Online Optimization of Teaching Sequences with Multi-Armed Bandits

Benjamin Clément, Didier Roy, Pierre-Yves Oudeyer, Manuel Lopes

Online Optimization of Teaching Sequences with Multi-Armed Bandits

Benjamin Clement Didier Roy Pierre-Yves Oudeyer

Manuel Lopes
Inria Bordeaux Sud-Ouest,
France
{FirstName.LastName}@inria.fr

ABSTRACT

We present an approach to Intelligent Tutoring Systems which adaptively personalizes sequences of learning activities to maximize skills acquired by each student, taking into account limited time and motivational resources. At a given point in time, the system tries to propose to the student the activity which makes him progress best. We introduce two algorithms that rely on the empirical estimation of the learning progress, one that uses information about the difficulty of each exercise **RiARiT** and another that does not use any knowledge about the problem **ZPDES**.

The system is based on the combination of three approaches. First, it leverages recent models of intrinsically motivated learning by transposing them to active teaching, relying on empirical estimation of learning progress provided by specific activities to particular students. Second, it uses state-of-the-art Multi-Arm Bandit (MAB) techniques to efficiently manage the exploration/exploitation challenge of this optimization process. Third, it leverages expert knowledge to constrain and bootstrap initial exploration of the MAB, while requiring only coarse guidance information of the expert and allowing the system to deal with didactic gaps in its knowledge.

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) have been proposed to make education more accessible, more effective and simultaneously as a way to provide useful objective metrics on learning. Recently, online learning systems have further raised the interest in these systems and several recent projects started on Massive Open Online Course (MOOC) for web-based teaching of university level courses. For a broad coverage on the field of ITS see [9] and [13].

According to [13], there are four main components of an ITS: i) a *cognitive model* that defines the domain knowledge or which steps need to be made to solve problems in a particular domain; ii) a *student model* that considers how students learn, what is the evolution of their cognitive state depending on particular teaching activities; iii) a *tutoring model* that defines, based on the cognitive and the student model, what teaching activities to present to students and iv) a *user interface model* that represents how the interaction with the students occurs and how problems are proposed to the learners.

In this work we are more focused on the *tutoring model*, that is, how to choose the activities that provide a better learning experience based on the estimation of the student competence levels and progression, and some knowledge about the cognitive and student model. We can imagine a student wanting to acquire many different skills, e.g. adding, subtracting and multiplying numbers. A teacher can help by proposing activities such as: multiple choice questions, abstract operations to compute with a pencil, games where items need to be counted through manipulation, videos, or others. The challenge is to decide what is the optimal sequence of activities that maximizes the average competence level over all skills.

There are several approaches to develop a *Tutoring Model*. A first approach is based on hand-made optimization and on pedagogical theory, experience and domain knowledge. There are many works that followed this line, see the recent surveys on the field by [9, 13]. A second approach considers particular forms of knowledge to be acquired and creates didactic sequences that are optimal for those particular classes of problems [2, 6, 7]. A third approach, and more relevant for our work, is that the optimization is made automatically without particular assumptions about the students or the knowledge domain. The framework of partial-observable Markov decision process (POMDP) has been proposed to select the optimal activities to propose to the students based on the estimation of their level of acquisition of each KC [14].

Our ITS aims at providing to each particular student the activities that are giving the highest learning progress. We do not consider that these activities are necessarily the ones defined a-priori in the cognitive and student model, but the ones that are estimated, at runtime and based on the students results, to provide the maximum learning gain. This approach has three main advantages:

Weaker dependency on the cognitive/student model
In most cases the tutoring model incorporates the student model inside. Given students' particularities, it is often highly difficult or impossible for a teacher to understand

all the difficulties and strengths of individual students and thus predict which activities provide them with maximal learning progress. Also, typically, these models have many parameters, and identifying all such parameters for a single student is a very hard problem due to the lack of data, the intractability of the problem and the lack of identifiability of many parameters that often results in models which are inaccurate in practice [3]. It has been shown that a sequence that is optimal for the average student is often suboptimal for most students, from the least to the most skilled [11].

