graphical Models^{*}

Anna Saranti¹, Behnam Taraghi¹, Martin Ebner¹, and Andreas Holzinger⁴

¹ A. Saranti

² B. Taraghi

³ M. Ebner Department Educational Technology, Graz University of Technology
Münzgrabenstrasse 36/I, 8010, Graz, Austria
`s0473056@sbox.tugraz.at`

`b.taraghi@tugraz.at`

`martin.ebner@tugraz.at`

⁴ A. Holzinger Institute of Interactive Systems and Data Science, Graz University of
Technology

Inffeldgasse 16c/I, 8010, Graz, Austria

⁵ Institute for Medical Informatics, Statistics & Documentation, Medical University
Graz, Auenbruggerplatz 2, 8036 Graz, Austria

`andreas.holzinger@medunigraz.at`

Abstract. One-digit multiplication problems is one of the major fields in learning mathematics at the level of primary school that has been studied over and over. However, the majority of related work is focusing on descriptive statistics on data from multiple surveys. The goal of our research is to gain insights into multiplication misconceptions by applying machine learning techniques. To reach this goal, we trained a probabilistic graphical model of the students' misconceptions from data of an application for learning multiplication. The use of this model facilitates the exploration of insights into human learning competence and the personalization of tutoring according to individual learner's knowledge states. The detection of all relevant causal factors of the erroneous students answers as well as their corresponding relative weight is a valuable insight for teachers. Furthermore, the similarity between different multiplication problems - according to the students behavior - is quantified and used for their grouping into clusters. Overall, the proposed model facilitates real-time learning insights that lead to more informed decisions.

Keywords: Bayesian networks · Probabilistic graphical models · Learning analytics

^{*} The authors declare that there are no conflicts of interest and no ethical issues, no particular funding was achieved.

1 Introduction

1.1 Previous Work

The field of learning analytics is of increasing importance for educational research [46], [17], [20]. Moreover, it aims to assist the learning process by providing teachers a deeper insight into learning processes and learning results. Teachers play an essential role because they are responsible for intervening in a pedagogical adequate manner. Recently, the field makes heavy use of statistical machine learning [42]. Whilst educational data mining targets on automating learning activities, learning analytics supports educators in their daily routine.

The so called “1x1 trainer”⁶ is a learning analytics application developed by the department Educational Technology of Graz University of Technology, Austria. It uses the benefits of both fields, learning analytics as well as educational data mining [18], [19], [44].

The application poses exercises to students from the multiplication table with one decimal digit operands. The algorithm of the “1x1 trainer” adapts the sequence of given questions subject to the students answers individually, in order to improve individual learning progress. However, technically, this needs to react adaptively to the changes of the learning progress of a student. In that way, it would support each student according to the user’s distinct learning progress over the whole learning period. This underlying personalized adaptive learning algorithm shall discover weak mathematical knowledge of single students and alerts teachers just in time to adequately intervene.

The work is based on previous research that used data gathered by the “1x1 trainer”. Firstly, different mathematical questions were roughly classified according to the learners answers. Questions were considered to be more difficult than others when students required more attempts to answer them [49], [52]. Some specific questions could be identified as difficult for the majority of the users. The next step was to analyze the explanation of the errors made. Therefore, error types were assigned to falsely answered questions, which correspond to the innate cognitive and conceptual learning shortcomings of the users [51]. The “relative difficulty” of those questions - 2×3 seems to be simpler than 7×8 - played no role in identifying the error types. More explanation on error types follows in section 2.

1.2 Bayesian Student Models

There is a plethora of learning applications that use probabilistic graphical models (also called Bayesian models/networks [40]) to model student’s knowledge. These models have started making an impact in the research of causal learning and inference generally, but there are good arguments that even children’s causal learning could also be modeled in this way [11].

Most applications belong to the category of intelligent tutoring systems (ITS) [14] or adaptive educational systems (AES) [4]. The main goal of an ITS is to

⁶ <https://schule.learninglab.tugraz.at/einmaleins/>, Last accessed 18 June 2019

provide personalized recommendations according to the different learning styles, whereas AES adapt the learning content as well as its sequence according to the student's profile [43]. As explained in the literature review by Chrysafiadi & Virvou [12], there are two classes of intelligent tutoring systems: Systems that make diagnosis with the student's knowledge, misconceptions, learning style or cognitive state, and systems that plan a personalized strategy using diagnosis for each learner individually. Student modelling is considered a subproblem of user modelling which is of central importance to ITS since otherwise each student is treated the same [38].

Primarily, we are interested in Bayesian networks because of their ability to model uncertainty [37] and, at the same time, to support a decision making process. The user modelling goals of a Bayesian network for knowledge modelling is mainly to have an adaptive estimation of the knowledge itself, since it may increase or decrease during the learning process [5]. Since scalar models and fuzzy logic approaches [15] have lower precision, structural models are built with the assumption that the knowledge is composed mainly by independent parts. On the other hand, bug/perturbation models [12] represent errors and misconceptions of the student. In this case, the Bayesian network is used to find the error that most probably caused the observable behavior (also called evidence) [37] which is called credit/blame assignment problem [39]. Bayesian networks can model the assumption that a wrongly answered question having two potential causes is most probably caused by the one that is more prevalent, according to the data provided so far. Sometimes, random slips or typos are included in the model and do not rely on assumptions as for example: A wrongly answered question does not necessarily mean that the student does not know a concept completely, or a correctly answered one wasn't a guess. The structure in both cases constitutes the qualitative model; its definition uses domain knowledge and (optionally) data. The parametrization is learned from the data during a training phase and constitutes the quantitative model.

The reason for the creation of the model is in some cases to assist the teachers of large classes that suffer from a high dropout rate [53]. A model recognizes the student's knowledge faster and more accurate [37] which is primarily beneficial when the class has a large number of students. In other cases that are summed up in [6], the goal is to provide a personalized optimal sequence of the learning material or even to sequence the curriculum according to the student's individual needs. And yet, further cases [48], [47] show that the learning application that is based on the model provides long-term learning effects as opposed to traditional methods. This was studied by a post-test that was made several weeks after the learning sessions.

The issue of defining the prior beliefs, which consist the starting parameterization, is often coupled with user clustering; demographics, longitudinal data [48], pre-tests [26], [13], defining the prior beliefs as well as the starting groupings [5] with respect to the learners. In other cases, the teacher sets the prior beliefs from his/her experience [38] or a uniform prior is used [13], [39]. Another common characteristic is the definition of hidden structural elements that rep-

resent unobservable entities, which must be estimated from the observed ones. The design of the structure must take correct assumptions into account, based on a solid theoretical background, otherwise the model will not work correctly [37].

In the work of Eva Millán et al. [34], the researchers draw a parallel between medical diagnosis systems and student's knowledge diagnosis [35]. Actually, this is an important comparison as the development of clinical reasoning and decision-making skills is very similar [3].

