iTalk2Learn
2014-04-30

Deliverable 2.2.1

Initial report on methods and prototype for adaptive intelligence for robust learning support

30 April 2014

D2.2.1 Initial report on methods and prototype for adaptive intelligence for robust learning support

Project acronym: iTalk2Learn

Project full title: Talk, Tutor, Explore, Learn: Intelligent Tutoring and Exploration for Robust Learning

Work Package: **2**

Document title: Deliverable 2.2.1

Version: **1.2**

Official delivery date: **30 April 2014**

Actual publication date: 30 April 2014

Type of document: Report

Nature: Public

Main Authors: Carlotta Schatten (UHi), Beate Grawemeyer (BBK), Manolis Mavrikis (IOE), Lars Schmidt-Thieme (UHi), Ruth Janning (UHi), Sergio Gutiérrez Santos (BBK)

Reviewers: Claudia Mazziotti (RUB), Katharina Loibl (RUB)

Version	**Date**	**Sections Affected**
1.0	7 April 2014	Initial version
1.1	25 April 2014	Review comments processed
1.2	28 April 2014	Layout updates for submission

Executive Summary

In this deliverable we address the problems performance prediction, sequencing and adaptive support. For performance prediction we explain the machine learning method Matrix Factorization and how it can be applied to Intelligent Tutoring Systems, especially to the Maths-Whizz data set. The main advantages of Matrix Factorization for performance prediction are its domain independence, small computation costs and no need of an exploratory corpus. Our proposed sequencing approach uses this performance prediction method in addition to a policy based on the theory of Vygotsky's Zone of Proximal Development. The efficiency of the approach is proven by experiments and discussed in the paper "Adaptive Content Sequencing without Domain Information" (Schatten et al. 2014a), that was presented at CSEDU 2014. The platform integration of this approach is described in D4.2.1 as well as in "Minimal Invasive Integration of Learning Analytics Services in Intelligent Tutoring Systems" (Schatten et al. 2014b) which will be presented at ICALT 2014. Referring to the platform integration, we describe the first sequencer and adaptive content prototype which are integrated in the iTalk2Learn platform as achievement for MS61. In the update of planned for D2.2.1 in M24, we will also describe how we plan to use speech to help the sequencer.

The adaptive support is split into two sub-problems: task-dependent and task-independent support. The task-dependent support aims to provide feedback during the interaction with the ELE Fractions Lab, Maths-Whizz, and Fractions Tutor. The aim of the task-independent support is to use the children's speech to provide feedback on structured and unstructured tasks according to the used mathematical vocabulary as well as to the emotions of the students. In this deliverable we discuss how both types of adaptive support can be realized within the iTalk2Learn platform. Connections with WP1 and WP3 are described in Section 4. Main formative evaluations will be described in D5.2.

Table of Contents

Executive Summary 3

1. Introduction 7

1.1 WP2 within the project 9

2. Performance Prediction 10

2.1 Matrix Factorization 11

2.2 Applying MF to the Whizz Dataset 12

2.3 Future Work 15

3. Sequencer 15

3.1 Content Sequencing in ITS 16

3.2 Simulated Learning Process 17

3.3 Vygotsky Policy 19

3.4 Matrix Factorization as Performance Predictor in sequencing 20

3.5 Test Session 21

3.5.1 Tests on the Simulated Learning Process 21

3.5.2 Sensitivity Analysis on the Vygotsky Policy 22

3.5.3 Vygotsky Policy based Sequencer 22

3.5.4 Advanced Tests 30

3.6 Feasibility discussion 33

3.7 Next steps 34

4. Adaptive Support 34

4.1 Task-dependent support for Fractions Lab 35

4.1.1 Related work....35
4.1.2 Components of the task-dependent support....36
4.2 Task-independent support....38
4.2.1 Mathematical vocabulary....38
4.2.2 Emotional feedback....39
4.2.3 Components of the task-independent support....40
4.2.4 Modelling of vocabulary and emotions....42
4.2.5 Adapting to vocabulary and emotions....43
4.3 Next steps....44
5. Conclusions....45
References....46

List of Abbreviations

BKT - Bayesian Knowledge Tracing

GT – Ground truth

ITS - Intelligent Tutoring System

KC - Knowledge Component

MF - Matrix factorization

RANGE - Range policy

RL - Reinforcement Learning

RMSE - Root Mean Square Error

RND - Random

VP - Vygotsky Policy

VPS - Vygotsky Policy based Sequencer

ZPD - Zone of Proximal Development

1. Introduction

In Intelligent Tutoring Systems (ITS), Artificial Intelligence and Machine Learning are more and more used as decision method. Performance Prediction is one of their main subtasks with the final goal to predict student performances and identify the students' knowledge on specific skills. In adaptive support provisions, instead, one needs to decide what help should be provided and when it should be displayed. Adaptive sequencers take past student performances into account to select the next task which best fits the student's learning needs.

In sequencing as well as in Performance Prediction and Adaptive Support, iTalk2Learn is working to ameliorate state-of-the-art. UHi's work focused in this first year on performance prediction and sequencing methods.

Simple task scheduling is based on fixed sequences decided by a human expert. Adaptive policies, instead, rely on assumptions such as that a student will be able to solve the exercises of the achieved difficulty level but not the more difficult ones without having completed the ones of the previous level. Empirical observations suggest that this can be problematic as it requires students to go through all the topics in the current level even if they can answer them successfully with the first attempt. Although the power-law-of-practice would suggest that students should be provided with several opportunities to practice, unnecessary, i.e. excessive, repetitions can be detrimental in that it can lead to student frustration and influence their perception of the reliability of the system. One approach to the problem is based on assessing the student skills and matching them to the required skills and difficulties of the available tasks. For example, the less known skills by the students are selected to be practiced in the next session. In this scenario two problems arise:

1. Tagging tasks with required skills necessitate experts and thus is time-consuming, costly, and, especially for fine-grained skill levels, also potentially subjective.
2. Learning adaptive sequencing models requires online experiments with real students and specific data collection policies that consist, at the beginning, in many randomly proposed tasks.

Our main goal is to take the state-of-the-art in sequencing educational contents further while achieving a solution that relies on data rather than on expertise and does not put too many requirements on data collection modalities and students' effort. As a consequence, this work allows easier integration of a data-driven sequencer into an already existing ITS supporting our current work at the iTalk2Learn project that aims to incorporate content from different systems into an open architecture.

In "Adaptive Content Sequencing without Domain Information" (Schatten et al. 2014a) we showed how a score prediction method and a simple policy, inspired by Vygotsky's concept of Proximal Development (Vygotsky, 1978), could be used to ameliorate sequencing in a simulated environment.

UHi developed:

1. A content sequencer, the Vygotsky policy based Sequencer (VPS), based on a performance prediction systems that (1) can be set up and preliminary evaluated in a laboratory, (2) models multiple skills and individualization without engineering/authoring effort, (3) adapts to each combination of contents, levels and skills available.

2. A simulated environment with multiple skill contents and students' knowledge representation, where knowledge and performance are modeled in a continuous way.

BBK's work, instead, focused on task-independent and task-dependent support for iTalk2Learn Exploratory Learning Environment (ELE) Fractions Lab.

ELEs are able to support students to discover and understand underlying domain concepts, rather than supporting drill and practice activities to reinforce procedures as typically applied in ITS. Learning performance in an ELE depends on the learner's ability to formulate goals as well as their ability to reflect on the effectiveness of the means of achieving these goals, including planning and carrying out tasks. The task-dependent support aims to provide feedback during interaction with the ELE Fractions Lab. The assistance provided is based on Pólya's reasoning stages (Pólya, 1945).

Within the state-of-the-art literature in ELEs personalized support is only provided during carrying out task phase. Pre-formulated prompts are provided mainly on all the other different reasoning phases. The aim of the task-dependent support is to provide adaptive personalized support at all reasoning phases, based on its student model. The student model, described in Section 4.1, is informed by Wizard of Oz studies described in D5.1.

In order to enable learners to communicate more naturally with the interface, speech recognition for children is integrated into the platform. The aim of the task-independent support is to use the children's speech to provide feedback on structured and unstructured tasks according to: i) the mathematical vocabulary; and ii) emotions.

Using appropriate mathematical terminology is an indication of the student's knowledge. With the detection of not using certain terminology, the task-independent support is able to prompt the student to use the correct terminology and to enhance knowledge.

Emotions play a significant role in learning. While positive emotions can enhance learning, negative emotions can inhibit it. The task-independent support will provide feedback according to the student's emotion via its emotion reasoner. The state-of-the-art literature mainly focuses on how to detect emotions. With the implementation of the emotion reasoner we contribute towards not only how to detect emotions but also on how to adapt support features to emotions effectively (based on Wizard of Oz studies).

1.1 WP2 within the project

Currently, UHi is working on integrating the Vygotksy Sequencer in the Maths-Whizz platform, one of the iTalk2Learn use cases. The domain agnostic recommender allows:

1. A bigger formative evaluation with all Whizz lessons. This experiment would not be possible with other state of the art methods since they require a skill analysis for all 1000 lessons.
2. The possibility to extend the duration of the experiment in case of inconclusive results, which could be due to the reduce amount of interaction with the novel system.
3. The development of the first web service for learning analytics which increases exploitation possibilities after the lifetime of the project.
4. The transfer on Fractions Tutor contents with small effort.

In this Deliverable we are going to show how to apply the so called machine learning method Matrix Factorization for performance prediction to the Whizz dataset.

BBK is working on the implementation of the task-dependent support for Fractions Lab. This support will include a student model, which enables the provision of personalized adaptive feedback tailored to the student's needs.

Additionally, BBK is working on the task-independent support, which will provide feedback on structured as well as unstructured tasks based on children's speech. The task-independent support will use the output from Sail's speech recognition software to detect mathematical terminology as well as emotions. Based on the content of what is detected, the task-independent support will provide appropriate adaptive feedback for use within Fractions Lab as well as Whizz and Fractions Tutor.

2. Performance Prediction

In the following we will shortly summarize state of the art methods for performance prediction. Its most famous example is Bayesian Knowledge Tracing (BKT) and its extensions. The algorithm is built on a given prior knowledge of the students and a data set of binary student performances. It is assumed that there is a hidden state representing the knowledge of a student and an observed state given by the recorded performances. The model learned is composed by slip, guess, learning and not learning probability, which are then used to compute the predicted performances (Corbett & Anderson, 1994). In the BKT extensions also difficulty, multiple skill levels and personalization are taken into account separately (Wang & Heffernan, 2012, Pardos & Heffernan 2010, Pardos & Heffernan 2011, Baker et al. 2008). BKT researchers have discussed the problem of sequencing both in single and in multiple skill environments (Koedinger et al. 2011). In a single skill environment the most not mastered skill is selected, whereas in a multiple skill environment this behavior would present a too difficult content sequence. Consequently, the contents with a small number of not mastered skills are selected. Moreover, Koedinger et al. (2011) point out how in ITS multiple skill exercises are modeled as single skill ones in order to overcome BKT limitations. We would like to stress that the sequencing requires an internal skills representation and consequently, together with the performance prediction algorithm, is domain dependent.

Another domain dependent algorithm used for performance prediction is the Performance Factors Analysis (PFM). In the latter the probability of learning is computed using the previous number of failures and successes, i.e. the representation of score is binary like in BKT (Pavlik et al. 2009). Moreover, similarly to BKT, a table connecting contents and skills is required.

UHi instead selected Matrix Factorization (MF) for performance prediction, which will be described in the next sections. There are several reasons to choose MF:

- Domain independence: ability to model each skill, i.e. no engineering/authoring effort in individuating the skills involved in the contents.
- Thai-Nghe et al. (2012) presented promising results using MF that were comparable with the state of the art ones (i.e., BKT or PFM).
- Possibility to build the system with a common data set, i.e. without an exploratory corpus.
- Small computational time on a 3rd Gen Ci5/4GB laptop and Java implementation: 0.43 s for building the model with already 122000 lines, negligible time for performance prediction.

2.1 Matrix Factorization

As mentioned above, we used MF as predictor. Generally, it predicts, which are the future user ratings on a specific item according to his previous ratings and the previous ratings of other users. The concept has been extended to student performance prediction, where a student's next performance, or score, is predicted. The matrix $Y \in \mathbb{R}^{n_s \times n_c}$ can be seen as a table of n_c total contents and n_s students used to train the system, where for some contents and students performance measures are given. MF decomposes the matrix Y in two other ones $\Psi \in \mathbb{R}^{n_c \times P}$ and $\Phi \in \mathbb{R}^{n_s \times P}$, so that $Y \approx \hat{Y} = \Psi\Phi^T$. Ψ and Φ are matrices of latent features. Their elements are learned with gradient descend from the given performances. This allows computing the missing elements of Y for each student i in each task j of a dataset D (Fig. 1). The optimization function is represented by:

$$\min_{\psi_j, \varphi_i} \sum_{i \in \mathbb{S}, j \in \mathbb{C}} (y_{ij} - \hat{y}_{ij})^2 + \lambda(\|\Psi\|^2 + \|\Phi\|^2) \tag{1}$$

where one wants to minimize the regularized squared error on the set of known scores.

The prediction function is represented by:

$$\hat{y}_{ij} = \mu + \mu_{cj} + \mu_{si} + \sum_{p=0}^{P} \varphi_{ip}^T \psi_{jp} \tag{2}$$

where μ, μ_c, and μ_s are respectively the average performance of all contents of all students, the learned average performance of a content, and learned average performance of a student. The two last mentioned parameters are also learned with the gradient descend algorithm.

The MF problem does not deal with time, i.e. all the training performances are considered equally. In order to keep the model up to date, it is necessary to re-train the model at specific time steps. MF has a personalized prediction, i.e. a small number of exercises need to be shown to each student in order to avoid the so called cold-start problem. Although a solution to these problems has been proposed in Schatten et al. (2014a), in the following section we discuss the practical approach performing a feasibility analysis for the predictor and showing how we employ it in the real case.

