Muñoz-Merino, P. J., Ganzález Novillo, R. y Delgado Kloos, C. (2018). Assessment of skills and adaptive learning for parametric exercises combining knowledge spaces and item response theory. *Applied Soft Computing*, 68, pp. 110-124.

DOI: https://doi.org/10.1016/j.asoc.2018.03.045

Assessment of Skills and Adaptive Learning for Parametric Exercises Combining Knowledge Spaces and Item Response Theory

Pedro J. Muñoz-Merino[a,b], Ruth González Novillo[a], Carlos Delgado Kloos[a]

[a]Universidad Carlos III de Madrid, Av. Universidad 30, 28911 Leganés (Madrid), Spain

[b]UC3M-BS Institute of Financial Big Data, Spain

* Corresponding author info: Pedro J. Muñoz-Merino (phone: (+34) 916245972, email: pedmume@it.uc3m.es)

ABSTRACT

Many computer systems implement different methods for the estimation of students' skills and adapt the generated exercises depending on such skills. Knowledge Spaces (KS) is a method for curriculum sequencing but fine-grained decisions for selecting next exercises among the candidates are not taken into account, which can be obtained with the application of techniques such as Item Response Theory (IRT). The combination of KS and IRT can bring advantages since the semantics of both models are included but some issues such as the required local independence of IRT should be considered. In addition, an open issue is how to handle with parametric exercises for skill modelling, i.e. exercises which are not static content but that can change from instance to instance depending on some parameters and a student can try to solve them again with different parameters after correct resolution. The correct inclusion of several instances of the parametric exercises on the adaptive decisions is important since the adaptation process can improve. This work describes two new algorithms for skill modelling and for adaptation of exercises that integrate IRTand KS to have a more powerful approach with more knowledge in the models and at the same time provides a solution for taking into account parametric exercises where a student should solve an exercise correctly several times to get proficiency. We have evaluated the different skill modelling algorithms using real data of students from their interactions in an Intelligent Tutoring System, and the correspondent adaptation algorithms using a simulator. Results show that the accuracy of the prediction is good with values of RMSE under 0.35. Both proposed algorithms got similar results on the accuracy of the prediction but one of them is better regarding performance. Changes of the buffer size for the MLE in IRT did not have a significant effect on the accuracy and on the performance. There is a trade-off for selecting one of the two proposed algorithms: while the first algorithm has better performance time for the calculation of the ability (because there is no need of calculation of local abilities), the second algorithm has better performance time for the selection of the next exercise and better accuracy and depending on the scenario one or another should be selected.

Keywords

skill modelling, Item Response Theory, Knowledge Spaces, adaptive learning

1. INTRODUCTION

In a learning computer based system, one of the most important indicators is the students' level on the different skills. Teachers and systems should track the students' evolution on the cognitive level. First Intelligent Tutoring Systems were focused on the cognitive level [1]. The development of methods to assess the level of the students' skills is of key importance and computer systems should be able to calculate this in a precise manner. The most used resources to track it are exercises or problems. A precise assessment of skills usually involves taking into account the difficulty of exercises or the relationship among them.

The most important proposals for skill modeling and assessment of skills that exist can be divided into three main groups: Item Response Theory (IRT), Knowledge Spaces (KS) and Bayesian Networks (BN) [2]. Different variants of each of them have been proposed in the literature.

The systems that implement KS have a structure of relationships among exercises that allow the system to select which exercises the students can solve next, depending on the learning path that the student followed. This is an approach for curriculum sequence assessment that focuses on adaptation decisions as a whole but without fine-grained decisions at the local level. Therefore, in a specific moment, several exercises can be selected but among these possible candidate exercises, there is not a way to estimate which are easier or more difficult for a specific student or even if the system can jump several nodes over the structure to select an exercise. In this context, it would be important to know which are the best exercises among the possible learning paths defined by KS. The introduction of approaches such as IRT in combination with KS can achieve it. If thi additional level of decision is introduced then more powerful adaptive learning is enabled.

Another open issue for skill modelling is the treatment of parametric exercises. Several Intelligent Tutoring Systems use parametric exercises [3]. A parametric exercise is a problem which contents are not static but dynamic. The exercise can change from time to time as there are some parameters that can take different values. For example, let be

X+Y=Z a parametric exercise, being X, Y and Z the parameters. A parametric exercise can have different instances, i.e. the student can be presented with the same exercise but changing the values of X, Y and Z in a random way. Parametric exercises are different from the static traditional ones, because mastering a traditional exercise means that this exercise will be done correctly in the future (if we consider that the student will not forget it). When there are parametric exercises, if a student masters a specific instance of that exercise, this does not mean that the student will solve correctly another instance of that exercise later (even if the student does not forget it), but the likelihood of making another instance correctly is greater. This aspect has been noticed in Muñoz-Merino et al. [4]. For adaptation purposes, different instances of the same parametric exercise can be selected in different times, but this does not make sense for a traditional exercise. Students can learn new things with new instances of the exercise, but there might be a limit of repetitions for which we can consider the student will master any further instances of that exercise. Current skill modelling approaches do not take into account these particularities of parametric exercises when estimating students' skills. It is important to include these features because repetitions of parametric exercises by the same student should be considered when selecting the next items in adaptation systems, extending the possibilities of exercises to select and thus improving the adaptation process.

A specific typical case of KS is POKs [5, 6] in which every exercise can have several parents with an AND relationship, meaning that such exercise can only be solved correctly if all the parent exercises can be solved correctly. In this paper, we consider POKs networks. This work tries to overcome the commented current limitations of the last two paragraphs: increasing the power of fine-grained decisions in KS by the introduction of IRT, and increasing the number of possible exercises to select considering repetitions of parametric exercises and integrating the different repetitions with the proper parameters. We want to enable skill modelling and adaptation algorithms. Although the issue of estimating the probability of solving correctly the different candidate exercises in a POKs in a fined-grained is independent from the issue of parametric exercises, we have to point out that the issue of parametric exercises is related to POKs since a student should solve correctly a number of previous repetitions in order to solve a new repetition of the parametric exercise, which can be seen as a prerequisite relationship among the different instances of a parametric exercise.

Among the existing skill modelling techniques, IRT [7] and BN [8] approaches do not make use of direct relationships among exercises such as KS [9]. But it would be interesting to include approaches such as IRT or BN for the inclusion of a latent trait in KS that would enable fine grained decisions in KS. The work done by [10] proposes a new model to combine IRT and probabilistic networks. The case of KS can be seen as a specific one of that proposal. However, the specific conditions of POKs make that specific rules should be applied, which is analyzed and developed in this work. In addition that work [10] did not consider the inclusion of parametric exercises and the proposed evaluation had a different purpose.

Finally, the concept of adaptive systems in education for providing personalized contents or links have been presented and analyzed in different works [11, 12]. One of the most typical indicators used for adaptation is based on the students' skills. Computer Adaptive Testing (CAT) systems adapt a specific educational resource, which is the questions in a test [13]. There are several examples of these types of systems such as the presented in Muñoz-Merino et al. [14] or in Lin, Gong, & Zhang [15]. Lin, Gong, & Zhang [15] pointed out that the adaptation of items depends on the type of skill modelling used and they presented an adaptation algorithm for cognitive diagnosis models (CDMs) which is different from those based on IRT. Therefore, skill modelling is usually related to adaptation since the assessment of students' skills is used for adaptation purposes.

In this work, we propose new skill modelling algorithms that combine IRT and KS, including a solution that considers parametric exercises at the same time, in order to be able to make proper decisions about the next items to select. In addition, we propose new adaptive learning algorithms that use the previous proposed skill modelling algorithms. We validate and compare these skill modelling algorithms in terms of accuracy with real data of students, varying also the number of considered last interactions with the exercises for estimating the ability. Finally, we validate and compare these adaptive learning algorithms with fictitious students in terms of execution time and successful finishing of students.

2. RELATED WORK

2.1. ITEM RESPONSE THEORY

IRT is a solid theory that has been applied for years for a more precise scoring of tests, where an item of a test is considered different from others (e.g. different level of difficulty). IRT was first applied in Computer Adaptive Testing (CAT) [16, 17, 18] to select the items of a test in such a way that with the minimum number of items for a student, the most information about his/her scoring can be obtained.

Recently, IRT has also been applied in Intelligent Tutoring Systems [19] and web based learning systems [20, 21, 22] as a solution for adapting the next questions to show to the students.

IRT models the ability of a student in a specific skill as a latent trait. Let be θ the latent trait that models the ability of a learner. This variable can take values from -∞ to +∞, from less ability to more ability. In addition, each exercise or item is associated with a different Item Response Function (IRF) that gives the probability of solving correctly this item depending on the ability of the student, which is θ. The IRF has usually three parameters a_i , b_i and c_i , for an item *i* and its equation is the following:

$$p_i(\theta) = c_i + \frac{1 - c_i}{1 + e^{-a_i(\theta - b_i)}} \qquad (1)$$

This equation corresponds to the function of figure 1:

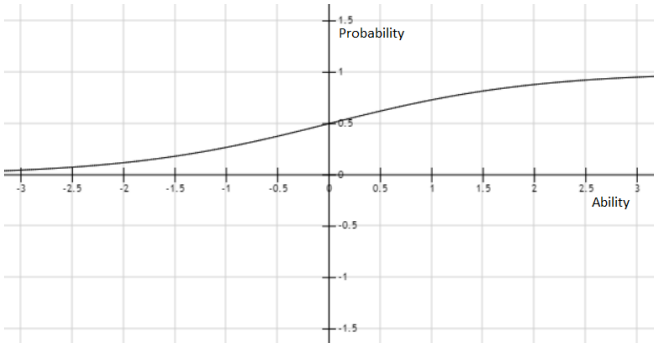

Fig 1 Graphical representation of an Item Response Function with three parameters.

