Exercises recommending for limited time learning

Pavel Michlík[a], Mária Bieliková[a,]*

[a]*Faculty of Informatics and Information Technologies, Slovak University of Technology, Ilkovičova 3, 842 16 Bratislava, Slovakia*

Abstract

In this paper we propose a method for personalized recommendation of assignments (tasks or exercises) in an adaptive educational system. Our main goal is to help students to achieve better performance in tests. To achieve this we enhance existing adaptive navigation approaches by considering the limited time for learning. Our strategy is to cover all the required topics at least to some extent rather than learn few topics perfectly. The proposed method uses utility-based recommending and concept-based knowledge modeling. We evaluate our approach in the domain of learning programming.

Keywords: Recommendation; educational system; exercise; limited time

1. Introduction

The objective of personalized recommendation in educational systems is to help students choose the best topic (learning object) to focus on, in order to maximize the learning efficiency. A personalization is inevitable as each student can have different level of knowledge concerning particular topics, different abilities, but also different ambitions and also an idea of time intended for learning. Adaptive navigation and personalized topics recommendation is currently common approach in support of effective education. Most educational systems recommend the most appropriate text for particular user considering learning goals, actual level of knowledge and often also activity of peers.

In this paper we focus on the educational content in form of exercises. Exercises are an important part of educational materials in many areas (e.g. math, or programming). They serve for practicing the concepts presented often in form of educational text (or multimedia). Often exercises compose main part of midterm exams. Since the time for exams preparation is always limited by the date of the exam, it might be not sufficient for learning all required topics perfectly, especially in case when a student has started his preparation a bit late.

Our goal is to help the student achieve as good exam result as possible. A common tactic used by students under time pressure is going through all topics very quickly rather than learning at least few topics in detail. Using such a strategy often there is no topic learnt at minimal level, i.e. the level which allows producing right answer for at least some tasks given during the exam. Our method for personalized recommendation presented in this paper is designed to help the students to prepare for the exam by covering as many topics as possible in limited time. It is implemented and evaluated as a part of the ALEF framework for adaptive web-based learning [1].

---

* Mária Bieliková. Tel.: +421-2-60291-473; fax: +421-2-65420-587.
E-mail address: bielik@fiit.stuba.sk

2. Related work

There exist many different approaches to adaptive navigation in educational systems, both content-based and collaborative. We are especially concerned with approaches in domain of learning programming.

In [2] the authors present a combination of content-based and collaborative approaches – a method based on learning objects content and good learners rating.

Brusilovsky et al. in [3] present an educational web-based system with adaptive navigation within exercises. They use personalized annotations for indication of the state of exploration of educational space by the student. An annotation symbol is assigned to each exercise depending on the portion of its source code that has not been explored by the student yet. No recommendation is made by the system – the student selects next exercise and his learning speed (the amount of new knowledge in each exercise) by himself.

The ALEA educational system for learning programming [4] provides adaptive navigation based on relationships in the concept layer of the domain model. The program examples (exercises) are presented as specific instances of program schemata which facilitate understanding of basic programming principles.

In [5] the authors present a method for adaptive selection of comprehension checking questions and its implementation in the system called Flip. Recommendation is based on Item Response Theory and utilizes the computer-adaptive testing method [6] for knowledge modeling. However, the system uses topic-based knowledge modeling and lacks a separate concept layer in the domain model, which does not facilitate effective representation of recommendation knowledge and recommendation in several related courses that share the concepts [7].

Educational content recommendation in [8] filters learning objects according to their relevance to the student's learning goals using a domain ontology organized in a hierarchy. Learning paths are generated using concept prerequisites.

Existing solutions present effective means for learning support. However, none of these navigation solutions consider time in their recommendation strategies or support in some way the preparation for a test considering limited time of a student for learning. The decision about when to stop learning a topic and move to another is left to the student. Our method for personalized recommendation of exercises is intended for providing assistance just in this case.

3. Models overview

We employ the domain model developed for the ALEF (Adaptive LEarning Framework) [1]. The domain model consists of content entities (*learning objects*) and metadata entities (*concepts*) that are connected via various types of relationships (see Fig. 1). Learning objects are further divided into the following types:

- *Explanation*, represents instructional content that describes a subject domain;
- *Question*, represents interactive part of a course with several types of questions (e.g., single choice, multi choice, text filling, sorting);
- *Exercise,* represents means for practicing knowledge in the explanation learning objects.