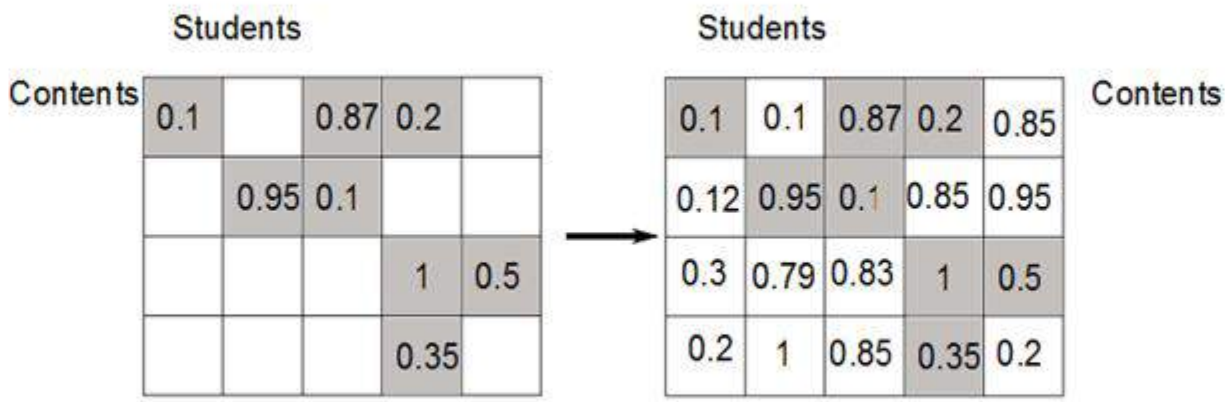

Figure 1: Computing of the missing values by means of MF. Students received a score on specific tasks, represented by the gray squares, and MF predicts the missing scores they could receive in unforeseen ones. The task, from a machine learning perspective, is similar to the recommendation problem. A user rates specific items and this information is used to predict the ratings on other items.

2.2 Applying MF to the Whizz Dataset

In this section we discuss how the MF can be applied to the Whizz dataset which has over 1000 lessons in 20 topics and was adapted to be used in several countries like United Kingdom, USA, and Russia. We performed a practical feasibility study using a dataset that is composed by data collected from children from five to fourteen years using the ITS in classrooms and at home. A lesson is composed of test and exercise sessions. The exercise session consists of approximately 10 exercises on a topic and specific learning objectives. While trying to solve those exercises a student can consult several hints, one of those is the bottom-out hint, which displays the solution. In order to pass the exercise a student must achieve a score of 7 out of 10. Only if he passes the exercise he can go to the test session. There he will have to show what he learned answering 5 questions with a score greater than 6 out of 10. The lesson sequencing policy relies on the assumption that a student will be able to solve the exercises of the achieved difficulty level but not the more difficult ones without having completed all the lessons of the previous level.

In contrast to state of the art performance prediction, where the main task is to predict the student's correct at first attempt answer, the commercial system uses the score as student's performance measurement. The data granularity level is low if compared with benchmark systems[1], since we possess a single score record for the 10 questions of the exercises and one record for the five test questions. Consequently, a multiple skill representation of the lessons would be the most plausible. Nevertheless, this information is not available for all the lessons, which are summarised with a single learning objective description.

Consequently, some preprocessing of the dataset was required.

In order to avoid sparseness in benchmark datasets (Thai-Nghe 2011) each line is abstracted to the Knowledge Component (KC) level, i.e. the algorithm predicts if the student is going to answer correctly to a specific KC or to a specific step. Since we did not have this information, we undertook two different preprocessing approaches for testing purposes: in the first one the algorithm predicts the score on the single lessons, whereas in the second we predict the score on a specific topic. We then distinguish, at each abstraction level, if the lesson was solved in exercise or test mode. This was done, since the use of hints strongly influences the outcome of the exercise session and modifies the experiment modalities. Moreover, we removed the skipped lessons in order to have only completed ones. We followed the standard approach in the field to divide the dataset temporally in two parts. The first two thirds, called the training dataset, were used for learning the model and the last third for testing it. The performances of the prediction methods are evaluated with the Root Mean Square Error (RMSE). The score, as in Schatten et al. (2014a), is represented in a continuous interval which goes from zero to one. Further dataset information are summarized in Table 2.

The average score obtained by the students in the training session is 8.1 out of 10. We used this value to have an indication of worst case performances of a prediction method called Global Average (see Table 1). The latter assumes that the student will always perform equally to the global score average computed on the training dataset. The Biased User-Item predictor, instead, uses only the biases μ, μ_c, and μ_s of Eq. (2), i.e. the latent features number P is set to zero. Consequently, Table 1 displays the contribution of the single components of Eq. (2) in ameliorating the performance prediction. According to the results, MF is able to predict a continuous interval performance in a multiple-topic and curricula scenario. This is different to what was done in Thai-Nghe et al. (2011), where the main task was to predict if the student was going to answer correctly at first attempt.

[1] pslc-datashop.web.cmu.edu/KDDCup

Experiment	RMSE
Global average	0.3032796
Biased User-Item Exercise	$0.2639167 \pm 3.6989 \ 10^{-5}$
Biased User-Item Topic	$0.26416832 \pm 3.36935 \quad 10^{-5}$
Topic Preprocessing	$0.260664942 \pm 7.77042 \quad 10^{-5}$
Exercise Preprocessing	$0.26061115 \pm 5.97504 \quad 10^{-5}$

Table 1: Results on the commercial dataset with standard deviation over five experiments and score normalized between 0 and 1.

Number of Items (Exercise/Topic)	9091/4169
Number of Students	258391
Total Student-Item Interactions	30813070
Total Exercise sessions	17512972
Exercise passed (Score 70-99)	9520278, i.e. 54%
Gaming the system (Score 100 + Bottom-out hint)	3988891, i.e. 23%
Total Test sessions	13300098
Test session passed (Score 60-99)	4378461, i.e. 33%
Average score obtained	8.1

Table 2: Statistics of the Dataset.

2.3 Future Work

Domain agnostic performance prediction methods are an advantage in designing novel systems and for integration in already existing Intelligent Tutoring System. Unfortunately, this causes a loss in interpretability of the skills of the student retrieved by the MF. In the following project months we will investigate how to modify the algorithm in order to increase the explicit physical meaning of the MF latent features and to be able to use MF also as user modeling system.

3. Sequencer

Many Machine Learning techniques have been used to ameliorate ITS, especially in order to extend learning potential for students and reduce engineering efforts for designing the ITS.

The most used technology for sequencing, as stated in D2.1. is Reinforcement Learning (RL), which computes the best sequence trying to maximize a previously defined reward function. Both model-free and model-based RL (Malpani et al. 2011, Beck et al. 2000) were tested for content sequencing.

Unfortunately, the model-based RL requires a special kind of data set called exploratory corpus. Such an exploratory corpus is not available for the ITS used in iTalk2Learn. Available are log files of ITS which have a fixed sequencing policy that teachers designed to grant learning. They explore a small part of the state - action space and yield to biased or limited information. For instance, since a novice student will never see an exercise of expert level, it is impossible to retrieve the probability of a novice student solving some tasks. Without these probabilities the RL model cannot be built (Chi et al. 2011). Model-free RL, instead, assumes a high availability of students on which one can perform an on-line training. The model does not require an exploratory corpus but needs to be built while the users are playing with the designed system. Given the high cost of an experiment with humans, most authors exploit simulated single skill students based on different technologies like Artificial Neural Networks or self developed student models (Sarma et al. 2007, Malpani et al. 2011). Particularly similar to our approach is (Malpani et al. 2011), where contents are sequenced with a particular model-free RL based on the actor critic algorithm (Konda et al, 2000), which was selected because of its faster convergence in comparison with the classic Q-Learning algorithm (Sutton et al. 1998). Unfortunately, RL algorithms still need many episodes to converge and will always need preliminary trainings on simulated students. Having a RL model-free algorithm also requires novel experiments.

In order to avoid RL problems we suggested a content sequencer based on student performance prediction that could be set up with the dataset available from the Whizz System. In conformity with the DOW, the RL tasks will be done using data collections from the project following the requirements of D2.1 and without frustrating the user. In particular we will exploit the data collection for the formative study of the German FT version to collect a more RL suitable dataset. How the variety of sequences will be created without frustrating the students with random sequences is discussed in D1.3.

3.1 Content Sequencing in ITS

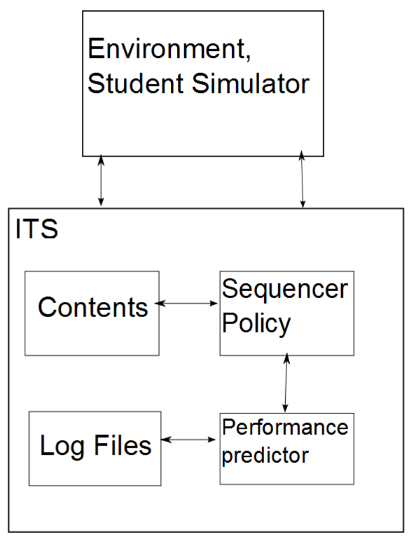

Figure 2: System structure in a block diagram.

The designed testing system consists of two main blocks. The first one is the environment are the students working with the ITS. In our case an on-line evaluation is required, i.e. the sequence optimality can be measured only after a student worked with it. However, in order to allow children interaction with the system one requires a certain degree of confidence in order not to frustrate the students. For the same reason we excluded the possibility of collecting an exploratory corpus because making practice with very easy and very difficult exercises in random order. Consequently we designed a simulated learning environment to perform on-line test in a first study phase. After validation with real students in a second study phase, only a common data set collection will be necessary to set up the system with new contents, giving also the possibility to calibrate the environment and later use it for new sequencing methods. The simulated environment will be presented in section 3.2.

The second block consists of different modules, i.e. the available contents, the previous interactions of the students with the system (log files), the student Performance Predictor and the Sequencer Policy. We chose a specific Performance Predictor and Policy, but this selection does not prevent us from using other ones in the future. The performance predictor was presented in chapter 2 and the Policy will be presented in section 3.3.

When a student interacts with the system the next exercise is proposed to him by the sequencer according to a policy. The Performance Predictor needs the log files of students working on the contents considered to predict their scores in the next contents. The policy is applied in an adaptive way thanks to the information on the predicted scores shared between Performance Predictor and Sequencer.

This architecture was used as a testing environment. How the Sequencer is integrated within the iTalk2Learn platform is explained in D4.2.1 and in Schatten (2014b). The latter also presents a more detailed discussion about the requirements for a lightweight machine learning integration, which became crucial to run a large scale experiment within the Maths-Whizz platform.

3.2 Simulated Learning Process

We designed a simulated student based on the following assumptions.

(1) A content is either of the adequate difficulty for a student, or too easy, or too difficult. (2) A student cannot learn from too easy contents and learns from difficult ones proportionally to his knowledge level. (3) It is impossible to learn from a content more than the required skills to solve it. (4) The total knowledge at the beginning is different than zero. (5) The general knowledge of connected skills helps solving and learning from a content. The last assumption is plausible because we assume to sequence activities of the same domain. For instance, in order to solve a fraction addition, a student needs more related skills: subtraction and equivalence. It is unlikely for a student to do a fraction expansion without knowing how multiplication works. At the same time the knowledge of multiplication will help him solving the steps on fraction expansion.

A student simulator is a tuple (S, C, y, τ) where, given a set $S \subseteq [0,1]^K$ of students, s_i is a specific student described as a vector $\boldsymbol{\varphi}^t$. The latter is of dimension K, where K is the number of skills involved. $C \subseteq [0,1]^K$ is a set of contents, where ${c_j}^t$ is the *j*-th content, defined with a vector $\boldsymbol{\psi}_j$ of K elements representing the skills required. $\varphi_{ik} = 0$ means student i has no knowledge from skill k, whereas $\varphi_{ik} = 1$ means having full knowledge. τ is a function defining the follow-up state $\boldsymbol{\varphi}^{t+1} = \boldsymbol{\varphi}^t + \tau$ of a student s_i after working on contents ${c_j}^t$. Finally, a function y defines the performance $y(\varphi_i, \psi_j)$.

y and τ can be formalized as follows:

$$y(\varphi_i, \psi_j) := \max(1 - \frac{||\alpha||}{||\varphi_i||}, 0)$$

$$\tau(\varphi_i, \psi_j)_k := y(\varphi_{ik}, \psi_{jk})\alpha_k$$

$$\tilde{y} := y\varepsilon$$

$$\alpha_k^{i,j} = \max(\psi_{jk} - \varphi_{ik}, 0) \tag{4}$$

and ε is proportional to the beta distribution *B(p,q)*. We selected p and q in order to have $\tilde{y} \sim B(y, \sigma^2)$, where σ^2 is the variance, i.e. the amount of noise. We chose the beta distribution because it is defined between zero and one as the score. Consequently, it will not change the codomain of the y function. The characteristic of the formulas are the following: (1) The performance of a student on a content decreases proportionally to his skill deficiencies w.r.t. the required skills. (2) The student will improve all the required skills of a content proportionally to his performance and his skill-specific deficiency up to the skill level a content requires. (3) As a consequence it is not possible to learn from a content more than the difference from the required and possessed skills. (4) A further property of this model is that contents requiring twice the skills level that a student has, i.e. $2\|\boldsymbol{\varphi_i}\| \leq \|\boldsymbol{\psi}_j\|$, are beyond the reach of a student. For this reason his performance will be zero (y=0).

With a simple experiment without noise, we can show the plausibility of the designed simulator. We inserted values in Eqs. (4) as follows. Let us consider a system with two skills and represent the student knowledge as $\boldsymbol{\varphi}$= [0.3,0.5].

$c_j{}^t$	d_{c_j}	Y	τ_k
[0.1,0.1]	0.2	1	[0,0]
[0.5, 0.6]	1.1	0.617	[0.12, 0.0617]
[0.5, 0.7]	1.2	0.515	[0.1, 0.1]
[0.9,0.9]	1.8	0	[0,0]

Table 3: Simulated learning process with two skills. A simulated student with $\boldsymbol{\varphi}$= [0.3,0.5], scores y and learning τ after interacting with different contents $c_j{}^t$.

As Table 3 displays that when the content difficulty increases the learning increases and the score decreases until $2\|\boldsymbol{\varphi}_i\| \leq \|\boldsymbol{\psi}_j\|$. The maximal difficulty level is equal to the number of skills since a single skill value cannot be greater than one.

3.3 Vygotsky Policy

The designed sequencer is defined as follows. Let C and S be respectively a set of contents and students as defined in Section 3.2, d_{c_j} be the difficulty of a content defined as $d_{c_j} = \sum_{k=0}^{K} \psi_{jk}$, $\tilde{y}$ be the performance or the score of a student working on the content, and T be the number of time steps assuming that the student is seeing one content every time step.

The content sequencing problem consists in finding a policy π^* that maximize the learning of a student within a given time T without any environment knowledge, i.e. without knowing the difficulties of the contents and the required skills to solve them.

A common problem in designing a policy for ITS is retrieving the knowledge of the student from the given information, e.g. score, time needed, previous exercises, etc. The previous mentioned data types are just an indirect representation of the knowledge, which cannot be automatically measured, but needs to be modeled inside the system. Hence, integrating the curriculum and skills structure is the cause of the high costs in designing the sequencer.

