*J. Basic. Appl. Sci. Res.*, 2(4)4333-4337, 2012
© 2012, TextRoad Publication

ISSN 2090-4304
**Journal of Basic and Applied Scientific Research**
*www.textroad.com*

# Path Recommendation Using Sequential Pattern Mining in Intelligent Tutoring System

**Yousef Moradi**

Computer Software Engineering,Islamic Azad University of Najafabad,Najafabad,Isfahan,Iran

**ABSTRACT**

Path recommendation plays an important role for learners to obtaining good action in e-learning environment. Without appropriate guiding service, learners might miss some resource and waste time. Therefore, how to provide visitors customized path becomes an important task for learners. To bridge the gap, this research uses sequential pattern mining for intelligent touring system to generate personalized path for learners. Through this paper, we will focus on guidance learners to appropriate path. This paper suggests the use of web mining techniques to build an agent that could recommend on-line learning activities or shortcuts in a e-learning environment based on learners' access history to improve course material navigation.
**KEYWORDS:** Recommender systems; Adaptive web based systems; Interactive web based systems; User profiles; User models.

## 1. INTRODUCTION

The continued growth and increasing complexity of Web-based applications, from traditional e-commerce, to Web services, to all kinds of dynamic content providers, has led to a proliferation of search tools. Personalized services, such as recommender systems, help engage visitors, turn casual browsers into customers, or help visitors to more effectively locate pertinent information. The goal of any recommendation in any area is to make a selection from among all the possible items by using certain attributes predefined by the context.

Recommender systems have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options (Burke, 2002). The last implies a clear main objective: to guide the user to useful/interesting objects. One of the applications of recommender systems is intelligent tutoring system. An intelligent tutoring system (ITS) is any computer programs thatincorporate techniques from the AI communityto provide direct customized instruction to students [1].

The term educational technology is often associated with, and encompasses, instructional theory and learning theory. While instructional is "the theory and practice of design, development, utilization, management, and evaluation of processes and resources for learning," according to the Association for Educational Communications and Technology (AECT) Definitions and Terminology Committee [2].

Educational Data Mining (called EDM) is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in.[1] A key area of EDM is mining computer logs of student performance.[2] Another key area is mining enrollment data.[3] Key uses of EDM includepredicting student performance, and studying learning in order to recommend improvements to current educational practice. EDM can be considered one of the learning sciences, as well as an area of data mining. A related field is learning analytics.

To satisfy learner's requirements, e-learning are now offering variant services to assist learners to learn. One of the simplest but conventional services is the recommendation path for learners that are personalized according to their requirement. Up to the very recent years, most e-learning systems have not been personalized. Several works have addressed the need for personalization in the e-learning domain. However, even today, personalization systems are still mostly confined to research labs, and most of the current e-learning platforms are still delivering the same educational resources in the same way to learners with different profiles. Recently, due to the impulsion of information technology, advanced personal digital assistant (PDA) devices are utilized to provide learners interactive experiences through multimedia and Internet functions.

In fact, one of the new forms of personalization in e-learning environment is to give recommendations to learners in order to support and help them through the e-learning process [4]. The task of delivering personalized e-learning material is often framed in terms of a recommendation task in which a system recommends items to an active user [5]. Several educational recommender systems have been proposed in the literature that the most of them

***Corresponding Author:** Yousef Moradi, Computer Software Engineering, Islamic Azad University of Najafabad, Najafabad, Isfahan, Iran. Email: su_moradi@yahoo.com

focus on recommending suitable materials or learning activities [6]. On the other hand, the popularity of e-learning has created huge amounts of educational materials. Hence, locating the suitable learning materials has become a big challenge. One way to address this challenge is the use of recommender systems [7].To solve the above difficulties, this research developed a path suggestion system that provides personalized learning path for learners when considering a set of learning materials.

