

A Comparative study on Intelligent Tutoring Systems fostering Self Regulated Learning.

Lekshmi A ¹, Prathibha.S Nair ²

¹(P G scholar, Department of Computer Science, Mohandas collg of Engineering and Technology, Kerala, India,)

²(Asst Professor, Department of Computer Science, Mohandas collg of Engineering and Tech, Kerala, India,)

Abstract: Educational data mining is an emerging research area where the traditional data mining techniques are applied on Intelligent Tutoring System (ITS) to study the learning pattern of students. This paper conducts a survey on various ITS that are used currently. An agent-based ITS that fosters the use of Self Regulated Learning (SRL) processes mainly concentrates on two questions: (1) How can students be grouped according to their performance and their type of interaction with the system? (2) How do specific learning behaviors of high and low performing students differ, in particular regarding their use of SRL processes?. The study helps in understanding how data can be mined and how the learners can be grouped into different clusters based on their learning activities.

Index Terms: Educational Data Mining, ITS, SRL, Clustering, Pattern Mining.

I. INTRODUCTION

Traditionally, most research on computer based (machine) learning has dealt with the development of techniques for solving engineering problems and many of the systems developed have been tested on simplified artificial problems (Reich, 1994)[1]. Consequently, the machine learning research field has historically been very rich in terms of theoretical developments but lacks practical applications with direct links between theory and practice. This is particularly noticeable in the field of adaptive educational hypermedia, where many developments are restricted to small, hand-crafted systems and omit thorough evaluations of utility.

The survey presented here is an attempt to bridge the gap between theory and practice in the domain of pedagogical tutoring systems. A number of these systems employ machine learning techniques and they are referred to as ITS(Intelligent Tutoring Systems).However, the key challenge in such systems is to be able to capture an individual user's preferences and specific information needs and utilize this information to adapt the environment to the user. These systems do not meet the criterion for the ideal adaptive system as described by Brusilovsky (1996):

“ while the user is simply working in an application system, the adaptation component watches what the user is doing, collects the data describing user's activity, processes these data to build the user model, then provides an adaptation. Unfortunately, such an ideal situation is very rarely met in adaptive hypermedia systems...”.

Intelligent Tutoring Systems (ITS) are instructional systems that use Artificial Intelligence (AI) techniques in computer programs to facilitate learning. These systems are based on cognitive psychology as an underlying theory of learning, which deals mainly with issues such as knowledge representation and organization within the human memory as well as the nature of human errors. The intelligent tutoring systems adopt a mixed-initiative teaching dialogue, which allows the system to initiate interactions with the learner, as well as interpret and respond meaningfully to learner-initiated interactions. A major challenge is to make the ITS more adaptive to individual learning characteristics, such as browsing behavior and initiative in performing appropriate SRL (Self Regulated Learning) processes.

The survey is situated at the intersection of three areas; machine learning, computer supported learning and Educational Data Mining (EDM) [2]. The emerging research community of Educational Data Mining exploits data from learners' interaction with learning tools. This new research area deals with issues that differ from the traditional data mining and machine learning environment. The educational data presents several difficulties for the data mining algorithms as it is temporal, noisy, and incomplete and may lack enough samples for mining. This area is establishing the new requirements for effective mining from scarce data and analysis of these learning data. Using this, different patterns of student learning can be recognized so that the categorization of learners is possible. The survey continues this exploration of foundations for this area, taking account of the particular demands of educational context.

II. CONTRIBUTED WORKS.

Numerous learning environments include meta cognitive support in order to improve domain-level learning. These general learning skills are often referred to as meta cognition, or self-regulated learning [3] suggest focusing on metacognition as one of the main principles that should be applied to educational research and design. Metacognition means the students need to be able to manage their learning, for example, by setting goals, planning their learning, monitoring their progress, and responding appropriately to difficulties and errors.

An agent-based ITS that fosters the use of SRL processes mainly concentrates on two questions: (1) How can students be grouped according to their performance and their type of interaction with the system? (2) How do specific learning behaviors of high and low performing students differ, in particular regarding their use of SRL processes? The answer to the first question is using a clustering approach that groups students with similar performance and scores on other system interaction metrics. The second answer is a pattern mining method which identifies statistically significant differences in frequent behaviors between clusters.