The student answers a set of questions that can only be answered correctly, when several concepts are known. In this case the knowledge of the concepts is the cause of the answer. The noise in the process, for instance when a student knows the concept but answers wrongly and vice versa a correct guess, is also modelled. The initialization of the model parameters is made by teachers-in-the-loop; afterwards the parameters are learned from the data. The model is used to efficiently determine those concepts the student knows less and the deductive proposal of the next question.

The "eTeacher" is a web-based education system [22], [43] that recognizes the learning style of a student according to the performance in exams as well as email, chat and messaging usage. The number of different learning strategies and their characteristics is the "domain knowledge" defining the structure of the Bayesian network. The initialization of the parameters uses in some cases uniform priors and in others priors defined by experts. After that initial phase, the parameters are continuously learned from the behaviour of the students. After identifying the learning style, a recommendation engine proposes different ways to learn the same material to each student according to his or her learning style.

"ANDES" is an ITS developed by Conati et al. [14], which mainly focuses on knowledge tracing but also on recognition of the learning plan of the user. The students solve Newtonian physics problems with different possible solution paths that define the Bayesian network's structure. Since each action may belong to different solution paths and the user does not provide its reasoning explicitly, the credit assignment problem is to find and quantify the most likely solution an action belongs to. This triggers personalized help instructions and hints in two cases: when a wrong answer is given or when the model predicts that the answer might be wrong. The parameters of the network change in an online manner while the student is solving the problems. Firstly, the evaluation was made by simulating students that have different knowledge profiles and measuring the accuracy of the predictions made by the model. In a second step, a post-test was carried out to compare real students having used "ANDES" to students who have not. Regression analysis was used to recognize the correlation between the use of the program and the learning gain [7].

Specifically for mathematical problems there are several approaches that specialize in dealing with decimals misconceptions. In the work of Stacey et. al. [48], [47] the misconceptions that define the structure of the model are provided by two main factors: the domain knowledge and data of a Decimal Conception Test

(DCT) that students had to go through. Wrongly answered questions provided by the students depend on their misconceptions. The researchers defined the distinct misconception by computing which of them has the highest probability according to the data. Although the model drives different question sequencing strategies, some of the misconceptions were not correctly recognized. Therefore, the researchers decided that the teacher and not the system should provide instructions.

Also, the research work of Goguadze concentrates on the modelling of decimals misconceptions [26], [25]. The “AdaptErrEx” project selected the most frequently occurring misconceptions and ordered them a taxonomy (higher and lower level misconceptions), which is reflected in the dependencies of the Bayesian network. As the previous application, a wrong answer may be caused by different misconceptions. The prior beliefs are defined by a pre-test; the researchers assert that sufficient training data diminish the role of the prior in the computation of the posterior. This prior defines the typical/average student and then each user’s parameters can be updated and individualized accordingly. One aspect that has not been considered in this model yet, is the difficulty of each question: easy questions will more likely be answered correctly than difficult ones, even if there is a high probability of misconception.

Several student modelling models track the progress of knowledge through time with Dynamic Bayesian Networks (DBN). The knowledge of the learner at each time point can be considered to be dependent on the knowledge and (optionally) the observed result of the interaction at the previous time point [36]. The project “LISTEN” [10] represents the hidden knowledge state of the student at each time point. The observable entities are the tutor interventions and the student’s performance which are used to infer the knowledge state. In the work of Käser et. al. [29] there is an overview and comparison of Bayesian Knowledge Tracing (BKT), which is a technique for student modelling using a Hidden Markov model (HMM) modelling and DBN for various learning topics, such as number representation, mathematical operations, physics, algebra and spelling. A HMM is a special case of a DBN, which, according to the researchers, cannot represent dependencies that would lead to hierarchies of skills; in these case DBNs create more adequate models.

All above described applications have a Bayesian network of the students model at its architecture core. There are a number of other components that either support the teacher or the student. One of them, for example, is the visualization of the model in the “VisMod” application [54], which is displayed in (among other things) color and arrow intensity instead of number-filled tables. This increases the readability of the model and enhances the tutor’s understanding. Gamification elements can also be found in “Maths Garden” [30], an application that lets users gain and loose points and coins depending on answering correctly or wrongly. A coaching component that provides feedback and hints to refresh the memory can be found in “ANDES” ’s architecture [7]. An overview about the design and architectural elements of intelligent tutoring systems that

have a Bayesian network as user model is provided in the work of Gamboa et. al. [21].

A detailed overview about intelligent techniques other than Bayesian networks, such as recommender systems for the computation of the learning path as well as clustering and classification for learner profiles that are used in e-learning systems, is provided in [33]. Specifically in [5], the demand for the most appropriate activity proposed - neither too easy nor too difficult - can only be fulfilled, if the used model is both accurate and adaptive.

1.3 Research Question

The main objective of this research work is to answer the research question, whether Bayesian networks can quantify the defined misconceptions of one-digit multiplication problems. In order to answer this question the “1x1 trainer” application is taken as the underlying data provider. The application focuses on the recognition of the current learning status. However learning aware applications maintain an adaptive learning model that represents the knowledge of the learner/user with regard to the learned topic. The application is expected to support individual learning needs and abilities as well as considering common characteristics in the learning process of different persons. The progress of the learning model itself will be used to transform the learning application into an adaptive one; that may change the content and sequence of assessments constantly to improve the learning process and to maximize the learning efforts.

The “1x1 trainer” application has a current overall report that is accessible to teachers. It contains information about the actual number as well as the proportion of correct and wrong answers of each posed question. It uses color encoding that helps distinguishing four sets of questions with similar proportion of correct and wrong answers. The implementation of the Bayesian model provides further insights to detailed cognitive information that enriches the information content of the current report. Furthermore, the new report can concentrate on individualized learning status, considering the causes of the wrong answers and can be updated in real time after each action of the student.

1.4 Outline

This research work proposes a Bayesian model for the learning competence of students using the “1x1 trainer” application. The first step is to specify the error types that are relevant for this research; their detailed description is made in section 2. Data analysis (specifically descriptive statistics) is used to guide the necessary assumptions about the modelled entities and their independences. Based on this information, the structure of the model and its parametrization is defined. The personalized model of each student and the method by which it adapts its parameters to new data is described in section 3. The usage of the model and the insights that are provided to the teachers in the form of an enhanced report are explained in section 4. Finally, a conclusion about future research and improvement possibilities is in section 5.

