Integration and Evaluation of a Matrix Factorization Sequencer in Large Commercial ITS

Carlotta Schatten, Ruth Janning, Lars Schmidt-Thieme
Information Systems and Machine Learning Lab, University of Hildesheim
Marienburger Platz 22
31148 Hildesheim, Germany

Abstract

Correct evaluation of Machine Learning based sequencers require large data availability, large scale experiments and consideration of different evaluation measures. Such constraints make the construction of ad-hoc Intelligent Tutoring Systems (ITS) unfeasible and impose early integration in already existing ITS, which possesses a large amount of tasks to be sequenced. However, such systems were not designed to be combined with Machine Learning methods and require several adjustments. As a consequence more than a half of the components based on recommender technology are never evaluated with an online experiment. In this paper we show how we adapted a Matrix Factorization based performance predictor and a score based policy for task sequencing to be integrated in a commercial ITS with over 2000 tasks on 20 topics. We evaluated the experiment under different perspectives in comparison with the ITS sequencer designed by experts over the years. As a result we achieve same post-test results and outperform the current sequencer in the perceived experience questionnaire with almost no curriculum authoring effort. We also showed that the sequencer possess a better user modeling, better adapting to the knowledge acquisition rate of the students.

Introduction

Intelligent Tutoring Systems (ITS) are more and more becoming of crucial importance in education. Apart from the possibility to practice any time, adaptivity and individualization are the main reasons for their widespread availability as app, web service and software. Unfortunately, many Machine Learning methods applied to Learning Analytics are born and die in laboratories due to the high integration requirements and constraints. As reported in (Manouselis et al. 2011) only one half of the developed task recommenders interacts, in a final evaluation stage, with students, whereas the other half stays at design or prototype stage of development. Adaptations of models to be integrated into already existing systems cannot be evaluated as a whole and need to be observed from different perspectives (Brusilovsky 2001). This is also important if we consider that in an integration experiment partners, especially if coming form different research areas, have different interests and qualitative measures. Integration within already existing ITS is even more challenging since those systems were not intended to be combined with Machine Learning methods, so the tendency is to develop ad-hoc learning environments. Such an integration is becoming unfeasible for Machine Learning sequencing experiments also because large data set are needed, time is required to show students' learning gains and it is necessary to have enough tasks to sequence. This impose large scale experiment in sufficiently large systems, that cannot be created ad-hoc.

In this paper we show how we adapted a machine learning based domain independent sequencer, composed of a performance predictor and a score based task sequencing policy, in order to be integrated in a large commercial online maths ITS. The latter is currently used in several countries by children aged from 6 to 14, who are practicing on over 2000 tasks and 20 topics. Moreover, we discuss the obtained online experiment's results from different perspectives. The developed sequencer showed the following promising characteristics:

1. Lightweight integrability of machine learning based sequencer in not ad-hoc constructed systems.
2. Having comparable response time as the actual rule based system.
3. Achieving the same post-test results with almost no curriculum authoring effort.
4. Possessing a better user modeling, better adapting to the knowledge acquisition rate of the students.
5. Outperforming the current sequencer in the perceived experience questionnaire.

In order to show how we achieved those results we present our work as follows. In Section we present state of the art of sequencing based on machine learning performance predictors and explain why they were not applicable for integration in our system. In Section we present the Vygotsky Sequencer and in Section how we adapted it to a real time working web service. In Section we present the designed experiments and discuss the results.

Background

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 is Reinforcement Learning (RL), which computes the best sequence trying to maximize a previously defined reward function. Both model–free and model–based (Malpani, Ravindran, and Murthy 2011; Beck, Woolf, and Beal 2000) RL were tested for content sequencing. Unfortunately, the model–based RL necessitates of special kind of data sets called exploratory corpus. Available data sets 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 advanced level, it is impossible to retrieve the probability of a novice student solving those tasks correctly. Without these probabilities the RL model cannot be built (Chi et al. 2011), moreover it excluded the already collected corpus (more than five years of data) we were planning to use for the experiment. Model–free RL 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, this technique is not advised in case of costly data collection (Chi et al. 2011).