Most of the systems that supports self regulation, attempts to help students to acquire or improve the metacognitive skills themselves. Few examples suggest that improving metacognitive and SRL skills can be done using educational technologies. Examples include the AutoTutor, ML Tutor, TRAC, Betty's Brain and Metatutor. This survey reveals the various data mining techniques used by these Tutors.

A. AutoTutor

AutoTutor [4] is a Natural Language Dialogue tutor developed by Graesser and colleagues at the University of Memphis Graesser, Person, et al., 2001. AutoTutor is a learning environment that tutors students by holding a conversation in natural language [5]. AutoTutor has been developed for Newtonian qualitative physics and computer literacy. It was designed on the basis of explanation-based constructivist theories of learning, ITS that adaptively responds to student knowledge in a dialogue pattern. AutoTutor posts question to the student where he/she has to answer it in a sentence or a paragraph. These tutors give immediate feedback to the students actions and guide the learner on what to do next. This is done in a manner what the system infers from what the learner knows. The learners will be given positive or negative responses based on their answer to the tutors questions. The patterns of discourse uncovered in naturalistic tutoring were imported into the dialogue management facilities of AutoTutor.

B. ML Tutor

MLTutor [6] is a hypertext system that provides suggestions to the user on the basis of their recent browsing history, indicating pages that are relevant to the user's current area of interest. MLTutor is designed for use in an educational context on the World Wide Web (WWW). A small popup window appears along with the browser which guides the user throughout his search. This paper discuss the technical feasibility and the utility of applying machine learning algorithms to generate personalized adaptation on the basis of a user's browsing history in hypertext, without additional input from the user.

C. TRAC

Group work [7] is commonplace in many aspects of life, particularly in the workplace where there are many situations which require small groups of people to work together to achieve a goal. Like for software engineering, various groups has to put different efforts and all should be collaborated together at the end to get the result, i.e., good flawless software. The importance of group work skills is reflected in education systems, where students are given opportunities to develop these valuable skills. Often, such groups are supported by software tools. The approach is to analyze the traces to create mirroring tools that enable the group members, their teachers or facilitators to see useful indicators of the health and progress of their group.

The learners were students who registered for a senior software development project course. They were give online projects and were monitored for their group work. Over 12 weeks, and working in groups of 5-7 students, they were required to develop a software solution for a client. The topics varied from creating a computer-based driving ability test to developing an object tracking system for an art installation. Student teams were required to use TRAC [8] for online collaboration. TRAC is an open source, professional software development tracking system. It runs in the background which keeps tracks of all the activities of the students. It supports collaboration by integrating three tools.

A group wiki for shared web pages, a task management system also known as a ticketing system, Subversion (SVN) control system that provides a repository for the software created by the group and manages the changes made over time. It also allows to recover the older versions of the software and a view of the history of how the files and directories were changed.

D. Betty's Brain

Betty's Brain[9], an open-ended Teachable Agent learning environment that requires students to gain familiarity with , and employ a number of cognitive and metacognitive skills to achieve success in their learning and teaching tasks[10]. This is a different approach where students learn by teaching a virtual agent. In developing a computer-based learning environment (CBLE) called Betty's Brain, the authors have adopted a self-regulated learning (SRL) framework to help students develop learning strategies. As they explores the resources on a science topic, they construct a causal map to teach Betty, their virtual Teachable Agent (TA)[11]. Betty only knows what she has been taught by the student, but once taught, she can use this information to answer questions like "if deforestation increases, what effect it has on polar sea ice?"and explain her answers as a chain of causal relations. The student can also ask their TA to take quizzes.TheTA's quiz performance helps the students to assess on their TA's, and, therefore, their own learning performance.

E. Meta Tutor

MetaTutor[12] is a multi-agent, adaptive hypermedia learning environment, which presents challenging human biology science content. The primary goal underlying this online tutorial is to investigate how multi-agent system can adaptively scaffold SRL and metacognitive skills within the context of learning about complex biological content. MetaTutor is grounded in a theory of SRL that views learning as an active, constructive process whereby learners set goals for their learning and then attempt to monitor, regulate, and control their cognitive and metacognitive processes in the service of those goals [13].Here, learners must regulate their cognitive and meta cognitive processes in order to integrate multiple informational representations available from the system.