We consider that it is important to be as independent as possible of the cognitive and student model when deciding which activities to propose. This requires that the ITS explores and experiments various activities to estimate their potential for learning progress for each student. The technical challenge is that these experiments should be not just sufficiently informative about the student's current competence but also to evaluate the effectiveness of each exercise to improve those competences (a form of stealth assessment [16]).

Efficient Optimization Methods We will rely on methods that do not make any specific assumptions about how students learn and only require information about the estimated learning progress of each activity. We make a simple assumption that activities that are currently estimated to provide a good learning gain, must be selected more often. A very efficient and well studied formalism for these kind of problems is Multi-Armed Bandits [5]. Following a casino analogy, at each step we can choose a slot-machine and we get to observe the payback we get, the goal it to find the best arm, but while we are trying to discover it we have to bet to test them.

More Motivating Experience Our approach considers that, at each time instance, the exercises that are providing the higher learning progress must be the ones proposed. This allows not only to use more efficient optimization algorithms but also to provide a more motivating experience to students. Several strands of work in psychology [4] and neuroscience [8] have argued that the human brain feels intrinsic pleasure in practicing activities of optimal difficulty or challenge, i.e. neither too easy nor too difficult, but slightly beyond the current abilities, also known as the zone of proximal development [10].

Our main contributions, when compared to other ITS systems, are: the use of highly performing Multi-Armed Bandit algorithms [5]; a simpler factored representation of the cognitive model that maps activities to the minimum necessary competence levels; and considering that the acquisition of a KC is not a binary variable but defined as the level of comprehension of that KC. The advantage of using MAB is that they are computationally efficient and require a weaker dependency between the tutoring and the cognitive and student models. Other contributions include an algorithm to estimate student competence levels; and the empirical learning progress of each activity. An extended version of this article is available at [12] including an initial user study.

2. ITS WITH MULTI-ARMED BANDITS

2.1 Relation between KC and pedagogical activities

In general, activities may differ along several dimensions and may take several forms (e.g. video lectures with questions at the end, or interactive games or exercises of various types). Each activity can provide opportunities to acquire different skills/knowledge components (KC), and may contribute differentially to improvement over several KCs (e.g. one activity may help a lot in progressing in KC_1 and only little in KC_2). Vice versa, succeeding in an activity may require to leverage differentially various KCs. While certain regularities of this relation may exist across individuals, it will differ in detail for every student.

First, we model here the competence level of a student in a given KC as a continuous number between 0 and 1 (e.g. 0 means not acquired at all, 0.6 means acquired at 60 percent, 1 means entirely acquired). We denote c_i the current estimation of this competence level for knowledge unit KC_i. In what we call a R Table, for each combination of an activity $\mathbf{a}$ and a KC_i, the expert then associates a $q-$value $(q_i(\mathbf{a}))$ which encodes the competence level required in this KC_i to have maximal success in this activity $\mathbf{a}$. This in turn provides a upper and lower bound on the competence level of the student: below $q_i(\mathbf{a})$ in case of mistake; above $q_i(\mathbf{a})$ in case of answering correctly.

We start by assuming that each activity is represented by a set of parameters $\mathbf{a} = (a_1, ..., a_{n_a})$. The R Table then uses a factorized representation of activity parameters, where instead of considering all $(\mathbf{a}, KC_i)$ combinations and their corresponding $q_i(\mathbf{a})$, we consider only (a_j, KC_i) combinations and their corresponding $q_i(a_j)$ values, where $q_i(a_j)$ denotes the competence level in KC_j required to succeed entirely in activity $\mathbf{a}$ which $j-th$ parameter value is a_j. This factorization makes the assumption that activity parameters are not correlated. The alternative would require a larger number of parameters and would also require more exploration in the optimization algorithm. We use the factorized R Table in the following manner to heuristically estimate the competence level $q_i(\mathbf{a})$ required in KC_i to succeed in an activity parameterized with $\mathbf{a}$: $q_i(\mathbf{a}) = \prod_{j=1}^{n_a} q_i(a_j)$

2.2 Estimating the impact of activities over students' competence level in knowledge units

Key to the approach is the estimation of the impact of each activity over the student's competence level in each knowledge unit. This requires an estimation of the current competence level of the student for each KC_i. We do not want to introduce regular tests that might interfere negatively with the learning experience of the student. Thus, competence levels need to be inferred through stealth assessment [16] that uses indirect information from the results on the exercises.