Every learning object has a fuzzy set of *related concepts* and a scalar *difficulty* property. Different types of relations between concepts are defined: similarity, prerequisites, *is-a* relation. All relations are weighed – a real number from $\langle 0, 1 \rangle$ sets the relation strength.

The students' knowledge model in ALEF is a concept-based [1] vector space model [9]. It consists of a set of *knowledge levels* – a student's knowledge of each concept in the domain model is represented by a real number from 0 (not understood) to 1 (perfectly understood).

An exercise consists of three parts

- task definition,
- hint and
- possible solution (with a brief explanation for clarification purposes if appropriate).

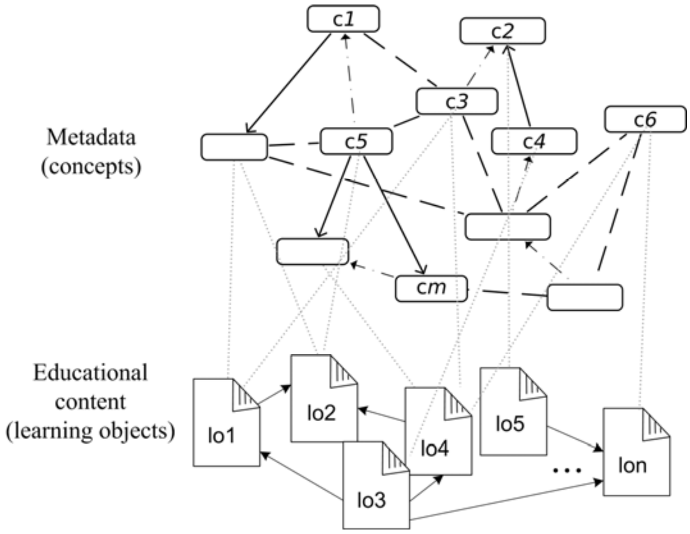

Fig. 1. An example of the domain model of ALEF. Core metadata entities – concepts – are assigned to learning objects. Concepts, like other metadata entities, are interlinked via various types of relationships.

After each attempt to solve an exercise, the student can provide a feedback, whether he could solve the exercise or at least understand the presented solution.

Four feedback options are possible:

1. the student has solved the exercise correctly,
2. the student has solved the exercise correctly with a help of the hint,
3. the student could not solve the exercise correctly, but understood the presented solution or
4. the student could not solve the exercise correctly, nor did he understand the solution.

Each of these feedback values has a corresponding knowledge level – the result of knowledge "measurement" which is carried out by the particular exercise. The third feedback option (understood the solution after not being able to solve the exercise) is assigned the knowledge level equal to the difficulty of particular exercise. We believe that this feedback value is the case where the student's knowledge exactly matches the exercise content and its difficulty.

After receiving a feedback, the student's concept-based knowledge model is updated. For updating the knowledge model we use the computer-adaptive testing method [6] with the exercise being a polytomous item. The knowledge levels of all concepts related to the exercise are moved towards the values given by the particular feedback option (with a coefficient lower than 1 to prevent rapid changes – the knowledge level is never moved all the way to the measured level).

The knowledge level change depends on the membership value of the particular concept in the exercise related concept set – knowledge levels of strongly related concepts are set closer to the measured level. Each change of concept knowledge is distributed to all related concepts according to the relation weight using the spreading activation algorithm.

In addition to the knowledge model, ALEF also has a history-based user model, which contains information about the content that the user accessed as well as the user's feedback on that content.

4. Expected knowledge levels for limited time learning

For an upcoming exam, *learning targets* are defined – a set of concepts which should be learned by the students to pass the test, each with a weight representing the concept's importance. In our experiments, these weights were set manually by a domain expert (teacher). To achieve proper learning time distribution between all required concepts, we attempt to estimate optimal knowledge levels of all concepts at the end of learning time. These knowledge levels should be achievable at the student's current learning speed and respect the concepts importance.

At the same time we need to prevent learning too many concepts to an inappropriate level, which otherwise could be a result of little time for learning and low entry knowledge. In this case, the tactic of learning as many concepts as possible is no longer acceptable.