As a learning tool, MetaTutor has various features that uses and fosters self-regulated learning. These include four pedagogical agents which guide students through the learning session and prompt students to engage in planning, monitoring, and strategic learning behaviors. In addition, the agents can provide feedback to guide the students to attempt their goals. The system also uses natural language processing which is incorporated along with their conversation with the agent, to allow learners to express metacognitive monitoring and control processes.

III. CLUSTERING AND SEQUENCE MINING.

A. Auto Tutor

Auto Tutor does not use the concept of clustering to group the students based on their learning behavior, instead it uses a pattern mining algorithms to evaluate each student. AutoTutor uses Latent Semantic Analysis (LSA) [14] as its primary, conceptual, pattern-matching algorithm when evaluating whether student input matches the expectations and misconceptions.LSA is a high-dimensional statistical technique that measures the conceptual similarity of any two pieces of text. The size of the text may vary from a single word to a sentence, paragraph, or lengthier document. The length of the text doesn't matter only the words contained in it matters. LSA-based similarity metrics are capable of evaluating the quality of learner contributions almost as well as graduate students in the subject matter,i.e., physics or computer literacy. This use of LSA makes AutoTutor robust and domain portable.For each words, curriculum script content, and the documents that are stored in the corpus has its own LSA vectors.

The conversation with the tutor are stored in storage registers. Storage registers are frequently updated as the tutoring process proceeds. For example, AutoTutor keeps track of student ability (as evaluated by LSA from student assertions), student initiative (such as the incidence of student questions), student verbosity (number of words per student turn), and the incremental progress in having a question answered as the dialogue history grows, turn by turn. No two conversations with AutoTutor are ever the same because the dialogue management module of AutoTutor flexibly adapts to the student by virtue of these parameters.The dialogue management module is an *augmented finite state transition network*. The *nodes* in the network refer to knowledge goal states/dialogue states. The *arcs* refer to categories of tutor dialogue moves (e.g., feedback, pumps, prompts, hints). A particular arc is traversed when particular conditions are met i.e when a goal is met. AutoTutor always respond very differently, when the learner makes an assertion than when the learner asks a question. A learner who asks "Could you repeat that?" would probably not like to have "yes" or "no" as a response, but would like to have the previous utterance repeated. The classifier system has listed out 20 categories identified by Graesser and Person (1994). These categories include assertions, meta communicative expressions like "Could you repeat that?", "I can't hear you" etc. The classifier uses a combination of syntactic templates and key words for framing these questions or assertions by the tutor. After one expectation is being covered, AutoTutor moves on to cover another expectation .

B. ML Tutor

The design of MLTutor aims to remove the need for pre-defined user profiles and replace them with a

dynamic user profile-building scheme in order to provide individual adaptation. In MLTutor, this adaptation is achieved by a combination of conceptual clustering and inductive machine learning algorithms. In the first phase of the learning process, clustering is used to find out the inherent patterns within the hypertext pages browsed by a user. To find these inherent patterns within the hypertext a simple conceptual clustering algorithm (Hutchinson, 1994) is used. Applying this method eliminates the need for initial stereotypical profiles or any additional input from a user. In the second phase of learning, a rule induction procedure [15] is employed to generate rules. These rules describe the concept of cluster membership or relationships. This eliminates the need of hard clustering here. This process works on the basis that the components of a cluster are in some way related to each other and as such are representative of a concept. Based on this, the content of a cluster is assumed to represent positive examples of a concept and anything beyond the boundary of the cluster is taken to be non-representative of that concept. The information concealed in clusters was initially interpreted by the ID3 algorithm [16] which is a decision tree based method, to create a set of rules which can be viewed as dynamically created user profiles. The rules generated by this phase are used to search for other hypertext pages within the original population of pages.

The clustering and rule induction processes within MLTutor make use of an attribute database. The first stage of attribute database construction is to assemble a list of keywords from the Web pages within the system. The ID3 algorithm was applied to the data in the two clusters to induce decision rules for each cluster. Each cluster in turn was treated as positive training data for the rule induction process; the negative training data was the pages in the other cluster. The ID3 algorithm uses an *information theoretic heuristic* to produce shallower trees by deciding the order in which to select attributes. The first stage in applying the information theoretic heuristic is to calculate the proportions of positive and negative training cases that are currently available at a node. In the case of the root node, this is all the cases in the training set. Within the context of the MLTutor this was not ideal as the suggestion rule failed to take into account all *primary determinants* of the cluster. In order to rectify this weakness the SG-1 [17] algorithm was developed as an enhancement of ID3. In contrast, SG-1 has the ability to produce multiple decision trees and treats each possible alternative equally, building a subtree for each of them. Conceptually, the SG-1 algorithm can be visualized as building three-dimensional trees. The SG1 algorithm was implemented and tested with the conceptual clustering algorithm, and found to perform satisfactorily, overcoming the shortcomings of ID3 .