The integrated and evaluated task sequencer is based on student performance predictions. An example of state of the art method 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 students' 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 and Anderson 1994). In the BKT extensions also difficulty, multiple skill levels and personalization are taken into account separately (Wang and Heffernan 2012; Pardos and Heffernan 2010; 2011; D Baker, Corbett, and Aleven 2008). BKT researchers have discussed the problem of sequencing both in single and in multiple skill environment in (Koedinger et al. 2011). 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 score representation is binary as in BKT (Pavlik and Koedinger 2009). Moreover, similarly to BKT, a table connecting contents and skills is required.

As we do not have access to the necessary skills information for the more than 2000 tasks in the chosen ITS, Matrix Factorization (MF) represents a good alternative to BKT and PFM. It has many applications like, for instance, dimensionality reduction, clustering and also classification (Cichocki et al. 2009). Its most common use is for Recommender Systems (Koren, Bell, and Volinsky 2009) and recently this concept was extended to performance prediction and to sequencing problems in ITS (Thai-Nghe et al. 2011; Schatten and Schmidt-Thieme 2014; Thai-Nghe et al. 2010; 2012), but all experiments were done with simulated students' interactions. (Manouselis et al. 2011) reports that recommender techniques for ITS are rarely integrated in larger systems. This happens because it is a common approach to build entire ITS to test sequencers making large scale experiments almost not affordable for universities (Schatten et al. 2014b). As discussed in (Schatten et al. 2014a; Janning, Schatten, and Schmidt-Thieme 2014; Janning, Schatten, and Lars 2014) the integration of Machine Learning components for applications in ITS, also if domain independent such as (Schatten and Schmidt-Thieme 2014), needs some tailoring. The first paper considers some preliminary analysis to evaluate feasibility of integration, whereas the second and the third suggest how to further personalize the sequencing parameter of (Schatten and Schmidt-Thieme 2014) with features coming from speech based emotion recognition.

Vygotsky Sequencer

The Vygotsky Policy Sequencer (VPS) is the domain independent sequencer we tested in our experiment. It was presented the first time in (Schatten and Schmidt-Thieme 2014) and is composed of two components: a performance prediction method and a score based policy. Because of its domain independence, the authors chose Matrix Factorization (MF) as score predictor, but other performance predictors returning a continuous value between 0 and 1 could be used. The matrix $Y_t \in \mathbb{R}^{n_s \times n_c}$ can be seen as a table of n_c total tasks and n_s students used to learn the students' model, where for some tasks and students performance measures are given at time t. 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 $Y_t \approx \hat{Y}_t = \Psi\Phi^T$. 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 the dataset Y_t. The optimization function is represented by:

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

where one wants to minimize the regularized squared error on the set of known scores optimizing variables ψ and φ. MF prediction is computed as:

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

where μ, μ_c and μ_s are respectively the average performance of all tasks 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 second component is a policy π that exploits the information of the performance predictor to select the next task. Given the predicted score for each student on each task, the VPS wants to sequence tasks in a way that attempts to keep students in this so-called zone of proximal development (ZPD) (Vygotsky 1978), which, in this context, is associated with tasks that are neither too easy nor too difficult to accomplish without much help. This concept is formalized by the following formula:

$$c^{t*} = \text{argmin}_c \left| y_{th} - \hat{y}^t\left(c\right) \right| , \qquad (3)$$

