Mining and Visualizing Visited Trails in Web-Based Educational Systems

Cristóbal Romero [1], Sergio Gutiérrez[2], Manuel Freire[3] and Sebastián Ventura[1]
cromero@uco.es, sergut@lkl.ac.uk, manuel.freire@uam.es, sventura@uco.es
[1]Department of Computer Science, Córdoba University, Córdoba, Spain
[2] London Knowledge Lab - Birkbeck College, London, UK
[3] EPS-Universidad Autonoma de Madrid, Madrid ES-28049, Spain

Abstract. A data mining and visualization tool for the discovery of student trails in web-based educational systems is presented and described. The tool uses graphs to visualize results, allowing non-expert users, such as course instructors, to interpret its output. Several experiments have been conducted, using real data collected from a web-based intelligent tutoring system. The results of the data mining algorithm can be adjusted by tuning its parameters or filtering to concentrate on specific trails, or to focus only on the most significant paths.

1 Introduction

The concept of trails is closely related to the concept of curriculum. A curriculum is essentially a course or learning plan. A trail is the plan or route that a learner follows within a curriculum element, such as a subject, a competence or a learning goal [11]. Data mining techniques have been used to discover the patterns of sequences or visited trails of students from log files [10]. Server log files store a vast amount of information that contains the page requests of each individual user/student. This information can be seen as a per-user ordered set of web page requests from which it is possible to infer user navigation sessions. Lessons learnt from examining log files can then be fed back into the learning environment for optimization purposes. The extraction of sequential patterns or trails has been proven to be particularly useful, and has been applied to many different educational tasks. For example, trails have been used to evaluate a learner's progress, and to adapt and customize resource delivery [13]. Other uses include comparing trails to expected behavioral patterns (specified by the instructor, designer or educator) that describe an "ideal" learning path [8], providing indications on how to best to organize an educational web space, suggesting alternative paths to learners who share similar characteristics [4], the generation of personalized activities for different groups of learners [12], supporting the evaluation and validation of learning site designs [7], the identification of interaction sequences which suggest problems and/or identify patterns that are markers of success [5], supporting navigational learning, or testing the effectiveness of certain trails of learning objects [11].

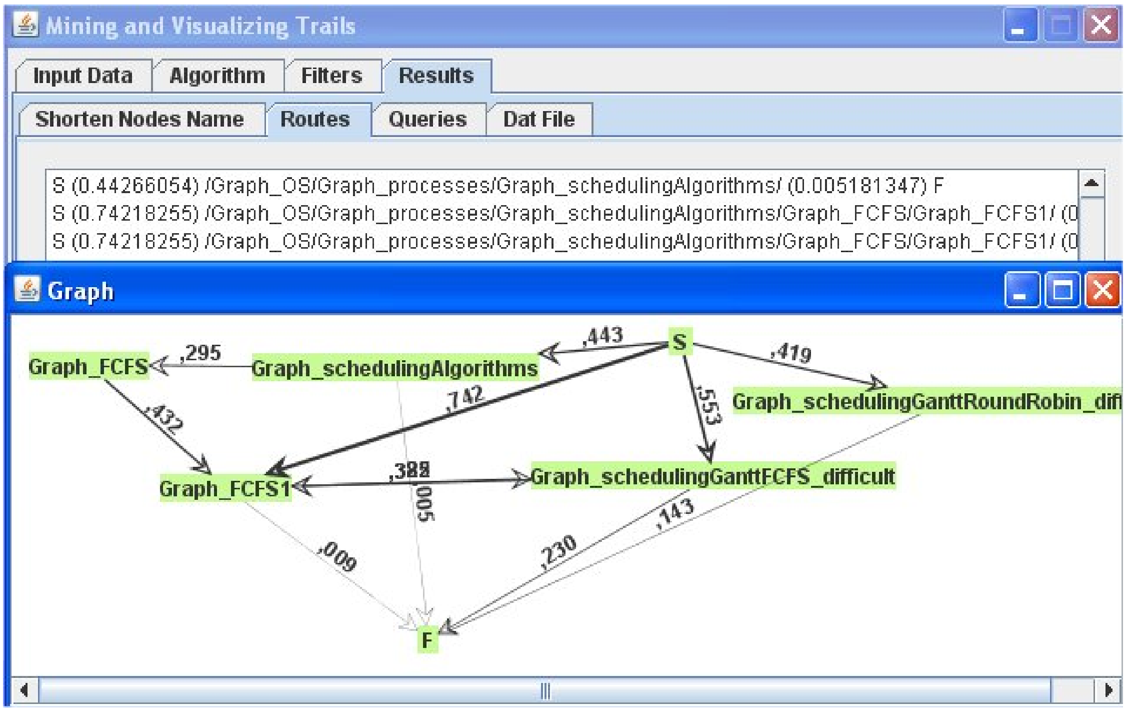
In this work, we describe a tool for mining and visualizing visit trails in web-based educational systems. The next section introduces the functionality of the tool to mine and visualize trails from user logs. Experimental results achieved with real data obtained from a web-based intelligent tutoring system are then discussed. Finally, conclusions and future work are presented.