2 Error types of one-digit multiplication and descriptive statistics

2.1 Error types of one-digit multiplication problems

The bug library [12] of the proposed learning competence model contains six error types: operand, intrusion, consistency, off-by, add/sub/div, and pattern. Any false answer that does not belong to one of those six categories is assigned to the unclassified category. The description of the error types is explained in detail in [50]; a brief description follows here:

1. Operand error: It occurs, when the student mistakes at least one operand for one of its neighbours [8]. In the implementation only a neighbourhood of overall absolute distance of 2 from the correct operands was considered. One example is the answer 48 to the question 7×8 since the user may mistakenly multiply 6×8 . Research shows that this is the most frequently occurred error, but it occurs with a different proportion in each posed question [9].
2. Operand intrusion (abbreviated intrusion) error: It happens, when the decades digit and/or the unit digit of the result equals one of the two operands of the posed question, for example $7 \times 8 = 78$. It is argued by [8] that the two operands of the multiplication question are perceived as one number by the student (the first operand corresponding to the decades digit and the second to the unit digit).
3. Consistency: The student's answer has either the unit digit or the decade digit of the correct answer [45], [16]. For example, the answer 46 to the question 7×8 indicates that the unit digit is correct, but the decades digit is false.
4. Off-by- ± 1 , Off-by- ± 2 : It occurs, when the answer of the student deviates from the correct one by ± 1 or ± 2 , for example, when the answer of the question $5 * 8$ is one of the following: $\{38, 39, 41, 42\}$.
5. Add/Sub/Div: The student confuses the operation itself and performs for example an addition instead of a multiplication; in that case the answer to 7×8 is 15.
6. Pattern: The student mistakes the order of the digits of the result, for example, question 7×8 provides the answer 65 (the decades digit and the unit digit are permuted).
7. Unclassified: Any answer that can not be matched to one of the above error types.

All questions that have a correct answer with value smaller than 10 do not have consistency error. These are: $1 \times 1, 1 \times 2, 2 \times 1, 1 \times 3, 3 \times 1, 1 \times 4, 4 \times 1, 1 \times 5, 5 \times 1, 1 \times 6, 6 \times 1, 1 \times 7, 7 \times 1, 1 \times 8, 8 \times 1, 1 \times 9, 9 \times 1, 2 \times 2, 2 \times 3, 2 \times 4, 3 \times 2, 3 \times 3, 4 \times 2$.

One of the main reasons to use a probabilistic graphical model, is the fact that a specific false answer can be classified to multiple error types. The identification of the most probable error type causing a wrong answer is called credit

assignment. The table 1 shows the possible false answers for the question 7×8 . One can see that for example the answer 72 could occur because of an operand or an intrusion error.

Table 1. Answers for question 7×8 listed by error types

Error Type	Answers
operand	40, 42, 48, 49, 54, 63, 64, 72
intrusion	18, 28, 38, 48, 58, 68, 71, 72, 73, 74, 75, 76, 77, 78, 79, 88, 98
consistency	16, 26, 36, 46, 51, 52, 53, 54, 55, 57, 58, 59, 66, 76, 86, 96
off-by-± 1/off-by-± 2	54, 55, 57, 58
add/sub/div	1, 15
pattern	65
unclassified	4, 5, 6, 7, 8, 9, 10, 11, 12, 19, 20, 21, 22, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 37, 39, 41, 43, 44, 45, 47, 50, 60, 61, 62, 67, 69, 80, 81, 82, 83, 84, 85, 87, 89, 90, 91, 92, 93, 94, 95, 97, 99
correct	56

2.2 Data of the “1x1 trainer”

The data that were used for building the model were provided from the “1x1 trainer” application. The application is for both students and teachers. For this work it was also used for a preliminary categorization of learners. Users of this application are confronted with multiplication questions with both multiplicands being one-digit integer numbers. The possible questions range from 1×1 up to 10×9 (a total of 90 questions) and are posed in a pre-specified order. The application does not provide any means of help or hints to the students so far; the only feedback users get, is whether their answer is correct or not. It is expected that by repeated use of the application the students will learn and get better through exercise. But there is no individualisation that takes care of the personal needs of the learning style and knowledge level of the users. Furthermore, personal information such as age, gender, demographics, and educational level were not collected.

The data were cleaned in the preprocessing phase. The answers that did not lie in the interval $[0 - 100)$ were considered invalid and were removed. Overall there were 1179720 question-answer pairs with 1164786 valid. The number of unique users that gave at least one valid answer is 9058. The file covers eight columns providing the user ID, session ID, platform ID, date and time of the answer, as well as the reaction time of the student. Along with the posed question and the provided answer the ID of the result type as one of $\{R, WR, W, WWR, WW, WWWR, WWW, WWWW\}$, whereas W means “wrong” and R “right” is stored. The detailed description of the result type is shown in [50] and is basically a way to quantify the relative difficulty of each question by keeping the recent history of the user’s answering behaviour for each question. This information and the reaction time was not used in the model.

2.3 Data Analysis and Descriptive Statistics

To help designing the probabilistic graphical model, some analysis steps were necessary to be carried out with the data. The analysis and descriptive statistics provides insights about the overall similarities and differences between the students [23].

Firstly, not every user has answered the same number of questions. The vast majority of the users (98,6%) have ≤ 1000 valid answered questions. For the training of the model, the prior must have an equal amount of answers for each question. This does not take the sequence of posed questions into consideration.

Secondly, for each question the proportion of wrong answers was computed and depicted with a heatmap, whereas the x-axis is the first operand and the y-axis the second (see figure 1). As it turned out, the most difficult question is 6×8 with 26.8% of wrong answers given. It must be advised to remember that not all questions are posed the same number of times, because of the algorithm that chooses the question sequence. Therefore the belief about the relative difficulty of the questions has not an equal confidence for all the questions.

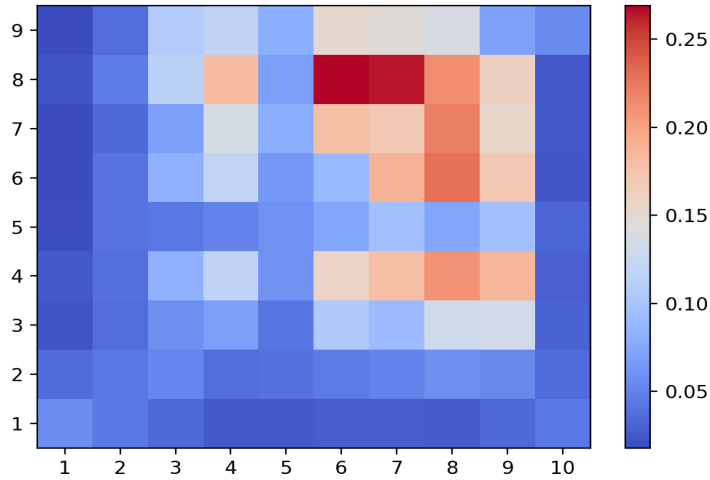


Fig. 1. Relative difficulty of the questions measured by the proportion of wrong answers. The x-axis is the first operand and the y-axis the second.

3 Probabilistic Graphical Model of Learning Competence

The use of a learning-aware data-driven application cannot assume that the user’s learning competence remains unchanged. Simple statistical descriptions are not practical in representing a continuous change and do not effectively capture the differences between the learning process of the users. Furthermore, the purpose is to choose intelligent actions (also called “actionable information” [1]) based on the data and this is not possible simply by one rigid and non-adaptive analysis of the data.