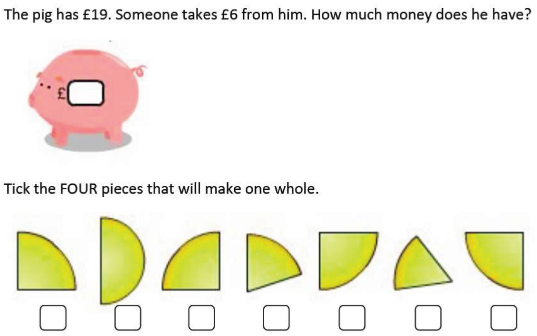
where y_{th} is a threshold score that will challenge the students and keep them in the ZPD. The so-called Vygotsky Policy (VP) will select at each time step the content c^{t*} with the predicted score $\hat{y}^t$ at time t most similar to y_{th} as described in Alg. 1, where s_i is the student currently interacting with the system and $\mathbb{C}$ is the number of tasks available. In (Schatten and Schmidt-Thieme 2014) evaluation on groups of simulated students in comparison to several sequences showed the advantages of such a system. There is no authoring effort for sequencing required, since the system is flexible enough to adapt to ITS with different number of tasks to practice per difficulty level. Most advantages are gained with large or small task availability per difficulty level. Given a curriculum that possesses the correct number of tasks to practice with for each difficulty level, the performances between a sequencer that selects tasks in order of difficulty and the VPS are comparable. On the contrary, if there are more tasks of the same difficulty level, VPS is able to skip the unnecessary ones. However, an evaluation with real students is required to confirm the results obtained in a simulated environment. The most interesting scenario for an evaluation would need an ITS with many tasks, where the sequencer could reduce the burden of pedagogical experts. Given the interdisciplinary knowledge required for creating such an ITS from scratch the experiment, without integration with an already existing system, would not have been affordable.

Algorithm 1: Vygotsky Policy based Sequencer

```
Input: $\mathbb{C}$, $Y_0$ $\pi$, $s_i$, T
Train the MF using $Y_0$;
for $t$ = 1 to T do
    for All $c \in \mathbb{C}$ do
        Predict $\hat{y}\left(c_j, s_i\right)$ Eq. 2;
    end
    Find $c^{t*}$ according to Eq. 3;
    Show $c^{t*}$ to $s_i$
    Add $y\left(s_i, c^{t*}\right)$ to $Y_t$;
    Retrain the MF solving Eq. 1
end
```

Integration in a commercial ITS

In order to integrate the system into a commercial ITS two steps need to be performed. The first one consists of an offline study and evaluation of the commercial system and its dataset, where the quality of the dataset is discussed. After a dataset preprocessing it is possible to create a first model and select the parameters of the VP. In the second step the system is taken to real time performances in order to be integrated in the commercial ITS. This involved the implementation of an online update for the MF algorithm and of a lightweight API.



The pig has £19. Someone takes £6 from him. How much money does he have?

Tick the FOUR pieces that will make one whole.

□ □ □ □ □ □ □

Figure 1: Two questions of the commercial ITS

Table 1: Dataset Statistics

Number of Tasks	2035
Number of Students	116843
Total Student-Task Interactions	15782070
Average score obtained	8.1

Commercial ITS and Dataset study

In order to design the experiment an understanding of the ITS and its dataset was required. The ITS has 20 topics about maths for children aged from 6 to 14, who can practice on over 2000 tasks at school or at home. The ITS possesses two kinds of tasks: exercises and tests. A test task cannot be shown if the student did not pass the corresponding exercise. Exercises and tests are composed of a set of questions that can be evaluated with correct and incorrect; an example can be found in Fig. 1. From these questions we do not know which ones were answered correctly since the ITS aggregates the information in a single score. In the exercises three hints are available and the last one is the bottom-out one displaying the solution. As a consequence students can pass exercises although they did not learn, but cannot proceed because they fail the tests. The score, as in (Schatten and Schmidt-Thieme 2014), is represented in a continuous interval which goes from 0 to 10. The topics and new skills to be acquired are introduced following the curriculum of the country. This also defines the Math Age which represents both the student's skills level on a topic and task's difficulty level. Tasks are assigned to topics and students have a Math Age per topic. Considering a specific topic, a task of Math Age 10 means that the difficulty should be solved by students of that age. It means also that a student with Math Age of 10 in a topic knows more than a student of Math Age 9. The commercial system possesses a rule–based adaptive sequencer that was designed and refined by pedagogical experts over the year that we call state of the art (SotA) sequencer. The Math Age is the indicator the SotA sequencer uses to monitor the progress within the curriculum and select the next tasks. A student can see tasks of the next difficulty level only if the ones of the previous level are completed.