The sequential pattern mining is a data mining technique focusing on discovering frequent sequences according to transaction time or order of events. Sequential pattern mining, first proposed by [8], is to find temporal, ordinal and frequent sequences. In their research, a large database consisted of a list of transactions, ordered by increasing transaction-time. The aim of the research was to find the maximal sequences among all sequences that had a certain user-specified minimum support. Each such maximal sequence was represented as a sequential pattern. The authors used Apriori property in association rule mining to solve this problem and presented AprioriAll, AprioriSome and DynamicSome algorithms. Some researchers proposed other Apriori-based algorithms such as GSP and SPADE. Subsequently, other researchers proposed projection-based algorithms of FreeSpan [4]. The difference between Apriori and projection-based algorithms is that the projection-based algorithm does not need candidate generation. The time-interval sequential pattern provides more valuable and meaningful information about time. Srikant and Agrawal (1996) provided definitions for the maximum interval (max-interval), the minimum interval (min-interval) and the sliding time window size (window-size). Wu, Peng, and Chen (2001) defined the interval between adjacent events in a sequential pattern is within the time range specified by users. Although this research sets a time-interval between sequential patterns, only a fixed time-interval between events is considered. Chen and Huang (2005) presented FTI-Apriori algorithm and FTI-Prefix Span algorithm to discover fuzzy time-interval sequential patterns. In their simulation results, the second algorithm outperforms the first one, not only in computing time but also in scalability with respect to various parameters. In this paper, we suggest the construction of such automatic recommendation system for Web-based learning environments that takes into account profiles of on-line learners, their access history and the collective navigation patterns, and uses simple data mining techniques, namely association rule mining.

## 2. METHODOLOGY

Whenever a recommender system displays a new recommendation, it must face two problems. On the first hand, it must choose a *useful* item to be recommended. The difficulty of this problem may be noticed by the numerous literature about it (for a survey, see Adomavicius&Tuzhilin, 2005). We will refer to this labor as *the task of filtering* useful items.

On the other hand, the recommender system must display recommendations to be *followed*. It must be noticed that the *usefulness* of a recommended item does not imply its *follow-ability*. For instance, a recommendation of a very *useful* book as “Romeo and Juliet” (Shakespeare) could be less *follow-able* than a recommendation which suggests “The Da Vinci Code” (Dan Brown). However, it must be highlighted that a *not followed* suggestion is, at least, as useless as a recommendation which leads to a *useless* item. Thus, another task must be considered.

We define *the task of guiding* the user the job of displaying *follow-able* recommendations to the user. Bearing this task in mind, the recommender systems must adapt the *display* of each recommendation regarded to its user model or user profile. Traditionally, machine learning is a common technique for user modeling [7]. In particular, for the field of recommender systems, bayesian classifiers, decision trees or instance based classifiers have been commonly employed for improving user models in the task of filtering useful items. For instance, bayesian classifiers and decision trees can be found in significative recommender systems. Instance based classifiers (i.e. nearest neighbor or neighborhood methods) are common in collaborative filtering approaches [1,7,8]).

However, the suitability of these algorithms for the task of guiding users remains to be studied. To this end, we have developed a recommender system centered on the task of guiding, which is to be described in the next section.

In order to implement and evaluate our approach, we have chosen a web-based learning environment. As usual, it is based on a web server, a set of scripts and a backed-database. Both the contents to be presented and the (personal and interaction) users’ data are stored in the database, so the state of the whole application is stored in the database. The scripts are the way of *showing* and *changing* the state of the whole application.

For the purposes of the experiment, we left students run free into a course (about data mining) built in this environment. To this end, users (students) had to fill a form previous to be registered into the course. As a result, each user who navigates through the course always holds an (static) user model filled by herself. Notice the problems related to prompting or asking for these data (obtrusive way of getting labels, e.g. see [8,9]). However, they are smoothed by the fact that they get back a course for free. Moreover, we can trust in what users filled because of two reasons: (1) there were only a few questions very easy to fill (just one selection between a few options), (2) the users were aware of the fact that the personalization of the course depended strongly on the way they filled the forms.

The interface of the whole course is built by means of four frames: header, index, content, and recommendation (recommender's interface). By interacting with the header frame, users can rate1 the current content or leave the course. Index frame is to be used for jumping to each piece of content, also called *content item*. Into the index frame, themes (see "Introduccion", "EvolucionHistorica", etc.) are used to organize semantically all single content items. Nonetheless, we did not number nor ordered them to leave as much freedom as possible in their access. Finally, the content frame renders the current content item into its limits.

**4. Building recommender system**

Web usage mining performs mining on web data, particularly data stored in logs managed by the web servers. The web log provides a raw trace of the learners' navigation and activities on the site. In order to process these log entries and extract valuable patterns that could be used to enhance the learning system or help in the learning evaluation, a significant cleaning and transformation phase needs to take place so as to prepare the information for data mining algorithms [10]. Web server log files of current common web servers contain insufficient data upon which tobase thorough analysis. However, they contain useful data from which a well-designed data mining system can discover beneficial information and which can provide a basis for model building. The model we use to construct our recommender system is based on association rules.

Association rules are one of the typical rule patterns thatdata mining tools aim at discovering. They are very usefulin many application domains, but are mainly appliedin the business world as in market-basket analysis. In a transactional database where each transaction is a set ofitems bought together, association rules are rules associatingitems that are frequently bought together. A rule consistsof an antecedent (left-hand side) and a consequent (right-hand side).