When doing an activity $\mathbf{a} = (a_1, ..., a_{n_a})$, the student can either succeed or fail. In the case of success, if the estimated competence level c_i in knowledge unit i is lower than $q_i(\mathbf{a})$, we are underestimating the competence level of the student in KC_i, and so should increase it. If the student fails and $q_i(\mathbf{a}) < c_i$, then we are overestimating the competence level of the student, and it should be decreased. For these two first cases we can define a reward:

$$r_i = q_i(\mathbf{a}) - c_i \quad (1)$$

and use it to update the estimated competence level of the student according to $c_i = c_i + \alpha r_i$ where α is a tunable

parameter that allows to adjust the confidence we have in each new piece of information.

A crucial point is that the quantity $r_i = q_i(\mathbf{a}) - c_i$ is not only used to update c_i, but is used to generate an internal reward $r = \sum r_i$ to be cumulatively optimized for the ITS (details below). Indeed, we assume here that this is a good indicator of the learning progress over KC_i resulting from doing an activity with parameters **a**. The intuition behind this is that if you have repeated successes in an activity for which the required competence level is higher than your current estimated competence level, this means you are probably progressing.

2.3 RiARiT: Right Activity at Right Time

To address the optimization challenge for ITS, we will rely on multi-arm bandit techniques (MAB)[5]. A particularity here is that the reward (learning progress) is non-stationary, which requires specific mechanisms to track its evolution. Indeed, here a given exercise will stop providing reward, or learning progress, after the student reaches a certain competence level. Also we cannot assume that the rewards are i.i.d. as different students will have different preferences and many human factors, i.e. distraction, mistakes on using the system, create several spurious effects. Thus, we rely here on a variant of the EXP4 algorithm [1, 5]. We consider a set of filters that track how much reward each exercise parameters is giving. Then the algorithm selects stochastically the teaching activities proportionally to the expected learning progress for each parameter.

Expert knowledge can also be used by incorporating *coarse* global constraints on the ITS. Indeed, for example the expert knows that for most students it will be useless to propose exercises about decomposition of real numbers if they do not know how to add simple integers. Thus, the expert can specify minimal competence levels in given KC_i that are required to allow the ITS to try a given parameter a_j of activities.

2.4 ZPDES: Zone of Proximal Development and Empirical Success

Our goal is to reduce the dependency on the cognitive and student models and so we will try to simplify further the algorithm. Our simplification will take two sources of inspiration: **zone of proximal development** and the **empirical estimation of learning progress**.

As discussed before focusing teaching in activities that are providing more learning progress can act as a strong motivational cue. Estimating explicitly how the success rate on each exercise is improving will remove the dependency on the R table. For this we replace Eq. 1 with $r = \sum_{k=1}^{t} \frac{C_k}{t} - \sum_{k=1}^{t-d} \frac{C_k}{t-d}$ where $C_k = 1$ if the exercise at time k was solved correctly. The equation compares the $d+1$ more recent success with all the previous past, providing an empirical measure of how the success rate is increasing. We no longer estimate the competence level of the student, and directly use the reward estimation.

The other inspiration is the concept of the zone of proximal development [10] that considers that activities that are slightly beyond the current abilities of the learner are the more motivating. This concept will provide three advantages: improve motivation; further reduce the need of quantitative measures for the educational design expert; and provide sequence of activities that follow a more sequential order. A first point is that there are some parameters that have a clear relation of increasing complexity (such as the parameter exercise type) and should be treated differently than other parameters that do not have such ordering (for instance the complexity in the modality presentation will change depending on each student and not on the problem itself). A final point is that we are choosing exercises based on the estimated (recent) past learning progress, and if we know which exercise is next in terms of complexity then we can use that one. This information, if correct, allows the MAB to propose the more complex exercises without requiring to estimate their value first. Providing a more predictive behavior and not just relying on the recent past.