The choice of a probabilistic graphical model has several benefits. Firstly, it allows the representation of conditional dependencies (and independencies correspondingly) in the graphical representation of the model of the data. Those are assumed to be the same for all users and stay stable over the course of application usage. Secondly, its parameters (that can be thought of as a configuration or instance) are adaptive and change with each new data sample that is observed. They may be a temporary snapshot description that characterizes the learning competence but unlike the statistics there is an effective way to adapt those and not recompute them from scratch each time the model confronts new data. Thirdly, they’ve already been extensively used for decision problems [1], [31] which are the forefronts of reinforcement learning algorithms.

3.1 Introduction to Probabilistic Graphical Models

Probabilistic Graphical Models are representations of joint probability distributions over random variables that have probabilistic relationships expressed through a graph. The random variables involved can be discrete which have categorical values or continuous with real values. The set of possible values that a random variable can take - sometimes also referred as the possible outcomes of the experiment described by the random variable - is its domain. The random variables can be either visible or hidden. The visible ones have outcomes that can be directly observed and their values are contained in the dataset. The hidden variables are defined by human experts using the domain knowledge of the problem, but their outcomes are not directly accessible. They usually represent latent causes of visible random variables and can improve the accuracy and the interpretability of the model [32].

To specify the dependencies of the variables in general, one needs to specify their direction, type and intensity. This is made with the use of graphs which provide the terminology and theory for understanding and reasoning about Probabilistic Graphical Models. The nodes (also called vertices) of the graph represent the random variables and the edges their dependencies which can be directed or undirected. Undirected models - also called Markov networks - on the other hand represent symmetric probabilistic interactions where there is no dependency with direction, only factors that represent the degree of the strength of the connection. In case where the dependencies are directed, the graph must be a directed acyclic graph (DAG), otherwise circular reasoning would be possible. These two categories are used in different applications.

3.2 Model Structure

Domain knowledge about the already described error types that are encountered in one-digit multiplication, as described in section 2, was used to define the model. This is in accordance with the data-driven approach of model construction [32] where the structure of the model is specified by the designer and the parameters are learned from the data.

A question is either answered correctly or faulty. The student can make one of the following errors: Operand, intrusion, consistency, off-by- ± 1 and off-by- ± 2 , pattern, confusion with addition, subtraction, division or an unclassified error (meaning none of the above). Therefore a multinomial random variable called **Learning State_q** - individual for each question q was chosen to represent the proportion of each of these misconceptions of the user, when he or she is answering a one-digit multiplication question. The variable follows the categorical distribution; in this case the **Learning State_q** has eight possible outcomes and the domain of this random variable is $\text{Val}(\mathbf{Learning\ State}_q) = \{\text{operand, intrusion, consistency, pattern, confusion, unclassified, correct}\}$ (meaning that 1 is the operand error, 2 the intrusion error and so on). The **Learning State_q** of a specific user can be described for example 5% operand error, 4% consistency error and 91% correct answering (the rest possible outcomes have 0%). This parametrization must be learned from the data.

In the previous section it is shown that a specific faulty answer may be classified to more than one error types. Although in reality the model does not assume that more than one error type created a particular answer, the model cannot know a priori which error type was more prevalent and played the decisive role in choosing the wrong answer. The **Learning State_q** is hidden and the percent of each error type is expected to be learned by the provided answers. Thereby, a dominant error type (for a specific user) can be still discovered and weaken the belief that multiple error types played a role for a specific faulty answer. In section 4 the inference of the most probable error type (credit assignment problem) of a specific wrong answer will be made after the learning of the parameters is completed.

The proportion of correct and false answers is different for each question. Even though each question is not posed the same number of times and the belief about the possibility of correctly answering each question is different, this was also taken into account. That means that the probability of answering correctly is not. Therefore, there are 90 random variables called **Correctness_{1×1}** to **Correctness_{10×9}** (abbreviated by **Correctness_q**) that have each two possible outcomes. Therefore the Bernoulli distribution was chosen, which is equivalent to a categorical distribution with a domain of two values.

Each question has a distinct random variable, named accordingly as **Answers_{1×1}** to **Answers_{10×9}** (abbreviated as **Answers_q**), which is a child of the **Learning State_q** random variable. The arrows from the **Learning State_q** to its children reflect the dependency of the answer to a question from the misconception or correct understanding of the user.

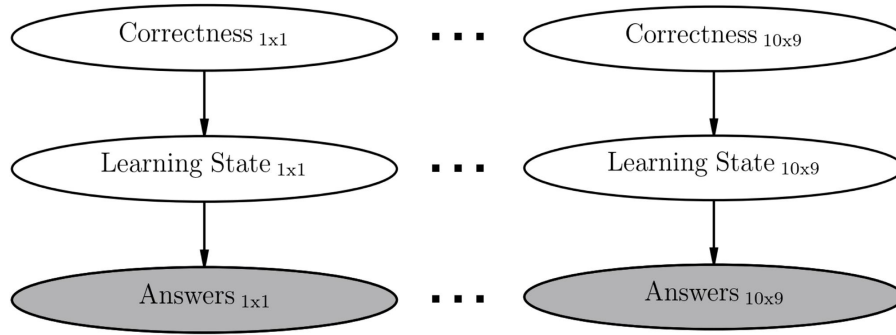


Fig. 2. The structure of all Probabilistic Graphical models for Learning Competence. The shaded $\mathbf{Answers}_q$ nodes are the ones that are observed, whereas the $\mathbf{Correctness}_q$, $\mathbf{Learning State}_q$ random variables remain unobserved.

The conditional independence property of each Learning Competence model is expressed by the following equation:

$$\mathbf{Answers}_q \perp \mathbf{Correctness}_q | \mathbf{Learning State}_q \quad (1)$$

The joint probability distribution for each question q has the following factorization:

$$\begin{aligned} P(\mathbf{Correctness}_q, \mathbf{Learning State}_q, \mathbf{Answers}_q) = \\ P(\mathbf{Correctness}_q) P(\mathbf{Learning State}_q | \mathbf{Correctness}_q) \\ P(\mathbf{Answers}_q | \mathbf{Learning State}_q) \end{aligned} \quad (2)$$

Each error type can only produce a specific subset of answers, so the others will have zero probability of occurring given this particular error type. Every row of the conditional probability tables of the $\mathbf{Answers}_q$ random variables has values that sum to one and the last row has only one entry with probability 1.0 at the column with the correct answer and 0.0 everywhere else. Figure 2 depicts the described structure of Learning Competence models.

The model needs to express the following procedure: First knowing if the question is answered correctly; this is provided by the $\mathbf{Correctness}_q$ random variable. If this is true then there are no more steps to follow. In the case where the answer is false, there must have been an error which belongs to the hidden $\mathbf{Learning State}_q$. One of the possible answers of this error, as seen and quantified by $\mathbf{Answers}_q$ will be the actual answer of the user. The possibility of guessing the answer is a valid one, but there is no way to get that kind of evidence in this application. The probability of continuously guessing the correct answer is very low and students that continuously provide random answers need to be discovered by the inconsistency of their model.

The model reflects our belief about the overall learning competence of the user. Its structure is considered to be the same for all users, but the conditional probability values (entries in the conditional probability tables) will differ for each individual user. Nevertheless the model can also reveal similarities between the users, meaning at this stage models that have similar parameter values.