We calculate the sum of all increases of the student's concept knowledge levels from the start of the learning session. Assuming steady progress over time, we use linear extrapolation to estimate the sum of potential knowledge level increases from present time to the end of learning. The potential overall increase is then divided between all concepts so that the final estimated knowledge levels of concepts correspond with the concepts importance. Every pair of concepts *(i, j)* has to satisfy the condition

$$\frac{\theta_i + \Delta\theta_{Ei}}{\theta_j + \Delta\theta_{Ej}} = \frac{G_i}{G_j}$$

where θ_i is the current knowledge of concept i, $\Delta\theta_{Ei}$ is the expected knowledge increase and G_i is the concept importance.

To prevent learning many concepts to an inappropriate level, we set a *minimal expected concept knowledge level.* The estimated knowledge level for every concept can never fall below this limit. Therefore, learning of a concept will not stop before the student's knowledge level reaches this minimum.

5. Criteria for recommendation

Our approach can be categorized as a *content-based* method according to [9] or [10] (it utilizes a user requirements model and properties of recommended items rather than other users' previous feedback) or as a *utility-based* method in finer classification introduced by Burke [11]. The recommender algorithm type is *memory-based* [9] and it adopts *attribute-based* technique [12]. The supported task [9] in our recommender system is to *find good items* – the system displays a list of *N* most suitable exercises.

To make personalized recommendation for a student, each exercise in the course is evaluated and assigned a scalar *appropriateness value* (or utility value) from ⟨0, 1⟩. Three criteria are used for each exercise evaluation:

- concept appropriateness for the student,
- exercise difficulty appropriateness and
- time period since the student's last attempt to solve the exercise.

These criteria are orthogonal and all of them are supposed to be met. Therefore, the final appropriateness of the exercise for the student is computed as the minimum of the three partial results. If we consider the partial results to be membership functions of fuzzy sets of appropriate exercises produced by these criteria, the final set of appropriate exercises is the intersection of those sets. This approach is a *value-focused* type of global preference model [13]. After evaluating all exercises in the course, one or more exercises with largest appropriateness values are chosen as recommendations for the student.

5.1. Concept appropriateness

In concept appropriateness evaluation we decide whether the student should learn the concepts covered by the particular exercise. Here, the exercise is characterized by its related concepts set and is compared to the user (student) profile – a fuzzy set of concepts which are appropriate for the student to learn at the moment.

The appropriateness value of a concept is determined by following three criteria:

5. *The concept is a member of the learning targets set.*
6. *The student's current knowledge level is less than the estimated optimal level.* Over-learning a concept would result in less achievable knowledge of other concepts. Therefore, a steep sigmoid appropriateness function is used to suppress further learning of a concept once the student reaches the estimated knowledge level.
7. *The concept's prerequisites (required knowledge of other concepts) are met.*

The required entry knowledge for learning a concept is represented by weighted prerequisite relations in the concept layer of the domain model. The higher the weight of a relation, the higher knowledge level of the corresponding concept is required to consider the prerequisite to be met. For each concept *i* which has an expected knowledge level θ_{Ei} and which depends on knowledge of concept j, we calculate the degree of fulfilling the prerequisite

$$R_{ij} = \min\left(1, \frac{\theta_j}{\theta_{Ei}} - p_{ij} + 1\right)$$

where θ_j is the student's knowledge level of concept j and p_{ij} is the weight of the prerequisite relation. For multiple prerequisites of one concept, all prerequisites are required to be met before learning the concept. Therefore, the final readiness for learning this concept is determined by the minimum of all corresponding R_{ij} values.

5.2. Exercise difficulty appropriateness

The exercise difficulty appropriateness evaluation ensures that the difficulty of the recommended exercise matches the student's knowledge of corresponding concepts, thus preventing the student from being uninterested or discouraged. Recommending exercises with their difficulty near the student's current knowledge level also improves user modeling because that is the case with the greatest information value [6]. For computing exercise difficulty appropriateness we utilize a Gaussian function with its peak set to the student's knowledge level

$$D = e^{\frac{(d-\theta)^2}{2B^2}}$$

where d is the difficulty of the exercise, θ is the student's knowledge level and B is a constant determining the function steepness.