The three parameters of the IRF have the following meaning:

- a_i : slope. It is the slope of the curve, which gives an idea of how an item can differenciate between students that have the skill and those who do not have it.
- b_i : difficulty. It is the difficulty of the item for a student with $\theta = 0$
- c_i : guessing. It is the probability that a student can guess the item by chance. It corresponds to the probability of answering correctly for a student with an ability -∞

The calculation of the ability of a student, given his/her responses to a collection of items is usually done using the Maximum Likelihood Estimation (MLE) [23]. Let assume that a student answered to a set of *k* items, being *i* a variable, denoting the item ranging from 1 to *k*. Let be $\overrightarrow{u_j} = (u_{1j}, u_{2j}, \ldots u_{kj})$ a vector denoting if the student *j* answered correctly or incorrectly to each one of the *k* exercises. It is a vector composed as values by 0s (incorrect responses) and 1s (correct responses). Let be P_{ij} the value of equation (1) for a student *j* interacting with exercise *i* at time *t* with ability θ_j^t . Let be $Q_{ij} = 1 - P_{ij}$, then the new ability θ_j^{t+1} of the student is calculated applying MLE as follows in equation (2):

$$\theta_j^{t+1} = \prod_{i=1}^{k} P_{ij}^{u_{ij}} Q_{ij}^{1-u_{ij}} \qquad (2)$$

As u_{ij} is either 0 or 1, then either Q remains or P, depending on the student answer to the exercise (incorrect or correct). By their definitions, P_{ij} and Q_{ij} only depend on the previous student skill level θ_j^t and the parameters associated to exercise *i*, i.e. a_i , b_i and c_i

It is important to note that IRT relies on the local independence of items as one of its assumptions. This means that IRT assumes that all the exercises involved in the test must be independent given the same value of the latent trait variable. Therefore, when there is a relation between two items for a given ability, they are said to be locally dependent and its violation affects the precise calculation of the ability [24].

2.2. BAYESIAN NETWORKS

BN can handle different skills of a student in the same model as a difference with traditional IRT, although there are some solutions for multi-dimensional IRT (e.g. [25]) but they are more complex and have not been used extensively in e-learning systems.

An introduction to BN and its connection with e-learning systems can be found in Millán, Loboda, & Pérez-de-la-Cruz [8]. BN are directed acyclic graphs composed by nodes which represent random variables (either quantitative or categorical). These nodes of the graph can have one type of relationship between them with a direction. Each relationship is between two nodes and establishes a source node and a destination one based on the direction. In most cases in the e-learning context, nodes represent categorical variables. In this context, each node has associated a conditional probability table (CPT) that can be composed of prior probabilities for each of the value possibilities (for root nodes) or conditional probabilities in other cases, denoting the probability of having some values for this variable/node depending on the different value combinations of the parent nodes.

Figure 2 shows an example of BN for estimating students' skills (hidden nodes) based on the students responses to different items (observable nodes). When using BN in e-learning systems, an event is usually an action done by the learner and it can be for instance answering an item in a correct or incorrect manner. This would correspond to the value of R1, R2, R3 and R4. Each item can be related to several skills on the network and a skill can be related with different items. There might also be relationships among skills, e.g. forming a hierarchy. The events of the students' responses update the network and the probability of each node, which allows estimating the learner's skills.

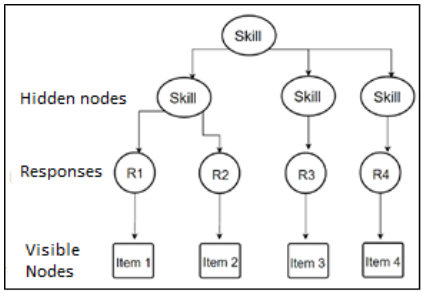

Fig 2 Example of a graphical representation of a Bayesian network for skill modelling.

The main tool for the Bayesian inference is the Bayes Theorem. For the Bayesian inference it is necessary to make one assumption: all nodes must be locally independent. An item is considered to be locally independent when its values are independent from all the values of the variables of the BN (skills or responses) given the value of all its parents nodes [26]. Several works have done proposals based on BN such as [27].

The aim is to obtain the probability of the nodes by means of the CPTs, the Bayesian inference and the information obtained in the events (correct and incorrect answers). This process can be costly and it is sometimes necessary to define some nodes or gates that simplify the network by reducing the number of parameters required to specify a conditional probability distribution [28, 29, 30].

Some Intelligent Tutoring Systems such as ANDES [31] for Newtonian physics and ASSISTment [32] for mathematical concepts use BNs to adapt the contents. The use of fine-grained models of skills in Bayesian Networks can increase the accuracy of the prediction of student state scores [32] so a careful design of the skills and the relationships with exercises is important.

Another approach of skill modelling that uses BNs is Bayesian Knowledge Tracing (BKT) [33]. BKT tries to estimate a student skill based on the past student actions done in a sequential way in which the temporal moment when the actions took place matter. These actions are all related to a single skill. One of the differences of BKT with respect to the previous commented BN approaches for skill modelling is that the time sequence matters in BKT for estimating the new skill. The probability of mastering a skill depends on a set of parameters (guessing, slip, probability the skill is already mastered, and probability the skill will be learned at an opportunity) and the previous state. BKT approach has been applied extensively (e.g. [34]).

Some modifications and extensions of BKT have been proposed to include student features in the model [35] [36] or item features such as the item difficulty [37]. This way, as well as IRT can include student features and item features, BKT can also include them although they are different models. Moreover, other works extend BKT including the relationships among skills [38]. These relationships are also present in Bayesian graphs as the commented in figure 2.

Other alternatives to BKT that also use the concept of temporal sequences to estimate student skills are Additive Factor Models [39] and Performance Factor Analysis [40] which demonstrated to perform better than BKT in specific conditions.

So far, all the presented approaches in this section do not make use of IRT which is a different model. This is a difference with respect to our work that uses IRT. Recent approaches have combined BKT with IRT to include user and item features in the model such as LFKT [41] and FAST [42].

All of these BN methods are more focused on fine-grained decisions on exercises, while methods like the KS are more focused on curriculum sequencing. A limitation of these previous works is that they do not consider the prerequisite relationships among exercises but if we consider this additional information we can make more precise decisions, taking this additional information into account.

Another limitation of these works based on BN is that they do not consider how to tackle with parametric exercises. A parametric exercise is something different to an exercise that is not related since we should be sure that the student can solve previous instance of the exercise in order to be able to solve a specific N repetition. With respect to this regard, it can be seen as a prerequisite relationship in a similar concept to the theory of Knowledge Spaces.

The main differences of all of these BNs approaches presented in this section regarding skill modelling with respect to our work are two:

- None of these references are considering the modeling of the theory of Knowledge Spaces (which will be summarized in next section), that we use in our work. In our work, we consider the direct relationship of the considered exercises as prerequisite relationships. This consideration imposes additional constraints in the model. This type of relationship is not taken into account into the other models of this section.
- None of these BNs works incorporates the concept of parametric exercises and a description of how parametric exercises can be included in the model. For example, there are not details about how the difficulty of a parametric exercise can change for a specific student when the exercise has been repeated N times by the same student with a previous history of interactions, taking into account

which are the interactions with the same parametric exercise and giving a different treatment with respect to the interactions with other parametric exercises. We take into account that the repetitions of an exercise are related to previous students interactions with that same parametric exercise, taking into account that the instances are related, which is a different relationship with other instances from other parametric exercises. This is not considered in the cited works of this section.

2.3. KNOWLEDGE SPACES

The Knowledge Space Theory (KS) allows representing the knowledge of a learner in a particular area as a graph [9]. As a difference with respect to IRT or BN approaches, KS models do not use hidden states. In this theory the knowledge is structured with logical and pedagogical relations and dependencies between exercises. A particular knowledge state can be defined as a set of exercises that the learner is capable of solving. Due to the dependencies between problems the knowledge states are finite which means that not all the possible states are possible. The collection of all the proper knowledge states is called knowledge structure. Each student can explore different knowledge states but not all the possible states. The set of knowledge states on which a student passes before arriving to the full state or full knowledge is called learning path. There are different variants to KS such as the addition of competencies in the process [43].

Different learning paths allow personalization so that certain paths are more suitable for some students. ITSs such as ALEKS [9] and RATH [44] use KS to provide content adaptation.The work by Maomi Ueno (2002) [10] proposes a skill modelling based on a probabilistic network and a latent trait (similar to IRT). Our work is also based on proposing a skill modelling combining a network with the IRT with a latent trait. But the main differences of our work with that work are the following:

1) The work by Maomi Ueno (2002) [10] proposes a model in which there is a probabilistic network and a latent trait such as the IRT. In our case, we propose a model in which we combine KS and IRT with prerequisite relationships among items. Our case with KS could be seen as a specific case of the work proposed by [10] in which their probabilistic network has some specific parameters and conditions. In this sense, the work by [10] is more general. But they do not analyze

the specific conditions of having a KS. In our work, we exploit, analyze and develop the model for this specific case, adding the specific conditions for KS.

2) Our work gives a solution for parametric exercises. The work by Maomi Ueno (2002) [10] does not integrate the concept of parametric exercises and do not give a solution for it.

3) The purpose of the evaluation in the work by Maomi Ueno (2002) [10] is different to our case. In our case, we compare two different skill modelling algorithms in terms of RMSE, analyze the change on the buffer size, create a simulator for making validations mainly on terms of execution time and percentage of students who finished with the adaptation proposals. These types of evaluations are not present in the work by Maomi Ueno (2002). [10].

3. DESCRIPTION OF THE PROPOSED SKILL MODELLING ALGORITHMS

This section describes two new proposed algorithms to assess the student skill making use of KS and IRT as well as to assess the probability of making any exercises correctly. The proposal forms a specific model where the Bayes Theorem is used but our approach does not form a typical BN applied in BN skill modeling approaches. In addition, the solutions are able to handle with parametric exercises.

3.1. ALGORITHM 1 PROPOSAL FOR SKILL MODELLING

Figure 3 shows a general overview of the first proposed new algorithm for skill modelling. The different steps of Figure 3 and their rationale are explained in each subsection.