We analyzed the dataset collecting the information summarized in Tab.1. In order to avoid sparseness, which strongly affects MF, each line is generally abstracted to Knowledge

Component (KC) level, i.e. the algorithm predicts if the student is going to answer correctly modeling his knowledge on a KC. Since we did not have this information, we preprocessed the dataset at task level predicting the future score for each task. In order to select the best performing model we followed the standard approach in the field to divide the dataset temporally in two thirds for training and one third for testing, evaluating the performances with the Root Mean Square Error (RMSE). We considered the last 3 years of data, excluding the oldest ones, since the tasks were slightly modified over time. We also removed the skipped tasks, where the score is automatically assigned to 7.5 out of 10 (7.5/10) by the ITS. Those data were considered noisy since there was no evidence that 7.5/10 could represent the knowledge of the student at that time. With a full grid search we selected the best hyperparameters' combination ($P = 50$,$\lambda = 0.01$, learn rate $= 0.02$, 100 iterations per training) with whom we obtained an average RMSE of 0.13 over 5 experiments.

Online update integration

It is well known to those working with recommender technology that MF does not deal with time, i.e. all the training performances are considered equally. As a consequence if the student is repeating a task his mark is computed as the weighted sum of the previous performances. Two past data on the same task are considered equally. Since the students'features change after each interaction in (Schatten and Schmidt-Thieme 2014) the model was retrained each time. Given the data amount this was not feasible while students were interacting with the system. In order to keep the model up to date, we implemented the online update proposed in (Rendle and Schmidt-Thieme 2008) that was previously tested for active learning problems in recommender systems. The method is an approximation of the retrain, the paper reports how the performances of the update deteriorates over time in comparison to the entire retrain. Considering Alg. 1, we solved again the minimization problem of Eq. 1 optimizing φ with gradient descend algorithm. We noticed that after approximately 20 interactions the model's update was not updating features as expected. This is coherent with the errors behavior reported in (Rendle and Schmidt-Thieme 2008). As a consequence, each night we retrained the model, assuming students would see approximately 10 tasks per day. Given the large data availability we had no cold–start problem, which is experienced in MF when not enough data on tasks or on students are available. The task cold–start problem is not common to the movie rating applications since there the data availability is higher, but could be experienced in ITS use. For this reason it was crucial to have a partner with large data availability both on tasks and on students. We were able to select 100 students coming from the same school and that had already experience with the system, so that also the student cold–start problem could be avoided.

Vygotsky Policy Integration

The policy integration consisted first of all in selecting the threshold score y_{th} (see Eq. 3). In (Schatten and Schmidt-Thieme 2014) a sensitiveness analysis of the VP to y_{th} was done with simulated students. The same approach in a real scenario was considered detrimental for children, consequently the threshold score was selected according to the authors experience with the system and according to following considerations. In (Schatten and Schmidt-Thieme 2014) was discussed that, in order to keep the student in the ZPD, the selected score should be good, in order to be able to assume that the student was learning something from the task, but also not excellent, to avoid unnecessary repetitions on already known concepts. (Schatten et al. 2014a) suggested to select the threshold in the middle of the passing range, i.e. given a passing score of 5.5/10 one should select as threshold score 8/10. Our decision was made also on further considerations. Given the characteristics of the MF we assumed that the model was going to underestimate the performances of the student, so 8 as threshold score would have been too high. Moreover, experts, that were analyzing the tasks, suggested that the path through the curriculum could be ameliorated by removing unnecessary repetitions rather than increasing practice. Consequently, we decided to choose a threshold score $y_{th} = 6.5$, i.e. the lowest possible in the passing range keeping a safe guard in case of overestimation by the model.