The knowledge level θ is an aggregation (weighted arithmetic mean) of student's knowledge levels of concepts which are related to the particular exercise. The appropriate B value depends on the number of exercises available and the difficulty distribution – few exercises and a steep Gaussian function would soon result in very low appropriateness results in this criterion (early repeating of exercises is suppressed) and recommendation would be degraded (partial criteria results are aggregated via minimum).

5.3. Repeating exercises

This final criterion considers the time period since the student last attempted to solve the particular exercise. Repeating the same exercise after a short interval is not suitable, therefore recommending recent exercises has to be suppressed. After visiting an exercise, its appropriateness value produced by this criterion drops to zero and gradually returns to 1 over time. For this purpose we chose a hyperbolic function

$$H = 1 - \frac{1}{(C \times t) + 1}$$

where t is the time period since the last attempt and C determines the function steepness. The value of C parameter can depend on the student's feedback on the last attempt – negative feedback will result in repeating the exercise sooner. In our experiments we did not use different C values according to the feedback values because the controlled learning sessions were very short with enough exercises available, so we did not expect the exercises to repeat at all.

6. Evaluation

We evaluated our method for recommending exercises in experiments within the Functional and logic programming course in academic year 2009/2010, using the ALEF adaptive learning framework. In addition to exercises, test questions provided by our educational course were recommended as well.

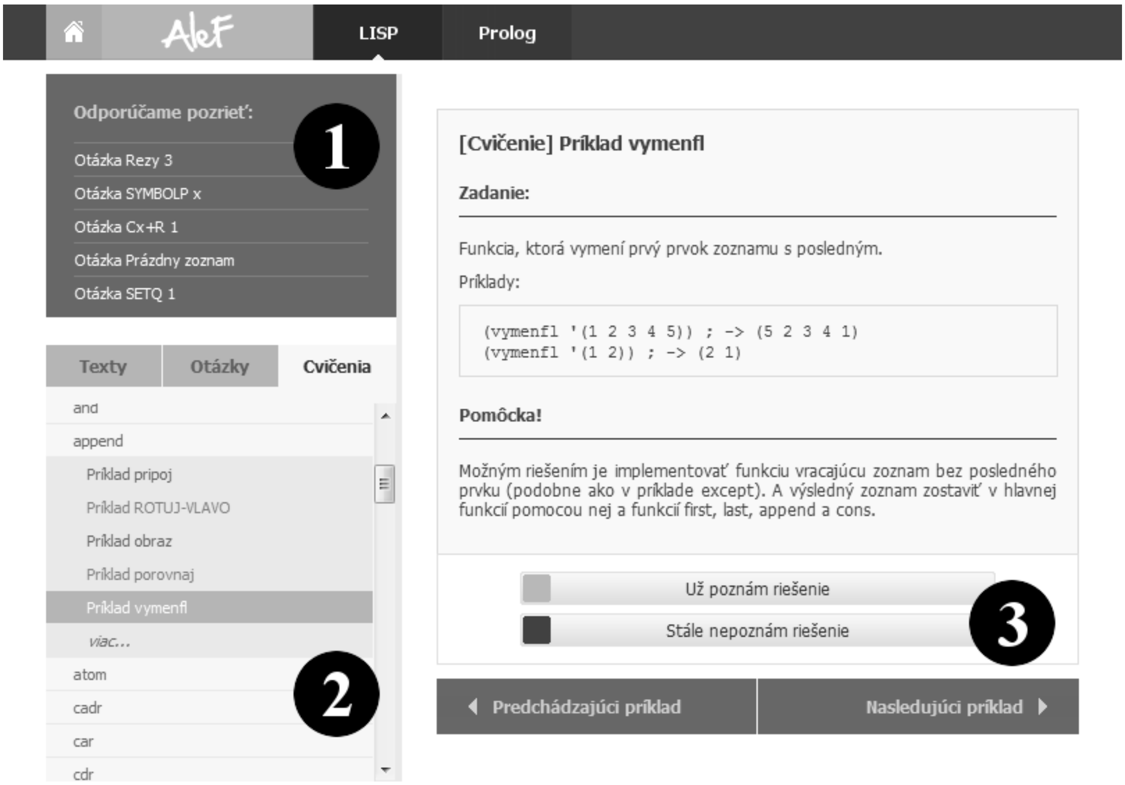

Fig. 2. Screenshot of ALEF user interface (in Slovak), (1) list of recommended exercises or questions, (2) standard non-adaptive menu, which serves as the main content of educational materials (texts) and lists all questions and exercises (in second and third tabs), (3) buttons for explicit user feedback: “I know a solution”, “I still do not know the solution”.