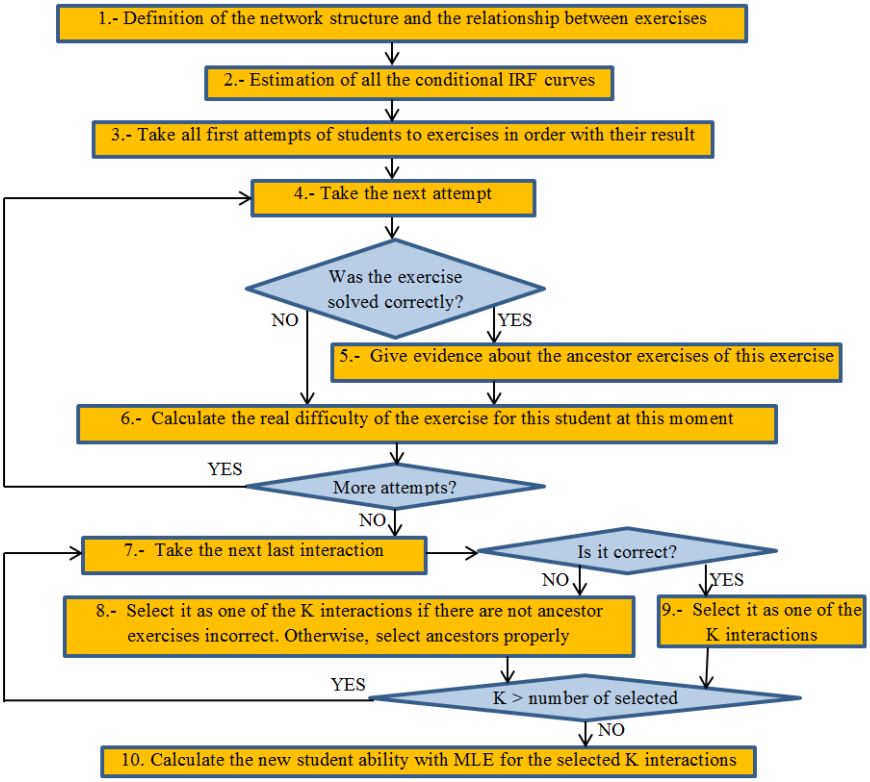

Fig 3 General overview of the first algorithm for skill modelling

Step 1: Definition of the network structure and the relationship between exercises. We propose to follow a network structure that is a particular case of the KS theory. The proposed network structure can be represented as a graph in which the different types of exercises are represented by circles and there can be relationships among them with arrows with a source and a destination (see figure 4 for a specific example). We denote e_i as a type of exercise *i* of this network.

A relationship between two nodes or types of exercises is represented by an arrow, which implies a prerequisite relationship. We consider that an arrow between two types of exercises implies that:

- For any instance of the destination exercise, there is an instance of the source exercise (parent node) such that if the instance of the source exercise cannot be solved correctly, then the instance of the destination exercise cannot be solved correctly. Therefore, extending this rule and applying it to all the ascendants recursivelly, for any instance of the destination exercise, there is an instance of all the ascendants exercises such that the student must solve all the ascendants

correctly in order to be able to solve correctly the instance of the destination exercise.

- If a student solves correctly an instance of an exercise, then this also gives evidence that the student would solve correctly an instance of all ascendants exercises.

These two assumptions are according to the fact that some prerequisite exercises are required to be able to solve the destination exercise or because there is some evidence that students who know some type of problem can solve other problems and this is done in a particular order. This reasoning is the same as in the KS theory, but the component of parametric exercises is added in this case.

The knowledge structure provides some semantic information, i.e. the prerequisite relationships among the different exercises. The addition of semantic information such as this one, might make possible to calculate more precisely the skill rather than not using this semantic information and applying only e.g. IRT which does not take into account semantic information regarding the relationship among different contents.

Moreover, an additional assumption of the model is that the structure is a Direct Acyclic Graph (DAG). Therefore, the knowledge structure should form a poly tree, where there cannot be cycles among the nodes. For example, a node A cannot be a prerequisite of node B and at the same time being B a prerequisite of A.

Therefore, a node or exercise can have multiple prerequisites, i.e. input arrows from other exercises. As a first step, content designers should select the different types of exercises and establishes for each exercise what are the prerequisite exercises and forming a graph like such as the one in figure 4.

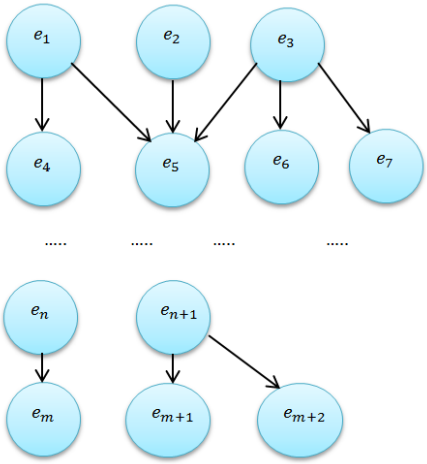

Fig 4 Example of a knowledge structure including relationship between exercises

Step 2: Estimation of all the conditional IRF curves. Our approach uses IRT (among other techniques) to estimate the student ability. For simplicity, let's consider first that exercises are not parametric. Then, each exercise e_i is going to have one associated IRF. In fact, our presented method would also be valid and novel (because it combines IRT and KS) even without parametric exercises, but the presence of parametric exercises add and additional novelty. As we have dependence relationships among the different exercises or items as explained in previous step 1 according to the knowledge structure (prerequisites relationships), then the assumptions of local independence of items required for the IRT are not fulfilled in case we use the IRF curves of traditional IRT. For example, if we have k interactions with exercises for estimating the ability of a student and it is included the interaction with an exercise e_i and some ascendants of it, then the calculation of the new ability would be flawed since the correct resolution of the parents of e_i would condition the resolution of e_i and this cannot only be explained by the latent trait.

In order to solve this issue, for the IRF of each item or exercise, we should consider the part of knowledge of the exercise which is new to the previous dependent exercises, i.e. we should consider an IRT curve for the exercise considering that the concepts of the previous ascendant exercises have already been mastered for that student. This way, we introduce the concept of conditional IRF curves. Conditional IRF curves are calculated to address the issue of the local independence assumption and allow us to compose the proper curves to be according to the independence of items assumption. This does not

mean that the algorithm for selecting the ability will always select for an item the conditional IRF curve for the estimation of the new ability because this would depend on the students' previous interactions, but this concept will enable the algorithm to compose the correct curve based on conditional IRF curves, which would not be possible without this concept.

For example, an exercise might have an IRF curve in the traditional way with a strong difficulty, but the conditional IRF curve with our approach might be with a weak difficulty. This would be possible by the fact that an exercise or item can be very easy to solve correctly if the previous exercises which are prerequisites of it are mastered by the student. If we have evidence that the student has already mastered the previous exercises, then we can use the conditional IRF for the estimation of the new ability. In case the student has not mastered the previous concepts, then in case there are not attempts on the parent exercises, we can use the traditional IRF curve with the strong difficulty.

Let now consider that each exercise is a parametric one, so several instances of this exercise can be presented to the student. A student will be presented with the same instance of the exercise (with the same values of the parameters) until the student solves it correctly. Let denote e_{ir} as the exercise i in the repetition r, which means the exercise i when the exercise has been previously correctly solved r times before by a student. If a student does not solve correctly an instance of an exercise, then the exactly same instance of the exercise is going to be presented to the student until the student solves it correctly.

A parametric exercise instance e_{ir+1} is dependent of another new instance e_{ir} of the same parametric exercise since if a student solves correctly an instance of a parametric exercise, then it is more probable that the student can solve correctly another instance of the same parametric exercise given a specific latent trait or ability. But on the other hand, it is a fact that solving correctly an instance of a parametric exercise does not imply that the student will solve correctly another new parametric exercise, since the own changes of the parametric exercise will add something new to the exercise. The more times a parametric exercise is solved, the most probable is that a student solves it correctly.

Taking into account all of this, we assume the following in our model to handle with parametric exercises:

- If a student does not solve correctly an instance of a parametric exercise, then he/she does not get any new knowledge for this parametric exercise.
- When a student solves correctly an instance of a parametric exercise, then the student acquire some knowledge regarding this parametric exercise but not all the knowledge of the parametric exercise.
- There are more than one IRF curve associated to a parametric exercise. The IRF curves represent the new knowledge for this parametric exercise assumed that the students have solved correctly r instances of this parametric exercise before. This way, there is also the concept of conditional IRF curves for parametric exercises, which was a similar concept as introduced before.

Figure 5 shows an example of IRF curves for an exercise i for repetitions *r* and *r+1*. It is clear that e_{ir} should be more difficult or equal than e_{ir+1} because it makes no sense that the more instances of exercises a student solve the less probable of solving correctly a new one. Let be e_i the following parametric exercise: X+Y=Z for which curve a) of figure 5 applies. In case e_{ir} is 3+2=Z and the student solve it incorrectly, then the same curve will apply for the following student attempt. Once the student solve the exercise correctly, then a new instance e_{ir+1} will be presented, e.g. 3+1=Z. This time the associated IRF curve which applies is b) of figure 5. Now, for the same ability of the student, it would be easier to solve this exercise instance because the student has already solved it correctly the previous instance.

Let analyze two extreme cases about parametric exercises:

- If a parametric exercise is always the same, e.g. a multiple choice exercise with always the same options, then there will be just one conditional IRF curve associated for the first repetition, and the next times we consider that the student will answer correctly the exercise, so it is very easy for all the students to solve this exercise once the exercise has been solved correctly once.
- If a parametric exercise can be any exercise by chance of a topic in the world, then there will be IRF curves for all the repetitions and all of them will be the

same without changing the difficulty of them, because answering r problems correctly before does not give more chances to answering correctly next time.

The advice is to design the parametric exercises in a way that exercises should have different parameters but that cover a specific concept (e.g. better the multiplication by multiples of two than any type of multiplication) so that the IRF curves can converge and after a number of repetitions solved correctly the exercise *i* can be very easy for any student.

We consider that once an IRF curve has a value of difficulty less than a threshold, e.g. -1.8, then this exercise after M_i repetitions done correctly is very easy and we consider that any student can solve correctly that exercise after these numbers of repetitions. This number of repetitions M_i is different for each exercise. At this moment, new curves for new repetitions are not calculated and we consider the knowledge of this exercise as mastered by any student that repeated such exercise this number of times. Therefore, each parametric exercise has a maximum number of conditional IRF curves which is different from each item. This maximum number of repetitions represents the number of times after solving correctly this exercise, when we consider that the probability of doing that exercise correctly is 1.

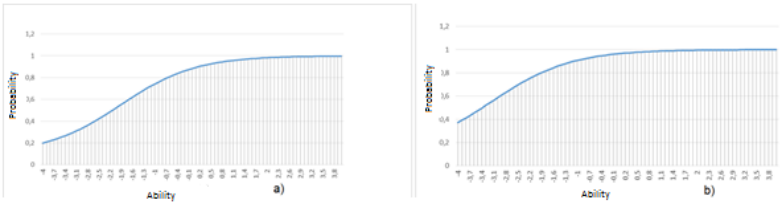

Fig 5 Example of conditional IRF curves for the same exercise in different repetitions

Let M_i be the maximum number of repetitions of a parametric exercise e_i. Then we need to provide all the conditional IRF for that parametric exercise, i.e. the parameters a_{ir}, b_{ir} and c_{ir} for $r = 0 .. M_i$. This estimation can be done directly by experts on the topic, but it is more usually to get these parameters from training data. The algorithm to retrieve these curves depends on the training data used. We will describe e.g. the calibration of these parameters for our evaluation in the results section.

