World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:3, 2008

Decision Rule Induction in a Learning Content Management System

Nittaya Kerdprasop, Narin Muenrat, and Kittisak Kerdprasop

***Abstract*—**A learning content management system (LCMS) is an environment to support web-based learning content development. Primary function of the system is to manage the learning process as well as to generate content customized to meet a unique requirement of each learner. Among the available supporting tools offered by several vendors, we propose to enhance the LCMS functionality to individualize the presented content with the induction ability. Our induction technique is based on rough set theory. The induced rules are intended to be the supportive knowledge for guiding the content flow planning. They can also be used as decision rules to help content developers on managing content delivered to individual learner.

***Keywords*—**Decision rules, Knowledge induction, Learning content management system, Rough set.

I. Introduction

THE term learning content management system (LCMS) refers to a suite of software tools designed to facilitate learning developers to create, manage and deliver learning content to distant learners [2]. The main features of an LCMS include content creation, content repository management, content delivery and interface, and learning process management such as course enrollment, assessment and performance tracking. An LCMS is adaptive and scalable in that creates proprietary content to meet the needs of individual learner.

The system offers course developers a feature to create and manage learning objects as customized content. Thus, the course development process can be viewed as a compilation of pieces of content retrieved from content repository to fit unique needs of different learners.

We, therefore, propose a knowledge induction technique to support course developers in designing flow of content appropriate to the ability of each learner. The induced knowledge is in the form of rules which are observed from the performance history of previous learners. These rules play the role of decision support in the course planning phase.

Decision support normally involves the integration of data and knowledge management to assist human on making effective and efficient choices [5], [16]. In the context of online course content delivery scalable to fit unique individual, decision making is on the basis of constant changing requirements that require a quick response. Traditional intuitive methods of decision making are no longer adequate to deal with such complicate situation. We consider rough set theory as a methodology to identify useful trends by exploring current and historical data.

Rough set theory is a new mathematical tool to deal with incomplete and inconsistent information. The theory was proposed by Pawlak, a Polish mathematician, in 1982 [12]. Since then it has drawn much attention from researchers interested in its theoretical aspects and applications [1], [4], [6], [8], [10], [18], [20]. Recent successful applications in the domains of machine learning and knowledge discovery have been reported [9], [11], [15], [19].

We study rough set theory within the framework of learning content management system. We focus on content creation that exploits rough set techniques as a tool to guide the decision on course content planning suitable to the learning performance of each learner.

The paper is organized as follows. Section 2 reviews the basic concepts of rough sets. Section 3 presents our idea of decision making with the induced patterns based on rough set approach. Section 4 illustrates the idea through the running examples. Section 5 concludes the paper.

II. Preliminaries on Rough Set

The notion of rough sets has been introduced by Zdzislaw Pawlak in the early 1980s [12], [13], [14] as a new concept of set with uncertain membership. Unlike fuzzy set, uncertainty in rough set theory does not need probability or the value of possibility to deal with vagueness. It is rather formalized through the simple concepts of lower and upper approximation, which are in turn defined on the basis of set. We present the basic concepts and terminology of rough sets within the framework of decision support system (DSS) [3].

Given the input data, the rough set-based DSS generates a list of certain and possible decision rules. The input data is a

Manuscript received March 26, 2008. This work was supported in part by the National Research Council of Thailand. DEKD research unit is fully supported by Suranaree University of Technology.

Nittaya Kerdprasop is a principal researcher of DEKD research unit and an associate professor at the School of Computer Engineering, Suranaree University of Technology, 111 University Ave., Nakhon Ratchasima 30000, Thailand (e-mail: nittaya@sut.ac.th, nittaya.k@gmail.com).

Narin Muenrat is a master student of Computer Engineering School, Suranaree University of Technology.