Figure 2 shows a screenshot from the ALEF with Lisp and Prolog programming exercises and questions. The dark box in the top left corner (1) lists the personalized recommendation of exercises and/or questions. Below is the standard non-adaptive menu (2) with exercises grouped by their related concepts. The main content at the right side is an exercise with its definition and hint displayed. Below the exercise are placed the buttons for explicit user feedback – an indication of understanding and knowing a solution (3).

Since a test question has set a set of related concepts and a difficulty value, and the feedback (correct or incorrect answer) can be processed the same way as feedback on an exercise, our method was used for recommending both exercises and test questions without any modifications.

In the first experiment the students took a pre-test, followed by a 60-minute learning session and a post-test (difficulty of both tests was equal). To verify the adaptive navigation impact on learning performance, the students were divided into 3 equivalent groups according to their previous results:

- *group A:* 17 students, provided with adaptive navigation based on a domain model with automatically generated concept layer [14],
- *group B:* 33 students provided with adaptive navigation based on a domain model with concept layer manually created by a domain expert,
- *group C:* 16 students, no adaptive features, only a standard menu with all exercises grouped by related concepts.

In both groups *A* and *B* the same recommending algorithm was used. The only difference was the concept layer creation method. Our hypothesis was: *Students who use adaptive navigation achieve better results in the test than students who have the access to the same learning resources, but navigate by themselves.*

Table 1 shows the test score comparison for the three student groups, including standard deviations. The greatest test score difference was achieved by the group *A*. The groups *B* and *C* achieved the same post-test results. However, students in the group *C* had higher entry knowledge. Neither of the groups *A* or *B* was outperformed by the group *C*, which confirms our hypothesis.

Table 1. Results of experiment 1

	pre-test (%)	*post-test (%)*	*difference (%)*
A: recommending *automatic model*	50,2 (±21,2)	70,5 (±15,2)	+20,2 (±15,2)
B: recommending *manual model*	42,4 (±21,0)	58,3 (±20,4)	+16,0 (±13,8)
C: no recommending	48,2 (±25,4)	59,2 (±17,6)	+11,0 (±17,6)

The second experiment consisted of a 50 minute learning session followed by a programming test. In this experiment, every student had a non-adaptive menu with all exercises as well as personalized recommendations of 5 best exercises for further learning. The students were divided into 3 groups, *A* and *B* being the same as above. Group *C* received recommendations of 5 randomly chosen exercises with suppressing recently viewed exercises as described in section 5.3. Our hypothesis was: *Students who use adaptive navigation based on a domain model achieve better results in the test than students who receive random recommendations.*

Table 2 shows the post-test score comparison. Students with adaptive navigation based on domain models achieved slightly better test results, but large standard deviations make these results unreliable. This experiment was conducted later in the course where some students already started preparing for the midterm and ALEF was not the only source of learning materials. This might have caused bigger differences within groups in achieved results.

Table 2. Results of experiment 2

	students	*post-test (%)*
A: recommending with automatic model	21	43,45 (±34,83)
B: recommending manual model	21	39,04 (±28,36)
C: random recommending	22	36,62 (±36,56)

In both experiments, all groups which used personalized recommending outperformed the control groups (no recommendation or random recommendation). However, the score difference was small. This could have been caused by extremely short learning time – no great difference in learning performance can be expected in a one-hour learning session. An experiment with more students is needed to get reliable results with statistically significant difference in group performance. Also, both experiments were conducted at the end of the day, which certainly affected the students' concentration.

7. Conclusions

We presented a utility-based method for recommending exercises to students, which helps the students to prepare for an exam. The main contribution of our proposal is considering limited time for learning and optimizing the recommendations to ensure covering all required concepts or topics at least to some extent. Our approach includes knowledge modeling based on the computer-adaptive testing method. We evaluated our method in a programming course and the results show some learning performance improvement in groups which used the personalized recommendation. Our recommending method also works well with automatically generated domain model.