C. TRAC

In TRAC, before any data mining was carried out, the data was examined to see whether any simple statistics could distinguish the stronger from the weaker groups. Firstly, they checked the total number of ticket events for each group. Secondly, they looked at the distribution of the individual ticketing events (ticket created, accepted, reopened, and closed). Thirdly, they examined the usage span of the wiki pages, i.e. the time between the first and last event on the page. Clustering allows using multiple attributes to identify similar groups in an unsupervised fashion. It also provides the opportunity to mine the data at the level of individual learners i.e., to list groups of similar learners and then to examine the composition of each group. The most important problem was attribute selection. The performance of clustering algorithms is very sensitive to the quality of the attributes. Initially the authors chose a set of 8 numeric attributes representing ticketing behavior such as the number of tickets and ticket events; the number of days on which tickets were opened, closed, or a ticket event occurred; and the ticket usage span. Firstly they ran k-means with k=3 clusters. The number of clusters was set to 3 .

TABLE I
CLUSTERING USING K-MEANS K=3.

Clusters	Distinguishing characteristics
Groups 2,3,4 & 7	Over all ticketing activity
Groups 5 & 6	Many Tickets Fewer ticketing events Greater percentage of trivial and minor ticket priorities. Less accepting events.
Group1	Many tickets Many ticketing events. Lowest percentage of minor ticket priorities. More events where ticket priorities were changed or comments posted

Table I[7] shows the clustering of the groups, together with the extracted distinguishing characteristics of each cluster. The first cluster consists of Groups 2, 3, 4 and 7 and is characterized by overall low ticketing activity. Same clustering results was obtained for EM(Expectation Maximization and hierarchical agglomerative clustering [18].An important aspect of the data which is ignored by mining techniques such as clustering is the timing of events. They believe that certain *sequences of events* distinguish the better groups from the weaker ones. Such sequence may represent a characteristic team interaction on a specific resource, or group members displaying specific work patterns across the three aspects of TRAC.A data mining technique which considers this temporal aspect is sequential pattern mining [19] . It finds sequential patterns that occur in a dataset with at least a minimal level of frequency called support.

There are two main approaches to sequential pattern mining: apriori-based and pattern-growth methods [20]. Both of them find the same results but the pattern growth approach is much faster. They used an apriori based algorithm for two reasons. More specifically, They used a slightly modified version of the Generalized Sequential Pattern Mining [19] (GSP) algorithm. GSP is based on the so called apriori property which states that if a sequence is frequent then all its sub-sequences must be frequent as well. Based on this heuristic, GSP adopts a multiple-pass, candidate generation-and-test approach in sequential pattern mining

D. Betty Brain

This paper, Betty Brain, extends an exploratory data mining methodology for assessing and comparing students' learning behaviors from these interaction traces. The core algorithm employs sequence mining techniques to identify differentially frequent patterns between two predefined groups. The authors extend this technique by contextualizing the sequence mining with information on the student's task performance and learning activities. Specifically, they study transformation of action sequences using action features, such as activity categorizations, relevance and timing between actions, and repetition of analogous actions. They employ a linear segmentation algorithm in concert with the action changes and differential sequence mining [21] techniques to identify and compare segments of student productive and unproductive learning behaviors. The results of this methodology applied to a recent middle school class study, in which students learned about the climate change.

To effectively perform sequential data mining on learning interaction traces, raw logs must first be transformed into a set of sequence of actions. The methodology used here in corporate iterative refinement of the action abstraction step to focus the analysis on various learning activities and actions. Action abstraction is the first step of this data mining methodology, in which researcher identified categories of actions define an initial alphabet (set of action symbols) for the sequences. To apply the abstraction process, log events captured by the CBLE (Computer Based Learning Environment) are mapped to a sequence of actions taken by each student.

