# Guiding Students towards Frequent High-Utility Paths in an Ill-Defined Domain

Igor Jugo  Božidar Kovačić  Vanja Slavuj
Department of Informatics
University of Rijeka
Croatia
+385 51 584 711
ijugo@inf.uniri.hr  bkovacic@inf.uniri.hr  vslavuj@inf.uniri.hr

## ABSTRACT

This paper presents an exploratory data mining methodology for discovering frequent high-utility learning paths from a database of student interactions with an adaptable tutoring system. The discovered paths are used to present recommendations to students in order to make the learning process more efficient. The novelty of our approach is twofold: a) the process of data preparation, path evaluation and path discovery is completely autonomous; and b) the process is executed on a growing dataset of learning traces while the students are advancing through the knowledge domain. We present the system overview and the obtained results.

## Keywords

Sequential pattern mining, computer-based learning environment, high-utility patterns, recommendations.

## 1. INTRODUCTION

The objective of a tutoring system is to guide each student towards a predefined goal such as completing a lesson, task, or mastering a skill. Guiding students is more complex in ill-defined domains [4] where it is not possible to break down the learning units into single skill tasks, and the students have the freedom to choose/create their own path through the domain. One such web-based system has been developed at our institution to serve as an additional learning platform in a blended learning approach applied in a number of courses. The process presented in this paper is the third and final part (the first two being: 1) a communication layer that enables the system to communicate with DM tools, and 2) a clustering method [3] that discovers groups of students that use the system in a similar manner) of a new infrastructure developed with the goal of improving the adaptivity of our system [2].

While attempting to master the knowledge domain presented in our system, each student creates a large number of learning paths. Most of the students will need multiple interactions with a unit until it is mastered/completed, e.g., after a failed attempt they realize they need to learn some other (lower-level) units and then they come back to complete the first unit. The objective of the system is to offer recommendations to students about which unit to select (when the student is just starting a session or a new learning "run") or which unit to learn next (right after finishing learning a unit). For this, we need to discover productive frequent paths leading to, and following after, each unit. To discriminate between productive and unproductive frequent patterns we decided to construct a new dataset based on the database of learning paths and then feed that dataset into a high-utility sequential pattern mining algorithm USPAN [5] which requires a sequence database that contains both the unit IDs and their "profit" (in our case – the calculated efficiency of each path).

## 2. DISCOVERING FREQUENT HIGH-UTILITY PATHS

The system supports two types of learning activities:

a) LEARNING - presenting learning materials followed by a question about the unit, and initial questions about the connected underlying units (units below in the domain structure created by teacher). If the student answers all the questions correctly, the path is considered optimal and the change in the student's overlay model is calculated. If the student offers an incorrect answer to a question about a connected unit, the system will transfer the student to learn that unit, and the whole process is recursively repeated. Therefore, one learning "run" can consist of a number of learning units and a number of questions answered;

b) REPETITION - answering a series of questions about a unit without presenting learning materials. A visualization of four possible paths for a sample domain consisting of five units (A-E) is presented in Figure 1.

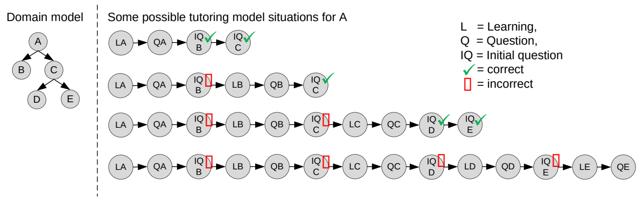

**Figure 1. Possible variations in learning path lengths**

The basic components for profit calculation, based on four paths presented in Figure 1, are presented in Table 2. Each unit has a set threshold value *t* that the student has to reach (by answering the questions). The current value of *t* for each unit in the domain model represents the student's model.

**Table 1. Path profit calculation**

| UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | ∑ | PL | Ls | Qs | IQs |
| *t*=10 | *t*=10 | *t*=6 | *t*=8 | *t*=10 | | | | | |
| +2 | +0.2 | +0.2 | | | 2.4 | 4 | 1 | 1 | 2 |
| +3 | -0.1 +2 | +0.2 | | | 5.1 | 6 | 2 | 2 | 2 |
| +2 | -0.1 +2 | -0.1 +2 | +0.2 | +0.2 | 6.2 | 10 | 3 | 3 | 4 |
| +1 | -0.1 +1 | -0.1 +1 | -0.1 +2 | -0.1 +2 | 6.6 | 14 | 5 | 5 | 4 |

The following expression is used to calculate "profit" *P* of path *x*:

$$P_x = \left(\frac{\sum_{i=1}^{Units} cq_i/t_i}{Q} + \frac{\sum_{i=1}^{Units} ciq_i/t_i}{IQ}\right) * \frac{PL_{min}}{PL}$$

We summarize the changes *c* that followed from answering a question about each of the units (Units) occurring in *x*, divided by the unit threshold value *t*. This accounts for the difficulty of the presented questions. The sum is then divided by the total number of questions answered (Q). The same is done for initial questions (IQ). Finally, the total change is multiplied by the difference between minimal and actual path length (PL). This penalizes longer paths as they are caused by incorrect answers to initial questions. Minimum path length ($PL_{min}$) is calculated based on the number of units added to the learning structure at the time the learning activity took place. The tutoring model determines the number of items in the learning structure based on the student's overlay model state, e.g., according to Figure 1, if the student starts the LEARNING activity with unit A, having previously completed units B and C, the tutoring module will not add any units to the learning structure (except for A, making $PL_{min} = 1$).