A further policy adjustment was required since in exercise modality bottom out hints are available. We decided to consider only test predictions to evaluate students'ability. As a consequence only tests are selected with the VP. Whether or not the correspondent exercise should be shown is decided considering the performance prediction, if the score predicted is higher than 9.5/10 the exercise is skipped.

Technical Integration

As proposed in (Schatten et al. 2014b) we integrated the VPS within the ITS with a single method API. The minimal integration effort, visible in Alg. 2, was crucial to convince the commercial partner to invest time and effort integrating a sequencer still not fully evaluated. The parameters passed by the method Get_Next_Task have the following function. The student ID, the task ID and the relative score obtained are required in order to maintain the VPS database up to date. The method Get_Next_Task was called after the use of the old sequencer by the ITS but before the sequencing decision is applied as shown in Alg. 2. $NextTaskID$ contained the task suggestion of the state of the art sequencer, so that the VPS was able to manage the A/B test, deciding which students were practicing with which sequencer. Moreover, to be robust to connection problems we used a timestamp indicating when the data was recorded in order to avoid inserting duplicates in the DB. Moreover, a timeout variable was used at ITS side for connection problems.

This single method was exposed as Web Service. The resulting architecture of the system can be seen in Fig. 2, where the three blocks, student interfaces, ITS platform and VPS, are connected through the web.

Experiment Session

The purposes of the trial with the students were several. Firstly, we wanted to show that it is possible to sequence

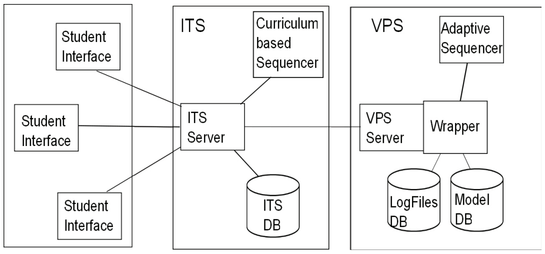


Figure 2: System Architecture

Algorithm 2: Implementing the Web Service, client side

```
Input: StudentID, PreviousTaskID, PreviousScore,
       TimeStamp
NextTask = Get_Next_Task_SotA();
if !timeout then
    Get_Next_Task(StudentID, PreviousTaskID,
    PreviousScore, NextTaskID, TimeStamp);
end
```

tasks by just considering students' score without frustrating them. Secondly, we wanted to evaluate sequencer performances in comparison with the current sequencer used by the ITS, which was adapted over the years to the tasks and countries curricula. In order to answer these questions we analyzed the following success criteria: learning gains evaluated with trial data analysis, comparison of questionnaire and post test scores with the SotA sequencer, and finally we evaluate integration performances. To demonstrate the last point is known to be problematic since the most used indicator is the learning gain of students. This is generally obtained with a large number of students and over extensive time. Schools agreed with us to let 98 children interact with the ITS 45 minutes a week for 4 weeks. Another appointment in the fifth week was granted for a 30 minutes post test and a five question questionnaire for the perceived experience. Students were able to practice also at home and use all other related features of the ITS, e.g. spend coins gained for passing tasks for decorating their virtual room. Of the 98 students that were assigned to the study we randomly assigned them to two groups one was practicing with the SotA sequencer and one with the VPS. Students did not know to which system they were assigned. Given the reduced amount of time we could not let students practice with all 20 topics in order to be able to monitor any learning gain. We selected 3 topics and one recall topic, i.e. Fractions, Properties of Numbers, Solving Problems as well as Rapid Recall on additions and subtractions. Moreover, we limited the VPS degree of freedom by defining an active range for each topic, i.e. we selected a subset of tasks between which the VPS could choose in order to limit difficulty jumps. The active range of each topic is initialized with the Math Age of the most difficult task of a topic a student could solve, this represents the center of the range. We then allowed only tasks around +/- one year Math Age from the center. Each time the student is able to solve a more difficult task in test mode the center of the active range is updated. Although from simulated experiments in (Schatten and Schmidt-Thieme 2014) an active range seemed not to be required we preferred to adopt this risk minimization procedure in order to avoid frustrating excessively the students in case of experiment failure. Experts considered the range adequate for an ethically correct experiment and large enough for being able to evaluate the ability of the VPS to construct a reasonable path. For introducing new topics we adopted the simple policy of showing them the easiest available task, further tasks on the topic are selected in the active range with the Vygotsky policy.