To identify important activity patterns in a comparison between two sets of action sequences (e.g READ, LINK, QUER, EXPL),it employs a novel combination of sequence mining techniques. Sequential pattern mining methods find the most frequent action patterns across a set of action sequences, while episode mining discovers the most frequently used action patterns within a given sequence. But, finding the patterns most important for interpreting learning behaviors or differentiating between groups of students is challenging, because of the need to limit the large set of frequent patterns to ones that are interesting.

In comparing across groups of action sequences, such as high-versus low-performing students, the differences between the groups provide a natural criterion for identifying important pattern that may find differences in learning behavior. To use this criterion for mining important frequent patterns, they define two measures of frequency and the corresponding differences calculated across the groups. The sequential pattern mining frequency is important for identifying pattern common to a group of action sequence. They refer to this as the "sequence support"(s-support) of the pattern following the convention of, and they call patterns meeting a given s-support thresholds-frequent.

First, a sequential pattern mining algorithm identifies the pattern meet a minimum s-support constraint within each group, employing a maximum gap constraint to account for noise, which is interpreted as a small number of irrelevant actions that may be interspersed in a pattern. To compare the identified s-frequent patterns across groups, they calculate the mean i-support of every pattern for each group. This comparison produce four distinct categories of frequent patterns: two categories where the pattern are s-frequent in only one group, illustrating patterns primarily employed by that group, and two categories where the pattern are common to both groups but used more often in one group than the other .The pattern in each of these qualitatively distinct categories are (separately)sorted by the difference in mean group i-support to focus the analysis on the most differentially frequent patterns. The results of this study presented an interesting opposing group in student performance at constructing their causal concept maps.

E. Meta Tutor

While data has been collected over a sample of 148 under graduate students from two large public universities in North America, they consider for this study only a sub-sample of 51 participants from the experimental condition that included the most prompts from the pedagogical agents to perform SRL actions and in which students were given some adaptive feedback after having performed those actions. For the analysis performed here, as justified inspection, they abstracted the set of collected interactions into three broad categories: reading, monitoring, and strategy.

They ran a cluster analysis over a subset of 13 variables extracted from the interaction log after the end of the student’s learning session: pretest and post test score, number of sub goal and page quizzes, mean first score in sub goal and page quizzes, proportion of sub-goals attempted among the 7 possible, number of sub goals changes, total session duration, time spent reading content, number of times the student took notes and checked notes, and the duration of the note-taking episodes. This analysis employed the Expectation-Maximization (EM) algorithm as implemented in the WEKA[23] data mining package. They used 1000 different initialization seeds for the EM algorithm, in order to compensate for its tendency to get stuck into local optima, and selected, among the 1000 partitions of students generated, the most frequent one among the most frequently obtained number of clusters. Three clusters were obtained, which characteristics are summarized in Table II [12], where clusters 0, 1 and 2 are made of 21,14 and 16 students, respectively. Generally, students from cluster 2 scored high on pretest, posttest and intermediary quizzes, and spent less time than others reading while attempting more sub goals, and took less notes and less time taking them. In contrast, students from cluster 1 scored low on pretest, posttest and intermediary quizzes, attempted less sub goals and took few notes and less time to take them. Students from cluster 0 occupied generally a intermediate position in terms of performance and sub goal uses, but took overall more notes and more time to take them.

TABLE II
SYNTHESIS OF CLUSTER DIFFERENCES.

Variables	Score for each cluster		
	0	1	2
Pretest Score	M	L	H
Posttest score	M	L	H
Session duration	M	M	M
Proportion of subgoals attempted	H	H	L
Number of subgoal changes	L	L	H
Number of subgoal quizzes	M	L	H
Mean first score in subgoal quizzes	M	M	M
Number of page quizzes	M	L	H
Mean first score in page quizzes	M	M	M
Number of note taking	H	L	L
Number of note checking	L	L	H
Time spent taking notes	H	L	L

To identify important activity patterns in comparison between student clusters, they employ a differential sequence mining technique. This technique uses sequence mining and two different measures of pattern frequency to identify differentially frequent patterns between two sets of action sequences. Differential sequence mining combines frequency measures and techniques from sequential pattern mining, which determines the most frequent action patterns across a set of action sequences, and episode mining, which determines the most frequently used action patterns within a given sequence. The sequential pattern mining frequency is used to identify patterns common to a group of students. They refer to this as the “sequence support”(s-support)of the pattern, and they call patterns meeting a given s-support threshold s-frequent. In this analysis, they employ a s-support threshold of 0.5 to focus on patterns exhibited by at least half of a given group of students.