In the next academic year, we plan to evaluate our recommendation solution as well as other components of ALEF (personalized annotator, navigation tracer, collaborative question creator [1]) in the Procedural programming course. Since the expected number of students in this course is about 300, much more accurate evaluation is possible.

In our future work we plan to design a user-friendly tool for easy creating of the learning targets set (setting the concept importance for the next milestone), since we cannot expect all this work to be done manually by the teacher for every test or milestone. Using such tool, the teacher could only select several key concepts and the tool would use the known relations between concepts to automatically find all closely related or required concepts and add them to the learning targets set. We also plan to experiment with using the information about students' learning progress for automatic acquiring of domain model attributes (e.g. exercise difficulty, concept difficulty or concept prerequisite weights), which currently cannot be obtained other way than setting them manually by a domain expert.

Acknowledgements

This work was supported by the Cultural and Educational Grant Agency of SR, grant No. 028-025STU-4/2010 and the Foundation of Tatrabanka, and it is a partial result of the Research & Development Operational Program for the project Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 25240120029, co-funded by ERDF.

References

1. M. Šimko, M. Barla, M. Bieliková, ALEF: A Framework for Adaptive Web-based Learning 2.0, In Proc. of Key Competencies in the Knowledge Society, KCKS 2010, IFIP AICT 324, 2010, pp. 367–378.
2. K. I. B. Ghauth, N. A. Abdullah, Building an E-Learning Recommender System using Vector Space Model and Good Learners Average Rating, In Proc. of 9th IEEE Int. Conf. on Advanced Learning Technologies, IEEE Computer Society, New York, 2009, pp. 194-196.
3. P. Brusilovsky, J. Ahn, T. Dumitriu, M. Yudelson, Adaptive Knowledge-Based Visualization for Accessing Educational Examples, In Proc. of 10th Int. Conf. on Information Visualization, IEEE Computer Society, New York, 2006, pp. 142-150.
4. M. Bieliková, An adaptive web-based system for learning programming. In Int. J. Continuing Engineering Education and Life-Long Learning, Inderscience, Vol. 16, Nos. 1/2 (2006) 122-136.
5. M. Barla, M. Bieliková, A. Bou Ezzeddinne, T. Kramár, M. Šimko, O. Vozár, On the Impact of Adaptive Test Question Selection for Learning Efficiency. Computers & Education. Vol. 55 (2010) 846-857.
6. J. M. Linacre, Computer-Adaptive Testing: A Methodology Whose Time Has Come, Development of Computerized Middle School Achievement Test (in Korean), Komesa Press, Seoul, 2000.
7. S. Sosnovsky, P. Brusilovsky, Layered Evaluation of Topic-Based Adaptation to Student Knowledge, In Proc. of 4th Workshop on the Evaluation of Adaptive Systems, Edinburgh, 2005, pp. 47-56.
8. Z. Yu, Y. Nakamura, D. Zhang, S. Kajita, K. Mase, Content Provisioning for Ubiquitous Learning, IEEE Pervasive Computing, Vol. 7, No. 4 (2008) 62-70.
9. N. Manouselis, C. Costopoulou, Analysis and Classification of Multi-Criteria Recommender Systems, World Wide Web, Vol. 10, No. 4, (2007) 415-441.
10. G. Adomavicius, A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Trans. on Knowledge and Data Engineering, Vol. 17 (2005) 734-749.
11. R. Burke, Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction, Vol. 12, No. 4 (2002) 331-370.
12. J. B. Schafer, J. A. Konstan, J. Riedl, E-Commerce Recommendation Applications, In Data Mining and Knowledge Discovery, Vol. 5, No. 1-2 (2001) 115-153.
13. G. Adomavicius, N. Manouselis, Y. Kwon, Multi-Criteria Recommender Systems, online [2010-08-06], http://ids.csom.umn.edu/faculty/gedas/NSFCareer/MCRS-chapter-2010.pdf
14. M. Šimko, M. Bieliková, Automatic Concept Relationships Discovery for an Adaptive E-course, In Proc. of Educational Data Mining 2009, 2nd Int. Conf. on Educational Data Mining, Cordoba, 2009, pp. 171-179.