3.3 Learning the Model’s Parameters

The answers of questions of the students which are already gathered comprise the data set denoted by \mathcal{D} . The goal of parameter learning is the estimation of the densities of all random variables in the model. The joint probability distribution $P_{\mathcal{M}}$ defined by the model \mathcal{M} with parameters Θ is expressed by equation 2. The parameter learning’s goal is to increase the likelihood of the data given the model: $P(\mathcal{D}|\mathcal{M})$ or equivalently the log-likelihood: $\log P(\mathcal{D}|\mathcal{M})$ with respect to the set of the parameters Θ of the model. The likelihood expresses the probability of the data given a particular model; a model that assigns a higher likelihood to the data \mathcal{D} approximates the true distribution (the one that has generated the data) better.

The algorithm that is used to estimate the parameters in cases where some of the variables are hidden is expectation-maximization (EM). Since the latent variables **Correctness_q** and **Learning State_q** are not observed, the direct maximization of the likelihood is not possible. The EM algorithm initializes all model’s parameters randomly and iteratively increases the likelihood by step-wise maximizing the expected value of the currently estimated parameters [2]. If the likelihood’s increase or the parameters’ change is not significant compared to the previous iteration, then the algorithm can be stopped. The procedure of updating the log-likelihood in this manner is shown to guarantee convergence to a stationary point, which can be a local minimum, local maximum or saddle point. Fortunately, by initializing the iterations from different starting parameters and injecting small changes to the parameters, the local minima and saddle points can be avoided [32].

The models are simple enough; therefore the EM-algorithm has a straightforward analytical solution. The available data were divided into a training and test set, with a dataset containing data from users that have answered all the questions at least one time (The number of users that have answered all questions exactly once is 2218). The models parameters are computed by the EM-algorithm on the training data and it iterates 4 times. After 4 iterations the likelihood of the training set increases, but the likelihood of the test set decreases which consists an indication of overfitting. The diagram in figure 3 describes the main computational blocks of this process.

Figure 4 depicts the learned parameters for the Learning Competence model of question 8×5 as an example.

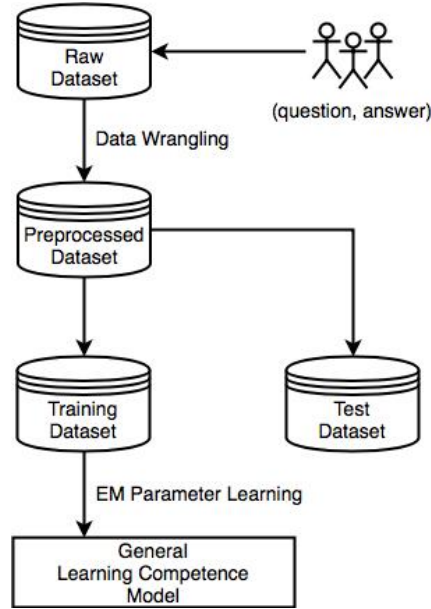


Fig. 3. Computational blocks diagram of data preprocessing, splitting into training and test set until the computation of the learned model.

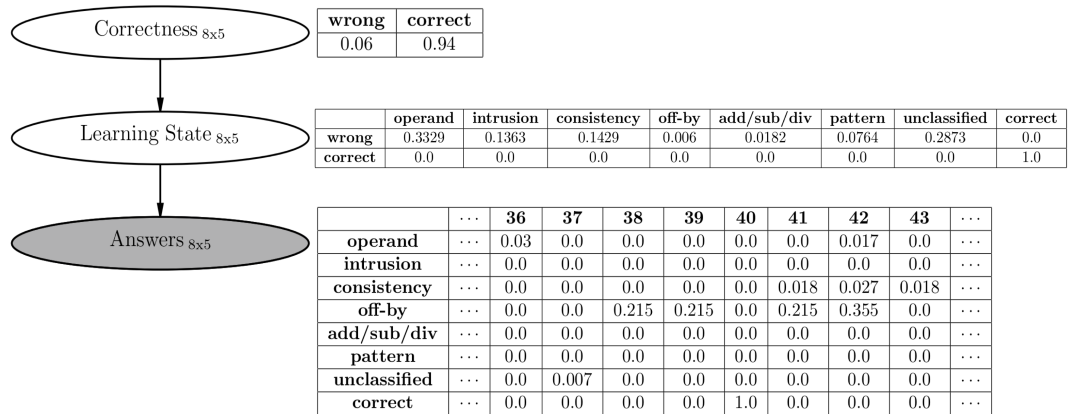


Fig. 4. Learned parameters of Learning Competence model for the question 8×5 .

4 Insights

After the model of a particular student is learned - by using the informed prior as starting point and as evidence the answers he or she has given so far. The better and more accurate the model captures the learning competence of the student, the better the performance of the predictions of the answers will be. In some probabilistic modelling frameworks such as Figaro⁷ the parameter learning part is made by the offline component and the probability queries by the online component.

There are three types of reasoning one can make with probabilistic graphical models: causal, evidential and explaining away. Causal reasoning (also called prediction) consists of statements that start with the knowledge of the causes as evidence and provide information about the effects. In our model this would be possible if the **Correctness_q** and **Learning State_q** were known: the computation of the answer to a posed question would be accurately determinable. The direction of causal reasoning in directed graphical models goes from parent to child variables (“downstream”) in general and is used to predict future events.

Evidential reasoning (also called explanation) on the other hand has the opposite direction and involves situations where effects lead to the specification of causes. This is the most important reasoning in our case because the answers of the students provide the information to do evidential reasoning and learn the hidden variables **Correctness_q** and **Learning State_q** which in turn can be used for causal reasoning to predict the future answers of each student. The difference in causal and evidential reasoning can be understood by considering the direction of time; evidential reasoning infers the past probability distribution from the current set of data whereas causal reasoning makes a prediction for the future given the data. The great benefit of graphical models over statistics is that the same model is used for both backward and forward reasoning (with respect to the perception of time).

Intercausal reasoning occurs when one random variable depends on two or more parents. In this case, the observation of the value of one parent influences the belief about the value of the other(s) (either strengthen or weaken). In this situation it is said that one reason explains away the other. The Learning Competence model’s structure does not contain such cases; further discussion about this reasoning type can be found in [28], [32], [2].

The upcoming sections proceed with an analytical implementation of probabilistic queries which is specific to the designed Learning Competence models. Personalized insights computed by the latent explanations of wrong answers of each student are made possible by exact and efficient inference as described in section 4.2.