Let e^t_{iaj} be the active instance of exercise in repetition a of a student *j* in type of exercise *i* at time *t*. In addition, the student might have answered incorrectly the exercise an

undefined number of times. In a moment *t*, the curve that should be considered for a type of exercise should be the one for the repetition that is active in that moment, i.e. we consider the student is able to solve correctly the instance of that exercise a number of times *a*.

Step 3: Take all first attempts of students to exercises in order including their result. For the calculation of the student ability, we only consider first attempts of students in an instance of exercise, i.e. first attempts in an exercise or their first attempts in an exercise after solving correctly another instance of the exercise (because the change of instance of exercise takes place when the student solves correctly a previous instance of the same exercise). This is done in this way because according to the IRT independent assumption, we should only include independent exercises for the calculation of the ability and an exercise that is repeated after solving it incorrectly is not independent and would use the same conditional IRF. In a parametric exercise, we only take into account the part that is new for the understanding of such exercise since this part should be taken into account and give evidence about the ability. We only take into account the first attempt after solving the exercise correctly some times before because the first attempt is the most indicative about the student ability and the curves are calibrated with the new knowledge of the student taking into account that the student already solved correctly the exercise some times before.

All these first attempts of students are ordered according to the time when they happened. Let be $\vec{v} = (v_{1r_1j}, v_{1r_2j}, \ldots v_{ir_ij}, v_{nr_dj})$ a vector with length L, being L the total number of first attempts, i the number of type of exercise, *j* the identifier of the user, and r_i the number of repetition of an instance of exercise *i*. Let be *n* the number of different types of exercises that the student attempted. L is different to *n*, because a student can repeat different instances of the same type of exercise several times. An element v_{ir_ij} can be either 0 or 1 and represents if the student answered incorrectly or correctly that first attempt. The elements have a correct order in $\vec{v}$ according to their order of appearance. For any v_{ir_ij} it is not required that all sub-indexes from 0 to r_i must be in vector $\vec{v}$ because it might be the case that several descendant exercises might be solved correctly, which gives evidence of solving correctly instances of ascendants exercises. But these are not real attempts, so they do not appear in $\vec{v}$.

Step 4:Take next attempt. In chronological order according to the events, the next student attempt of vector $\vec{v}$ is taken. If the student answered correctly this attempt, then the instance which is the active repetition for that exercise would be increased by 1, i.e. let be $e^{t+1}_{ia_i'j}$ and $e^{t}_{ia_ij}$, then $a_i' = a_i + 1$, and step 5 is done. In case, the student answered incorrectly this attempt, then the student would go directly to step 6, advancing one position in vector $\vec{v}$. This process is done as many times as the number of first students' attempts.

Step 5: Give evidence about the ascendants exercises of this exercise. As presented previously, one of the algorithm assumptions is that when a student solves correctly an instance of an exercise then he/she will know the concepts to solve one instance of each of the prerequisite exercises (i.e. the ascendants exercises). Therefore, when a student solves correctly a repetition of an exercise, then we consider that this student would solve correctly one additional repetition for each ascendant exercise. So if a student solves correctly an instance of an exercise once, then this would give evidence that the student would solve correctly once instance more of all its ascendant exercises.

This evidence about the ascendant exercises, does not mean that this evidence will create new elements of $\vec{v}$, but the present active repetition and associated IRF curve for the student will change in that type of exercise, summing 1. Let be:

- $specasc_{iq}$: denotes a specific exercise q which is an ascendant of exercise i. But this does not denote the own instance.
- $specasc^{t}_{ib_iqj}$: denotes the instance of the exercise q (which is an ascendant of i) at time instant t for the student j, being this instance the active repetition b_i
- $specasc^{t+1}_{ib_i'qj}$: denotes the intance of exercise q (which is an ascendant of i) at time instant t+1 for the student j, being this instance the active repetition b_i'

Then for any $specasc_{iq}$, we have that $b_i' = b_i + 1$

Step 6: Calculate the real difficulty of the exercise for this student at this moment. The student ability is the latent trait that can explain the performance of the student in the different exercises of the graph of the knowledge structure. Given a specific student latent trait for a student j (fixed value), and a set of exercise instances, which are the active repetitions. Then, for solving correctly an exercise instance which is the active repetition of that exercise, it is needed to solve correctly the exercise instances which are the parent active repetitions. Under these conditions, the graph of the knowledge structure forms a network in which the following conditions are fulfilled:

- It is a Direct Acyclic Graph (DAG).
- For any two nodes (instances of exercises) of the graph of the knowledge structure e_i and $e_{i\prime}$, the probability of solving correctly the active repetition of exercise e_i is independent from solving correctly the active repetition of exercise $e_{i\prime}$ given the values of the parents of e_i (i.e. if the student solved correctly those instances of parent exercises or not).

Let be G a knowledge structure where we can calculate the probability of solving correctly any instance of exercise of the graph, which is the active repetition given a specific student ability. And let be the following:

- e_{ia_ij} represents the active instance (which is the repetition a_i) of the exercise i for the student j. Once the student answers, this value can be 0 (the student solved it incorrectly) or 1 (the student solved it correctly).
- $e_{ia_ij} = 1$ represents the fact that a student j solves correctly his/her active instance (which is the repetition a_i) of exercise i.
- $\text{parents}_{\text{i}p_ij}$ represents any possible combination of values for active instance of the parent exercises (which are at repetitions p_i) of exercise e_i, for a student j.
- $\text{asc}_{\text{i}s_ij}$ represents any posible combination of values for active instances of ascendant exercises (which are at repetitions s_i) of exercise e_i, for a student j.
- $\text{parents}_{\text{i}j} = 1$ represents the event of answering correctly all the instances of parent exercises (which are the active repetitions) of exercise i for student j
- $\text{asc}_{\text{i}j} = 1$ represents the event of answering correctly all the instances of ascendant exercises (which are the active repetitions) of exercise i for student j

- $specasc_{ib_iqj}$= 1: represents the event that a student j solves correctly his/her active repetiton b_i of exercise q which is ascendant of exercise i.
- $parentsspecasc_{iqj}$= 1: represents the event that a student j solves correctly all the instances of active repetitions of parent exercises of the exercise q which is parent of exercise j.

With these conditions and this notation, and given a specific student latent trait, we can calculate according to [45] the probability of doing correctly or incorrectly a set of any n instances of active repetitions of different types of exercises of the graph as:

$$P(e_{1a_1j}, e_{2a_2j}, .. e_{na_nj}) = \prod_i P(e_{ia_i} | parents_{ip_ij}) \quad (3)$$

Only when all the parent instances of exercises have been solved correctly, then there is some chance to solve correctly the next instance of the selected exercise (AND condition). If there is some parent node which is not solved correctly, then the probability of solving correctly that instance will be 0.

In case that the instance of exercise e_i was solved correctly by student j, then all the ascendants of this exercise would be solved correctly for that student because this is a property of the graph knowledge structure, which can be inferred from the conditional tables as we pointed out before. Therefore, if a student can solve correctly an instance of an exercise, then this student can solve correctly all the exercise instances of the ascendant nodes. Therefore, the following equation (4) can be applied.

$$P(asc_{ij} = 1 \mid e_{ia_ij} = 1) = 1 \quad (4)$$

In addition, according to [45], we consider equation (3), and we choose to apply it for calculating that the ascendant nodes can be solved correctly. Then, the probability of solving correctly all the ascendant nodes is given by equation (5). The part on the right side of this equation (5) is in this way because the parents of an ascendant node are also ascendants. So, by the fact of solving correctly all the ascendant nodes, the parents of the ascendants would be solved correctly too. This can explain why the conditioning of parents in equation (5) are as correctly solved but not with any value.

$$P(asc_{ij} = 1) = \prod_q P(specasc_{ib_iqj} = 1 | parents_{iq} = 1) \quad (5)$$

In case that specific parents are taken for equation (5) instead of specific ascendants, then the product would also be among all the ascendants because if the parents are solved correctly, the ascendants would be solved correctly too but not only the parents.

Applying the Bayes Theorem, and putting $p(e_{ia_ij} = 1)$ on the left side, we can obtain equation (6)

$$p(e_{ia_ij} = 1) = \frac{P(e_{ia_ij} = 1 \mid asc_{ij} = 1)\, p(asc_{ij} = 1)}{P(asc_{ij} = 1 \mid e_{ia_ij} = 1)} \quad (6)$$

Combining the previous equations we can obtain equation (7).

$$p(e_{ia_ij} = 1) = P(e_{ia_ij} = 1 \mid asc_{ij} = 1) \prod_{q} P(specasc_{ib_iqj} = 1 \mid parents_{iq} = 1) \quad (7)$$

This means that the probability of solving correctly an instance of an exercise is the product of the probabilities of mastering each node without parents and the conditional probabilities of solving correctly an instance of an exercise conditioned that their parents were mastered (i.e. the knowledge of these nodes were mastered). This is the same to say that we need to multiply the conditional probabilities of all the parent nodes without repeating nodes for calculating the probability of solving correctly an instance of exercises.

Once we have the real probability of solving correctly an instance of an exercise, then we can infer the real difficulty for this exercise, using also the latent trait of the student. It is important to note that the real difficulty is not always the same, but depends on the previous students' interactions and what the student has solved correctly as ancestor nodes. The real difficulty is conditioned to the previous student interactions.

Step 7:Take the next last interaction. In this case, first attempt interactions are taken in an inverse order, so the interactions that took place last are the first to be taken, because we want to calculate the ability based on the last K interactions. Therefore, we form a vector that is the same as $\vec{v}$ but with the inverse order of the elements.

Steps 8 and 9: Selection of the K interactions. In case the attempt is correct, then this attempt will form part of a vector $\overrightarrow{u_j} = (u_{1j}, u_{2j}, \ldots u_{kj})$ where u_{ij} can be either 0 or 1. For this attempt, we will take into account the real probability and real difficulty of the student regarding this exercise at that moment, as specified in step 6 and the correspondent IRF curve to that real difficulty will also be considered. This is this way because in case a student solves correctly an exercise, although vector $\overrightarrow{u_j}$ includes an ascendant node of another, the conditional independence of IRT is maintained because the information provided would be independent. This is due to the fact that when a student answers correctly, the IRF curves are updated for each one of the exercises when the active repetition of that instance of exercise has been solved correctly. Each exercise solved correctly will give evidence in different moments and the real difficulty of both items would be in a way that the conditional independence required by IRT is fulfilled.