For a given student, it refer to this as the “instance support”(i-support),and they call patterns meeting a given i-support threshold i-frequent. To calculate the i-support of a pattern for a group of students, they use the mean of the pattern’s i-support values across all traces in the group. The differential sequence mining technique first uses a sequential pattern mining algorithm to identify the patterns that meet a minimum s-support constraint within each group. To compare the identified frequent patterns across groups, they calculate the i-support of each pattern for each student (in each group). Using a t-test, they filter the s-frequent patterns to identify those for which there is a statistically significant difference in i-support values between groups. Comparing the mean i-support value for each pattern between groups then allows us to focus the comparison on patterns that are employed significantly more often by one group than the other. This comparison produces four distinct categories of frequent patterns. The patterns in each of these qualitatively distinct categories are (separately) sorted by the difference in mean group i-support to focus the analysis on the most differentially frequent patterns .

To summarize the results obtained, they conclude that students from cluster H are more inclined to follow the system prompts and to follow the suggestions to take notes or summarize what they have just learned. On the other hand, students from cluster L appear particularly unable to identify pages relevant to their sub goal, which is probably linked to their lower prior knowledge. .

IV. DISCUSSIONS AND REMARKS

Auto Tutor uses natural language dialogue facilities are not expected to do a reasonable job in all conversational contexts. The quality of the discourse depends on the subject matter, the knowledge of the learner, the expected depth of comprehension, and the expected sophistication of the dialogue strategies. Natural language dialogue facilities are unlikely to be impressive when the subject matter requires mathematical or analytical precision and when the user would like to converse with a witty or humorous conversation partner. In contrast, a natural-language dialogue facility is more feasible in applications that involve imprecise verbal content, low-to-medium user knowledge about a topic, and earnest literal replies. There is no grouping for students based on their behavior. Each student is evaluated independently.

A number of technical issues with the conceptual clustering algorithm used in the MLTutor were identified. Although faults in the experimental design limit the conclusions that can be drawn about MLTutor, the results of the evaluation do show that MLTutor is a robust and functional system and suggest the potential benefits of using a machine learning approach to provide adaptively based on an analysis of browsing patterns.

In TRAC The main limitation was the small data sample, especially in the first task, clustering of groups. Although the data contained more than 15000 events, they had only 7 groups and 43 students. Nevertheless, they think that the collected data and selected attributes allowed for uncovering useful patterns characterizing the work of stronger and weaker students as discussed above. How to select the most appropriate clustering algorithm and how to set its parameters is another important issue. There are methods for determining a good number of clusters and evaluating the clustering quality in terms of cohesion and separation of the clusters found. For larger datasets, hierarchical clustering may not be applicable due to its high time and memory requirements; k-means may be still a good choice, especially some of its modifications, such as bi-secting k-means which is less sensitive to initialization and is also more efficient. Also there was a number of issues emerged during the use of sequential pattern mining, ranging from limitations in the data to how output was interpreted. Currently the data contains only modification and creation events. The common situation where a team member views another's work but does not feel the need to modify it was thus effectively ignored. This emphasizes the need to incorporate data from sources such as web logs. A problem with the mining program itself is the lack of gap constraints.

In Betty's Brain performance was linked to monitoring behaviors that incorporated re-reading of the resource material. In particular, high-performers were more likely to employ various types of reading actions both before and after assessments of progress/correctness using the quiz. Low performers, on the other hand, had a differential tendency to use irrelevant, extended re-reads of pages in the resources during productive periods. They may have also had more difficulties identifying the less obvious causal relations, as suggested by their greater reliance on initial reads of a resource page during productive periods.

In Meta Tutor, the analysis performed allows to more accurately identify the group to which a student belongs during their use of Meta Tutor and dynamically adapt the scaffolding and feedback mechanisms accordingly. However there is no mechanism to enhance the identification and understanding of phases when low-performing students are unable to properly monitor their learning.