Kittisak Kerdprasop is a director of the Data Engineering and Knowledge Discovery (DEKD) research unit, School of Computer Engineering, Suranaree University of Technology, 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand (phone: +66-44-224349; fax: +66-44-224602; e-mail: kerdpras@ sut.ac.th, KittisakThailand@gmail.com).

decision table comprising of conditional attributes, or conditions for short, and a decision attribute. Table I gives an example of a decision table containing information of eight students. Conditions are number of times the students log-in the system to access the online course and two pretest scores (intervals of numeric values). The level attribute (either basic or advanced) is a decision. Conditions together with decision attribute form a decision system.

TABLE I
A DECISION TABLE JUDGING STUDENTS' PERFORMANCE

	Conditions			*Decision*
	log-in (c1)	score1 (c2)	score2 (c3)	level
s1	15	0-20	0-20	Basic
s2	15	0-20	21-40	Basic
s3	20	0-20	41-60	Basic
s4	20	0-20	41-60	Basic
s5	15	0-20	81-100	Advanced
s6	15	41-60	41-60	Advanced
s7	15	21-40	61-80	Advanced
s8	20	21-40	21-40	Advanced

A decision table is a representation of real-world data. Each row represents one object. Rough set theory is based on the formation of equivalence relations [7], [17] within the given data**.**

Definition 1: A *decision system* is any system of the form $\mathcal{A}$ = <U, A, d>, where U is a non-empty finite set of objects called the universe, A is a non-empty finite set of conditions, and d $\notin$ A is the decision attribute.

Definition 2: Given a decision system $\mathcal{A}$ = <U, A, d>, then with any B $\subseteq$ A there exists an *equivalence* or *indiscernibility relation* $I_{\mathcal{A}}$(B) such that

$$I_{\mathcal{A}}(B) = \{(x, x') \in UxU \mid \forall a \in B\ [a(x) = a(x')]\ \}.$$

From the data samples in Table I, the followings are equivalent relations.

I(c1) = {{s1, s2, s5, s6, s7}, {s3, s4, s8}}
I(c2) = {{ s1, s2, s3, s4, s5}, {s6}, {s7, s8}}
I(c3) = {{s1},{s2, s8},{s3, s4, s6}, {s5}, {s7}}
I(c1,c2) = {{s1,s2,s5},{s3,s4},{s6},{s7},{s8}}
I(c1,c3) = {{s1},{s2},{s3,s4},{s5},{s6},{s7},{s8}}
I(c2,c3) = {{s1},{s2},{s3,s4},{s5},{s6},{s7},{s8}}
I(c1,c2,c3) = {{s1},{s2},{s3,s4},{s5},{s6},{s7},{s8}}

These equivalence relations partition the universe into groups of similar objects based on the values of some attributes. The question often arises is whether one can remove some attributes and still preserve the same equivalence relations. This question leads to the notion of *reduct* [7].

Definition 3: Let $\mathcal{A}$ = <U, A, d> be a decision system and P, Q $\subseteq$ A be sets of conditions, P $\neq$ Q. The set P is the *reduct* of set Q if P is minimal (i.e. no redundant attributes in P) and the equivalence relations defined by P and Q are the same.

It can be seen from the listed equivalence relations that I(c1,c3) = I(c2,c3) = I(c1,c2,c3). Therefore, (log-in, score1) and (score1, score2) are reducts of (log-in, score1, score2). Either reduct can be used as a representative set of attributes. The intersection of all reducts produces *core attributes*. According to our example, score2 is a core attribute. A reduct table of (score1, score2) and its partitions are shown in Fig. 1.

	score1	score2	level
s1	0-20	0-20	basic
s2	0-20	21-40	basic
s3	0-20	41-60	basic
s4	0-20	41-60	basic
s5	0-20	81-100	advanced
s6	41-60	41-60	advanced
s7	21-40	61-80	advanced
s8	21-40	21-40	advanced

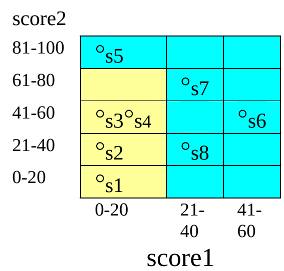

Fig. 1 A reduct table and its partition into equivalence relations, each equivalence relation is represented by a rectangular region

If we are interested in the decision criteria for the advanced-level students, we can infer decision rules from Fig. 1 as follows.