4.1 Probability Queries

A conditional probability query $P(\mathbf{Y}|\mathbf{E} = e)$ - also called probabilistic inference - computes the posterior of the subset of random variables represented by \mathbf{Y} (tar-

⁷ <https://www.cra.com/work/case-studies/figaro>, Last accessed 18 June 2019

get of query) given observations e of the subset of evidence variables denoted by \mathbf{E} (of course there may be a subset of variables \mathbf{Z} in the model not belonging to either of these two subsets). By using the Bayes rule, the conditional probability is written as:

$$P(\mathbf{Y}|\mathbf{E} = e) = \frac{P(\mathbf{Y}, e)}{P(e)} \quad (3)$$

The MAP query, which is also called most probable explanation (MPE) [32], [41] is a query that maximizes the posterior of the joint distribution of a subset of random variables \mathbf{Y} :

$$\text{MAP}(\mathbf{Y}|\mathbf{E} = e) = \arg \max_y P(y, e) \quad (4)$$

In the case of MAP Query the whole set of random variables is $\mathcal{X} = \{\mathbf{Y}, \mathbf{E}\}$. In other words the MPE, after observing (clamping) a subset of variables, it computes the most likely values of the rest of them jointly.

A slightly different query is the marginal MAP which is written as follows:

$$\begin{aligned} \text{Marginal MAP}(\mathbf{Y}|\mathbf{E} = e) &= \arg \max_y P(y|e) = \\ &= \arg \max_{\mathbf{Y}} \sum_{\mathbf{Z}} P(\mathbf{Y}, \mathbf{Z}|\mathbf{E} = e) \end{aligned} \quad (5)$$

which directly follows from the fact that $\mathcal{X} = \{\mathbf{Y}, \mathbf{E}, \mathbf{Z}\}$.

The computation of the query result can be made with the variable elimination algorithm which is described in the following section 4.2; in this case the exact value of equation 3 is computed by dividing $P(y, e) = \sum_w P(y, e, w)$ and $P(e) = \sum_y P(y, e)$. Alternatively, the normalization of a vector containing all $P(y^k, e)$ (where y^k are all possible outcomes of the variables \mathbf{Y}) so that it has sum that equals to one, provides also the desired result. For more complex Bayesian networks approximate inference algorithms are applied since exact inference is NP-hard [31], but even those can be in the worst-case also NP-hard [32]. The Learning Competence models are simple and the variable elimination algorithm is fast enough.

4.2 Variable elimination in the Learning Competence Model

The probability query that is of high relevance for the teachers is the probability of error types regarded as causes of a specific wrong answer. The sum of products expression in equation 6 computes the distribution of the **Learning State_q** by means of the joint distribution $P(\mathbf{C}_q, \mathbf{LS}_q, \mathbf{A}_q)$:

$$P(\mathbf{LS}_q) = \sum_{\mathbf{C}_q, \mathbf{A}_q} P(\mathbf{C}_q, \mathbf{LS}_q, \mathbf{A}_q) = \sum_{\mathbf{C}_q} P(\mathbf{C}_q) P(\mathbf{LS}_q|\mathbf{C}_q) \sum_{\mathbf{A}_q} P(\mathbf{A}_q|\mathbf{LS}_q) \quad (6)$$

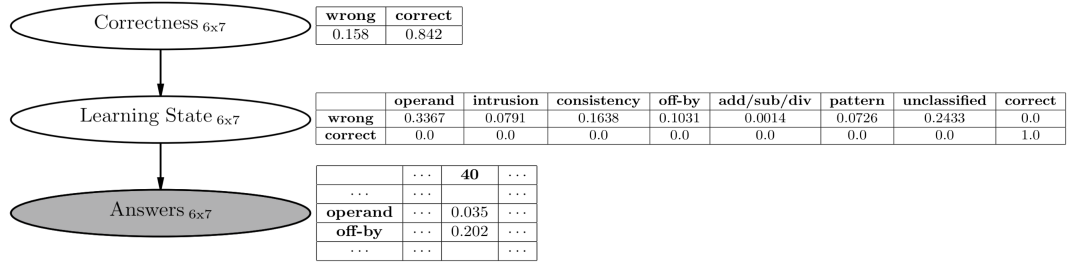


Fig. 5. Parameters of Learning Competence model of question 6×7 that are relevant to the computation of the MAP query when the answer is 40

The first step of the Variable Elimination algorithm, in case it is applied where an evidence exists, is to compute the unnormalized joint distribution $P(\mathbf{C}_{6 \times 7}, \mathbf{LS}_{6 \times 7}, \mathbf{A}_{6 \times 7} = 40)$. For example, the faulty answer 40 for the question 6×7 eliminates all cases for which the answer is not equal to 40; it can belong only to two potential error types: consistency and off-by. The remaining rows of the joint distribution - those containing the unnormalized proportion unequal to 0 are listed in table 2. The computations use the corresponding parameters of the Learning Competence Model of question 6×7 depicted in figure 5.

Table 2. Unnormalized joint distribution $P(\mathbf{C}_{6 \times 7}, \mathbf{LS}_{6 \times 7}, \mathbf{A}_{6 \times 7} = 40)$

$\mathbf{C}_{6 \times 7}$	$\mathbf{LS}_{6 \times 7}$	$\mathbf{A}_{6 \times 7}$	unnormalized proportions
wrong operand	40	40	$0.158 \cdot 0.336 \cdot 0.035 = 1.85 \cdot 10^{-3}$
wrong off-by	40	40	$0.158 \cdot 0.103 \cdot 0.202 = 3.28 \cdot 10^{-3}$

The sum of the unnormalized proportions, $1.85 \cdot 10^{-3} + 3.28 \cdot 10^{-3} = 5.14 \cdot 10^{-3}$ (which is the value of $P(\mathbf{A}_{6 \times 7} = 40)$), can be used to compute the normalized probabilities of the causes of answer 40 as depicted in table 3.

Table 3. Normalized joint distribution $P(\mathbf{C}_{6 \times 7}, \mathbf{LS}_{6 \times 7}, \mathbf{A}_{6 \times 7} = 40)$

$\mathbf{C}_{6 \times 7}$	$\mathbf{LS}_{6 \times 7}$	$\mathbf{A}_{6 \times 7}$	normalized probabilities
wrong operand	40	40	$1.85 \cdot 10^{-3} / 5.14 \cdot 10^{-3} = 0.36$
wrong off-by	40	40	$3.28 \cdot 10^{-3} / 5.14 \cdot 10^{-3} = 0.64$

The process eventually performs the following computation in equation 7 which is in accordance to equation 3.

$$P(\mathbf{C}_{6 \times 7}, \mathbf{LS}_{6 \times 7} | \mathbf{A}_{6 \times 7} = 40) = \frac{P(\mathbf{C}_{6 \times 7}, \mathbf{LS}_{6 \times 7}, \mathbf{A}_{6 \times 7} = 40)}{P(\mathbf{A}_{6 \times 7} = 40)} \quad (7)$$

The distributions **Correctness**_{6×7} and **Learning State**_{6×7} in the Learning Competence model is as follows:

Table 4. **Learning State**_{6×7} distribution of wrong answers in question 6 × 7 before the user answers 40

		wrong	correct				
		0.158	0.842				
operand	intrusion	consistency	off-by	add/sub/div	pattern	unclassified	
0.336	0.079	0.163	0.103	0.0014	0.072	0.243	

After observing 40, the Explanations probability distributions are as follows:

Table 5. Explanations distribution of wrong answers in question 6 × 7 after the user answers 40

wrong	operand	off-by
1.0	0.36	0.64

The result of the MAP Query (most probable explanation) is the joint assignment $\text{MAP}(\mathbf{Correctness}_{6 \times 7}, \mathbf{Learning State}_{6 \times 7}) = (\text{wrong}, \text{off-by})$. The result of the Marginal MAP query over the **Learning State**_{6×7} only, states that the most probable cause of the answer is the off-by error, as seen in figure 6.