2 A tool for Mining and Visualizing Educational Trails

We have developed a mining and visualizing tool in order to help instructors to discover the most visited trails. Course authors or instructors can execute the tool whenever enough usage information from students has been collected. First, the user has to create a data file from the web-based educational system's log files. The tool's user has to indicate the name, type of database, user, password, server and port. Optional restrictions such as a time period (a start date and an end date, empty by default) and session time in minutes (25 minutes by default) can also be indicated. Then, the system creates a .dat file using the Weka format. This file contains all the visited links and the total number of times that students use each of them.

Next, the user can apply a web data mining algorithm to locate navigation sessions and trails. Currently, we have implemented the HPG (Hypertext Probabilistic Grammar) model to efficiently mine the trails [1]. HPG uses a one-to-one mapping between the sets of non-terminal and terminal symbols. Each non-terminal symbol corresponds to a link between web pages. Moreover, there are two additional artificial states, S and F, which represent the start and finish states of the navigation sessions. The number of times a page was requested, and the number of times it was the first and the last page (state) in a session, can easily be obtained from the collection of student navigation sessions. The number of times a sequence of two pages appears in the sessions gives the number of times the corresponding link was traversed. The aim is to identify the subset of these trails that correspond to the rules that best characterize a student's behavior when visiting the web-based course. A trail is included only if its derivation probability is above the parameter λ (cut-point). The cut-point is composed of two distinct thresholds (support and confidentiality). The support value is for pruning out the strings whose first derivation step has low probability, corresponding to a subset of the hypertext system rarely visited. The confidence value is used to prune out strings whose derivation contains transitive productions with small probabilities. Therefore, in order to use the HPG algorithm, the user has to select a *.dat* file, set three parameters (α, support and confidence) and the desired name of the routes file that will be created. This output text file will contain all the routes and the probabilities of the sequences to go from one node to another node. The user can also reduce the size of the generated route files by using three types of filters. The *accumulated transition probability* filter eliminates sequences that have a value lower than the user-specified value (between 0 and 100%). The *non-accumulated transition probability* filter is similar to the previous one, but does not accumulate the probability values. The *route length filter* limits the length of the routes to a maximum value, specified by the user (integer greater than 1).

Then, we visualize the resulting trails as a graph (considering each hypertext page as a node and each hyperlink as an edge) in order to display the amount of traffic for each route to instructors (see Figure 1). This graph is zoomable, and the nodes can be manually displaced, allowing users to modify the automatically generated layout or examine the graph at varying degrees of detail. The application also shows, in different panes, the name of the nodes (reduced and full name of the nodes), the routes in text mode, and the used data file.



```
S (0.44266054) /Graph_OS/Graph_processes/Graph_schedulingAlgorithms/ (0.005181347) F
S (0.74218255) /Graph_OS/Graph_processes/Graph_schedulingAlgorithms/Graph_FCFS/Graph_FCFS1/ (0
S (0.74218255) /Graph_OS/Graph_processes/Graph_schedulingAlgorithms/Graph_FCFS/Graph_FCFS1/ (0
```

Graph
Graph_FCFS
,295
Graph_schedulingAlgorithms
,443
S
,419
Graph_schedulingGanttRoundRobin_dif
,742
,553
,432
Graph_FCFS1
,388
,005
Graph_schedulingGanttFCFS_difficult
,009
,143
,230
F

Figure 1. Results showing routes in text mode and in the graph visualization window.

In Figure 1, each node represents a web page, and the directed edges (arrows) indicate how the students have moved between them. The tool's graph visualizations are generated using the CLOVER framework, described in greater detail in [3]. Edge thickness varies according to edge weight; this allows users to quickly focus on the most important edges, ignoring those that have very low weights. In addition to line widths, numerical weights are also available. This information can be useful to a learning designer in different ways. First, it can be used as a limited *auditing tool*, providing a deeper understanding of the learning paths effectively followed by the students. Additionally, comparing this information with expected *a priori* paths allows the designer to refine the sequencing strategy. The results in the graph can show information that was not know in the first place, e.g. which activities are the most difficult, which are easier than expected (shown as more common transitions), etc.

3 Experimental Results

This study uses real data collected from a web-based intelligent tutoring system [9] for the domain of Operating Systems. Although the original log file contains sessions from 88 students that used the system in 2006, the study covers only the subset of "good users" (those with more than two sessions). The study covers data from 67 students, with 754 sessions (using 25 minute timeouts) and 1121 records in total. We have carried out several experiments focused on HPG's sensibility to its parameter values in order to obtain different configurations of the graph (number of nodes, links, routes, and average route length). Results with varying parameters are displayed in Table 1.

Table 1. HPG: Comparison varying alpha and cut-point parameters.