In this paper we try to keep the contents in the Vygotsky's Zone of Proximal Development (ZPD)

(Vygotsky, 1978), i.e. the area that students can solve with some support. In other words, the contents is neither very easy for the student nor is it too hard to be solved even when support is given. We mathematically formalized the concept with the following policy, that we called Vygotsky Policy (VP):

$$c^{t*} = arg\, \min_c \left\| y_{th} - \widetilde{\widehat{y}}^{t}\, (c) \right\| \tag{5}$$

where y_{th} is the threshold score, i.e. the score that keeps the contents in the ZPD. The policy will select at each time step the content with the predicted score $\widetilde{\widehat{y}}^{t}$ at time t most similar to y_{th}. We will discuss further in the experiment session how to tune this hyper parameter and its meaning.

The peculiarity of the VP is the absence of the difficulty concept. Defining the difficulty for a content in a simulated environment as ours is easy, because we mathematically define the skills required. In the real case it is not trivial and quite subjective. Also the required skills are considered as given in the other state of the art methods like PFM and BKT, where a table represents the connection between contents and skills required. Without skills information not only BKT and PFM performance prediction cannot be used in our formalization, also sequencing methods (Koedinger et al. 2011) have no information to work with.

3.4 Matrix Factorization as Performance Predictor in sequencing

As mentioned above, the MF problem does not deal with time, i.e. all the training performances are considered equally and in order to keep the model up to date, it is necessary to re-train the model at each time step. Furthermore, MF has a personalized prediction, which means that in order to avoid the so called cold-start problem a small number of exercises need to be shown to each student. Although some solutions to these problems have been proposed (Thai-Nghe et al., 2011, Krohn-Grimberghe et al. 2011), we will show in the experiment session that these aspects do not affect the performance of the system, neither they reduce its applicability.

From now on we will call the sequencer utilizing the VP policy and the MF performance predictor Vygotsky Policy based Sequencer (VPS).

3.5 Test Session

In this section we show how the single elements work in detail. We start with the student simulator, continue with the VP and end with some experiments with performance prediction in different scenarios and noise. A scenario is represented by a number of contents n_c, a number of difficulty levels n_d, a number of skills n_k, and a number of students for each group n_t[2]. All the first experiments will have no noise, i.e. $\tilde{y} = y$.

3.5.1 Tests on the Simulated Learning Process

To prove the operating principle of the simulator we tested basic sequencing methods in a particular scenario. The one we chose is described in Fig. 3, with n_d=7 and n_c=100. For representation purposes we created the contents with increasing difficulty, so that IDs implicitly indicates the level of difficulty[3]. The scenario mimics an interesting situation for sequencing, i.e. when more apparently equivalent exercises are available. The two policies we used are (1) Random (RND), where contents are selected randomly, and (2) the in range policy (RANGE), where each second content is selected in difficulty order. This strategy is informed on the domain because it knows the difficulty of the contents. We initialized the students and contents skills with a uniform random distribution between 0 and 1. Again for representation purposes we show the average total knowledge of the students that is represented by average of the students' skills sum at each time step. We chose to perform the tests on 10 skills, i.e. the maximal total knowledge possible is equal to 10. We considered the scenario mastered when the total knowledge of the student group is greater than or equal to the 95% of the maximal total knowledge.

Fig. 4 shows the total knowledge of two groups of n_t=200 students, one group was trained with random policy the other one with the in range policy. RANGE is characterized by a low variance in the learning process. RND, instead, has a high variance because the knowledge level of the students at each time step is given by chance. It is shown that the order in which the student practices on the contents is important for the total final learning. Fig. 4 also shows how the practice on too many contents of the same difficulty level, after a while, saturates the knowledge acquisition. This is coherent with the learning process of procedural tasks. As also reported in D1.3, students should be

[2] The MF was previously trained with n_s students that were used to learn the characteristic of the contents. Consequently, the dimensions of the MF during the simulated learning process are: $\Psi \in \mathbb{R}^{n_c \times P}$ and $\Phi \in \mathbb{R}^{n_s \times P}$, so that $Y \approx \hat{\tilde{Y}} = \Psi\Phi^T$

[3] E.g., a content with ID 2 is easier than a content with ID 100, see Fig. 3.

provided with more than a single structured practice task, because students need more practice to become fluent in the application of the problem-solving procedure (Newell & Rosenbloom, 1981), but, as stated by Rohrer and Taylor (2006), showing nine in comparison to three practice problems only lead to very little benefit.

During CSEDU we had a face to face discussion with Carniage and Mellon University Senior Researcher Bruce McLaren. From the discussion it emerged that student simulators are of great interest due to the difficulties in finding students for the experiments and preliminary assessments would be helpful.

3.5.2 Sensitivity Analysis on the Vygotsky Policy

In order to evaluate the VP we created two more sequencing methods that exploit information not available in reality.

The best sequencing knows exactly which is the content maximizing the learning for a student, for this reason we called it Ground Truth (GT). Vygotsky Policy Sequencer Ground Truth (VPSGT), instead, uses the Vygotsky Policy and the true score y of a student to select the following content. GT and VPSGT can be considered the upper bound of the sequencer potential in a scenario. In order to select the correct value of y_{th}we plot the average knowledge level at time *t=11* for the policy with different y_{th}. From the relationship between Eqs. (4) of the student simulator displayed in Fig. 6 one can see that the policy is working for $y_{th} \in [0.4,0.7]$. In a real environment the interpretation of these results is twofold. First we assume y_{th} will be approximately the score keeping the students in the ZDP. Second, from a RL perspective, this value would allow finding the trade-off between exploring new concepts and exploiting the already possessed knowledge. Moreover, as shown in Fig. 5, the policy obtains good results if compared with GT for some y_{th}, but for others the policy is outside the ZPD and the students do not reach the total knowledge of the scenario. In some experiments we noticed that the width of the curve in Fig. 6 decreased so that the outer limits of the y_{th} interval create a sequence outside the ZPD. As consequence we selected the value y_{th}=0.5 that was successful in most of the scenarios.

3.5.3 Vygotsky Policy based Sequencer

The scenario we selected for the tests with the VPS has n_c=200, n_d=6, n_k =10 and n_t=400. In order to train the MF-model a training and test data set need to be created. We used n_s=300 students who learned with all the contents in order of difficulty. We used 66% of the data to train the MF model and the remaining 34% to evaluate the Root Mean Squared Error (RMSE) for selecting the regularization factor λ and the learning rate of the gradient descent algorithm. We performed a full Grid Search and selected the parameters shown in Tab. 4. The sequencing experiments are done on a separate group of n_t students. In order to avoid the cold start problem 5 contents are shown to them and their scores added to the training set of the MF. For *T=40* the best content ${c_j}^{*t}$ is selected with the policy VP for the n_t students, using the predicted performance $\hat{\tilde{y}}_{ij}{}^{t}$. In order to avoid the deterioration of the model, after each time step the model is trained again once all students saw an exercise. A detailed description of the algorithm of the sequencer can be found in Alg. 1, where Y_0 is the initial data set.

Parameters	**Choise**
Learning rate	0.01
Latent features	60
Regularization	0.02
Number of iterations	10

Table 4: Parameters MF.

As one can see in Fig. 7 the VPS selects the first content similarly to RANGE. Then the prediction allows skipping unnecessary contents speeding up the learning. Once the total knowledge arrives around 95%, the selection policy cannot find contents that fit to the requirements. Consequently the students learn as slow as the RND group, as one can see from the saturating curve. In Fig. 8 GT selects the contents in difficulty order skipping the unnecessary ones. The average sequence of the VPS, instead, is also with approximately increasing difficulty but in an irregular way. This is due to the error in the prediction performance. In conclusion the proposed sequencer gains 63% over RANGE and 150% over RND.

Algorithm 1: Vygotski Policy based Sequencer

Input: $\mathbb{C}$, Y_0 π, s_i, T

Train the MF using Y_0;
for t = *1 to T* **do**
    for *All* $c \in \mathbb{C}$ **do**
        Predict $\hat{y}(c_j, s_i)$ Eq. 6;
    end
    Find c^{t*} according to Eq. 5;
    Show c^{t*} to s_i with Eq. 1;
    Add $y(s_i, c^{t*})$ to Y_t;
    Retrain the MF; // Corrects over- or underestimation by the MF
end

Algorithm 1: Vygotsky Policy based Sequencer.

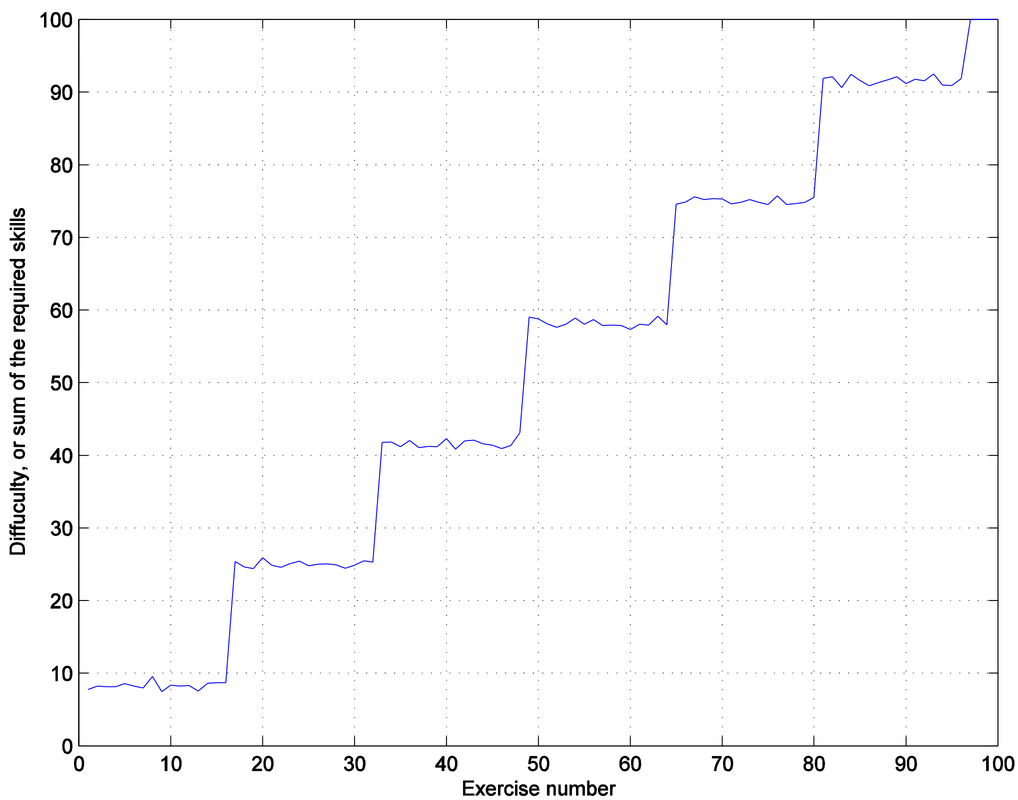

Figure 3: Scenario, Content Number and difficulty level.

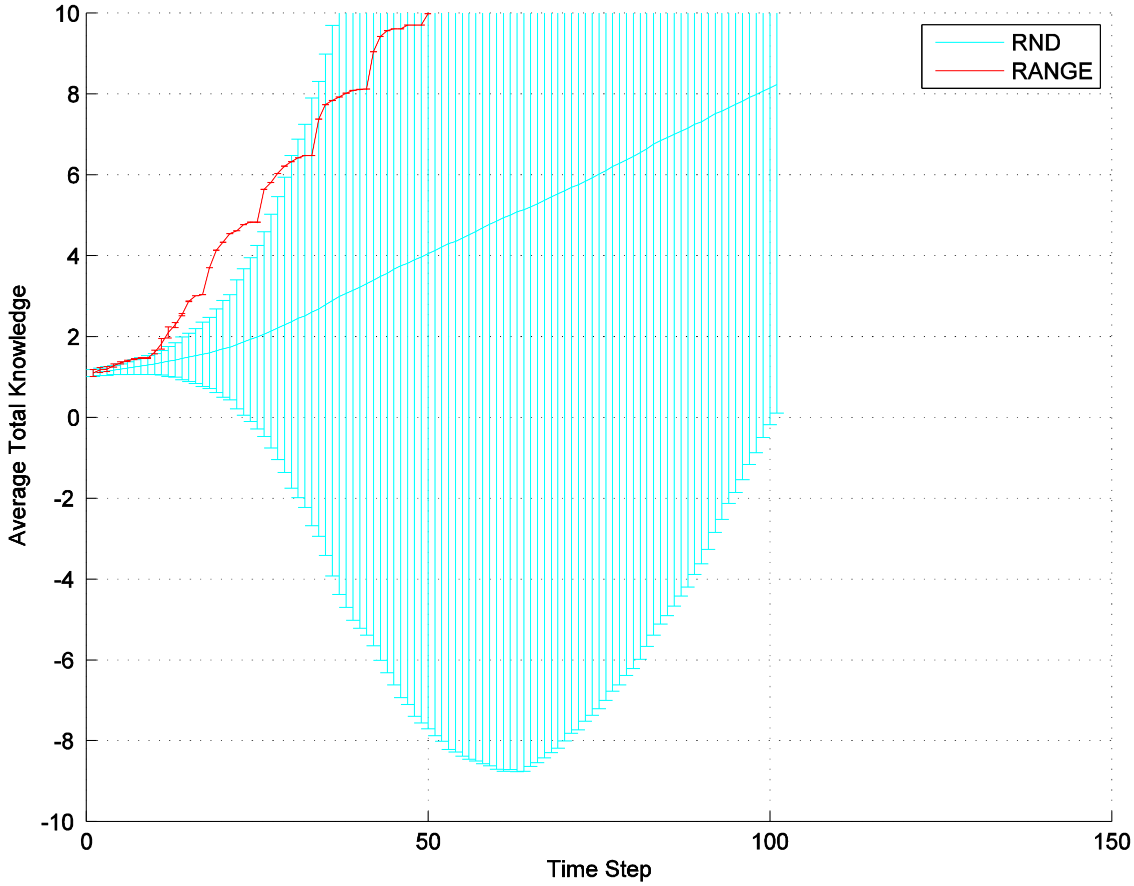

Figure 4: Comparison between RANGE and RND. Average skills sum, i.e. knowledge, over all the students with variance.