	VPS	SotA
Avg Math Age Improvement	0.06±0.038	0.03±0.0354
Avg Start Math Age	8.41±1.42	8.26±1.94
Avg End Math Age	9.72±1.92	8.84±2.02
Avg Tasks Score	6.82±0.901	7.9±0.992
Inter Topic StD	0.64 ±0.30	0.32±0.44
MF Error	0.317±0.10	-

Table 2: Trial Data Analysis. Values are indicated with ± standard deviation

Results from Data set Analysis

From the 98 students we filtered those that practiced on less than 10 tasks and/or did not participate to all tests having 80 students left. From the trial data analysis we could notice that there was no big usage difference. Both groups approximately saw 2000 tasks in a month. In order to have a learning gain comparison we computed the average Math Age per student and per topic at the beginning (Avg Start Math Age) and at the end (Avg End Math Age) of the experiment. We then normalized the difference, i.e. the students' learning gain, with the number of tasks seen, in order to exclude also amount of practice differences (Avg Math Age Improvement). As one can see from in Tab. 2 the average improvement of VPS students per task is double as much as the SotA ones. This proved that the VPS is able to propose tasks in a way that students can proceed in the curriculum also if this is composed by different topics. However, by observing Fig. 3 it is possible to see how the average standard deviation (Inter Topic StD) between topics' Math Age (Inter Topic Standard Deviation) is higher for those who interacted more with the system. This means that the inter knowledge standard deviation between topics' knowledge will increase until the student finishes the tasks of some topics, i.e. when the tasks of the mastered topics will be too easy to be in the ZPD and the VPS will select those of the uncompleted topics. The performance prediction was working correctly. As one can see in Tab. 2 the average score for VPS students is 6.82, i.e. the threshold score 6.5 plus a slight overestimation, as expected. The average score of the SotA students is coherent with those of the data used to train the model as one could see comparing Tab. 2, 1.

Post Test

The trial post-test comprised 15 questions; 5 corresponding to each of the three topics, excluding the recall. The questions were sourced by experts from the ITS library of tasks

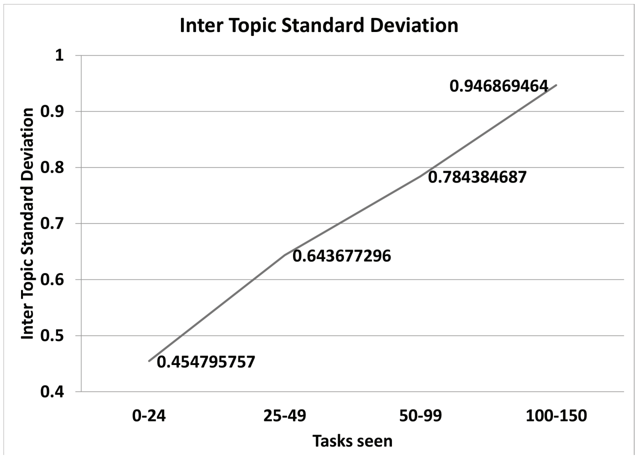


Figure 3: Inter topic standard deviation, i.e. average of the standard deviation between Topic Math Age for students that interacted with 10-24, 25-49, 50-99, or 100-150 tasks.