This algorithm is identical to RiARiT but we treat the parameters that have a clear relation of increasing complexity differently. For the parameter i, when the expected learning progress of parameter j is below the level of the more complex parameter value, $w_i(j) < w_i(j+1)/\theta$, and the success rate is higher than a pre-defined threshold : $\sum_{k=1}^{t} \frac{C_k(j)}{t} > \omega$, we allow the parameter value $j+3$ to be chosen and initiate it with: $w_i(j) = 0$ and $w_i(j+3) = w_i(j+2)$.

3. TEACHING SCENARIO

We will now describe a specific teaching scenario about learning how to use money, typically targeted to students of 7-8 years old. The parameters of the activities are commonly used in schools for acquiring these competences and there are already well studied teaching sequences validated in several studies [15].

In each exercise, one object is presented with a given tagged price and the learner has to choose which combination of bank notes, coins or abstract tokens need to be taken from the wallet to buy the object, with various constraints depending on exercises parameters. The five Knowledge Components aimed at in these experiments are:
KnowMoney: Global skill characterizing the capability to handle money to buy objects in an autonomous manner; **SumInteger**: Capability to add and subtract integer numbers; **DecomposeInteger**: Capability to decompose integer numbers into groups of 10 and units; **SumCents:** Capability to add and subtract real numbers (cents); **DecomposeCents**: Capability to decompose real numbers (cents); **Memory**: Capability to memorize a number which is presented and then removed from visual field.

The various activities can be parameterized with the following properties: **Exercise Type** depending on the complexity of decomposing a price[1] that can be read directly by making the correspondence to a real note/coin $a = (1, 2, 5)$ and those that need a decomposition that requires more than one item $b = (3, 4, 6, 7, 8, 9)$. The exercises will be generated by choosing prices with these properties in a set of six levels of increasing difficulty and picking an object that is priced realistically.; **Price Presentation**: i) written and spoken; ii) written; iii) spoken; **Cents Notation**: i) x€x; ii) x, x€; **Money Type**: i) Real euros; ii) Money Tokens.

[1]In the euro money system the money items (bills and coins) have the values 1, 2 and 5 for the different scales.

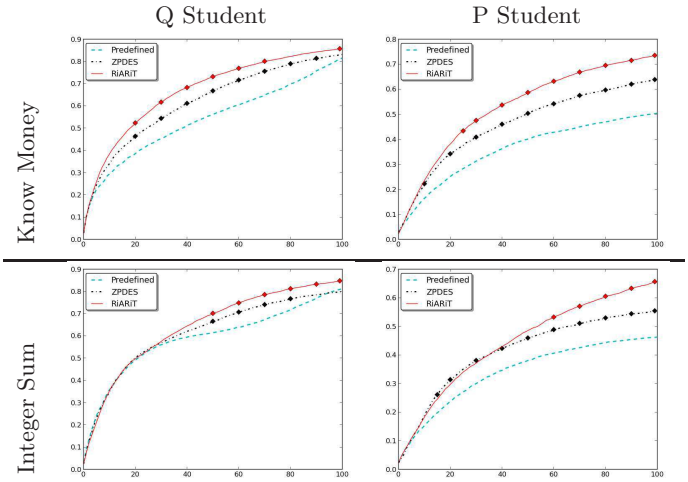

Figure 1: The evolution of the comprehension of two knowledge components with time for population . Markers on the curve mean that the difference is significative.