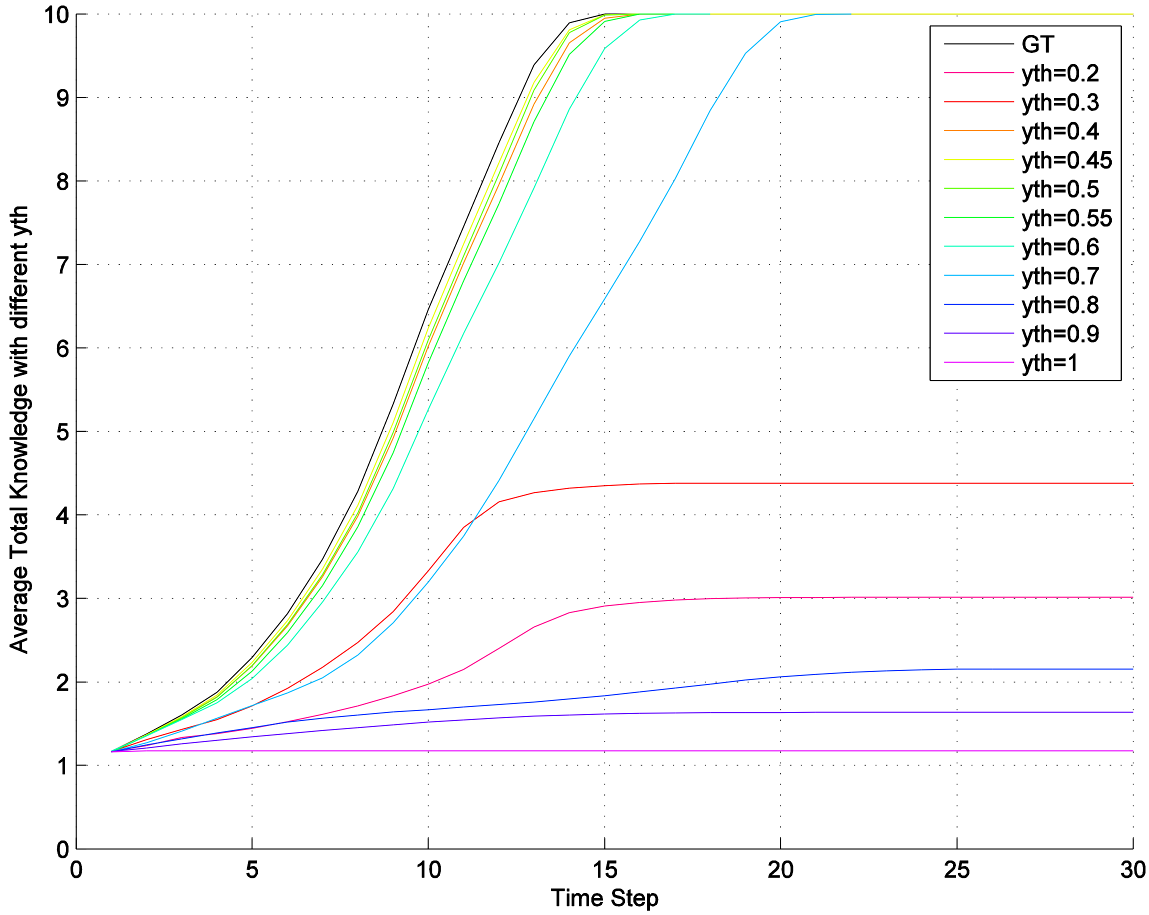

Figure 5: Effects of the different y_{th} on the final knowledge of the students. The learning curves of the student groups that learned with the different Vygotsky policies.

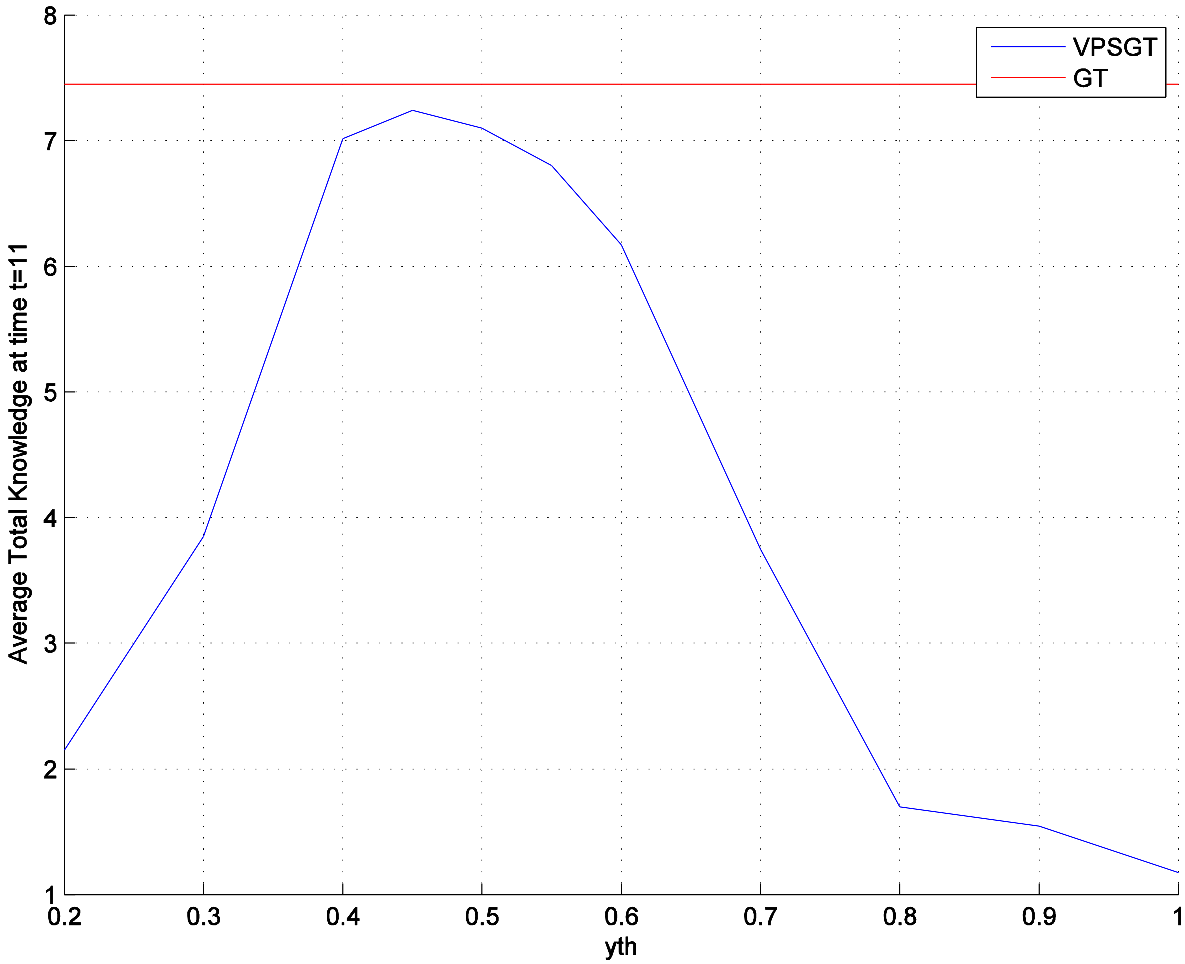

Figure 6: Policy selection, i.e. the performance of the Vygotsky policy with different y_{th} at the same time step. Different groups of students learned with the Vygotsky policy with y_{th} values going from 0.1 to 0.9.

Policy	Description
Random (RND)	Contents are selected randomly
In Range (RANGE)	Each second content is selected
Ground Truth (GT)	Selects the contents according to which is the one maximizing the learning
Vygotsky Policy based Sequencer Ground Truth (VPSGT)	Chooses the next content using the policy and the real score of a student
Vygotsky Policy based Sequencer (VPS)	Chooses the next content using the policy and the predicted score of a student

Table 5: Tested sequencers.

The presented experiments show how the MF is able, without domain information, to model the different skills of students and contents and partially mimics the best sequence, which is the one selected by GT in Fig. 8.

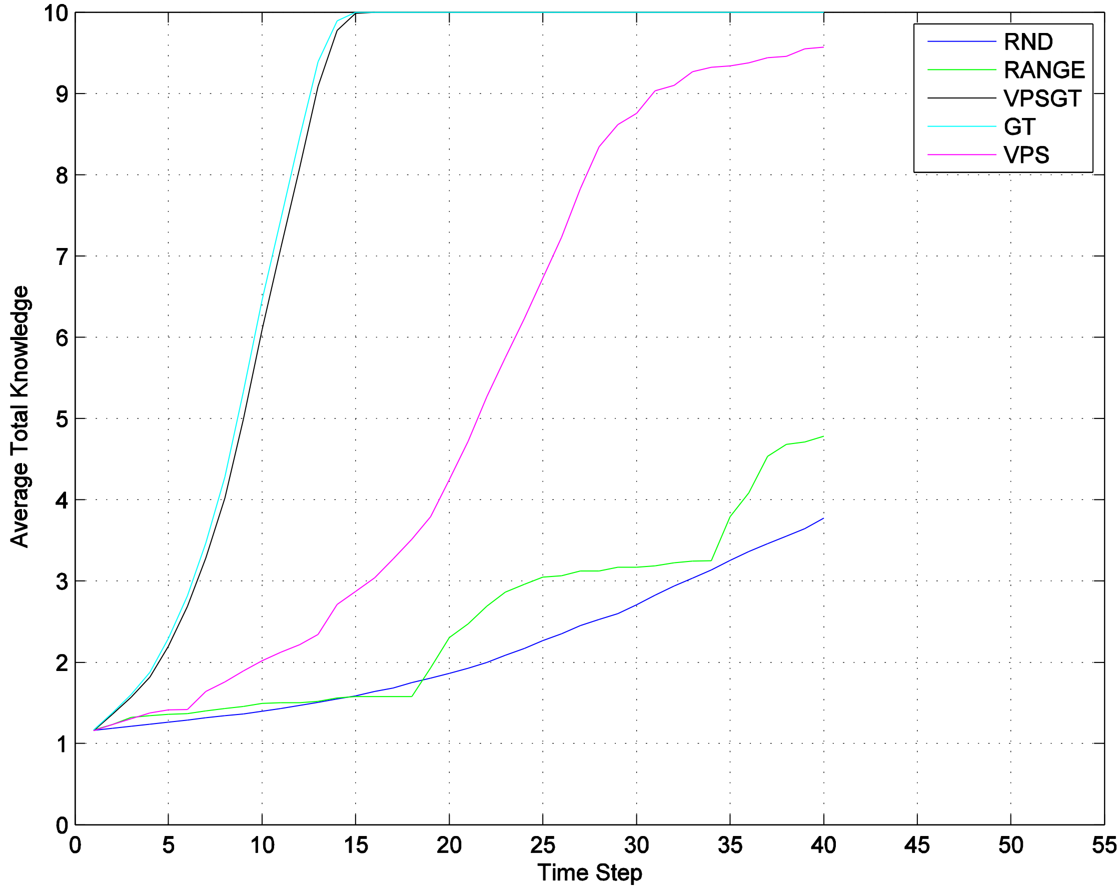

Figure 7: Average Total Knowledge. How the average learning curve of the students changes over time.

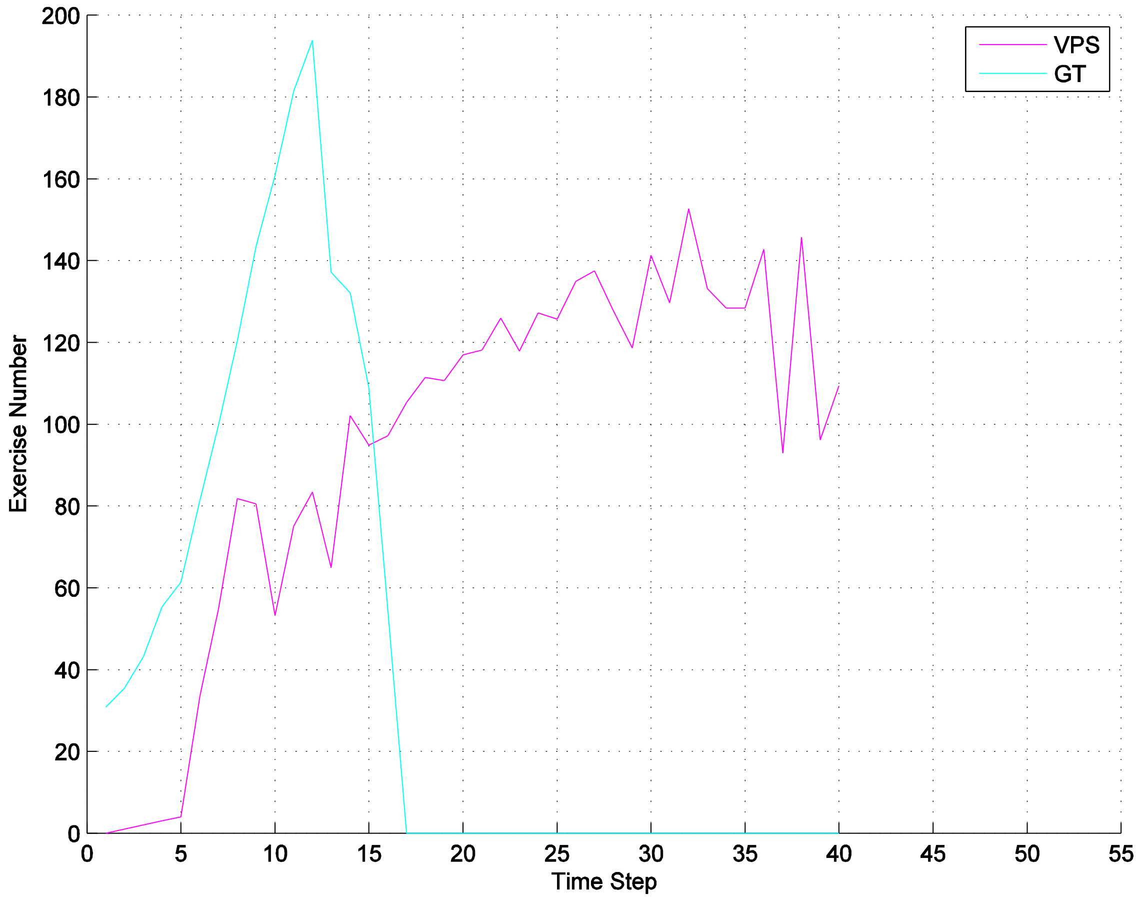

Figure 8: Average sequence selected by the GT and the VPS. The VPS approximate the optimal sequence that GT computes thanks to the real skills of the students.

3.5.4 Advanced Tests

In this section we want to show the correct working of the sequencer changing the parameters of the scenario n_k and n_c and later adding noise.