Alpha	Support	Confidence	Nodes	Links	Routes	Avg. Route Length
0	**0.08**	**0.3**	6	8	4	4.5
0	**0.08**	**0.2**	48	123	97	16.26
0	**0.08**	**0.18**	117	340	76443	32.19
0.5	**0.5**	**0.7**	4	4	2	3.0
0.5	**0.3**	**0.7**	19	32	17	2.88
0.5	**0.1**	**0.7**	136	244	144	2.83
0.5	**0.5**	**0.2**	4	6	6	3.66
0.5	**0.3**	**0.2**	62	177	2190	14.45
0.5	**0.1**	**0.2**	162	551	46913	18.62
1	**0.7**	**0.7**	11	17	9	2.88
1	**0.5**	**0.7**	30	54	28	2.92
1	**0.3**	**0.7**	89	167	94	2.92

Support and confidence thresholds give the user control over the quantity and quality of the obtained trails, while α modifies the weight of the first node in a user navigation session. In Table 1, the support must be set very low in order to obtain routes with $\alpha = 0$. This is due to the fact that there are few start nodes. It shows that students have started their sessions in different nodes, and none of these have a significantly higher probability. This changes as α increases, since there will progressively be more visited nodes. The number of routes, nodes and links is increased as time support is decreased. On the other hand, the number of resulting nodes, links, routes and average route lengths is greatly increased when the confidence value is decreased. This effect is more evident on links and routes. This can be traced to the fact that the confidence threshold prunes the intermediate transactions that do not have a derivation probability above the cut-point. It must be noted that the user of the HPG algorithm can use both the alpha, support and confidence thresholds, and the three available filters, in order to obtain a suitable number of trails. The learning designer must work with the course lecturer in order to tune these parameters to a particular community of learners. Then, combining the information on the table with that displayed in the graph, the instructor can focus on the most visited routes in order to make decisions on the organization of the educational web space, or recommend paths and shortcuts to learners.

4 Conclusions

This paper has described a data mining and information visualization tool that aids authors and instructors to discover the trails followed by students within web-based educational systems. The resulting networks are then visualized using a graph representation with edges of varying thickness, which is more compelling to non-specialized users than textual output. Future plans include the addition of other sequential pattern mining algorithms such as AprioriAll and PrefixSpan. Our goal is to use the tool to provide personalized trails to students, delivering on the promise of personalized learning within adaptive e-learning systems.

Acknowledgments. The authors gratefully acknowledge the financial subsidy provided by the Spanish Department of Research under TIN2005-08386-C05-02 and TIN2007-64718, and the British Teaching and Learning Research under grant RES-139-25-0381.

References

[1] Borges, J., Levene, M. Data Mining of user navigation patterns. Proc. of Workshop Web Usage Analysis and User Profiling. San Diego, 2000. pp. 31-36.

[2] Dichev, C., Dicheva, D., Aroyo, L. Using Topic Maps for Web-based Education *Advanced Technology for Learning*, 2004, 1, pp. 1-7.

[3] Freire, M. An Approach to the Visualization of Adaptive Hypermedia Structures and other Small-World Networks based on Hierarchically Clustered Graphs. PhD Thesis presented at Universidad Autónoma de Madrid, 2007.

[4] Ha, S., Bae, S., Park, S. Web Mining for Distance Education. IEEE Conf. on Management of Innovation and Technology, Singapore, 2000. pp. 715–719.

[5] Kay, J., Maisonneuve, N., Yacef, K., Zaiane, O.R. Mining Patterns of Events in Students' Teamwork Data. Educational Data Mining Workshop, Taiwan, 2006. pp. 1-8.

[6] Keenoy, K., Levene, M. A Taxonomy of trails of learning objects. Trails of Digital and non-digital Los. EU Sixth Framework programme priority 2. 2004. pp. 97-106.

[7] Machado, L., Becker, K. Distance Education: A Web Usage Mining Case Study for the Evaluation of Learning Sites. In Proc. Int. Conf. on Advanced Learning Technologies, Athens, Greece, 2003. pp. 360-361.

[8] Pahl, C., Donnellan, C. Data mining technology for the evaluation of web-based teaching and learning systems. In Proc. E-learning. Montreal, Canada, 2003. pp. 1-7.

[9] Prieto Linillos, P., Gutiérrez, S., Pardo, A., Delgado Kloos, C. Sequencing Parametric Exercises for an Operating System Course. Proc. of AIAI 2006, pp. 450-458.

[10] Romero, C., Ventura, S. Educational Data Mining: a Survey from 1995 to 2005. Expert Systems with Applications, 2007, 33(1), pp. 135-146.

[11] Schoonenboom, J., Levene, M., Heller, J., Keenoy, K, Turcsanyi-Szabo, M. Trails in Education. Technologies that Support Navigational Learning. Sense Publishers. 2007.

[12] Wang, W., Weng, J., Su, J., Tseng, S. Learning portfolio analysis and mining in SCORM compliant environment. Proc. ASEE/IEEE Frontiers in Education Conference, Savannah, Georgia, 2004. pp. 17–24.

[13] Zaïane, O., Luo, J. Web usage mining for a better web-based learning environment. In Proc. Conf. Advanced Technology For Education, Banff, Alberta, 2001. pp. 60–64.