	Post Test Score	Average Math Age
VPS	6.68±1.53	9.94±0.60
SotA	6.70±1.79	9.96±0.71

Table 3: Post Test Comparison. Values are indicated with ± standard deviation

ensuring that the Math Age assigned to the task was consistent with their perceived difficulty of each chosen question, and that the questions range of difficulty level was broad and considering each difficulty level. Given the average age of student this range was from Math Age 7.5 to 11.0. All 15 questions were taken from the test tasks. These originally existed in paper form and were adapted to the online ITS tutor. Thus, converting the digital questions back to their original form required no work, and the questions lost no key form factors (such as digital interactivity) in the conversion process. For each question one point was assigned so that the sum of points obtained, normalized in a 0 to 10 range, represents the score of the post test. Moreover, taking the tests from the ITS curriculum, allowed the experts to reassess the student knowledge in order to compute the actual Math Age. As expected due to the short time period of the trial, no difference in average score and Math Age can be seen between the two groups (Tab. 3). This means that the sequencer does not damage learning. This result could appear at first glance in contrast with the average Math Age improvement, however the final average Math age reported in Tab. 2 of VPS students, i.e. the most difficult task students could solve, is more similar to those evaluated by experts in the post test. This means the VPS system was able to better model the current knowledge of the student and adapt to it.

Questionnaire

In order to evaluate the perceived experience of the students following questions were posed to them.

- Q1: Was the ITS fun?

	Q1	Q2	Q3	Q4	Q5
VPS	3.76 ±1.03	3.69 ±1.22	3.26 ±1.13	3.56 ±0.98	4 ±0.95
SotA	3.59 ±1.23	3.9 ±1.12	3.49 ±1.13	3.51 ±1.29	3.49 ±1.11

Table 4: Questionnaire comparison. 1: strong disagreement, 5: strong agreement. Values are indicated with ± standard deviation

- Q2: Were the exercises repetitive?
- Q3: Were the exercises easy?
- Q4: Was the ITS helpful?
- Q5: Was the ITS easy to understand?

The students could give a vote between 1 and 5 where 5 meant strong agreement and 1 strong disagreement.
As one can see in Tab.4 in almost all questions the VPS is slightly better than the SotA sequencer except from Q4 where the outcome is equal. In general the experience was positive, as one can see from Q1 and Q4. There where no usability issues related to introduction of the sequencer as reported from Q5. In Q3 students stated that tasks were between the adequate difficulty and too easy. This is coherent with the outcome of the post tests since the tasks proposed were too easy at the beginning. The VPS was better at adapting to the students' learn rate, so the sequence proposed by the VPS was perceived by the students to be of a more correct difficulty level. This agrees also with the data analysis where the average score of the VPS students is lower than the those of the SotA group. The only negative comment reported was the repetitiveness that could have been perceived by both groups for several reasons. Firstly because the set of questions in the tasks cannot be interrupted, secondly because the recall tasks are similar with one another but in the commercial version are presented interleaved with more topics.

Integration

We rented a server with 8 virtual CPUs, 30Gb RAM and two SSD of 80Gb each. The adapted VPS required $6s$ worst case to: identify the student, his appertaining group, update the model, and select the next task from the subset with the VP. The last action was not always necessary, if, for instance, the student had to practice with the correspondent test of an exercise or if he was of the SotA group and then the suggestion of the ITS needed just to be forwarded. The sequencing time could be further reduce by indexing the DB, but we did not require to do so, since response times were already comparable to those of the current SotA sequencer. The implementation was tested with 30 students practicing at the same time, but we do not exclude it could work with more.

Conclusions and Future Work

In this paper we showed how we integrated the Vygotsky sequencer in a commercial ITS and we analyzed its potential from different perspectives. Although there was a high standard deviation due to the small sample, results are promising, showing how the Vygotsky sequencer was able to better model the students'knowledge. This result is coherent with the results of the questionnaire and the post test. The latter tests also show that the VPS is not damaging learning and that children had an experience comparable with the SotA sequencer, a sequencer that was modeled by experts over the years. Our future planned work is to test with more students and for a longer period of time so that we get the chance to monitor learning gains also in the post test. Moreover, we will develop a policy for topic sequencing.