IF (score1=0-20 $\wedge$ score2=81-100) THEN level = advanced
IF (score1=21-40 $\wedge$ score2=21-40) THEN level = advanced
IF (score1=21-40 $\wedge$ score2=61-80) THEN level = advanced
IF (score1=41-60 $\wedge$ score2=41-60) THEN level = advanced

The decision criteria for basic-level students can be inferred accordingly.

IF (score1=0-20 $\wedge$ score2=0-20) THEN level = basic
IF (score1=0-20 $\wedge$ score2=21-40) THEN level = basic
IF (score1=0-20 $\wedge$ score2=41-60) THEN level = basic

Suppose we are given additional information of the ninth student as shown in Fig. 2, then the above decision rules for the advanced-level students is no longer valid. It can be seen from Fig. 2 that s8 and s9 are in the same equivalence relation but their performance levels are different. It is such conflicting cases that inspire the rough set concept. Given the two decision sets of advanced/basic level, the uncertain cases such as s8 and s9 can be approximated their membership by means of lower and upper approximation [7].

Definition 4: Let $\mathcal{A}$ = <U, A, d> be a decision system, B ⊆ A, X ⊆ U and $[x]_B$ denote the equivalence class of $I_{\mathcal{A}}(B)$. The *B-lower approximation* and *B-upper approximation* of X, denoted by *bX* and *BX* respectively, are defined by $bX = \{x \mid [x]_B \subseteq X\}$ and $BX = \{x \mid [x]_B \cap X \neq \varnothing\}$.

	score 1	score 2	level
s1	0-20	0-20	basic
s2	0-20	21-40	basic
s3	0-20	41-60	basic
s4	0-20	41-60	basic
s5	0-20	81-100	advanced
s6	41-60	41-60	advanced
s7	21-40	61-80	advanced
s8	21-40	21-40	**advanced**
s9	21-40	21-40	**basic**

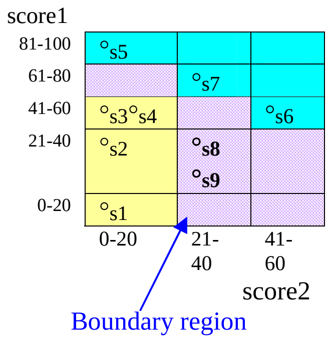

Fig. 2 A decision table with conflict cases on students' performance level

Given the information as shown in Fig. 2, B = {score1, score2} and X = {s5, s6, s7, s8} is set of students with advanced-level performance, then the lower approximation *bX* = {s5, s6, s7} and the upper approximation *BX* = {s5, s6, s7, s8, s9}. That is, the lower approximation of X is the set of all objects that are certainly belong to X. This set is also called *B-positive region* of X. The *B-negative region* of X is defined as U−BX, i.e. {s1, s2, s3, s4} or the set of all objects that definitely not belong to X. The area between these two sets is called *B-boundary region* of X, denoted by *BN*, and defined as *BN* = *BX*−*bX*. It is the set of all objects that cannot be classified as not belonging to X.

If the boundary region is empty, it is a *crisp* (precise) set; otherwise, the set is *rough*. The set of advanced-level students in Fig. 1 is a crisp set, but it is a rough set in Fig. 2. Decision rules generated from a rough set comprise of certain rules generated from the positive and negative regions, and possible rules generated from the boundary region.

A method to generate decision rules as explained above is static. The decision attribute is defined in advance. Within the framework of DSS that decision problems are usually not known in advance, such classical rough set methodology is inadequate. We thus propose in the next section our method of dynamic decision rule induction.