In case the attempt is incorrect, then we look if there are any ascendants of this exercise incorrect in the present active instance. If this is not the case, then this attempt will take part of vector $\overrightarrow{u_j}$. If there are some ascendants of this exercise which are solved incorrectly in the present active instance of these ascendants, then we select among all the ascendants, the ones that do not have other ascendants solved incorrectly in the present instance of these exercises. All of them will take part of vector $\overrightarrow{u_j}$ if this does not exceed the total length of k for this vector. In case this length is exceeded, then we only take the number of interactions to have a total of k, taking the ones that happened the latest. All of this is done to keep the conditional independence of the items.

In any case, if vector $\overrightarrow{u_j}$ does not have the total number of k interactions required, then the algorithm takes the next last interaction and continues this process, going back to step 7.

Step 10:Calculate the new student ability. Once vector $\overrightarrow{u_j}$ is formed, the new student ability is calculated according to (1) applying MLE. In this case, we should take into account:

- There can be different u_{ij} elements that can belong to the same instance of exercise, but that correspond to different repetitions.

- All the elements u_{ij} belong to real students' interactions (i.e. it is not because of evidence of ascendants while being the real interaction in a descendant node).
- The IRF curves associated to u_{ij} are the real ones and the associated difficulty is the real one, i.e. it is not directly the IRF conditional curves but the correspondents to the calculation of step 6, because we should take into account the results with the ancestor exercises.

3.2. ALGORITHM 2 PROPOSAL FOR SKILL MODELLING

One of the limitations of the previous commented algorithm 1 is that it only takes into account the first attempts for the calculation of the ability. This was done this way in order not to break the local independence assumptions of IRT. In order to track the learner's ability with parametric exercises, this algorithm 2 proposes to add a new ability level which is local to each one of the exercises. Therefore, we will have many abilities related to one student, i.e. as many as the number of exercises. This local ability will not only take into account first attempts but all of them.

This proposed algorithm 2 maintains the concept of global ability considering all the exercises and its computation is the same as described in previous algorithm 1.

In addition, in order to try to track better the different student responses to parametric exercises, we propose to add a new latent trait variable for each parametric exercise, which we denote as local ability. The local ability is computed for each type of problem. If there are enough number of students' interactions with that type of problem, then we only take into account the attempts of the learner to the instances of that type of problem for the calculation of the local ability. In case, there are not enough number of students' interactions with a type of problem, then we should also take into account the global ability of the student for the calculation of the local ability, because there would not be enough evidence of the local ability with just the students' interactions with that exercise.

In this algorithm, the local ability is used to compute the real probability of an item while the global ability is used to reflect the average knowledge of the learner in the topic covered by the different exercises. When we do not have any information about

the local ability in an item, then the global ability is used to compute the real probability and with the subsequent attempts the local ability is updated as follows:

- Let be θ_{gj} the global ability of a student *j* in a topic related to a set of exercises.
- Let be θ_{lij} the local ability of a student *j* in an exercise *i*
- Let be θ'_{lij} the local ability of a student j in an exercise i considering only attempts in that specific exerice.
- Let be T_{ij} the number of attempts that a student *j* has done in an exercise *i*. In this case, this does not only include the first attempts in an exercise instance but all of them.
- Let be K_2 the minimum number of interactions of a student with an exercise to consider that we can compute the local ability only with these local interactions.
- Let be W_1 and W_2 the weights used to make an average of the local ability, considering the global interactions and also the local ones in the exercise. In this case $W_1 = 1 - \frac{T_{ij}}{K_2}$ y $W_2 = \frac{T_{ij}}{K_2}$

Then, the local ability for an exercise *i* is computed as follows:

$$\begin{cases} \theta_{lij} = W_1\theta_{gj} + W_2\theta'_{lij} \text{ in case } T_{ij} < K_2 \\ \theta_{lij} = \theta'_{lij} \text{ in case } T_{ij} \geq K_2 \end{cases}$$

The θ'_{lij} is computed taking the last k_l interactions of a student with that exercise, being $k_l > K_2$, , and applying the MLE. It is remarkable that the calculation of the local ability follows the principle of local independency because the ability is local to the exercise but not the global ability. Therefore, although it is clear that two instances of the same exercise are dependent, they are independent once we consider a given value of the latent trait, i.e. the latent trait is what explains the difference in student level for solving that exercise.

With the local ability, we expect more accuracy to compute the real probability of solving an exercise correctly. If a learner has failed several times an item, the local ability is very low, therefore the real probability will be low too. However, the calculation of the local skills for a student in an instant of time requires some additional processing time, because this algorithm executes the same code as algorithm 1 but in

addition additional processing for the calculation of the local skills for each one of the items.

4. DESCRIPTION OF THE ADAPTATION ALGORITHMS

In this section, we describe two algorithms for adapting the exercises that are presented to the students based on the level of skills estimated with the previous algorithms in section 3. The adaptive algorithms depend on the skill modelling algorithms since the specific modeling and conditions applied for skill modelling should be taken into account for the adaptation phase. Algorithm 1 for adaptation will use algorithm 1 for skill modelling, while algorithm 2 for adaptation will use algorithm 2 for skill modelling.

4.1. ALGORITHM 1 FOR ADAPTATION

Figure 6 shows the general overview of algorithm 1 for adaptation. For each new student attempt in an exercise, first the algorithm recalculates the new ability of the student using the explained algorithm 1 for skill modelling (from previous section). This will give the new global ability of the student in the topic.

Next, the algorithm has some steps for treating with blocking exercises. Blocking exercise is a concept that emerges because it is possible a case in which a student can be blocked in a specific exercise and he/she cannot solve correctly this exercise after several attempts. The probability that the model can give that this student should solve this exercise correctly in some attempts might be very high but even under this situation the student might fail at doing this exercise. This special situation is a blocking situation and our algorithm has a specific method to detect and react under this condition. The blocking situation might be due to two different causes:

- The student needs to previously master the knowledge of an exercise where he/she has a lack of knowledge which is required to solve the blocking exercise. As our algorithm might do a student jump some exercises in the tree, then it is possible that the student requires doing those previous exercises first to solve the blocking exercise.

- The student is stuck in this exercise and needs some knowledge that cannot be obtained in the platform, so some new content should be created at this point.

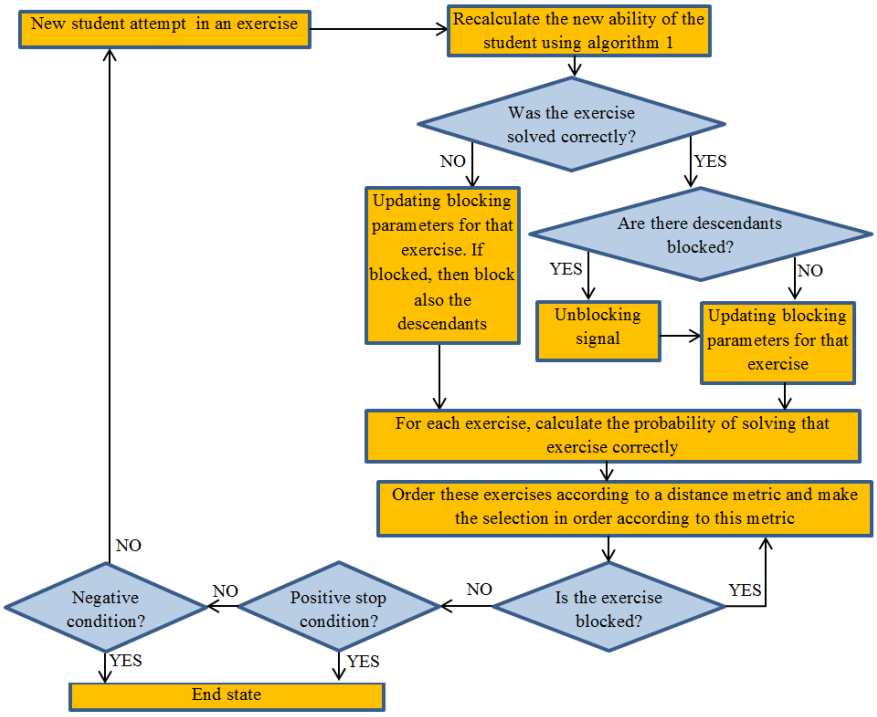

Fig 6 Diagram for algorithm 1 for adapting exercises

As only first attempts after correct resolution are taken (this is a reason based on skill modelling to guarantee the local independence assumption when combining KS and IRT), then if a local blocking probability is not calculated for the adaptation decision, we would not take into account other attempts rather than first attempts after correct resolution and an exercise could be repeated forever in the adaptation decision, which makes no sense, so an adaptive criteria to stop it should be included.

Taking into account this aspect, in the proposed algorithm, each exercise i has a related "blocking probability" B_{ij}^{t} for each student j at an instance of time t. This is the probability that after N attempts on that exercise by the student, the student should solve correctly this exercise. This is the probability that is estimated by the model. When this probability is greater than a threshold (e.g. 0.98) but the student did not solve correctly the exercise, then the algorithm will decide that this exercise is blocked. When this happens, this exercise cannot be selected again (even if the adaptation algorithm selects the exercise according to the set up criteria) after the exercise is unblocked again.

Let be B_{ij}^{t} the blocking probability of a student in instant *t*, p_{ij}^{t} the probability of a student of solving this exercise correctly at time *t*, and B_{ij}^{t+1} the blocking probability of the student in an instant *t+1*. If the student fails an exercise, then B_{ij}^{t+1} is calculated as follows:

$$\begin{cases} B_{ij}^{t+1} = p_{ij}^{t} \text{ in case it is the first attempt} \\ B_{ij}^{t+1} = P\left(B_{ij}^{t} \cup p_{ij}^{t}\right) = B_{ij}^{t} + p_{ij}^{t} - B_{ij}^{t} \text{ in case it is not the first attempt} \end{cases}$$

For the first attempt on the exercise, the value of the new blocking probability is the probability of answering correctly that exercise at that moment. For the following incorrect attempts, the new blocking probability is the union of probabilities of the past blocking probability and the probability of solving correctly that exercise at that moment. This is, the blocking probability represents the probability that at least one of the N interactions of the student with the exercise was correct. If the blocking probability is greater than a threshold, then this exercise is considered as blocked as well as all its descendants.