4. SIMULATIONS

We present a set of simulations with virtual students. We consider two populations. A population "Q" where the students have different learning rates and maximum comprehension levels for each KC and another population "P" where, in addition to this, the students have limitations in the comprehension of specific parameterizations of the activities. We expect that in the population "Q" an optimization will not provide big gains because all students are able to use all exercises to progress. On the other hand, the population "P" will require that the algorithm finds a specific teaching sequence for each particular student. We note that the algorithm itself is not provided with any a-priori information about the properties of the students. We present here the results showing how fast and efficiently our algorithms estimate and propose exercises at the correct level of the students. Each experiment considers a population of 1000 students generated using the previous methods and lets each student solve 100 exercises.

Figure 1 shows the skill's levels evolution during 100 steps. For Q student, learning with RiARiT and ZPDES is faster than with the predefined sequence, but at the end, Predefined catch up with ZPDES. For P simulations, as students can not understand particular parameter values, they block on stages where the predefined sequence does not propose exercises adequate to their level, while ZDPES, by estimating learning progress, and RiARiT, by considering the estimated level on all KC and parameter's impact, are able to propose more adapted exercises.

5. CONCLUSIONS AND FUTURE WORK

In this work we proposed a new approach to intelligent tutoring systems. We showed through simulations and empirical results that a very efficient algorithm, that tracks the learning progress of students and proposes exercises proportionally to the learning progress, can achieve very good results. Using as baseline a teaching sequence designed by an expert in education [15], we showed that we can achieve comparable results for homogeneous populations of students, but a great gain in learning for populations of students with larger variety and stronger difficulties. In most cases, we showed that it is possible to propose different teaching sequences that are fast to adapt and personalized. We introduced two algorithms RiARiT that uses some information about the difficulty about the task, an another algorithm ZPDES that does not use any information about the problem. It is expected that RiARiT, as it uses more information, behaves better when the assumptions are valid, while ZPDES, without any information can not achieve as high performance in well behaved cases but is surprisingly good without any information. Even when compared with a hand optimized teaching sequence ZPDES shows better adaptation to the particular students' difficulties.

6. REFERENCES

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2003.

[2] F. J. Balbach and T. Zeugmann. Recent developments in algorithmic teaching. In *Inter. Conf. on Language and Automata Theory and Applications*, 2009.

[3] J. E. Beck and X. Xiong. Limits to accuracy: How well can we do at student modeling? In *Educational Data Mining*, 2013.

[4] D. Berlyne. *Conflict, arousal, and curiosity.* McGraw-Hill Book Company, 1960.

[5] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Stochastic Systems*, 1(4), 2012.

[6] M. Cakmak and M. Lopes. Algorithmic and human teaching of sequential decision tasks. In *AAAI Conference on Artificial Intelligence*, 2012.

[7] J. Davenport, A. Rafferty, M. Timms, D. Yaron, and M. Karabinos. Chemvlab+: evaluating a virtual lab tutor for high school chemistry. In *Inter. Conf. of the Learning Sciences (ICLS)*, 2012.

[8] J. Gottlieb, P.-Y. Oudeyer, M. Lopes, and A. Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in Cognitive Sciences*, 17(11):585–593, 2013.

[9] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 2013.

[10] C. D. Lee. Signifying in the zone of proximal development. *An introduction to Vygotsky*, 2:253–284, 2005.

[11] J. Lee and E. Brunskill. The impact on individualizing student models on necessary practice opportunities. In *Inter. Conf. on Educational Data Mining*, 2012.

[12] M. Lopes, B. Clement, D. Roy, and P.-Y. Oudeyer. Multi-armed bandits for intelligent tutoring systems. *arXiv:1310.3174 [cs.AI]*, 2013.

[13] R. Nkambou, R. Mizoguchi, and J. Bourdeau. *Advances in intelligent tutoring systems*, volume 308. Springer, 2010.

[14] A. Rafferty, E. Brunskill, T. Griffiths, and P. Shafto. Faster teaching by pomdp planning. In *Artificial Intelligence in Education*, 2011.

[15] D. Roy. Usage d'un robot pour la remédiation en mathématiques. Technical report, 2012.

[16] V. J. Shute. Stealth assessment in computer-based games to support learning. *Computer games and instruction*, 55(2):503–524, 2011.