The intersection between the antecedent and the consequent is empty. If items in the antecedent are bought then there is a probability that the items in the consequent would be bought as well at the same time. An efficient algorithm to discover these association rules was first introduced in [11,12]. The algorithm constructs a candidate set of frequent itemsets of length k, counts the number of occurrences, keeps only the frequent ones, then constructs a candidate set of itemsets of length k+1 from the frequent itemsets of smaller length. It continues iteratively until no candidate itemset can be constructed. In other words, every subset of a frequent itemset must also be frequent. The rules are then generated from the frequent itemsets with probabilities attached to them indicating the likelyhood (called support) that the association occurs. We use this idea of association rules to train our recommender agent to build a model representing the web page access behaviour or associations between on-line learning activities.

A recommender system suggests possible actions or web resources based on its understanding of the user's access. To do so we have to translate the entries in the web log into either known actions (i.e. learning activities such as accessing a course notes module, posting a message on the forum, doing a test, trying a simulation, etc.) or URLs of a web resource. This mapping is a significant processing phase that in itself presents a considerable challenge [13-15]. Moreover, these identified actions and URLs are grouped into a session which is yet another difficult and delicate task [16]. These sessions are then modeled into transactions as sets of actions and URLs. The association rule mining technique is applied on such transactions to discover associations between actions, associations between URLs and associations between actions and URLs, as well as associations between sequences of actions and/or URLs. This process usually leads to a very large number of association rules even after filtering out those that do not satisfy the requirement of minimum support [17-19]. We use other specific filtering approaches to eliminate such discovered rules that associate two URLs that are directly linked from each other. Indeed, it is useless to recommend a page that is directly linked from the current page as a shortcut. Moreover, we give higher weights to rules that have as a consequent a URL or a set of URLs that are frequently towards the end of a session. For the rules that associate actions, we keep only rules that have as a consequent an action that terminated successfully. For instance, if the action is taking an on-line test, it is only useful to recommend that action after a sequence of actions if that test was successful. In other words, actions are labeled whenever possible with "successful" or "unsuccessful" using the users' profiles [20].

When the recommender agent is activated by a triggering event, the association rules are consulted to check for matches between the triggering event, or sequence of events, with the rule antecedents. When a match is found, the consequent of the rule is suggested. If more matches are found, the suggestions are ranked and only a small set (highest ranked) is displayed.

## 5. Conclusions

In this paper, we have considered the recommendation problem as formed by two tasks. Also, we have focused on the one that, in our opinion, needs a more closer attention in the field: the task of guiding.Also, notice the general terms in which the experiment has been developed (as a summary, see Table 2). They allow us to generalize the results and think just in terms of the task (of guiding) involved. In addition, in an experiment centered on this task, we have observed a really bad behavior of the machine learning techniques commonly employed in this area.As a final conclusion, we claim that this behavior has to do with the following distinguishing characteristics of the task of guiding:The task of guiding pursues two objectives at the same time: (i) it tries to recommend as much as possible, but (ii) it tries to recommend (only) follow-able recommendations. Opposite to this, the task of filtering has commonly pursued *accurate* recommendations in terms of usefulness/interest. In other words, the task of guiding pursues balancing the *cost* of displaying a recommendation against the possibility of annoying the user with it. Therefore, the task of guiding is a cost-oriented duty. Although, the task of filtering fits better in an accuracy-maximization duty.The latter is one of the keys for explaining why the behavior of conventional algorithms (for the task of filtering) are so degraded when applied to the task of guiding. In fact, the task of filtering employs classifiers that maximize their accuracy, though the task of guiding needs classifiers that maximize the benefits (minimize costs). This partially explains the shy behavior of adaptive recommenders in the experiment: once they are left alone (after the first 50 visits), they tend to prefer the *accurate* model "never recommend"2 over a more profitable one.Evidently, there is a necessity of new training cases to improve the recommendation task as the work moves along. In the task of filtering, the training cases come from the users' activity on items (by rating on them). Thus, it is expected an increase of cases as the time goes along. However, in the task of guiding, a new training case comes *only* when there is an act of recommendation. In fact, the number of training cases totally depends on the recommender's activity. For instance, an extreme adaptive recommender whose primer behavior was similar to "never recommend" would never improve its conduct because the recommender would never get a single training case.Again, the latter partially explains the experiment. As the recommenders tend to recommend less frequently (tend to "never recommend"), they get less number of training cases. Therefore, their models remain bad and constant over the experiment. However, as the time runs, a constant bad behavior is worse considered. In the next section, we introduce some ideas which look promising to face the above problems.