After the learning traces of all the students that are using the knowledge domain have been evaluated, they can be transformed into a sequence database for the USPAN algorithm. The transformation algorithm creates two databases for each unit in the domain – a set of paths consisting of units learned before the current unit ("prefix") and a set of paths consisting of units learned after ("suffix"). Each transformed sequence has a maximum length of 6 units. Both datasets are then converted to the correct format of the USPAN algorithm implementation in SPMF [1]. The system is now ready to discover high-utility frequent paths (HUFP). We run the algorithm on each dataset under the condition that a unit has been learned by at least five students, i.e., we must have a minimum of five paths in the dataset, although there can be much more if the students have been struggling with the unit. When the process is complete, all the discovered high-utility frequent paths are written to the database. Once the system has updated the HUF paths database the recommendation selection algorithm chooses the unit to be recommended at the beginning and the end of each learning activity. The algorithm considers: a) the student model; b) whether the unit has already been recommended and/or followed by this student; and c) which recommendation was most followed by other students.

## 3. RESULTS

We tested our system in two different knowledge domains, with 31 and 69 learning units, used by 30 and 20 students, respectively. The results are presented in Table 2. The "D.SET" column contains the number of learning traces in the system at the time the process was executed. The number of HUFPs discovered at each execution (divided by "prefix" and "suffix" paths) is presented in the next three columns. Column "UNITS" presents the number of units for which HUFPs were discovered. As expected, the unique number of units reached the total number of units in the domain in last two executions. The number of recommendations presented to students and the unique number of units for which the recommendations were presented are displayed in the next two columns.

**Table 2. Results for first domain**

| No | D.SET | HUFPs | PRE | SUF | UNITS | REC. | UN. REC | FOL. |
|---|---|---|---|---|---|---|---|---|
| 1 | 1206 | 21 | 5 | 16 | 7 | 12 | 5 | 5 |
| 2 | 1616 | 35 | 20 | 15 | 15 | 83 | 24 | 39 |
| 3 | 2068 | 115 | 65 | 50 | 22 | 204 | 15 | 36 |
| 4 | 2504 | 121 | 79 | 42 | 18 | 225 | 15 | 12 |
| 5 | 2912 | 89 | 52 | 37 | 21 | 47 | 12 | 13 |
| 6 | 3314 | 418 | 227 | 191 | 31 | 417 | 23 | 35 |
| 7 | 3604 | 538 | 289 | 249 | 31 | 332 | 24 | 22 |

Finally, the last column presents the number of recommendations followed (clicked) by students. The percentage of followed versus total number of recommendations varied from 5 to 47 percent. Further analysis will be performed to evaluate the overall impact of the recommendation mechanism on the learning process.

## 4. CONCLUSION

The presented methodology was implemented in a web-based ITS and tested on two different domains. We believe that the main improvements to the system can be made in: a) the interaction-to-path transformation algorithm, by implementing additional logic to recognize branch/level changes in the domain hierarchy which can reflect student's strategy, and b) the recommendation selection algorithm, by implementing additional logic to minimize repetition and optimize the selection process.

## 5. ACKNOWLEDGMENTS

This research is a part of the Project "Enhancing the efficiency of an e-learning system based on data mining", code: 13.13.1.2.02., funded by the University of Rijeka, Croatia.

## 6. REFERENCES

[1] Fournier-Viger, P. et al. 2014. SPMF: a Java Open-Source Pattern Mining Library. *Journal of Machine Learning Research*, 15, 3389-3393.

[2] Jugo, I., Kovačić B. and Slavuj, V. 2016. Increasing the Adaptivity of an Intelligent Tutoring System with Educational Data Mining: a System Overview, in *Int. Journal of Emerging Technologies in Learning*, 11, 3.

[3] Jugo, I., Kovačić, B. and Tijan, E.. 2015. Cluster analysis of student activity in a web-based intelligent tutoring system, in *Pomorstvo: journal of maritime studies*, 29, 80-88.

[4] Lynch, C. et al. 2006. Defining Ill-Defined Domains; A literature survey. *In ITS2006: Proc. of Intelligent Tutoring Systems Ill-Defined Domains Workshop*, Taiwan, 1-10.

[5] Yin, J., Zheng, Z., & Cao, L. 2012. Uspan: an efficient algorithm for mining high utility sequential patterns. In *18th ACM SIGKDD international conference on Knowledge discovery and data mining,* 660-668.