This is an example of a case where the an error type has a higher probability than another one in the $P(\mathbf{Learning State}_q | \mathbf{Correctness}_q = \text{wrong})$ distribution, but the probability query could state that the most probable cause of a particular answer is the second one.

The results of the probability queries depend on the parameters of the model, which in turn are influenced by the prior distribution and the number of EM-iterations.

5 Future Work

The learned probabilistic model can be used in a generative scheme where the learning application will sample the model to predict the answer of the student. There are several algorithms that compute samples from the models with different characteristics [41], [32]. Particularly for this model where the dataset is highly unbalanced and the number of correctly answered questions is predominant, the metric to measure prediction performance should particularly take this fact into account. Although this feature does not provide an insight per

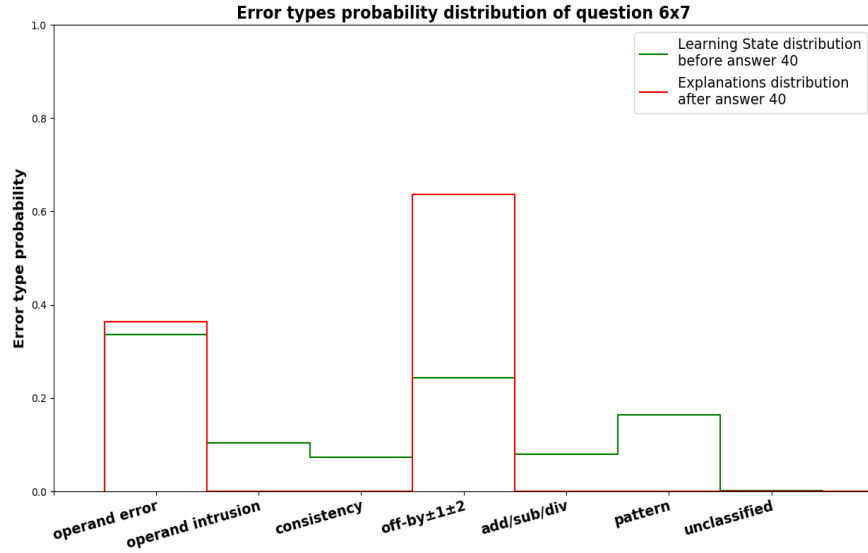


Fig. 6. Learning State_{6×7} and explanations distribution of question 6 × 7 before and after the user answers 40

se, it can be a starting point for other informative learning aspects. One aspect is explainable-AI which combines Bayesian learning approaches with classic logical approaches and ontologies, thereby making use of re-traceability and transparency [24].

Even though the proposed research extends the capabilities of the current learning application considerably, it cannot answer the fundamental question of which should be the most appropriate question to pose to the student. After the different learning competences are derived, the handling is delegated to the teacher, not the application itself, thereby applying the “human-in-the-loop” principle [27]. Further considerations apply to whether the models of learning competences that could be grouped together are the ones where the students will have the same learning path till they’ve learned to answer all questions correctly. The goal of this learning-aware application is not to group the learning competences by similarity of their parameters (expressing the current situation), but to find which ones will lead to similar optimal learning paths. This learning-aware application could benefit from an answer prediction component that accurately simulates students learning paths.

References

1. Barga, R., Fontama, V., Tok, W.H., Cabrera-Cordon, L.: Predictive analytics with Microsoft Azure machine learning. Springer (2015)

2. Bishop, C.: Pattern recognition and machine learning. Springer (2006)
3. Bloice, M., Simonic, K.M., Holzinger, A.: On the usage of health records for the teaching of decision-making to students of medicine. In: Huang, R., Kinshuk, Chen, N.S. (eds.) *The New Development of Technology Enhanced Learning*, pp. 185–201. Springer Berlin Heidelberg (2014). https://doi.org/10.1007/978-3-642-38291-8_11
4. Brusilovsky, P., Millan, E.: User models for adaptive hypermedia and adaptive educational systems. In: *The adaptive web*, pp. 3–53. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_1
5. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: *The adaptive web*, pp. 3–53. Springer (2007)
6. Brusilovsky, P., Peylo, C.: Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education (IJAIED)* **13**(2-4), 159–172 (2003)
7. Bunt, A., Conati, C.: Probabilistic student modelling to improve exploratory behaviour. *User Modeling and User-Adapted Interaction* **13**(3), 269–309 (2003)
8. Campbell, J.I.: Mechanisms of simple addition and multiplication: A modified network-interference theory and simulation. *Mathematical cognition* **1**(2), 121–164 (1995)
9. Campbell, J.I.: On the relation between skilled performance of simple division and multiplication. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **23**(5), 1140–1159 (1997)
10. Chang, K.m., Beck, J., Mostow, J., Corbett, A.: A bayes net toolkit for student modeling in intelligent tutoring systems. In: *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. pp. 104–113. Springer (2006)
11. Chater, N., Tenenbaum, J.B., Yuille, A.: Probabilistic models of cognition: Conceptual foundations. *Trends in cognitive sciences* **10**(7), 287–291 (2006)
12. Chrysaftadi, K., Virvou, M.: Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications* **40**(11), 4715–4729 (2013)
13. Conati, C., Gertner, A., Vanlehn, K.: Using bayesian networks to manage uncertainty in student modeling. *User modeling and user-adapted interaction* **12**(4), 371–417 (2002)
14. Conati, C., Gertner, A.S., VanLehn, K., Druzdzel, M.J.: On-line student modeling for coached problem solving using bayesian networks. In: *User Modeling UM 97*. pp. 231–242. Springer (1997). https://doi.org/10.1007/978-3-7091-2670-7_24
15. Danaparamita, M., Gaol, F.L.: Comparing student model accuracy with bayesian network and fuzzy logic in predicting student knowledge level. *International Journal of Multimedia and Ubiquitous Engineering* **9**(4), 109–120 (2014)
16. Domahs, F., Delazer, M., Nuerk, H.C.: What makes multiplication facts difficult: Problem size or neighborhood consistency? *Experimental Psychology* **53**(4), 275–282 (2006)
17. Ebner, M., Neuhold, B., Schön, M.: Learning analytics—wie datenanalyse helfen kann, das lernen gezielt zu verbessern. In: Wilbers, K., Hohenstein, A. (eds.) *Handbuch E-Learning-Expertenwissen aus Wissenschaft und Praxis-Strategie, Instrumente, Fallstudien*, pp. 1–20. Deutscher Wirtschaftsdienst (Wolters Kluwer Deutschland), 48, erg.-lfg edn. (2013)
18. Ebner, M., Schön, M.: Why learning analytics in primary education matters. *Bulletin of the Technical Committee on Learning Technology* **15**(2), 14–17 (2013)
19. Ebner, M., Schön, M., Taraghi, B., Steyre, M.: Teachers little helper: Multi-math-coach. *International Association for Development of the Information Society* (2013)