When a student solves correctly an exercise, then the blocking probability B_{ij}^{t+1} is reset to 0 because there is evidence that the student is not blocked in that exercise. In addition, there is a signal to all the descendants to unblock them in case there is someone blocked. If an exercise is blocked, then it is sure that at some moment the student will have to do some ascendant exercises which are not completely mastered and which are not blocked. The logic under this is that if a student solved some prerequisite, then the student might be able to do some descendant exercise which was blocked. At this moment, the exercises that were blocked are unblocked again and can be selected.

The concept of blocking exercise in the adaptive algorithm is important in relationship with the skill modelling algorithm. As there are prerequisite relationships among exercises, then if an exercise cannot be solved, then the descendants cannot be solved so they should not be selected for adaptation. And if an exercise is solved correctly, then this gives an indication for the ascendant exercises. In other skill models (e.g. traditional IRT) this should not be taken into account because there are not such relationships in the model.

An exercise is considered as blocked permanently when that exercise is blocked and has all its ascendants as mastered or blocked. If an exercise is blocked permanently then the algorithm will never select any exercises which are descendant of it, since the descendant exercises are supposed to require the knowledge of the blocked exercise. Therefore, if an exercise is permanently blocked then the descendant exercises of this exercise will also be blocked.

Once the blocking parameters have been updated, next step is the calculation of the probability of solving correctly any of the considered exercises. This calculation is done as explained in skill modelling algorithm 1. Next, the algorithm selects the next exercise, which can be done with different distance metrics. For example, we propose to select as next exercise, these ones which probability is closer to some value. We set e.g. an initial value of 0.80 for this probability. Therefore, in each step we know the probability of a student of solving correctly each of the different exercises, and we select the exercise which probability is closer to this threshold and that is not blocked. Many other previous works (e.g. [20]) use the same concept of calculating some function taking into account the difficulty of exercises and the student ability to make the decision of the next exercise to select.

We consider that a student pass a topic or didactic unit when the student does not have any exercises blocked and the probability of doing correctly any of the exercises of the didactic unit is greater than a threshold, e.g. greater than 0.95 or 0.98. This is the positive stop condition.

We consider that a student does not pass a topic or didactic unit when the student has at least one exercise permanently blocked o when without having an exercise permanently blocked, the probability of doing correctly any of the remaining exercises is lower than a threshold, e.g. lower than 0.20. This is the negative stop condition.

The consequence of a finishing condition on a topic or didactic unit is that the student exits the topic or the didactic unit. Positive and negative stop conditions criteria can be changed.

4.2. ALGORITHM 2 FOR ADAPTATION

Algorithm 2 for adaptation makes use of algorithm 2 for skill modelling. Algorithm 2 for adaptation works as the algorithm 1 but introducing some slight differences, which

are explained next. These differences are related to the differences between skill modelling algorithms 1 and 2, which have also an implication in the adaptation algorithms.

In algorithm 2, for considering an item as mastered the algorithm checks if the student has answered correctly the selected item k times (e.g. three) in a row, if so, the item is marked as mastered, its conditional probability is 1 in next times and it cannot be selected again.

The algorithm calculates the local ability of the students in all the exercises and associates the correspondent probability to each exercise. The next exercise is selected based on the probability which is closer to some threshold as in algorithm 1 for adaptation. Here, the difference is that local abilities are used instead of the global one because the skill modelling algorithm 2 includes these local abilities in its model, which were not present in algorithm 1 so they cannot be used for adaptation purposes.

The learner can finish the topic or didactic unit with a master or a non-master condition as in algorithm 1. The positive stop criteria remains the same but the negative stop criteria is slightly modified. In algorithm 2, blocks disappear so there is only one chance to finish with the negative criteria. This happens when the probability of the remaining items (the ones that are not mastered) is below a threshold after a configurable number of times.

Unlike algorithm 1, blockings are unnecessary in the second algorithm because of the local ability that tracks the mastery for a particular item. If a learner has a low local ability in a particular exercise the probability of solving correctly that item is also low. This probability is used to compute the real probabilities of the descendants, therefore all the descendants will have a slower value of probability than the original item and so they won't be selected. This behavior is similar to the explained for algorithm 1 but omitting blockings. As in algorithm 1, each parametric exercise has associated multiple IRFs but in this case doing the last IRF correctly does not imply that the student has mastered the exercise. As it is explained before, the curve indicates the probability to master an item. The algorithm considers the item as mastered only when this probability is above a threshold after the student has done correctly the item.

If the blocking concept was not used, then algorithm 1 for adaptation would not change the probability of solving correctly an exercise when an instance of exercise is solved incorrectly several times (because we only consider first attempts). In the algorithm 2 for adaptation, this probability changes when an exercise is solved incorrectly several times in a row, because not only first attempts are taken into account but all and there is a local ability. To do so, algorithm 2 calculates all the local abilities for a student in each exercise, which is computationally more expensive. However, on the other hand, algorithm 2 does not manage the blocking concept, which makes it simpler and less computationally expensive. Depending on the situation (e.g. the number of blocking conditions) algorithm 1 or 2 might be better in terms of computational performance.

5. RESULTS AND DISCUSSION

This section is divided into two subsections. Subsection 6.1 is devoted to the evaluation with real students in order to validate the accuracy of the prediction model. Subsection 6.2 is devoted to the evaluation using a simulator in order to evaluate the performance in terms of execution time and percentage of student finishing in different conditions.

5.1. EVALUATION WITH DATA FROM REAL STUDENTS

We use a dataset with interactions of students (who are children) with exercises. The dataset contains a total number of 170.254 interactions from 1.068 different students with 254 different types of exercises about maths.

For the calibration of the IRF curves, the dataset was separated into two groups: 30% of the dataset was used for training, while 70% of the data was used to test the approach.

Each IRF curve is defined by its slope, guessing and difficulty parameters. In our case, the slope parameter is always fixed to 1. The guessing parameter depends on the type of exercise. If the exercise is a multiple choice, then the number of options of the type of problem determines the guessing parameter, e.g. if there are five options, then guessing is set to 1/5. In general, depending on the type of exercise, the guessing parameter is estimated by the experts who estimate the probability of solving an exercise correctly

even if a student does not have any knowledge about it, i.e. by chance. Finally, the difficulty of each IRF curve is calibrated based on previous data, using the training data.

A recent study of Pelánek, Rihak & Papousek [46] shows that the way data is collected can have an important impact on the evaluation and the results. The manuscript warns that researchers do not typically take this aspect into account. In particular, the factors of mastery attrition bias and adaptive choice [46] are studied and inaccuracies in the results are proved with these two issues. Mastery attrition bias is based on the fact that many tutors present different exercises to a student until the student masters them. Therefore, there are some exercises that are not presented to the student because the student already mastered them. This introduces a gap because it is probable that these students might solve correctly these exercises but as there are not any interactions then this is not taken into account in the calibration. The calibration usually only takes into account students' interactions with those exercises, but these are of students who did not master the topic, so the calibration is bounded because we do not take the students who interacted with that exercise by chance to infer its parameters, but a set of specific students who did not master the topic so that it is more difficult for them such exercise.

For the used dataset, this issue of mastery attrition is present, since when a student is considered to master a type of exercise, then a new instance of a new type of exercise is presented and no more instances of the initial type of exercise will be presented. Therefore, in a parametric exercise, the number of real students interacting with that exercise, considering that the student has solved correctly the exercise r times, decreases as r is greater, because when a student master the type of exercise, the student will not be presented with more instances of this exercise. If we only take the real interactions of students with exercises, a bias will be introduced because we are only considering students who interact with a repetition r with the type of exercise, i.e. students who did not master the type of exercise before, and cases that make no sense might happen such as considering an exercise more difficult when a student has solved it correctly r +1 times than when the student answered it correctly r times.

This mastery attrition issue is taken into account in our calibration algorithm to avoid these types of problems during the validation. The calibration algorithm to estimate the difficulty of each IRF is as follows. First, we estimate an initial value of the difficulty for each IRF using the Proportion Correct Method, which has been evaluated as the best

in a previous study [47]. This method takes into account the number of correct responses (n) in relation to the total number of responses (N). The mathematical function to compute the difficulty is obtained by clearing the difficulty parameter in the Rasch model as is shown in

$$\beta_i = \log\left(\frac{1 - P(\theta)_i}{P(\theta)_i}\right)$$

In this case, $P(\theta)_i$ can be replaced by the proportion correct (n/N) therefore the final function is given by

$$\beta_i = \log\left(\frac{1 - \frac{n}{N}}{\frac{n}{N}}\right)$$

After obtaining the initial parameters of the IRF curves, the calibration algorithm also use the train subset. First, the calibration algorithm uses the real students' events with the exercises. The algorithm considers the real students' interactions with a repetition of a type of exercise to estimate the proportion of corrects for that repetition of type of exercise. In addition, it also estimates what the students' interactions would be with exercises that have not been attempted to avoid the mastery bias. To do so, n is estimated as the probability of a student for solving correctly a repetition r of that exercise with his/her past level of ability, while N is estimated as the product of the probabilities of all its ancestors, i.e. the probability of knowing all the previous prerequisites to tackle this exercise. Moreover, the aim of the calibration algorithm is also discovering the number of IRF associated with each parametric exercise creating or deleting curves during its execution. The curves obtained by means of the calibration algorithm are fixed and used for the analysis of both algorithms 1 and 2.

For the evaluation of the accuracy of the prediction of the two proposed algorithms for skill modelling, we use the RMSE parameter, as suggested as the best metric in our case by a recent study [48]. The study reported by Pelanek [48] warns that in many occasions parameters such as AUC or MAE should not be used for the evaluation of skill modelling methods, although this is the common practice in many studies. For predictors of binary outcomes (this is the case in what we want to evaluate, i.e. if the student will answer correctly or not an exercise), the MAE metric is not adequate [48], while the AUC metric is less suitable for the evaluation of skill models [48].

Figure 7 shows a comparison in terms of RMSE between algorithm 1 and algorithm 2 for different buffer sizes, i.e. the *k* value that corresponds to the last students' interactions to estimate the new ability following the MLE method.