III. Rough Set Based Rule Induction

In the traditional environment of DSS, decision rules are manually encoding by knowledge engineers. It is a time consuming process that requires close collaboration between experts of the field and the computer professionals. With the emerging of data warehousing technology, we have a huge valuable resource of knowledge from which we can induce useful decision rules. But with the classical rough set method, the number of generated decision rules is tremendous. We propose a different approach of inducing certain and possible decision rules; the induction process is triggered by the query. The information regarding learners is stored in a table form and decision rules on content management guiding are induced by posing query on any students' attribute. By this scheme, we can limit the induction to only relevance rules. The framework of our approach is shown in Fig. 3.

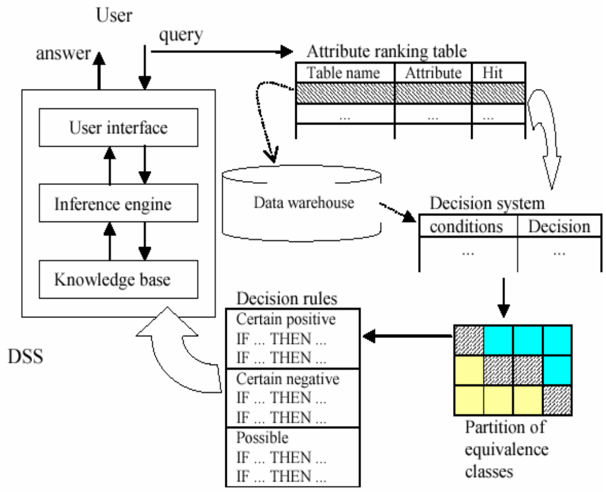

Fig. 3 The induction of rough and precise knowledge

Our proposed framework of decision rule induction is invoked by query. Once the query has been posted, the auxiliary data structure named *attribute ranking table* has been updated with the table's and attribute's name extracted from the query. The column *hit* counts the number of times that attributes has been used. The counter is sorted in descending order to place the most frequently used attribute in the first row. This attribute always referred to by users' queries, therefore it is worth generating decision rules based on this attribute value. The approach of inducing decision rules based on the most frequently asked attribute is described in the following algorithm.

Algorithm Decision Rule Induction

Input: User's query and a data warehouse
Output: Decision rules

Step:
1. Extract table names T_i and attribute names A_j from the query
2. Access the attribute ranking (AR) table and update the hit counter identified by each T_i and A_j
3. Descending sort the AR-table according to the hit value
4. Extract the top row of AR-table to obtain T_1 and A_1
5. Create a decision table $\mathcal{A}$ = <U, A, d> where d = A_1, A = a set of attributes in T_1, U = a set of records in T_1
6. Pre-process $\mathcal{A}$ by
   - removing attributes with number of distinct values = $|T_1|$

- discretizing attributes with real values

7. Partition U into equivalence classes
8. Search for the first reduct R
9. Identify *bX*, *BX*, *BN* regions
10. From R, *bX*, *BX*, and *BN*, generate certain, negative, possible rules
11. Generalize all three classes of decision rules using dimension tables and hierarchical information from the data warehouse
12. Insert rules into the knowledge base

IV. Running Examples

We use the student data shown in Table I with additional record <s9, 20, 21-40, 21-40, basic> as our running example. The hierarchical information on interval order that 81-100 > 61-80 > 41-60 > 21-40 > 0-20 is used as background knowledge for decision rule generalization.

Example 1: Suppose there is a query consulting the system whether the score1 = 55 is high enough for the assigning the student to the advanced level.

Method:

(1) This query asks about student's level with score1 as a condition. Hence, a reduct decision table as in Fig. 2 is constructed.

(2) Then, the following decision rules are generated.