In order to do so we consider the percentage of gain of VPS with respect to RANGE considering a specific time step *t=30* with n_k=10 and n_d=6. As one can see in Fig. 10 the gain obtained by the sequencer depends on the available number of contents. Since in RANGE each second content is selected, with n_c < 60 there are not enough contents for all time steps. Our sequencer can adapt without problems to the situation. The optimal point for the in range policy is when n_c = 60 because there is exactly the necessary number of contents for the student to learn. When n_c > 60 the students see many unnecessary contents and consequently learn slower. Fig. 9 with n_c=60, *t=30* and n_d=6 shows the dependencies between skills and gain. The experiments demonstrated a high adaptability of the sequencer to the different scenarios.

Lastly, we experimented the results robustness adding noise, i.e. $\tilde{y} = y\epsilon$. We experimented with $\sigma^2 \in [0,0.5]$. As one can see in Fig. 11 with σ^2=0.1 the Vygotsky sequencers are still able to produce a correct learning sequence but more time is required. The VPSGT is the one that suffered the most from the introduction of noise, probably related to the selection of y_{th}.

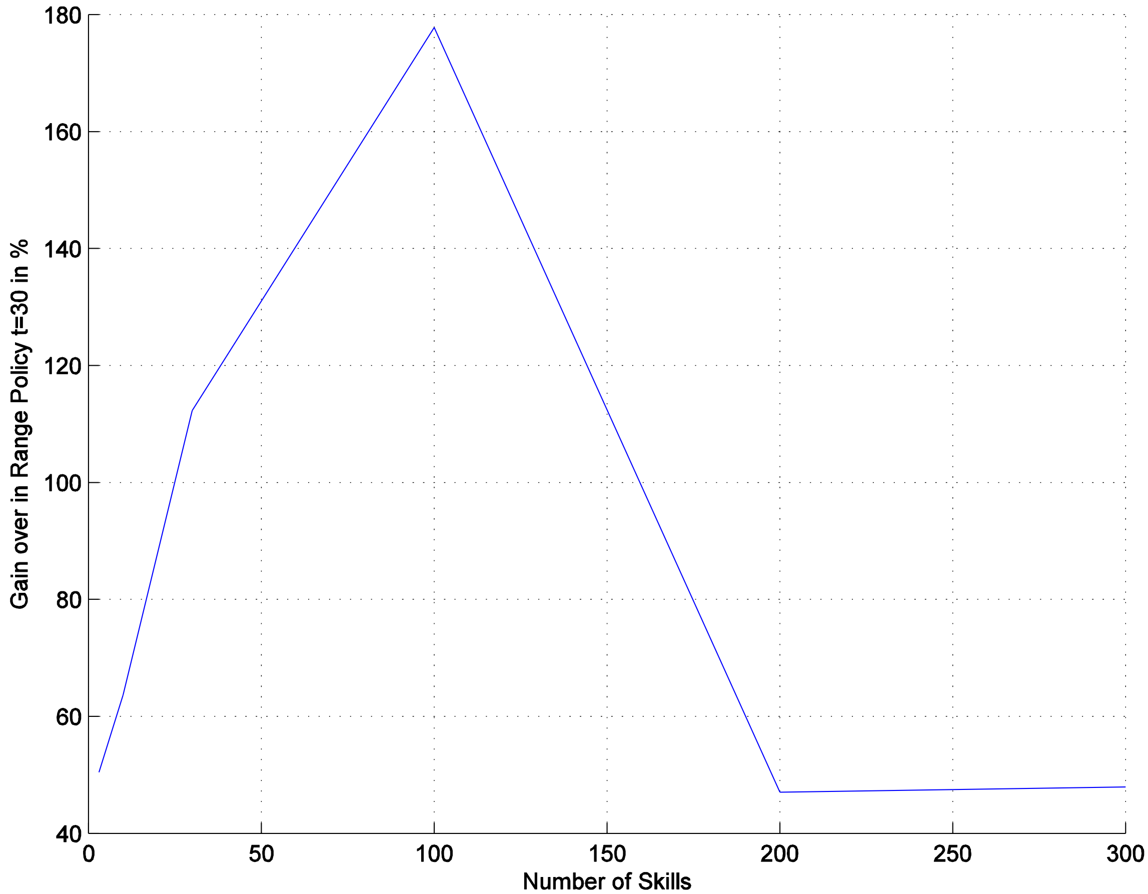

Figure 9: Gain over RANGE policy varying n_k. The gain is measured at a specific time step in percentage, considering the average knowledge level of the two groups of students, one practicing with the RANGE sequencer and one with the VPS.

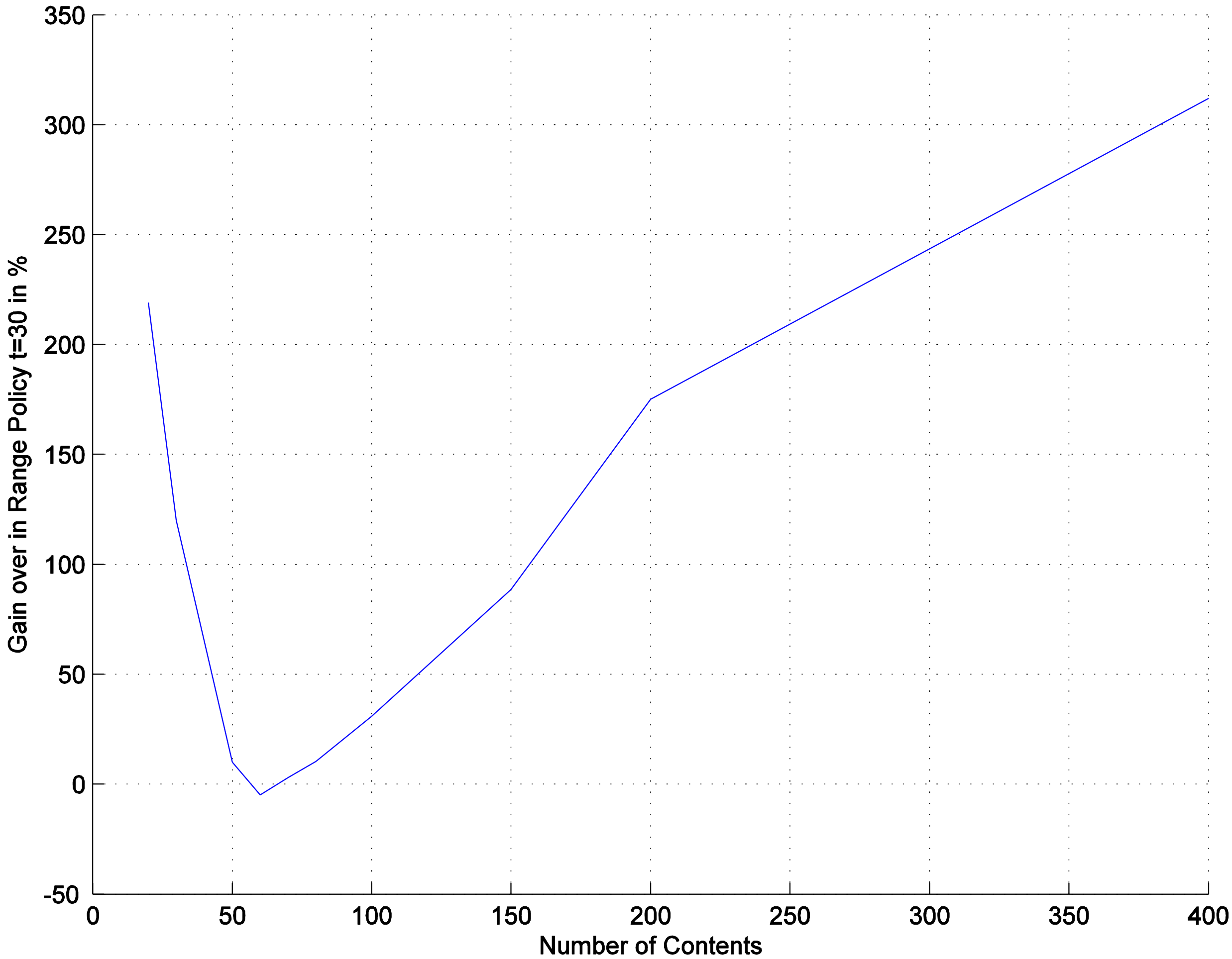

Figure 10: Gain over RANGE policy varying n_c. The gain is measured at a specific time step in percentage, considering the average knowledge level of the two groups of students, one practicing with the RANGE sequencer and one with the VPS.

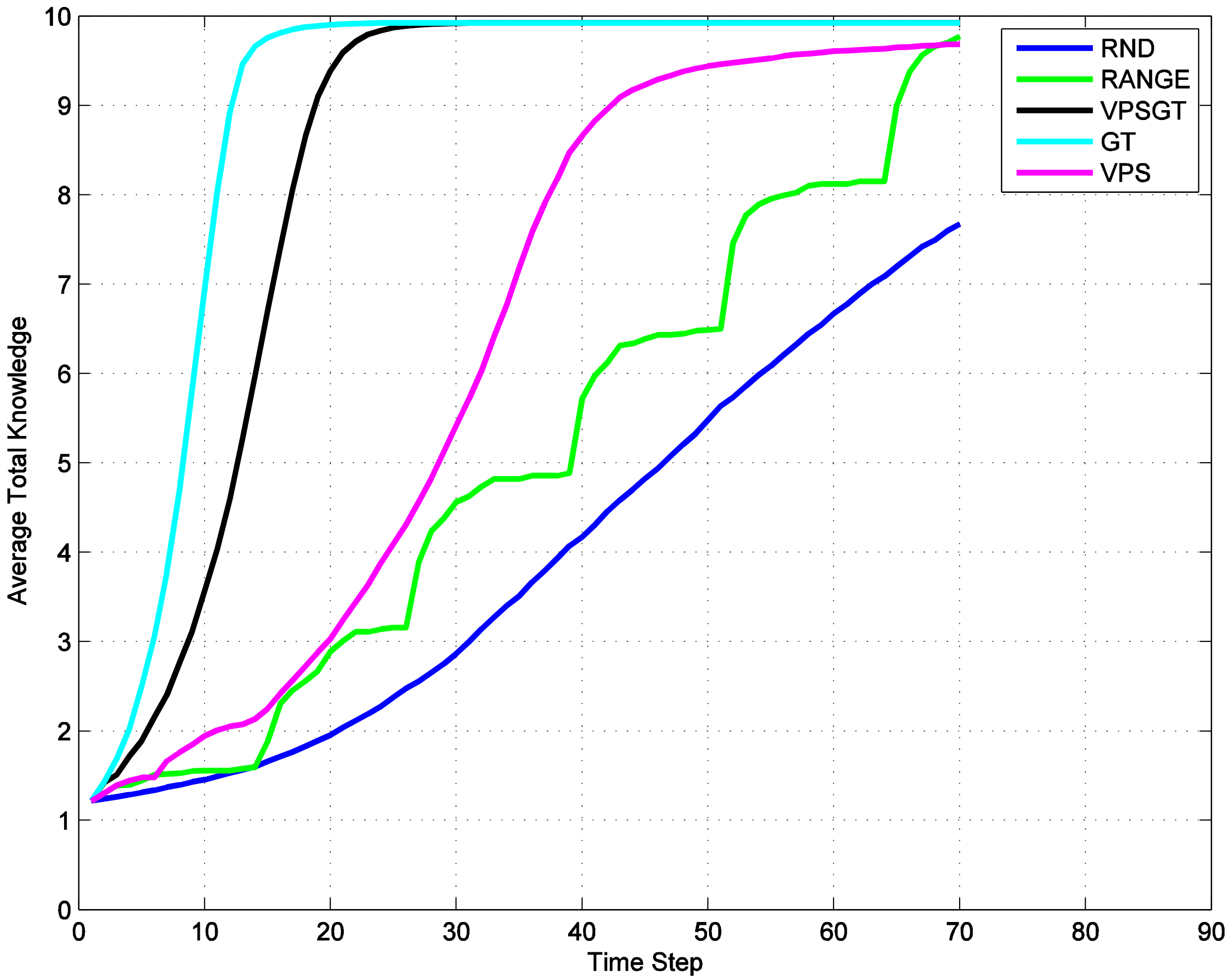

Figure 11: Effect of noise in the simulated learning process. Beta distribution noise with σ^2=0.1

3.6 Feasibility discussion

In Schatten et al. (2014a) we discussed how to tune the threshold score parameter on simulated students. Here, instead, we would like to discuss how this value should be selected, for instance within the commercial system earlier described, and address some of the open issues with respect of its usage in a realistic scenario. Considering a score range for passing of 6 to 10 out of 10, y_{th} should be set in the middle of the interval, so that the most exercise selected are predicted with a score of 8. This avoids that in case of no available tasks predicted with exactly y_{th} the policy does not select exercises which are out of the score range for passing. Moreover, it minimizes the risk of MF incorrect prediction. With an RMSE of ± 2.6 (see Table 1), the selected lessons will approximately always be in the aforementioned range. Once y_{th} is selected other aspects need to be discussed. To avoid the cold start problem, the assessment procedure, generally used by the ITS to determine the starting difficulty level, can be exploited to get initial information about the students. Without any information the first prediction of the MF on the student will be comparable to the ones of the Global Average in Table 1. In Schatten et al. (2014a) we provided the simulated students five contents in order of difficulty and then re-build the model to take the new information into account. This method was sufficient to get initial information about the students if the model is recreated at time intervals. Rebuilding the model is computationally demanding, consequently the length of the time interval should be calibrated according to the amount of new available data.

3.7 Next steps

We are going to experiment the sequencer on real students as a joint action of UHi and Whizz. The details of the experiment together with the first results are going to be shown in D5.2.

Please note that in this Section we talked about sequencing of contents and not of tasks. We did so because we envisage the possibility to use performance prediction and Vygotsky policy also for sequencing other contents like hints and feedbacks.

Moreover, we will discuss how different speech analysis could be uses to support our sequencer. This could be done in many different ways: it could interact with the MF algorithm, it could be used to select the y_{th} in the Vygotsky policy or it could be used to further represent the student's state within a RL-based sequencer (Folsom-Kovarik et al. 2013).