Acknowledgments

The research leading to the results reported here has received funding from the European Union Seventh Framework Programme (FP7/20072013) under grant agreement No. 318051 iTalk2Learn project (www.italk2learn.eu). Furthermore, we thank our project partners the London Institute of Education and Whizz, in particular Junaid Mubeen, who organized the trial with the students and helped preparing the post assessment and the perceived experience questionnaire.

References

Beck, J.; Woolf, B. P.; and Beal, C. R. 2000. Advisor: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI* 2000:552–557.

Brusilovsky, P. 2001. Adaptive hypermedia. *User modeling and user-adapted interaction* 11(1-2).

Chi, M.; VanLehn, K.; Litman, D.; and Jordan, P. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *UMAI*.

Cichocki, A.; Zdunek, R.; Phan, A. H.; and Amari, S.-i. 2009. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley. com.

Corbett, A., and Anderson, J. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMAI*.

D Baker, R. S.; Corbett, A. T.; and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *ITS*, 406–415. Springer.

Janning, R.; Schatten, C.; and Lars, S.-T. 2014. Feature analysis for affect recognition supporting task sequencing. In *ECTEL*.

Janning, R.; Schatten, C.; and Schmidt-Thieme, L. 2014. Multimodal affect recognition for adaptive intelligent tutoring systems. In *FFMI EDM*.

Koedinger, K.; Pavlik, P.; Stamper, J.; Nixon, T.; and Ritter, S. 2011. Avoiding problem selection thrashing with conjunctive knowledge tracing. In *EDM*.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37.

Malpani, A.; Ravindran, B.; and Murthy, H. 2011. Personalized intelligent tutoring system using reinforcement learning. In *FLAIRS*.

Manouselis, N.; Drachsler, H.; Vuorikari, R.; Hummel, H.; and Koper, R. 2011. Recommender systems in technology enhanced learning. In *Recommender systems handbook*. Springer. 387–415.

Pardos, Z. A., and Heffernan, N. T. 2010. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *UMAP*. Springer.

Pardos, Z. A., and Heffernan, N. T. 2011. Kt-idem: introducing item difficulty to the knowledge tracing model. In *UMAP*. Springer. 243–254.

Pavlik, P., C. H., and Koedinger, K. 2009. Performance factors analysis-a new alternative to knowledge tracing. In *AIED*.

Rendle, S., and Schmidt-Thieme, L. 2008. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, 251–258. ACM.

Schatten, C., and Schmidt-Thieme, L. 2014. Adaptive content sequencing without domain information. In *CSEDU*.

Schatten, C.; Mavrikis, M.; Janning, R.; and Schmidt-Thieme, L. 2014a. Matrix factorization feasibility for sequencing and adaptive support in its. In *EDM*.

Schatten, C.; Wistuba, M.; Schmidt-Thieme, L.; and Gutirrez-Santos, S. 2014b. Minimal invasive integration of learning analytics services in its. In *ICALT*.

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

Thai-Nghe, N.; Drumond, L.; Horvath, T.; Krohn-Grimberghe, A.; Nanopoulos, A.; and Schmidt-Thieme, L. 2011. Factorization techniques for predicting student performance. *Educational Recommender Systems and Technologies: Practices and Challenges. IGI Global*.

Thai-Nghe, N.; Drumond, L.; Horvath, T.; and Schmidt-Thieme, L. 2012. Using factorization machines for student modeling. In *UMAP Workshops*.

Vygotsky, L. L. S. 1978. *Mind in society: The development of higher psychological processes*. HUP.

Wang, Y., and Heffernan, N. T. 2012. The student skill model. In *ITS2012*.