20. Ebner, M., Taraghi, B., Saranti, A., Schön, S.: Seven features of smart learning analytics—lessons learned from four years of research with learning analytics. *eLearning papers* **40**, 51–55 (2015)
21. Gamboa, H., Fred, A.: Designing intelligent tutoring systems: a bayesian approach. *Enterprise information systems* **3**, 452–458 (2002)
22. García, P., Amandi, A., Schiaffino, S., Campo, M.: Evaluating bayesian networks precision for detecting students learning styles. *Computers & Education* **49**(3), 794–808 (2007)
23. Godsey, B.: *Think Like a Data Scientist*. Manning Publications (2017)
24. Goebel, R., Chander, A., Holzinger, K., Lecue, F., Akata, Z., Stumpf, S., Kieseberg, P., Holzinger, A.: Explainable ai: the new 42? In: *Springer Lecture Notes in Computer Science LNCS 11015*. pp. 295–303. Springer (2018). https://doi.org/10.1007/978-3-319-99740-7_21
25. Gogvadze, G., Sosnovsky, S., Isotani, S., McLaren, B.M.: Towards a bayesian student model for detecting decimal misconceptions. In: *Proceedings of the 19th International Conference on Computers in Education*. pp. 34–41. Chiang Mai, Thailand (2011)
26. Gogvadze, G., Sosnovsky, S.A., Isotani, S., McLaren, B.M.: Evaluating a bayesian student model of decimal misconceptions. In: *Proceedings of the 4th International Conference on Educational Data Mining*. pp. 301–306. Citeseer (2011)
27. Holzinger, A., Plass, M., Holzinger, K., Crisan, G.C., Pintea, C.M., Palade, V.: Towards interactive machine learning (iml): Applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In: *Springer Lecture Notes in Computer Science LNCS 9817*, pp. 81–95. Springer (2016). <https://doi.org/10.1007/978-3-319-45507-56>
28. Karkera, K.R.: *Building probabilistic graphical models with Python*. Packt Publishing Ltd (2014)
29. Käser, T., Klingler, S., Schwing, A.G., Gross, M.: Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies* **10**(4), 450–462 (2017)
30. Klinkenberg, S., Straatemeier, M., van der Maas, H.L.: Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education* **57**(2), 1813–1824 (2011)
31. Kochenderfer, M.J.: *Decision making under uncertainty: theory and application*. MIT Press (2015)
32. Koller, D., Friedman, N.: *Probabilistic graphical models: principles and techniques*. MIT Press (2009)
33. Markowska-Kaczmar, U., Kwasnicka, H., Paradowski, M.: Intelligent techniques in personalization of learning in e-learning systems. In: *Computational Intelligence for Technology Enhanced Learning*, pp. 1–23. Springer (2010)
34. Millán, E., Agosta, J.M., Pérez de la Cruz, J.L.: Bayesian student modeling and the problem of parameter specification. *British Journal of Educational Technology* **32**(2), 171–181 (2001)
35. Millán, E., Loboda, T., Pérez-De-La-Cruz, J.L.: Bayesian networks for student model engineering. *Computers & Education* **55**(4), 1663–1683 (2010)
36. Millán, E., Pérez-De-La-Cruz, J.L.: A bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction* **12**(2-3), 281–330 (2002)
37. Millán, E., Trella, M., Pérez-de-la Cruz, J.L., Conejo, R.: Using bayesian networks in computerized adaptive tests. In: *Computers and Education in the 21st Century*, pp. 217–228. Springer (2000)

38. Nouh, Y., Karthikeyani, P., Nadarajan, R.: Intelligent tutoring system-bayesian student model. In: 1st International Conference on Digital Information Management. pp. 257–262. IEEE (2006)
39. Pardos, Z.A., Heffernan, N.T., Anderson, B., Heffernan, C.L.: Using fine-grained skill models to fit student performance with bayesian networks. Handbook of educational data mining pp. 417–426 (2010)
40. Pearl, J.: Embracing causality in default reasoning. *Artificial Intelligence* **35**(2), 259–271 (1988)
41. Pfeffer, A.: *Practical Probabilistic Programming*. Manning Publications (2016)
42. Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **40**(6), 601–618 (2010)
43. Schiaffino, S., Garcia, P., Amandi, A.: eteacher: Providing personalized assistance to e-learning students. *Computers & Education* **51**(4), 1744–1754 (2008)
44. Schön, M., Ebner, M., Kothmeier, G.: It’s just about learning the multiplication table. In: Buckingham Shum, S., Gasevic, D., Ferguson, R. (eds.) *Proceedings of the 2nd international conference on learning analytics and knowledge*. pp. 73–81. ACM, New York, NY, USA (2012)
45. Seidenberg, M.S., McClelland, J.L.: A distributed, developmental model of word recognition and naming. *Psychological review* **96**(4), 523–568 (1989)
46. Siemens, G., d Baker, R.S.: Learning analytics and educational data mining: towards communication and collaboration. In: *Proceedings of the 2nd international conference on learning analytics and knowledge*. pp. 252–254. ACM (2012)
47. Stacey, K., Flynn, J.: Evaluating an adaptive computer system for teaching about decimals: Two case studies. In: *AI-ED2003 Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education*. pp. 454–460. Cite-seer (2003)
48. Stacey, K., Sonenberg, E., Nicholson, A., Boneh, T., Steinle, V.: A teaching model exploiting cognitive conflict driven by a bayesian network. In: *International Conference on User Modeling*. pp. 352–362. Springer (2003)
49. Taraghi, B., Ebner, M., Saranti, A., Schön, M.: On using markov chain to evidence the learning structures and difficulty levels of one digit multiplication. In: *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*. pp. 68–72. ACM (2014)
50. Taraghi, B., Frey, M., Saranti, A., Ebner, M., Müller, V., Großmann, A.: Determining the causing factors of errors for multiplication problems. In: *European Summit on Immersive Education*. pp. 27–38. Springer (2014)
51. Taraghi, B., Saranti, A., Ebner, M., Mueller, V., Grossmann, A.: Towards a learning-aware application guided by hierarchical classification of learner profiles. *J. UCS* **21**(1), 93–109 (2015)
52. Taraghi, B., Saranti, A., Ebner, M., Schön, M.: Markov chain and classification of difficulty levels enhances the learning path in one digit multiplication. In: *International Conference on Learning and Collaboration Technologies*. pp. 322–333. Springer (2014)
53. Xenos, M.: Prediction and assessment of student behaviour in open and distance education in computers using bayesian networks. *Computers & Education* **43**(4), 345–359 (2004)
54. Zapata-Rivera, J.D., Greer, J.E.: Interacting with inspectable bayesian student models. *International Journal of Artificial Intelligence in Education* **14**(2), 127–163 (2004)