V. CONCLUSION

The Intelligent Tutoring Systems discussed in this survey points out how the traditional clustering algorithm can be used in a different context. They identify different groups of students based on their learning behavior. It helps in guiding the students to achieve their goals..Reluctantly all the tutors do not enhance the performance of low category students so that they achieve their goals within the time limit. In addition, the initial hard clustering is based on a written pretest or previous examination results, which forms the seed to the clustering. The choice of this data plays an important role in further clustering

REFERENCES

- [1] Michell T.M, "Does Machine Learning Work?" *AI magazine* Vol.18,No.3pp 11-20,1997.
- [2] <http://www.educationaldatamining.org>.. Educational Datamining,2008.
- [3] VincentA.W.M.M.Aleven,, KennethR.Koedinger "An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor" *CognitiveScience*, Vol-26,pp147-179, Jan 2002.
- [4] Aurthur C Grasser,,ShulanLU,George Tanner Jackson "AutoTutor:A tutor with dialogue in natural language"*IEEE Transactions on education*,vol. 48, no. 4,pp 612 -618, November 2005.
- [5] Aurthur C Grasser, A C Wiemer,"Autotutor : A simulation of a human tutor ", *Journal of Cognitive Systems Research*, pp 35-37, 1999

- [6] A SerengulGuvenSmith ,AnnBlandford “MLTutor:An Application of Machine Learning Algorithms for an Adaptive Web-based Information System” *International Journal on Artificial Intelligence in Education,version 14*,pp 51-54,Dec 2007.
- [7] DilhanPerera,JudyKay,IrenaKoprinska “Clustering and Sequential pattern mining of Online collaborated data” *IEEE transactions on Knowledge and Data Engineering*, 21(6):759-772,June 2009.
- [8] <http://trac.edgewall.org>.
- [9] James R. Segedy,John S Kinnebrew,GautamBiswas,”Supporting Student Learning using conversational agents in a teachable agent environment.”,*International Journal of Artificial Intelligence in Education* 18(3),pp181-208,2008.
- [10] J.Bransford,A.Brown,andR.Cocking,editors. “How people learn “. *National Academy PressWashington,DC*,2000.
- [11] G.Biswas, K.Leelawong, D.Schwartz,N.Vye, and T. Vanderbilt. “Learning by teaching:A new agent Paradigm for educational software”. *Applied Artificial Intelligence*, 19(3):363{392,2005.
- [12] FrancisBouchet,RogerAzevedo,JohnS.Kinnebrew,GautamBiswas “Identifying Students’Characteristic Learning Behaviors in an Intelligent Tutoring System Fostering Self-Regulated Learning “. *International Journal on Educational Data Mining*, 2012.
- [13] R.Azevedo, D.C.Moos, A.Johnson, and A.Chauncey “Measuring cognitive and metacognitive regulatory processes used during hypermedia learning:Issues and challenges”.*Educational Psychologist*,45(4):pp210-223,2010.
- [14] B.Zimmerman and D.Schunk, editors. “Handbook of self-regulation of learning and performance”.*Routledge*,NewYork,2011.
- [15] Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N. K., “Using latent semantic analysis to evaluate the contributions of students in AutoTutor”. *Interactive Learning Environments*Vol8, pp129-148,The Tutoring Research Group (2000).
- [16] Quinlan J. R. “Induction of Decision Trees”. *Machine Learning*, Vol.1, pp 81-106.
- [17] Michalski R. S.” Pattern recognition as rule-guided inductive inference”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-2, pp 349-361. 1986.
- [18] P.-N. Tan, M. Steinback, and V. Kumar, “ Introduction to Data Mining:“ Pearson Addison Wesley, 2006.
- [19] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," *Proc. Fifth Int'l Conf. Extending Database Technology (EDBT)*, Avignon, France, 1996.
- [20] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Datal, and M.-C. Hsu, "Freespan: Frequent Pattern-Projected Sequential Pattern Mining," *Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, Boston, USA, 2000.
- [21] R.Agrawal and R.Srikant.”Mining sequential patterns” In Proceedings of *The Eleventh IEEE International Conference on Data Engineering* (ICDE),pp3-14,1995.
- [22] J.S.Kinnebrew,K.M Loretz and G.Biswas. “A differential sequence mining method contextualized by student performance evolution to derive learning behaviour patterns.”*International Journal of Educational Data Mining*,2012
- [23] M.Hall,E.Frank,G.Holmes,B.Pfahringner, P.Reutemann, and I.H.Witten. “The WEKA datamining software:an update”. *SIGKDD Explor.Newsl*,11(1), pp 10-18Nov.2009.