4. Adaptive Support

We mentioned in the introduction that another use of prediction approaches is student modelling with respect to the interaction with the learning environment. This student model information could be used to enhance feedback provision to students. In previous work we have used Bayesian Networks to predict the necessity of help requests in a web-based ITS (Mavrikis 2008). This fed to the prediction of effective interaction that would benefit from additional information on the predicted success on the task. Having this information gives the possibility of providing what is sometimes referred to as global feedback (Melis et al. 2003) or task-independent adaptive support, i.e. hints that relate to students' interaction overall rather than the specific problem-solving steps. At least in the case of the commercial ITS under investigation, problem-solving steps are dealt by different components and in fact operate as individual learning objects. Examples of such feedback include the provision of support at the beginning or end of the exercise but also during an exercise if, for example, there is no task-specific help to provide. Accordingly, when students start their experience, it is helpful to provide suggestions about which topic(s) they could choose to study based on the MF prediction. The topic having the most tasks in the ZPD, defined in Eq. (5), should be proposed. During an exercise, if students attempt to ask for help but the prediction above indicates that they do not seem to need it, the system can restrict help depending on the current answers and attempts on the exercise. Alternatively, this information can be used for penalising scoring and influencing subsequent sequencing. Similarly, if attempting to leave an exercise, students can be reminded of its goals and encouraged to complete it. However, if there are more exercises and the MF prediction indicates a successful completion, appropriate affective hints can encourage the student to try harder as in Mavrikis (2008).

4.1 Task-dependent support for Fractions Lab

The task-dependent support for Fractions Lab aims to provide feedback during interaction with the ELE. As described in D1.3, conceptual knowledge can be developed through student engagement with exploratory tasks related to domain-specific content. The assistance provided by the task-dependent support is based on Pólya's reasoning stages (Pólya, 1945).

4.1.1 Related work

According to the different reasoning stages, different approaches have been taken to supporting learners within an exploratory learning environment. Machine learning such as Bayesian Networks (e.g. Bunt and Conati, 2003) have been mainly applied while the plan and task performance is carried out, while formulating goals and planning mainly involved pre-formulated prompts (e.g. Moos and Azevedo 2008; Davis and Linn, 2000; Simons and Klein, 2007). Additionally, the self-reflection phase has mainly involved prompts or questions (e.g. Ergazaki et al., 2007, van der Meij & de Jong, 2011; Jones et al., 2013; Thillmann et al., 2009).

It can be seen that adaptive personalized support has mainly been provided during the carrying out the plan and tasks stage, while the other reasoning stages usually provide pre-formulated non adaptive prompts. The aim of the task-dependent support is to provide adaptive personalized support at all reasoning stages.

4.1.2 Components of the task-dependent support

The different layers of the task-dependent support can be seen in Figure X. Similar to Gutierrez-Santos et al. (2012), the support consists of three main layers: (1) the analysis or evidence detection layer; (2) the reasoning layer; and (3) the feedback generation layer. In the evidence detection layer the student's interactions with Fractions Lab are detected. This provides an overview of the current situation.

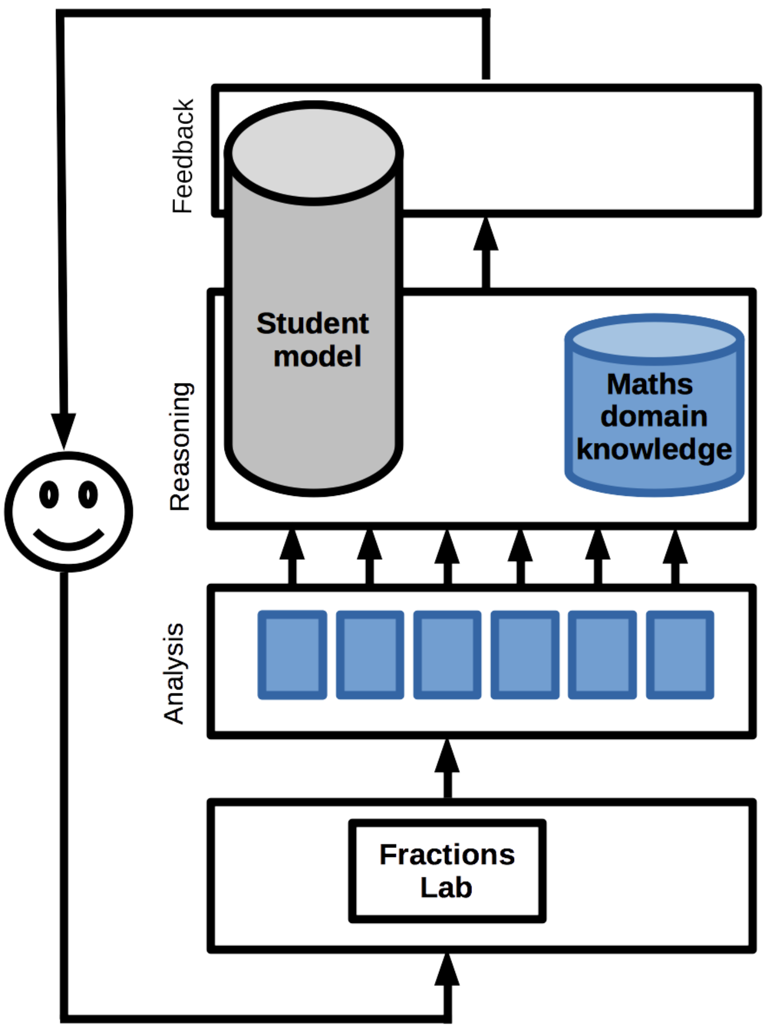

Figure 12: Components of the task-dependent support.

Based on the evidence detection component, the reasoning layer decides which aspects or concept needs support. This layer includes the maths domain knowledge and the student model.

The maths domain knowledge base includes rules about how to solve certain maths tasks within Fractions Lab, such as which actions you need to undertake in order to solve the task.

The student model is a copy of the maths domain knowledge that also includes common misconceptions associated with the particular task. The model receives data about the student's actions from the analysis layer, which is then mapped to the maths knowledge and the misconceptions. The student model also includes information about the current reasoning stages, including which reasoning stage the student is currently occupied with.

This information is the used by the reasoning layer to provide personalized adaptive support at the particular reasoning stage the student is currently at, which is based on his/her knowledge of the task, including misconceptions.

The feedback generation layer receives the output from the reasoning layer to decide how the support should be presented; for example high or low intrusion feedback.

4.2 Task-independent support

In order to enable learners to communicate more naturally with the interface, speech recognition for children is integrated into the platform. The aim of the task-independent support is to use the children's speech to provide feedback on structured and unstructured tasks according to: i) the mathematical vocabulary; and ii) emotions. The next sections describe the different feedback options in more detail.

4.2.1 Mathematical vocabulary

Using appropriate mathematical vocabulary is an indication of the student's knowledge. If the correct vocabulary is used then the student has high knowledge of the task, whereas if it is incorrect, this might indicate some knowledge gaps. The task-independent support will detect the mathematical vocabulary of the student and prompt the student to use the correct vocabulary if necessary.

This approach is very innovative has it has not been used in existing learning environments for children yet. Somewhat related to this goal is the work by Litman and Silliman (2004). They describe an intelligent tutoring system, ITSPOKE, which is able to engage the student in a spoken dialogue to provide feedback and correct misconceptions. AutoTutor (Graesser et al., 2005) is another example where students are able to have a dialogue with a conversational agent in an intelligent tutoring system. Here, the learner is assisted in the construction of an answer that is based on the learner's knowledge. However, both systems described are for adults, which focus on a dialogue, and do not take into account the use of specific domain vocabulary. Whereas we aim at supporting children in learning maths through the use of correct vocabulary.

4.2.2 Emotional feedback

As described in D1.3, emotions play a significant role in learning. While positive emotions can enhance learning, negative emotions can inhibit it. The task-independent support will provide feedback according to the student's emotion.

Different computational approaches have been taken into account in detecting emotions. These include, for example: speech-based approaches (e.g. Cowie et al., 1999; Vogt and André, 2005); using information from facial expressions (e.g. Kaliouby and Robinson, 2004); keystrokes or mouse movements (Epp et al., 2011); physiological sensors (e.g. Lang et al., 1993; Vyzas and Picard, 1998; Nasoz et al., 2003); or a combination of these (D'Mello et al., 2005).

In the area of education, Conati & MacLaren (2009) developed a model of emotions (Dynamic Bayesian network) based on students' bodily expressions for an educational game. The system uses six emotional states: joy, distress, pride, shame, admiration and reproach. A pedagogical agent provides support according to the emotional state of the students, and the user's personal goal, such as wanting help, having fun, learning maths, or succeeding by oneself.

Another example is Shen et al. (2009), who also used Bayesian Networks to classify students' emotions. Here biophysical signals, such as heart rate, skin conductance, blood pressure, and EEG brainwaves were used for the classification. The type of emotions used included: interest, engagement, confusion, frustration, boredom, hopefulness, satisfaction, and disappointment.

Woolf et al. (2009) developed an affective pedagogical agent which is able to mirror a student's emotional state, or to acknowledge a student's emotion if it is negative. They use hardware sensors (pressure sensitive seat cushion, pressure mouse, wireless conductance bracelet) and facial movements to detect students' emotions. The system discriminates between seven emotions: high/low pleasure, frustration, novelty, boredom, anxiety, and confidence. Different machine learning techniques were applied for the classification, including Bayesian Networks and Hidden Markov models.

Litman, & Forbes-Riley (2004) developed a physics text-based tutoring system called ITSPOKE. It uses spoken dialogue to classify emotions. Acoustic-prosodic and lexical features are used to predict student emotions. They apply boosted decision trees for their classification. Three emotion types are detected: negative, neutral and positive emotions.

Another example is the tutoring system described by D'Mello et al. (2005) which holds conversations with students in computer literacy and physics courses. The system classifies emotions based on natural language interaction, facial expressions, and gross body movements. Different machine learning techniques were applied, like Decision Trees, Bayesian, Neural Networks, Fuzzy, and genetic algorithms. They focus on three emotions, namely frustration, confusion, and boredom. The classification is used to respond to students via a conversation.

Most of the related work in the educational domain focusses on detecting emotions in different input stimuli, ranging from spoken dialogue to physiological sensors. However, little research has been done in how those detected emotions can be used in a tutoring system to enhance the learning experience. The only research we are aware of specifically targeting the question of responding to student affect is Woolf et al. (2009) and Baker et al. (2010). Woolf et al. (2009) describe how an embodied pedagogical agent is able to provide different types of interventions, such as praising or mirroring the student's emotional state. Baker et al. (2010) looks at the effect of cognitive-affective states on student's learning behaviour. This work contributes to how intelligent support can be used to turn negative emotions into positive emotions.

4.2.3 Components of the task-independent support

The platform is able to support the student, based on speech as well as on the interaction. Two different components are included within the task-independent support: a vocabulary and emotion detector, and an emotion reasoner. Figure 2 provides an overview of the different layers of task-independent support in the iTalk2Learn platform.

Similar to the task-dependent support, the task-independent support consists of three main layers: the analysis or evidence detection layer; the reasoning layer; and the feedback generation layer. The evidence detection layer includes a vocabulary detector as well as an emotion detector. The student model includes the student's interaction, provided by the task-dependent support. The student model is only filled with knowledge if Fractions Lab is used. If any of the other structured learning activities from Whizz or Fractions Tutor are used, then the student model does not contain any information.

The reasoning layer includes the emotion reasoner, which uses the results of the emotion detector and the student model to decide if any support needs to be provided based on the student's current emotional state. This layer also analyses the vocabulary and prompts the student to use the domain specific maths vocabulary if the student does not use it.

In addition to the feedback generation layer at the task-dependent support, here the feedback also takes emotions into account to decide how the support should be presented.

A use case scenario that describes the different layers and the data flow follows:

The student is confronted with a particular learning task. He or she reads and listens to the task and tries to understand the next steps necessary to deal with it. In order to come up with a plan of action, the student refers to their knowledge of the task. Let's assume the student has low knowledge of the task and struggles to formulate a plan of action. S/he feels bored, as the task seems too difficult. The system recognises that the student is hesitating to perform an action (some time has passed without any action taking place) and asks the student to express her/his feelings verbally by responding to the question: `How do you feel?'. The student then answers: `I am bored, I want to do something else'. This answer is processed by the speech recognition software, which provides a list of words. This list of words is then used by the emotions detector for classification. The result of the classification is that the student is bored. This is then used by the emotion reasoner to decide how to transform the negative emotion into a positive one. It uses additional knowledge from the student model, such as the current knowledge level of the student. The emotion reasoner tries to align the student goal with the learning task by supporting the student in formulating a plan of action. Here, the plan of action entails a set of activities that the student needs to perform in order to address the task. This information is then used by the feedback strategy component to display the plan of action. The student is provided with a set of activities, such as recommending a different representation to solve the task, which aim at turning the negative emotion of boredom into a positive emotion such as enjoyment.

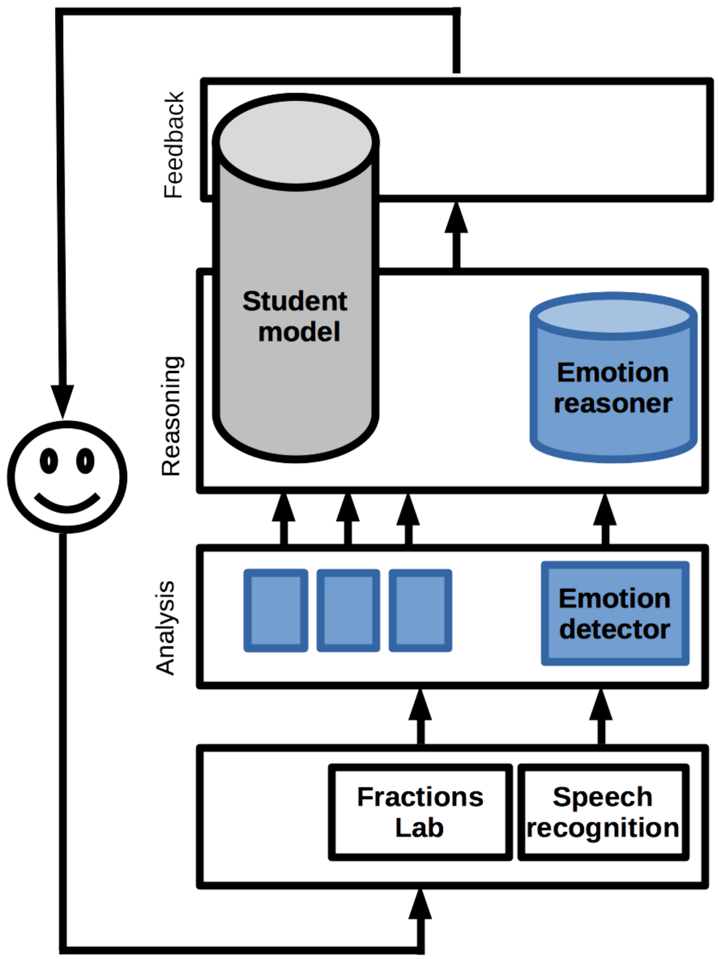

Figure 13: Components of the task-independent support.