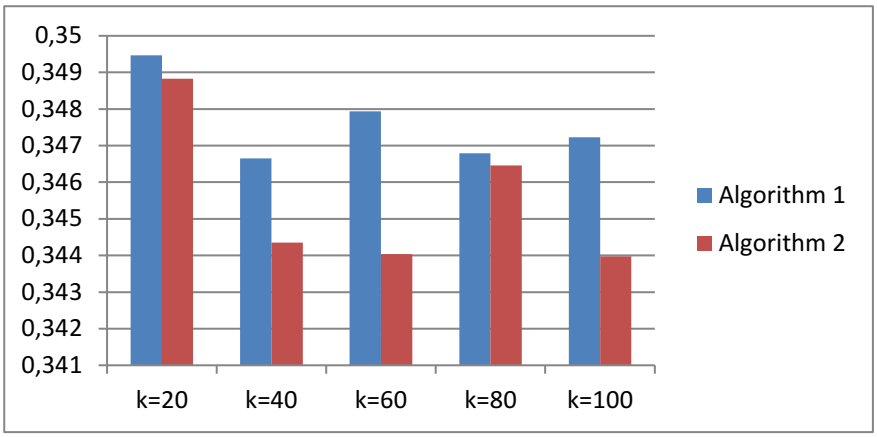

Fig 7 RMSE comparison between the two algorithms for different buffer sizes

The RMSE values are good with values under 0.35 in all cases. Therefore, our prediction algorithms are good to estimate if the students will answer correctly the different exercises. These values validate that the proposed algorithms work well for estimating students' skills within the selected dataset. If we compare these RMSE values with others found in other previous works, we can see e.g. from recent previous works that the best algorithms in Pelánek, Rihak & Papousek, 2016 [46] obtained values of RMSE over 0.37, and the algorithms tested by Kaser, Klingler, Schwing, & Gross [49] got RMSE values from 0.324 to 0.465.

However, it is important to note that the values of RMSE depend a lot on the dataset and type of contexts. For example, Kaser, Klingler, Schwing, & Gross [49] proved that RMSE values changed a lot depending on the considered topic. Therefore, we can say that our algorithms make a good prediction of student skills, with values that are good compared with other previous works, but we cannot make comparison with other algorithms since we should use the same dataset, the same conditions, etc.

Another aspect that we can see from figure 7 is that although RMSE values are better for algorithm 2, the difference is very low. Therefore, algorithms 1 and 2 are almost equally accurate. Initially, we expected an improvement with algorithm 2 because we

thought that algorithm 2 would track better the local responses of items, but this proves that algorithm 1 can do it almost as well. In a different context, e.g. with different topics this might change. However, the adaptive algorithm based on skill modelling algorithm 2 is easier to implement since there are not e.g. blockings, so it is worth to use this algorithm 2.

The variation of the buffer size (k) did not have an influence on the accuracy of the prediction. Therefore, we can use a low number of last students' interactions without having an effect on the RMSE, since accuracy is not clearly improved by increasing the number of last interactions. Taking also k=3, k=6, k=9 and k=12 for algorithm 2, we got RMSE values of (0.381, 0,360, 0.353, 0.350) respectively. Therefore, when k is low the differences of accuracy might be greater, but as k is equal or greater than 6, the differences of RMSE are very low and we can conclude that there is not almost any effect of k in the accuracy.

Figure 8 shows a comparison in terms of RMSE between algorithm 1 and algorithm 2 but only for first attempts in questions or first attempts in questions after a student solved it correctly an instance of the same type of exercise.

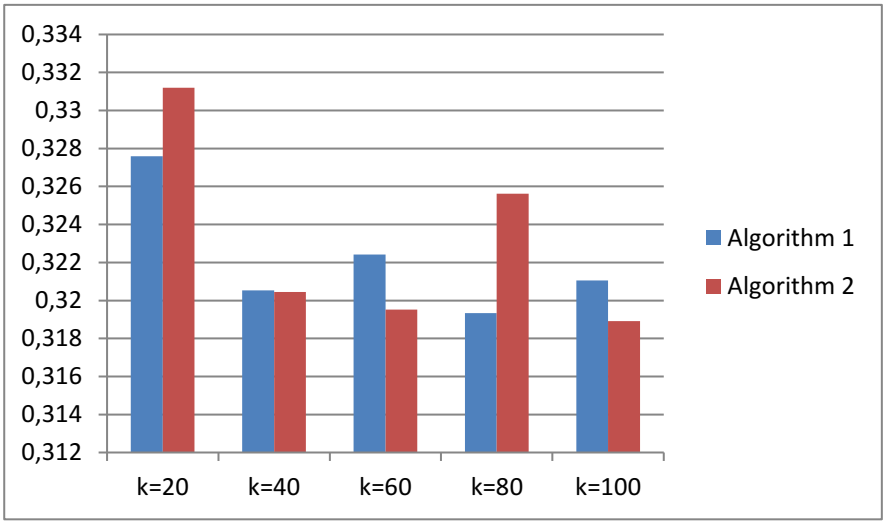

Fig 8 RMSE comparison for first attempts between the two algorithms for different buffer sizes

Figure 8 shows that RMSE differences are very small between algorithm 1 and 2, as expected, since the prediction of first attempts is quite similar in both algorithms. There is also almost no difference when varying the buffer size k. There is an improvement on the prediction of first attempts with respect to the general prediction as expected (since the algorithms take track of the first attempts for some of the predictions), but the difference is not high.

Finally, figure 9 is associated with algorithm 1 and represents the estimated probability of answering a question correctly given by the model using algorithm 1 (axis x) with respect to the average frequency of real correct answers by students when the condition of that probability takes place (axis y). In this case the number of last interactions (k) considered for estimating the ability has been fixed to 10. In an analogous way, figure 10 represents the same for algorithm 2. An ideal algorithm would have all the points on the line. In any case, we can see that the algorithms follow well the tendency. It is also important to note that most of the estimated probability values by the algorithms are high values (as we can see in the probability density functions). In these cases, the algorithms fit very well with the line. The estimated probabilities are quite good for values over 0.60 and they are not so good as the estimated probabilities decrease. However, there are only a few number of cases with low probabilities estimated by the model, so for these cases it is difficult to extract any conclusions because we would need more data. As future work, it would be interesting to have more values of probabilities bellow 0.60 to test how well the model is working there. We would need a different dataset where this might happen, e.g. where the questions are more difficult to students or with students with a lower level who answer incorrectly a greater number of the dataset questions.

Fig 9 Representation of the frequency of students' responses vs estimated probability, and the distribution of estimated probabilities for algorithm 1

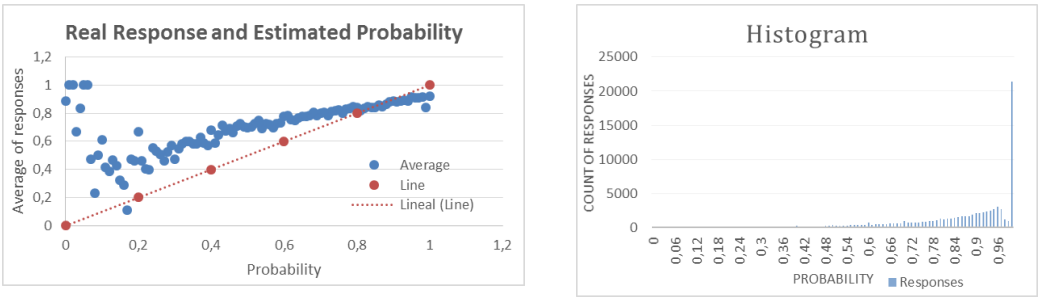

Fig 10 Representation of the frequency of students' responses vs estimated probability, and the distribution of estimated probabilities for algorithm 2

5.2. EVALUATION WITH A SIMULATOR

A student simulator has been designed and implemented to simulate the skill modelling algorithms but also the proposed adaptation algorithms for the adaptive selection of the next item. The main purpose of the created simulator is to validate the performance of the adaptation algorithms in terms of time response, and number of students that can finish with the master condition under some simulated conditions. The designed simulator creates fictitious students.

The designed simulator assigns an initial ability to each student using a normal distribution N(0,1) and emulates its response in each iteration depending on the probability of the selected item using a Bernoulli distribution. The selection threshold to select the next item is fixed to 0.8 (i.e. the algorithm will select items that are closer to 80% of probability of solving them correctly) and the buffer size of the last interactions to make the calculation is fixed to 10 (k=10). The considered exercises are exactly the same as the ones for the evaluation with real students and are taken from the Smartick company.

In these conditions, measuring the time of execution for calculating the next item (including also the time for calculating the new skill of the student) had a performance of 132 ms. for algorithm 1, while algorithm 2 gave a performance of 392 ms. This can be explained because algorithm 2 of skill modelling adds new processing for the calculation of the skill, which is the calculation of the local ability.

A total of six different simulations with algorithm 1 gave a 40% of students who completed the contents with master condition, while a 70% of students who completed the contents with master condition for algorithm 2. Therefore, a good number of students completed the topics with a global master condition.

6. CONCLUSIONS

This paper presents new algorithms for skill modelling combining KS and IRT for estimating the student ability and including parametric exercises. The original KS and IRT have been adapted to work together and be according to the assumptions. In addition, the paper presents new algorithms for the adaptation of exercises based on the skill modelling proposals. The proposals take into account the best features of KS and IRT. KS adds the semantic relationships between the different exercises (prerequisite relationships) while IRT adds a powerful formulation for calculating the ability taking into account different exercise parameters. The combination of both methods (KS and IRT) let us have a more complete model.

In addition, the presented algorithms offer a solution for considering parametric exercises as exercises that should be solved some number of times to be mastered and exercises that can change their difficulty depending on the number of times that students have solved them correctly. Parametric exercises are completely integrated in our models in combination with KS and IRT.

The evaluation of the proposed skill modelling algorithms with real students provided good results of RMSE, so their prediction accuracy is good. As a future work, we aim at comparing our proposed algorithms with a common dataset, integrating other state of the art approaches (such as BKT, LFKT or FAST) with KS and parametric exercises.

The proposed algorithms 1 and 2 share the same base for the calculation of the global ability. However, algorithm 2 adds the concept of local ability to try to track better local student answers in exercises to be according to the local independence of items. This additional modelling adds new calculations for the estimation of student ability. However, adaptation algorithm 1 (associated with skill modelling algorithm 1) implies a heavier calculation. In the validation scenario with real students, both algorithms got almost the same accuracy on prediction. In addition, algorithm 1 got better results on

performance time with the simulator. In this case, it is better to use algorithm 1 for skill modelling and for adaptation.

The results of similar accuracy can be explained by the specific scenario: the probability of students to solve the exercises correctly is high. In a different scenario where there is a greater variety of exercises, the local ability can be used to track better the ability and algorithm 2 can make a difference. In addition, in this scenario, because of the high probability of solving exercises correctly, the blocking of exercises is low so there is no extra calculation for algorithm 1 but in other scenarios algorithm 1 might require more performance time than algorithm 1. Future work can study the accuracy and performance time in other scenarios for both algorithms to extract conclusions of when it is better to use one or another.