Certain positive rules

IF (score1=0-20 ∧ score2=81-100)
THEN level = advanced
IF (score1=21-40 ∧ score2=61-80)
THEN level = advanced
IF (score1=41-60 ∧ score2=41-60)
THEN level = advanced

Certain negative rules

IF (score1=0-20 ∧ score2=0-20)
THEN level = basic
IF (score1=0-20 ∧ score2=21-40)
THEN level = basic
IF (score1=0-20 ∧ score2=41-60)
THEN level = basic

Possible rules

IF (score1=21-40 ∧ score2=21-40)
THEN level = advanced

(3) The three classes of decision rules are generalized according to the background knowledge. The final decision rules are as follow.

R1: IF (score1 > 20 ∧ score2 > 60)
THEN level = advanced
R2: IF (score1 > 40 ∧ score2 > 40)
THEN level = advanced
R3: IF (score1 > 20 ∧ score2 > 20)
THEN level = possibly advanced

Notice that with the given information there is no matching rules from the negative class and R2 can be applied to answer this query.

Answer:

IF score2 > 40 THEN level = advanced.
IF score2 > 20 THEN level = possibly advanced.

Example 2: From the response of example 1, suppose the user wants to consult the system further that based on the information of her first pretest score, could the system predicts her second pretest score.

Method:

(1) The query asks the value of score2, given the value of score1=55. Thus, a decision attribute is score2 and a decision table is as shown in Table II.

TABLE II
A Decision Table with Respect to Example 2

	Conditions			*Decision*
	log-in	score1	level	score2
s1	15	0-20	Basic	0-20
s2	15	0-20	Basic	21-40
s3	20	0-20	Basic	41-60
s4	20	0-20	Basic	41-60
s5	15	0-20	Advanced	81-100
s6	15	41-60	Advanced	41-60
s7	15	21-40	Advanced	61-80
s8	20	21-40	Basic	21-40
s9	20	21-40	Advanced	21-40

(2) There is no reduct. So, all conditional attributes are used in the approximation of *bX*, *BX*, and *BN* regions. The decision objectives (X) are set of all students whose score2 values are in the range 0-20, 21-40, 41-60, 61-80, and 81-100. From the approximation, these rules are induced:

Certain rules

IF (log-in=15 ∧ level = advanced ∧ score1=0-20)
THEN score2= 81-100
IF (log-in=15 ∧ level = advanced ∧ score1=21-20)
THEN score2=61-80
IF (log-in=20 ∧ level = basic ∧ score1=0-20)
THEN score2=41-60
IF (log-in=15 ∧ level = advanced ∧ score1=0-20)
THEN score2=41-60
IF (log-in=20 ∧ score1=21-40)
THEN score2=21-40

Possible rules

IF (log-in=19 ∧ score1=20 ∧ level=basic)
THEN score2=0-20 ∨ 21-40

(3) Generalized decision rules are as follow.

R1: IF (score1 = 0-20)
THEN score2 = 81-100
R2: IF (score1 = 21-40)
THEN score2 = 61-80
R3: IF (score1 = 0-20 ∨ 41-60)

THEN score2 = 41-60
R4: IF (log-in=20 $\wedge$ score1 = 21-40)
THEN score2 = 21-40
R5: IF (log-in=15 $\wedge$ score1 = 20 $\wedge$ level=basic)
THEN possibly score2 = 0-40

Answer:
IF score1 = 55 THEN score2 = 41-60.

V. Conclusion

In this paper we propose a technique of decision rule induction to induce knowledge that can facilitate the content management in the learning content management system. The induction process is based on the rough set theory. Our assumption is that system with the availability of a warehouse as a data and knowledge repository may produce tremendous amount of decision rules. We thus limit the number of rules by inducing only rules that are relevant to user's need. Relevancy is guided by query predicates. We propose the framework of the system and the algorithm of decision rule induction. The intuitive idea is illustrated trough running examples. Our proposed idea is general, so it can be applied to any kind of domain. We plan to test the effectiveness of our framework with the real-world data in the future.