4.2.4 Modelling of vocabulary and emotions

The vocabulary detector is based on the vocabulary that is commonly used when working with fractions.

Regarding the emotions, we focus on emotions that arise from a learning situation. The emotion detector is based on the achievement emotions described in Pekrun (2006) as well as emotions which were detected in the Wizard of Oz studies described in D5.1. The following five emotions are included in the task-independent support: enjoyment, surprise, confusion, frustration and boredom.

The input for the vocabulary and emotions detector is the result of the speech recognition software from Sail. It creates an array of words based on speech input. This word array is used by both the detectors for classification.

The classification is based on the `Bags-of-Words' model (e.g. Schuller et al., 2005; Batliner et al., 2006), which is also mentioned in D3.4.1 as a possible feature source. For the maths vocabulary as well as for each emotion a set of words is assigned. Those words were gathered in experimental studies where students used the iTalk2Learn platform or paper-based tasks for naming fractions or adding and subtracting fractions.

Examples for the maths vocabulary include the words: `nominator', `denominator'. Examples for the emotion boredom are: `this is too simple', `this is boring, `can we do something else'. Examples for enjoyment are: `Yes', `I got it', `Yeah', `I am having fun', `Done'.

For the maths vocabulary as well as for each of the five achievement emotions mentioned above a vector is created that includes a set of words. The dimensionality of the vector is the number of words in the vocabulary. The occurrence of particular words in those vectors is used for classification.

We apply a naive Bayes classifier for classifying the emotions. The result of the classification is then used by the system to decide whether and how adaptive feedback should be provided via the reasoning layer.

4.2.5 Adapting to vocabulary and emotions

If the student does not use the maths vocabulary then feedback will be provided to the student prompting him/her to use appropriate vocabulary, such as `Can you explain that again using the terms denominator, numerator?'

To adapt to emotions, the emotion reasoner is used. This is similar to an affect consequent model, described by Marsella et al. (in press) where affect is mapped to some behaviour or cognitive change. It is based on appraisal theory, which assumes that emotions are based upon patterns of individual judgement concerning the relationship between events and beliefs, desires and intentions. These judgements refer to the individual significance of events. Appraisal can trigger cognitive responses that feed back into a cycle of appraisal. In this theory, appraisal is seen as the cause of behavioural and cognitive changes that are associated with emotion.

Within computational appraisal theory, an affect consequent model maps some affect into a behavioural or cognitive change. This can include changing features in an environment which led to undesirable appraisal, in order to transfer a negative into a positive emotion.

Our emotion reasoner tries to reduce negative emotions by changing the environment via adaptive support. It includes pedagogical affect rules about how negative emotions can be transferred into positive emotions by aligning the student's reasoning process with the learning task. The rules are based on Wizard of Oz studies where the platform was used as a tool to investigate what type of support is effective for a particular emotion.

For structured tasks, the support provided will be mood boosts, such as telling the student that s/he is doing great if s/he is frustrated and finds the current task hard.

The adaptation according to emotions will be more specific on unstructured tasks (i.e. Fractions Lab) because knowledge about the student can be integrated within the emotional support. For example, the student is frustrated as the actions he or she performed did not lead to a desired outcome. The student has a particular misconception which led to the set of actions performed, which were wrong. In order for the student to overcome the frustration, the support prompts him to reflect on his plan of action and the resulting outcome. With this the student then reconsiders his actions and comes up with a new plan of action to perform the learning task.

4.3 Next steps

For the task-dependent support new rules on how to provide adaptive feedback to students need to be added. Those rules will be gathered in the Wizard-of-Oz studies planned this summer. We are particularly interested to explore how the feedback can be adapted to the specific needs of the student (based on the student model that includes misconceptions), as well as how the feedback should be presented (high or low intrusive feedback).

The task-independent support needs to be implemented and tested in autumn. Pedagogical rules need to be added to the task-independent support which can enhance the learning experience based on student's emotional state (while they are performing a learning task).

5. Conclusions

In this deliverable we presented the work of WP2 in the principal topics of user modeling for ITS and ELE: Adaptive Support, Sequencing and Performance Prediction. We showed MF performances on the Whizz dataset, moreover we showed how a performance prediction method could be applied for sequencing exploiting a score based policy. We discussed the feasibility of employing a Matrix Factorization prediction to sequencing and providing adaptive support. In particular, we presented the requirements for the applications within the Whizz system exploiting the already available dataset and multi-topic and curricula lessons. To ameliorate this system and many others, a major role could be played by data-driven sequencers and hints, where the personalization and statistical characterization of the tasks are crucial. To move towards an automatized process, with a consequent increase of the available amount of data, novel success criteria needs to be defined. We suggested to maximize the number of exercises passed within a specific score range and are currently working in defining other ones. Moreover, by monitoring the behaviour of the students in the available dataset, we showed also how correct hinting could be ameliorated by the VP. Finally, we described how the task-dependent support will be able to provide personalized support based on its student model and how the task-independent support is able to provide support according to maths terminology and emotions. We are interested to explore how the emotion detector from the task-independent support can also provide relevant information to the task-dependent support for Fractions Lab, such as providing certain types of feedback according to student's emotional state.

As future work we will investigate if, while interacting with the system, other indicators could be used to measure the sequencing quality: like hours spent within the system and number of exercises skipped. Moreover, we will try to ameliorate the performance prediction with the final goal to increase MF latent feature interpretability.

For the task-dependent support more rules about the maths domain knowledge and misconceptions will be added. Those rules will be gathered in the Wizard of Oz study planned for this summer. It is planned to test the prototype at the school twice for iterative development.

The task-independent support needs to be implemented and pedagogical affect rules added to the emotion reasoner. It is planned to test task-independent support in autumn in the school to iteratively design the emotion reasoner.

References

Batliner, A., Steidl, S., Schuller, B., Seppi, D., Laskowski, K., Vogt, T., Devillers, L., Vidrascu, L., Amir, N., Kessous, L., Aharonson, V. (2006) Combining efforts for improving automatic classification of emotional user states. In: Proc. IS-LTC 2006, Ljubliana, Slovenia, pp. 240—245.

Beck, Joseph (1997) Modeling the student with reinforcement learning. In Machine learning for User Modeling Workshop at the Sixth International Conference on User Modeling.

Beck, Joseph, Beverly Park Woolf, and Carole R. Beal (2000) ADVISOR: A machine learning architecture for intelligent tutor construction. In AAAI/IAAI 2000, 552-557.

Beck, Joseph E., and Beverly Park Woolf (2000) High-level student modeling with machine learning. In Intelligent tutoring systems. Springer Berlin Heidelberg.

Baker, Ryan SJ, Albert T. Corbett, and Vincent Aleven (2008) More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In Intelligent Tutoring Systems. Springer Berlin Heidelberg.

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why Students Engage in "Gaming the System. In Behavior in Interactive Learning Environments. Journal of Interactive Learning Research, 19(2), 185-224.

Baker, R.S.J.D, D'Mello, S.K., Rodrigo, M.T., Graesser, A.C. (2010) Better to be frustrated and bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environemnts. Int. J. Hum.-Comput. Stud., 68(4):223-241.

Baker, R.S.J.,Pardos, Z.A., Gowda, S.M., Nooraei, B.B., Heffernan, N.T. (2011) Ensembling predictions of student knowledge within intelligent tutoring systems. In User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, 2011. 13-24.

Bunt A. and Conati C. (2003) Probabilistic Student Modelling to Improve Exploratory Behaviour. In User Modeling and User-Adapted Interaction, 3.

Cen, Hao, Kenneth Koedinger, and Brian Junker (2006) Learning factors analysis–a general method for cognitive model evaluation and improvement. In Intelligent Tutoring Systems. Springer Berlin Heidelberg..

Chang, M. M. (2007) Enhancing web-based language learning through self-monitoring. In Journal of Computer Assisted Learning, 23 (3), 187-196.

Chang, K., Beck, J., Mostow, J., Corbett, A. (2006) A bayes net toolkit for student modeling in intelligent tutoring systems. In Intelligent Tutoring Systems. Springer Berlin Heidelberg.

Chavhan, Yashpalsing, M. L. Dhore, and Pallavi Yesaware (2010) Speech emotion recognition using support vector machine. In International Journal of Computer Applications 1.20, 6-9.

Chi, M., VanLehn, K., Litman, D., Jordan, P. (2010) Inducing effective pedagogical strategies using learning context features. In User Modeling, Adaptation, and Personalization. Springer Berlin Heidelberg. 147-158.

Chi, M., VanLehn, K., Litman, D., Jordan, P. (2011) Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. In User Modeling and User-Adapted Interaction 21.1-2, 137-180.

Chieu, V.M., Luengo, V., Vadcard, L., Tometti, J. (2010) Student Modeling in Orthopedic Surgery Training: Exploiting Symbiosis between Temporal Bayesian Networks and Fine-grained Didactic Analysis. In International Journal of Artificial Intelligence in Education, 20(3), 269-301.

Cocea, M., Gutierrez-Santos, S., Magoulas, D. (2010) Adaptive Modelling of Users' Strategies in Exploratory Learning Using Case-Based Reasoning. In Proceedings of the 14th International Conference of Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, Vol. 6277, 124-134.

Colancies, J.D., Nussbaum, E.M. (2008) Enhancing online collaborative argumentation through question elaboration and goal instructions. Journal of Computer Assisted Learning, 24 (3), 167-180.

Conati, Cristina, Abigail Gertner, and Kurt Vanlehn. (2002) Using Bayesian networks to manage uncertainty in student modeling. In User modeling and user-adapted interaction, 12(4), 371-417.

Conati C. and Zhou X. (2002). Modeling Students' Emotions from Cognitive Appraisal in Educational Games. In Proceedings of ITS 2002, 6th International Conference on Intelligent Tutoring Systems, Biarritz, France.

Conati, C., MacLaren, H. (2009) Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction.

Conati, C., Jaques, N., Muir, M. (2013) Understanding attention to adaptive hints in educational games: an eye-tracking study. In International Journal of Artificial Intelligence in Education, 23.

Corbett, Albert T., and Anderson, John R. (1994) Knowledge tracing: Modeling the acquisition of procedural knowledge. In User modeling and user-adapted interaction, 4(4), 253-278.

Cowie, R., Douglas-cowie, E., Apolloni, B., Taylor, J., Romano, A., Fellenz, W. (1999) What a neural net needs to know about emotion words. In: N. Mastorakis (eds.), Computational Intelligence and

Applications, pp. 109-114.

Crippen, K.J., Earl, B.L. (2007) The impact of web-based worked examples and self-explanation on performance, problem solving, and self-efficacy. In Computers & Education, 49, 809-821.

Davis, E.A., Linn, M.C. (2000) Scaffolding students' knowledge integration: prompts for reflection in KIE. In International Journal of Science Education, 22.

Diziol, D., Walker, E., Rummel, N., Koedinger, K.R. (2010) Using intelligent tutor technology to implement adaptive support for student collaboration. In Educational Psychology Review, 22(1), 89-102.

Devolder, A., van Braak, J., Tondeur, J. (2012) Supporting self-regulated learning in computer-based learning environments: systematic review of effects of scaffolding in the domain of science education. In Journal of Computer Assisted Learning, 28(6), 557-573.

Epp, C., Lippold, M., Mandryk, R.L. (2011) Identifying emotional states using keystroke dynamics. In: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems, pp. 715—724.

Ergazaki, M., Zogza, V., Komis, V. (2007) Analysing students' shared activity while modeling a biological process in a computer-supported educational environment. In Journal of Computer Assisted Learning, 23(2), 158-168.

Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. The Journal of User Modeling and User-Adapted Interaction, 19, 243-266.

Folsom-Kovarik, J. T., Sukthankar, G., & Schatz, S. (2013). Tractable POMDP representations for intelligent tutoring systems. ACM Transactions on Intelligent Systems and Technology (TIST), 4(2), 29.

Fund, Z. (2007) The effects of scaffolded computerized science problem-solving on achievement outcomes: a comparative study of support programs. In Journal of Computer Assisted Learning, 23 (5), 410-424.

Furberg, A. (2009) Socio-cultural aspects of prompting student reflection in Web-based inquiry learning environments. In Journal of Computer Assisted Learning, 25(4), 397-409.

Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.(2010) Learning attribute-to-feature mappings for cold-start recommendations. In Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE.

Gong, Y., Beck, J.E., Heffernan, N.T. (2010) Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In Intelligent Tutoring Systems. Springer Berlin Heidelberg.

González-Brenes, J.P., Mostow, J. (2013) What and When do Students Learn? Fully Data-Driven Joint Estimation of Cognitive and Student Models. In Proceeding of the 6th International Conference on Educational Data Mining, 236-240.

González-Brenes, J.P., Mostow, J. (2012) Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. EDM. 2012.

Graesser, A.C., Chipman, P., Haynes, B.C., Olney, A. (2005) AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. IEEE Transactions on Education, 48(4), 612-618.

Gurlitt, J., Renkl, A. (2008) Are high-coherent concept maps better for prior knowldge activation? Different effects of concept mapping tasks on high school vs. university students. In Journal of Computer Assisted Learning, 24.

Gutierrez-Santos, S., Mavrikis, M., Magoulas, D. (2012) A Separation of Concerns for Engineering Intelligent Support for Exploratory Learning Environments. In Journal of Research and Practice in Information Technology, 44(3), 347-360.

Harrison, B., and Roberts, D.L. (2012) A Review of Student Modeling Techniques in Intelligent Tutoring Systems. In Proceedings of Eighth Artificial Intelligence and Interactive Digital Entertainment Conference.

Iglesias, A., Martinez, P., Fernández, F. (2003) An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. In Informatics in Education, 2(2), 223-240.

Iglesias, A., Martinez, P., Aler, R., Fernández, F (2009) Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning.In Applied Intelligence 31(1), 89-106.

Joolingen van, W. (2012) Supporting inquiry learning based on Emerging Learning Objects. In Proceedings of the Intelligent Support for Exploratory Environments workshop at ITS 2012.

Jones, A., Bull, S., Castellano, G. (2013) Teacher Perspectives on the Potential for Scaffolding with an Open Learner Model and an Empathic Robot. In Proceedings of Scaffolding in Open-Ended Learning Environments at AIED 2013.

Joshi, A., Kaur, K. (2013) A Study of Speech Emotion Recognition Methods.In International Journal of Computer Science and Information Technology (IJCSMC), 2(4).

Joshi, D.D., Zalte, M.B. (2013) Speech Emotion Recognition: A Review. In Journal of Electronics and Communication Engineering (IOSR-JECE), 4(4).