The variation of the buffer size for the calculation of the new ability using MLE did not have a considerable effect on the accuracy if this buffer size is greater than 3. Again, this might change if the proposal is tested in a different scenario, e.g. with greater probability of solving correctly the exercises.

The estimated probability using the algorithms of solving the exercises correctly is very good for estimations with probability greater than 60%. For estimations of probabilities less than 60%, the algorithms follow the tendency but the prediction is not so good. However, because of the tested environment, the number of interactions in these conditions is low so we would need another new scenario (with students who interact with exercises where they answer incorrectly more frequently) in order to reach conclusions. In any case, from the evaluation with real students, we can hypothesize that the algorithms are quite conservative, i.e. the estimated probability is bellow the real probability of solving the exercises correctly. This can be explained because one of the assumptions of the model, that a student should master all the ascendants to be able to solve the exercise, might not be 100% true in all the cases of contents in the data set. There might be some exercises for which there is some possibility to solve them correctly, even if students did not master the ascendant exercises. In order to solve this issue, we might include a correction factor in the algorithms so that the estimation of the probability can be lower.

ACKNOWLEDGMENTS

This work was supported by the Smartick company (project entitled “ANÁLISIS, EVOLUCIÓN, PROPUESTAS DE MEJORA y DESARROLLO DEL SISTEMA DE APRENDIZAJE ADAPTATIVO Y ANALÍTICA DEL APRENDIZAJE DE LA PLATAFORMA SMARTICK”), by the Madrid Regional Government (eMadrid project, grant number S2013/ICE-2715), and by the Spanish Ministry of Economy and Competiveness (Smartlet project, grant number TIN2017-85179-C3-1-R) funded by the Agencia Estatal de Investigación (AEI) and Fondo Europeo de Desarrollo Regional (FEDER).

REFERENCES

1.Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. The journal of the learning sciences, 4(2), 167-207.

2.Desmarais, M. C., & Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling and User-Adapted Interaction, 22(1-2), 9-38.

3.Saludes, J., Martin, S., & Dalmau, M. (2002). Parametric Exercises with Zope Server.

4.Muñoz-Merino, P. J., Ruipérez-Valiente, J. A., Alario-Hoyos, C., Pérez-Sanagustín, M., & Delgado Kloos, C.. Precise Effectiveness Strategy for analyzing the effectiveness of students with educational resources and activities in MOOCs. Computers in Human Behavior, 47, 108-118.

5.Desmarais, M. C., Maluf, A. & Liu, J. (1996). User-Expertise Modeling with Empirically Derived Probabilistic Implication Networks. User Modeling and User-Adapted Interaction 5(3-4), 283-315

6.Desmarais, M. C., Meshkinfam, P. & Gagnon, M. (2006). Learned student models with item to item knowledge structures'. User Modeling and User-Adapted Interaction 16(5), 403-434.

7.Embretson, S. E., & Reise, S. P. (2013). Item response theory. Psychology Press.

8.Millán, E., Loboda, T., & Pérez-de-la-Cruz, J. L. (2010). Bayesian networks for student model engineering. Computers & Education, 55(4), 1663-1683.

9.Falmagne, J. C., Cosyn, E., Doignon, J. P., & Thiéry, N. (2006). The assessment of knowledge, in theory and in practice. In Formal concept analysis (pp. 61-79). Springer Berlin Heidelberg.

10. Ueno, M. (2002). An extension of the IRT to a network model. Behaviormetrika, 29(1), 59-79.

11.Brusilovsky, P., & Maybury, M. T. (2002). From adaptive hypermedia to the adaptive web. Communications of the ACM, 45(5), 30-33.

12.Muñoz-Merino, P. J.,Delgado Kloos, C., Muñoz-Organero, M., & Pardo, A. (2015). A software engineering model for the development of adaptation rules and its application in a hinting adaptive e-learning system. Computer Science and Information Systems, 12(1), 203-231.

13.Van der Linden, W. J., & Glas, C. A. (Eds.). (2000). Computerized adaptive testing: Theory and practice. Dordrecht: Kluwer Academic

14.Muñoz-Merino, P. J., Fernández Molina, M., Muñoz-Organero, M., & Delgado Kloos, C. (2012). An adaptive and innovative question-driven competition-based intelligent tutoring system for learning. Expert Systems with Applications, 39(8), 6932-6948.

15.Lin, Y., Gong, Y., & Zhang, J. (2017). An adaptive ant colony optimization algorithm for constructing cognitive diagnosis tests. Applied Soft Computing, 52, 1-13.

16.Lord, F. M. (1980). Application of item response theory to practical testing problems. Hillsdale, NJ: Lawerence Erwrence Erlbaum Associates.

17.Hsu, T. C., & Sadock, S. F. (1985). Computer-assisted test construction: A state of art. TME Report 88, Princeton, NJ, Eric on Test, Measurement, and Evaluation, Educational Testing Service.

18.Horward, W. (1990). Computerized adaptive testing: A primer, Hillsdale. New Jersey: Lawerence Erwrence Erlbaum Associates.

19.Guzmán, E., Conejo, R., & Pérez-de-la-Cruz, J. L. (2007). Adaptive testing for hierarchical student models. User Modeling and User-Adapted Interaction, 17(1-2), 119-157.

20.Chen, C. M., Lee, H. M., & Chen, Y. H. (2005). Personalized e-learning system using item response theory. Computers & Education, 44(3), 237-255.

21.Chen, C. M. (2008). Intelligent web-based learning system with personalized learning path guidance. Computers & Education, 51(2), 787-814.

22.Huang, S. L., & Shiu, J. H. (2012). A user-centric adaptive learning system for e-learning 2.0. Educational Technology & Society, 15(3), 214-225.

23.Scholz, F. W. (1985). Maximum likelihood estimation. Encyclopedia of statistical sciences.

24.Lee, Y. W. (2004). Examining passage-related local item dependence (LID) and measurement construct using Q3 statistics in an EFL reading comprehension test. Language testing, 21(1), 74-100.

25.Briggs, D. C. & Wilson, M. (2003). An introduction to multidimensional measurement using Rasch models. Journal of Applied Measurement, 87-100.

26.Culbertson, M. J. (2014). Graphical models for student knowledge: Networks, parameters, and item selection (Doctoral dissertation, University of Illinois at Urbana-Champaign)

27.Almond, R., Yan, D., & Hemat, L. (2007). Parameter recovery studies with a diagnostic Bayesian network model. Behaviormetrika, 35(2), 159-185.

28.Zagorecki, A., & Druzdzel, M. J. (2013). Knowledge engineering for Bayesian networks: How common are noisy-MAX distributions in practice?. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(1), 186-195.

29.Oniśko, A., Druzdzel, M. J., & Wasyluk, H. (2001). Learning Bayesian network parameters from small data sets: Application of Noisy-OR gates. International Journal of Approximate Reasoning, 27(2), 165-182.

30.Galán, S. F., & Díez, F. J. (2000). Modeling dynamic causal interaction with Bayesian networks: temporal noisy gates. In Proc. 2nd Inter. Workshop on Causal Networks (pp. 1-5).

31.Conati, C., Gertner, A., & Vanlehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. User modeling and user-adapted interaction, 12(4), 371-417.

32.Pardos, Z. A., Heffernan, N. T., Anderson, B., Heffernan, C. L., & Schools, W. P. (2010). Using fine-grained skill models to fit student performance with Bayesian networks. Handbook of educational data mining, 417.

33.Corbett, A. T., & Anderson, J. R. (1995). Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction 4(4), 253-278.

34.Pardos, Z. A., & Heffernan, N. T. (2010). Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. Journal of Machine Learning Research W & CP.

35.Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. (2013). Individualized bayesian knowledge tracing models. In Proc. of the International Conference on Artificial Intelligence in Education (pp. 171-180).

36.Pardos, Z. A., & Heffernan, N. T. (2010). Modeling individualization in a bayesian networks implementation of knowledge tracing. In Proc. of the International Conference on User Modeling, Adaptation, and Personalization (pp. 255-266).

37.Pardos, Z., & Heffernan, N. (2011). KT-IDEM: introducing item difficulty to the knowledge tracing model. User Modeling, Adaption and Personalization, 243-254.

38.Käser, T., Klingler, S., Schwing, A. G., & Gross, M. (2014). Beyond knowledge tracing: Modeling skill topologies with bayesian networks. In Proc. of the International Conference on Intelligent Tutoring Systems (pp. 188-198).

39.Cen, H., Koedinger, K.R., & Junker, B. (2006). Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement. In Proc. of the 8th Intelligent Tutoring Systems Conference, (pp. 164-175).

40.Pavlik, P. I., Cen, H. & Koedinger, K.R. (2009). Performance Factors Analysis - A New Alternative to Knowledge Tracing. In: Proc. of the 2009 conference on Articial Intelligence in Education (pp. 531-538).

41.Khajah, M., Wing, R., Lindsey, R., & Mozer, M. (2014). Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In Proc. of the Educational Data Mining conference.González-Brenes, J., Huang, Y., & Brusilovsky, P. (2014). General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In Proc. of the 7th International Conference on Educational Data Mining (pp. 84-91).

42.González-Brenes, J., Huang, Y., & Brusilovsky, P. (2014). General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In Proc. of the 7th International Conference on Educational Data Mining (pp. 84-91)

43.Heller, J., Steiner, C., Hockemeyer, C., & Dietrich, A. (2006). Competence-based knowledge structures for personalised learning. International Journal on ELearning, 5(1), 75.

44.Hockemeyer, C., & Albert, D. (1999). The adaptive tutoring system RATH. In ICL99 Workshop Interactive Computer aided Learning: Tools and Applications. Villach, Austria: Carinthia Tech Institute.

45.Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann Publishers.

46.Pelánek, R., Rihák, J., & Papoušek, J. (2016). Impact of data collection on interpretation and evaluation of student models. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (pp. 40-47). ACM.

47.Wauters, K., Desmet, P., & Van Den Noortgate, W. (2012). Item difficulty estimation: An auspicious collaboration between data and judgment. Computers & Education, 58(4), 1183-1193.

48.Pelánek, R. (2015). Metrics for evaluation of student models. JEDM-Journal of Educational Data Mining, 7(2), 1-19.

49.Kaser, T., Klingler, S., Schwing, A. G., & Gross, M. (2017). Dynamic Bayesian Networks for Student Modeling. IEEE Transactions on Learning Technologies.