References

[1] Y. Caballero, R. Bello, A. Taboada, A. Nowe, M. Garcia, and G. Casas, "A new measure based in the rough set theory to estimate the training set quality", *Proc.8th Int. Symp. Symbolic and Numeric Algorithms for Scientific Computing*, pp.133-140, 2006.
[2] B. Chapman and B. Hall, *Learning Content Management System.* Brandonhall.com, New York, 2005.
[3] Z. Chen, *Computational Intelligence for Decision Support*, CRC Press, 2000.
[4] K. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publishers, 1998.
[5] C.W. Holsapple and A.B. Whinston, *Decision Support Systems: A Knowledge-Based Approach*, West Publishing Company, 1996.
[6] X. Hu, "Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining application", *Proc.IEEE ICDM*, pp.233-240, 2001.
[7] J. Komorowski, L. Polkowski, and A. Skowron, "Rough sets: A tutorial", In S.K. Pal and A. Skowron (eds.), *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, Springer, pp.3-98, 1999.
[8] A. Lenarcik and Z. Piasta, "Probabilistic rough classifiers with mixture of discrete and continuous variables", In T.Y. Lin and N. Cercone (eds.), *Rough Sets and Data Mining: Analysis for Imprecise Data*, Kluwer Academic Publishers, pp.373-383, 1997.
[9] D. Miao and L. Hou, "A comparison of rough set methods and representative inductive learning algorithms", *Fundamenta Informaticae*, vol.59, pp.203-218, 2004.
[10] P. Pattaraintakorn, N. Cercone, and K. Naruedomkul, "Hybrid intelligent systems: Selecting attributes for soft computing analysis", *Proc.29th Int. Conf. Computer Software and Applications*, pp.319-325, 2006.
[11] S. Pal and P. Mitra, "Case generation using rough sets with fuzzy representation", *IEEE Trans. Knowledge and Data Engineering*, vol.16, no.3, pp.292-300, 2004.
[12] Z. Pawlak, "Rough sets", *Int. Journal of Information and Computer Science*, vol.11, no.5, pp.341-356, 1982.
[13] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, 1991.
[14] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, "Rough sets", *Communications of the ACM*, vol.38, no.11, pp.88-95, 1995.
[15] J. Peters, D. Lockery, and S. Ramanna, "Monte Carlo off-policy reinforcement learning; A rough set approach", *Proc. 5th Int. Conf. Hybrid Intelligent Systems*, pp.187-192, 2005.
[16] F. Radermacher, "Decision support systems: Scope and potential", *Decision Support Systems*, vol.12, pp.257-265, 1994.
[17] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems", In R. Slowinski (ed.), *Intelligent Decision Support, Handbook of Applications and advances of the Rough Set Theory*, Kluwer Academic Publishers, pp.331-362, 1992.
[18] R. Swiniarski, "Rough sets and principal component analysis and their applications in feature extraction and selection, data model building and classification", In S. Pal and A. Skowron (eds.), *Fuzzy Sets, Rough Sets and Decision Making Processes*, Springer, 1998.
[19] L. Yang and L. Yang, "Study of a cluster algorithm based on rough sets theory", *Proc. 6th Int. Conf. Intelligent Systems Design and Applications*, pp.492-496, 2006.
[20] W. Ziarko, "The discovery, analysis, and representation of data dependencies in databases", In G. Piatetsky-Shapiro and W.J. Frawley (eds.), *Knowledge Discovery in Databases*, AAAI Press, pp.195-209, 1991.

Nittaya Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. She received her B.S. from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, USA, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, AI, Logic Programming, Deductive and Active Databases.

Narin Muenrat received his bachelor degree in Information Technology in 2003 from Suranaree University of Technology. He is currently pursuing a master degree in computer engineering at Suranaree University of Technology. His research interests are content and learning management systems, web technology and open source programming.

Kittisak Kerdprasop is an associate professor at the school of computer engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, USA., in 1999. His current research includes Data mining, Artificial Intelligence, Functional Programming and Computational Statistics.