Kaliouby, R.E., Robinson, P. (2004) Real-time inference of complex mental states from facial expressions and head gestures. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition Workshop, pp. 154.

Karakostas, A., Demetriadis, S. (2011) Enhancing collaborative learning through dynamic forms of support: the impact of an adaptive domain-specific support strategy. In Journal of Computer Assisted Learning, 27(3), 243-258.

Kardan, S., and Conati, C. (2012) Providing Adaptive Support in an Exploratory Learning Environment Using Interaction Data. In Proceedings of the Intelligent Support for Exploratory Environments workshop at ITS 2012.

Kickmeier-Rust, M.D., Albert, D. (2010) Micro-adaptivity: protecting immersion in didactically adaptive digital educational games. In Journal of Computer Assisted Learning, 26(2), 95-105.

Kim, J., Hill, R.W., Durlach, P., Lane, H.C., Forbell, E., Core, M., Marsella, S., Pynadath, D., Hart, J. (2009) BiLAT: A Game-Based Environment for Practicing Negotiation in a Cultural Context. In International Journal of Artificial Intelligence in Education, 19(3), 289-308.

Koedinger, K.R., Pavlik, P.I., Stamper, J., Nixon, T., Ritter, S. (2010) Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing. In EDM, 91-100 .

Krohn-Grimberghe, A., Busche, A., Nanopoulos, A., & Schmidt-Thieme, L. (2011). Active learning for technology enhanced learning. In Towards Ubiquitous Learning (pp. 512-518). Springer Berlin Heidelberg.

Krothapalli, S.R., Koolagudi, S.G. (2013) Speech Emotion Recognition: A Review. In Emotion Recognition using Speech Features. Springer New York, 15-34.

Krohn-Grimberghe, A., Busche, A., Nanopoulos, A. (2011) Active learning for technology enhanced learning. Towards Ubiquitous Learning. Springer Berlin Heidelberg, 512-518.

Lallé, S., Mostow, J., Vanda, L., Guin, N. (2013) Comparing Student Models in Different Formalisms by Predicting their Impact on Help Success. In Proceedings of the 16th International Conference, AIED 2013.

Lang, P.J., Greenwald, M.K., Bradley, M.M., Hamm, A.O. (1993) Look at Pictures: Affective, Facial, Visceral, and Behavioral Reactions. Psychophysiology, 30, pp. 261—273.

Lee, J.I., Brunskill, E. (2012) The Impact on Individualizing Student Models on Necessary Practice

Opportunities." In EDM.

Liao, C.C.Y., Chen, Z.-H., Cheng, N.H.H., Chen, F.-C., Chan, T.-W. (2011) My-Mini-Pet: a handheld pet-nurturing game to engage students in arithmetic practices. In Journal of Computer Assisted Learning, 27(1), 76-89.

Lindstoem, P., Gulz, A., Haake, M., Sjoeden, B. (2011) Matching and mismatching between the pedagogical design principles of a math game and the actual practices of play. In Journal of Computer Assisted Learning, 27(1), 90-102.

Lintean, M., Rus, V., Azevedo, R. (2011) Automatic Detection of Student Mental Models Based on Natural Language Student Input During Metacognitive Skill Training. In International Journal of Artificial Intelligence in Education, 21 (3), 169-190.

Litman, D.J., Silliman, S. (2004) ITSPOKE: An intelligent tutoring spoken dialogue system. In Demonstration Papers at HLT-NAACL 2004. Association for Computational Linguistics.

Litman, D.J., Forbes-Riley, K. (2004) Predicting student emotions in computer-human tutoring dialogues. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL '04), Association for Computational Linguistics.

Manlove, S., Lazonder, A.W., de Jong, T. (2007) Software scaffolds to promote regulation during scientific inquiry learning. In Metacognition and Learning, 2, 141-155.

Manlove, S., Lazonder, A.W., de Jong, T. (2009) Trends and issues of regulative support use during inquiry learning: patterns from three studies. In Computers in Human Behavior, 25.

Marsella, S., Gratch, J., Petta, P. (in press)Computational Models of Emotion. In: Scherer, K.R., Baenziger, T., Roesch, E. (eds.) A blueprint for an affectively competent agent: Cross-fertilization between Emotion Psychology, Affective Neuroscience, and Affective Computing, Oxford University Press, Oxford

Martin, K.N., Arroyo, I. (2004) AgentX: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In Intelligent Tutoring Systems. Springer Berlin Heidelberg.

Mavrikis, M. (2008). Data-driven modelling of students' interactions in an ILE. InEDM (pp. 87-96).

Mavrikis, M. (2010) Modelling Student Interactions in Intelligent Learning Environments: Constructing Bayesian Networks from Data. In International Journal on Artificial Intelligence Tools, 19(6), 733-753.

Mavrikis, M., Gutierrez-Santos, S., Geraniou, E., Noss, R. (in press) Design Requirements and Validation Metrics for Adaptive Exploratory Learning Environments: From pedagogic strategies to computer-based support. Journal of Personal and Ubiquitous Computing.

Mao, X., Chen, L., Fu, L. (2009) Multi-level speech emotion recognition based on HMM and ANN. In Computer Science and Information Engineering, 2009 WRI World Congress, 7.

van der Meij, J., de Jong, T. (2011) The effects of directive self-explanation prompts to support active processing of multiple representations in a simulation-based learning environment. In Journal of Computer Assisted Learning, 27(5), 411-423.

Melis, E., & Ullrich, C. (2003). Local and global feedback. In Proceedings of AIED2003, 11th International Conference in Artificial Intelligence in Education, Sydney, Australia (pp. 476-478).

D'Mello, S.K., Craig, S.D., Gholson, B., Franklin, S., Picard, R.W., Graesser, A.C. (2005) Integrating Affect Sensors in an Intelligent Tutoring System. In Affective Interactions: The Computer in the Affective Loop Workshop at 2005 International Conference on Intelligent User Interfaces, pp. 7—13.

Muir, M.M.A. (2012) Prime Climb: an analysis of attention to student-adaptive hints in an educational game. MSc Thesis. University of British Columbia.

Moos, D.C., Azevedo, R. (2008) Exploring the fluctuation of motivation and use of self-regulatory processes during learning with hypermedia. In Instructional Science, 36.

Nash, P., Willamson Shaffer, D. (2011) Mentor modeling: the internalization of modeled professional thinking in an epistemic game. In Journal of Computer Assisted Learning, 27(2), 173-189.

Nasoz, F., Alvarez,K., Lisetti,C.L., Finkelstein, N. (2003) Emotion Recognition from Physiological Signals for Presence Technologies. International Journal of Cognition, Technology and Work, Special Issue on Presence, 6(1), pp.4--14

Newell, A.Hillsdale, NJ: Erlbaum & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1-55).

Nooraei, B., Pardos, Z.A., Heffernan, N.T., Baker, R.S.J.D (2011) Less is More: Improving the Speed and Prediction Power of Knowledge Tracing by Using Less Data. In EDM.

Pavlik, P.I., Cen,H., and Koedinger, K.R. (2009) Performance Factors Analysis-A New Alternative to Knowledge Tracing. In AIED.

Pardos, Z.A., Heffernan, N.T. (2010) Modeling individualization in a bayesian networks implementation of knowledge tracing. In User Modeling, Adaptation, and Personalization. Springer

Berlin Heidelberg. 255-266.

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. In Journal of Machine Learning Research W & CP.

Pardos, Z.A., Heffernan, N.T. (2011) KT-IDEM: introducing item difficulty to the knowledge tracing model.In User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, 243-254.

Pekrun, R. (2006) The Control-Value Theory of Achievement Emotions: Assumptions, Corollaries, and Implications for Educational Research and Practice. J. Edu. Psych.. Rev., pp. 315--341

Gowda, S.M., Baker, R.S.J.D, Pardos, Z., Heffernan, N.T. (2012) The sum is greater than the parts: ensembling models of student knowledge in educational software. In ACM SIGKDD Explorations Newsletter, 13(2), 37-44.

Pintrich, P.R. (2000) The role of goal orientation in self-regulated learning. In Handbook of Self-Regulatio, 451-502. Academic Press, San Diego, CA.

Pólya, G. (1945) How to Solve It. Princeton University Press. ISBN 0-691-08097-6.

Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C., Baker, R.S.J.D. (2013) Knowledge Elicitation Methods for Affect Modelling in Education. In International Journal of Artificial Intelligence in Education, 23.

Qiu, Y., Qi, Y., Lu, H., Pardos, Z.A., Heffernan, N.T. (2011) Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing. In EDM. 2011.

Ravindran, B., Sarma, B.H. (2010) Intelligent Tutoring Systems using Reinforcement Learning to teach Autistic Students." International Federation for Information Processing Digital Library 241(1).

Rendle, S. (2010) Factorization machines. In Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE.

Reye, J. (2004) Student modelling based on belief networks. In International Journal of Artificial Intelligence in Education 14(1), 63-96.

Rohrer, D., & Taylor, K. (2006). The effects of overlearning and distributed practice on the retention of mathematical knowledge. *Applied Cognitive Psychology, 20,* 1209-1224.

Roll, I., McLaren, B.M., Koedinger, K.R. (2011) Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. In Learning and Instruction, 21(2), 267-

280.

Sabourin, J.L., Shores, L.R., Mott, B.W., Lester, J. (2013) Understanding and Predicting Student Self-Regulated Learning Strategies in Game-Based Learning Environments. In International Journal of Artificial Intelligence in Education, 23.

Sarma, B. S., & Ravindran, B. (2007). Intelligent Tutoring Systems using Reinforcement Learning to teach Autistic Students. In Home Informatics and Telematics: ICT for The Next Billion (pp. 65-78). Springer US.

Schatten C. and Schmidt-Thieme, L. (2014) Adaptive content sequencing without domain information. In proceedings of CSEDU 2014.

Schatten C., Wistuba M., Schmidt-Thieme, L. and Gutiérrez-Santos S. (2014) Minimal Invasive Integration of Learning Analytics Services in Intelligent Tutoring Systems, 14th IEEE International Conference on Advanced Learning Technologies - ICALT2014.

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In D. Grouws (Ed.), Handbook for Research on Mathematics Teaching and Learning (pp. 334-370). New York: MacMillan.

Schuller, B., Müller, R., Lang, M., Rigoll, G. (2005) Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensemble. In: Proc. Interspeech, Lisbon, Portugal, pp. 805—808.

Shen, L., Wang, M., Shen, R. (2009) Affective e-Learning: Using “Emotional” Data to Improve Learning in Pervasive Learning Environment. Educational Technology \& Society, 12 (2), pp. 176—189.

Simons, K.D., Klein, J.D. (2007) The impact of scaffolding and student achievement levels in a problem-based learning environment. In Instructional Science, 35.

Sutton, R.S., Barto, A.G. (1998) Reinforcement learning: An introduction. Vol. 1. No. 1. Cambridge: MIT press.

Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L. (2010) Recommender system for predicting student performance. In Procedia Computer Science, 1(2), 2811-2819.

Thai-Nghe, N., Drumond, L., Horvath, T., Schmidt-Thieme, L. (2011) Multi-relational factorization models for predicting student performance. In KDD 2011 Workshop on Knowledge Discovery in Educational Data, KDDinED.

Thai-Nghe, N., Drumond, L., Horvath, T., Krohn-Grimberghe, A., Nanopoulos, A., Schmidt-Thieme, L.

(2011) Factorization techniques for predicting student performance. In Educational Recommender Systems and Technologies: Practices and Challenges (In press). IGI Global (2011).

Thai-Nghe, N., Drumond, L., Horvath, T., Nanopoulos, A., Schmidt-Thieme, L. (20100) Matrix and Tensor Factorization for Predicting Student Performance. In CSEDU (1).

Thai-Nghe, N. Horvath, T. Schmidt-Thieme, L. (2011) Context-Aware Factorization for Personalized Student's Task Recommendation. In Proceedings of the International Workshop on Personalization Approaches in Learning Environments. Vol. 732.

Thai-Nghe, N., Horváth, T., Schmidt-Thieme, L. (2011) Personalized forecasting student performance. In Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on. IEEE.

Thillmann, H., Kunsting, J., Wirth, J., Leutner, D. (2009) Is it merely a question of 'what' to a prompt or also 'when' to a prompt? The role of point of presentation time of prompts in self-regulated learning. Zeitschrift Fuer Pedagogische Psychologie, 23, 105-115.

Trudel, C., Payne, S.J. (1995) Reflection and goal management in exploratory learning. In International Journal of Human-computer Studies, 42(3), 307-339.

VanLehn, K., Jordan, P.W., Rose, C.P., Bhembe, D., Bottner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S., Srivastava, R. (2002) The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In Intelligent tutoring systems. Springer Berlin Heidelberg.

VanLehn, K., Jordan, P. Litman, D. (2007) Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In Proceedings of SLaTE Workshop on Speech and Language Technology in Education ISCA Tutorial and Research Workshop.

Vapnik, V.N. (1998) Statistical learning theory. Wiley, New York. NY.

Ververidis, D., Kotropoulos, C. (2006) Emotional speech recognition: Resources, features, and methods. In Speech communication, 48(9), 1162-1181.

Vygotsky, L. L. S. (1978). Mind in society: The development of higher psychological processes. Harvard university press.

Wang, Y., Heffernan, N.T. (2011) Extending Knowledge Tracing to Allow Partial Credit: Using Continuous versus Binary Nodes. Springer-Verlag.

Wang, Y., Heffernan, N.T. (2012) The student skill model. In Intelligent Tutoring Systems. Springer

Berlin Heidelberg.

Webber, C., Pesty, S., Balacheff, N. (2002) A multi-agent and emergent approach to learner modelling. In Proceedings of the 8th Iberoamerican Conference on Artificial Intelligence.

Whitehill, J., Bartlett, M., Movellan, J. (2008) Automatic facial expression recognition for intelligent tutoring systems. In Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on. IEEE.

Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, C., Picard, R. (2009) Affect-aware tutors: recognizing and responding to student affect. Int. J. Learning Technology, 4(3-4), 129-164.

Worsley, M., Blikstein, P. (2011) What's an Expert? Using Learning Analytics to Identify Emergent Markers of Expertise through Automated Speech, Sentiment and Sketch Analysis. In EDM.

Xu, Y., Mostow, J. (2011) Logistic Regression in a Dynamic Bayes Net Models Multiple Subskills Better!. In EDM.

Xu, Y., Mostow, J. (2012) Comparison of methods to trace multiple subskills: Is LR-DBN best?. In EDM. 2012.

Xu, Y., Mostow, J. (2013) Using Item Response Theory to Refine Knowledge Tracing. In EDM.