## REFERENCES


[1] Lumer, E. and B. Faieta. Lumer E.D. and Faieta B. Diversity andAdaptation in Populations of Clustering Ants.Procs.of SAB´94 – 3rd Conf. on Simulation ofAdaptive Behavior: From Animal to Animats, The MIT Press/Bradford Books, Cambridge,MA. 1994

[2] Lumer, E. and B. Faieta. Exploratory Database Analysis via Self-Organization.Unpublished manuscript, 1995. Referred to in [Bonabeau et al, 1999]

[3] Montaner, M., López, B., de la Rosa, J. Ll., "A Taxonomy ofRecommender Agents on the Internet", Artificial Intelligence Review, Kluwer AcademicPublishers. Volume 19, Issue 4, pp. 285-330. June, 2003

[Mooney and Roy, 2000] Mooney, R. J.; Roy, L., Content-Based Book Recommending UsingLearning for Text Categorization, Proceedings of the V ACM Conference on Digital Libraries,San Antonio, USA, pp.195-204, 2000

[4] I. Cadez, D. Heckerman, and C. Meek. Visualization of navigationpatterns on web site using model based clustering. In*ACM Int. Conf. on Knowledge Discovery and Data Mining(SIGKDD'00)*, pages 280–284, Boston, USA, August 2000.

[5] S. Chee, J. Han, and K. Wang.Rectree: An efficient collaborativefiltering method. In *3rd Int. Conf. On Data Warehousingand Knowledge Discovery (DAWAK 2001)*, LNCS2114, pages 141–151, Munich, Germany, September 2001.Springer Verlag.

[6] E. Chi, J. Pitkow, J. Mackinlay, P. Pirolli, and J. Konstan.Visualizing the evolution of web ecology.In*ACM CHI'98Conference on Human Factors in Computing Systems*, page644.645, Los Angeles, USA, 1998.

[7] R. Cooley, B. Mobasher, and J. Srivastava. Web mining:information and pattern discovery on the world wide web. In*9th IEEE International Conference on Tools with ArtificialIntelligence*, pages 558–567, 1997.

[8] R. Cooley, B. Mobasher, and J. Srivastava. Data preparationfor mining world wide web browsing patterns. *Knowledgeand Information Systems*, 1(1):5–32, 1999.

[9] H. A. Edelstein. Pan for gold in the clickstream, March2001. http://www.informationweek.com/828/mining.htm.

[10] U. Fayyad, G. Grinstein, and A.Wierse. *Information Visualizationin Data Mining and Knowledge Discovery*.MorganKaufmann Publisher, 2001.

[11] A. Foss, W. Wang, and O. R. Za¨ıane. A non-parametricapproach to web log analysis. In *Web Mining Workshopin conjunction with the SIAM International Conference onDataMining*, pages 41–50, Chicago, IL, USA, April 2001.

[12] J. Han and M. Kamber.*Data Mining, Concepts and Techniques*. Morgan Kaufmann, 2001.

[13] T. Munzner and P. Burchard.Visualizing the structure of theworld-wide web in 3d hyperbolic space.In*Proceedings ofVRML'95*, 1995.

[14] M. Spiliopoulou, L. C. Faulstich, and K. Winkler.Adataminer analyzing the navigational behaviour of web users.In*workshop on Machine Learning in User Modeling of theACAI'99*, Creta, Greece, July 1999.

[15] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan.Webusage mining: Discovery and applications of usage patternsform web data. *SIGKDD Explorations*, 1(2), January 2000.

[16] R. G. T. Zheng, Y. Niu. Webframe: in pursuit of computationallyandcongnitively efficient web mining. In *6$^{th}$Pacific-Asia Conference on Knowledge Discovery and DataMining*, pages 264–275, Taipei, Taiwan, May 2002.

[17] Webct, available on August 2002. http://www.webct.com/.

[18] O. R. Za¨ıane. Web usage mining for a better web-basedlearning environment. In *Proc. of Conference on AdvancedTechnology for Education*, pages 60–64, Banff, AB, June2001.

[19] O. R. Za¨ıane and J. Luo.Towards evaluating learners' behavior in a web-based distance learning environment. In*Proc.of IEEE International Conference on Advanced LearningTechnologies (ICALT01)*, pages 357–360, Madison, WI,August 2001.

[20] O. R. Za¨ıane, M. Xin, and J. Han.Discovering web accesspatterns and trends by applying OLAP and data mining technologyon web logs. In *Proc. Advances in Digital LibrariesADL'98*, pages 19–29, Santa Barbara, CA, USA, April 1998.