International Journal of Engineering Education Vol. 27, No. 5, pp. 992–1002, 2011
Printed in Great Britain

0949-149X/91 $3.00+0.00
© 2011 TEMPUS Publications.

Predicting Academic Performance Using a Rough Set Theory-Based Knowledge Discovery Methodology*

ELEAZAR GIL-HERRERA,[1] ATHANASIOS TSALATSANIS,[2] ALI YALCIN[1] and AUTAR KAW[3]

[1] University of South Florida, Industrial and Management Systems Engineering Department, 4202 East Fowler Ave. ENB 118, Tampa, FL 33620 USA. E-mail: eleazar@mail.usf.edu ayalcin@usf.edu
[2] University of South Florida, Center for Evidence-based Medicine and Health Outcomes Research, 12901 Bruce B. Downs Blvd. MDC27, Tampa, FL 33612 USA. E-mail: atsalats@health.usf.edu
[3] University of South Florida, Mechanical Engineering Department, 4202 East Fowler Ave. ENB 118, Tampa, FL 33620 USA. E-mail: kaw@usf.edu

In an effort to predict student performance in an engineering course, Rough Set Theory (RST) is employed as the core of a knowledge discovery process. Student performance is captured in terms of successful course completion. Therefore, students are classified into two categories: those who pass a course and those who do not. The Rough Set Theory paradigm presented here analyzes each student based on a set of attributes. These attributes are collected through a series of surveys conducted in the first week of the course, allowing for early identification of potential unsuccessful students. Variations of the Rough Set approach are evaluated to determine the one most suited for the particular dataset. The results are promising since the accuracy of student performance prediction presents an *Area under the Receiver Operating Characteristic Curve* equal to 80%. The benefits anticipated from early identification of weak and/or potentially unsuccessful students will enable educators to engage these students at the onset of the course and enroll them in additional activities to improve their performance.

Keywords: academic performance prediction; linear systems; rough set theory; knowledge discovery

1. Introduction

Knowledge discovery is the research area concerned with analyzing existing information and extracting implicit, previously unknown, hidden and potentially useful knowledge in an automated manner [1, 2]. The core of the presented knowledge discovery process is Rough Set Theory (RST) [1], an extension to classical Set Theory used to represent incomplete or imperfect knowledge. RST combines theories such as fuzzy sets [3], evidence theory [4] and statistics, hence is able to cope with the shortcomings of these underlying theories.

In this paper, we describe the application of an RST-based knowledge discovery process in predicting student performance in an undergraduate engineering course. We measure student performance in terms of successful completion of a course. In this context, we classify students into two categories: *Passing students* are those who complete the course with a passing grade and *Failing students* are those who fail to complete the course or receive a failing grade. Note that the failing students category is used in the generic sense and includes those students who withdraw from it. The dataset for this study consists of information collected from two distinct groups of students enrolled in two different classes of the course. Student information was collected through a series of surveys conducted in the first week of the classes.

The rest of the paper is organized as follows: Section 2 presents a review of the recent work in predicting student performance in a single course. Section 3 describes the dataset utilized in this study and Section 4 presents in detail each of the steps involved in the RST-based knowledge discovery process and their application to predicting student performance. Section 5 discusses our results and finally, Section 6 concludes this paper.

2. Literature review

A variety of methodologies has been proposed to predict student performance in academic settings with the majority of them relying on statistical and soft computing techniques. The specific topic of academic performance prediction in a single course is dominated by regression-based statistical approaches. Recent notable efforts based on regression analysis appear in [5–15].

Soft computing techniques have found application in student performance prediction in the broader sense of overall academic success and retention in [16–19]. There are also notable efforts in applying these techniques to student performance prediction in a single course.

Hamalainen and Vinni [20] compared five student performance classification methods; two multiple linear regression and three versions of naïve Bayes classifiers. Students were classified into a passing

* Accepted 16 April 2011.

and failing group based on the final course grade. The factors considered for all five classifiers were based on six cognitive areas of programming courses. The authors show that the Bayes classifier had very good prediction accuracy.

Vandamme et al. [21] proposed three classification models to measure the probability of failing a course. The authors considered in their study sociological attributes, class attendance, prior academic experience regarding mathematics, study skill, and student self-confidence. The authors used data from three academic institutions from Belgium.

Fang and Lu [22] developed a prediction methodology based on a decision tree to predict student performance in a core engineering course. Based on the grades of four prerequisite courses and the cumulative GPA of the student, nine "if-then" decision rules were generated to predict student performance represented by the final course grade. It was revealed that a student's grade in one of the prerequisite courses and the cumulative GPA govern student performance. The prediction accuracy of the Decision Tree model was tested using data from two different semesters with remarkable accuracy. The results were superior to those of traditional multivariate statistical approaches.

Fan and Matsuyama [23] presented a rough set theory-based approach to analyze academic performance in a Web-based learning support system. The study included the analysis of 28 student profiles considering general attributes such as age, gender, financial aid, marital status, dependents, etc. No results regarding the predictive capability of the model were presented. The authors emphasized the importance of personalized learning particularly in web-based environments.

Most recently Pai et.al. [24], presented a model based on RST to analyze academic achievement in terms of overall course grades in junior high school students. To predict a student's performance, the authors considered external relationships, such as teacher–student interaction, parental expectations, learning styles, and socio-demographic attributes such as family income per month. Linear discriminant analysis was used to identify the nine attributes significant to academic performance. The authors compared the RST model based on linear discriminant analysis to five different data mining algorithms and concluded that the RST model performed better in terms of classification accuracy. While this effort is not necessarily in the same topic as addressed in this paper, to our knowledge, it is the only significant example of using RST-based knowledge discovery methodologies in educational research.

The work presented in this paper is unique in the sense that it is the first example of applying an RST-based knowledge discovery process for predicting student success in a course. To ensure that the prediction model is generally applicable, the data used in the prediction model are universal and not course specific. Furthermore, the model attributes are limited to data that are available before or at the time of course registration which allows the outcomes of the prediction model to be effectively used to benefit the students during the course.

3. Description of dataset

The dataset employed in this study consists of information collected from two distinct groups of students. The first group comprises 60 students enrolled in the *Introduction to Linear Systems* course during the spring term of the 2007–2008 academic year at the University of South Florida. The second group consists of 70 students enrolled in the same course during the spring term of the 2009–2010 academic year at the same university.

The datasets collected from each group of students have unique roles in the knowledge discovery process. Specifically, we use the data from the first group of students to develop the prediction model to classify students as passing or failing (training dataset) and the data from the second group to validate the accuracy of the developed model (testing dataset). By utilizing different datasets for development and validation, we overcome problems related to overfitting and, hence, enhance the robustness of the prediction model across different student populations within the same course.

We define *student profile* as the set of attributes that capture information regarding the demographics, workload, and student's previous performance. These are few candidate attributes which we believe to have a significant influence on the expected performance of the students in a particular course. A complete list of the attributes considered in this study is presented in Table 1. Student profiles are populated through a set of surveys administered at the beginning of both courses. Note that the generic aspect of the attributes considered will allow utilization of this basic student profile across various disciplines.

Analysis of the captured information was conducted based on RST. In the RST framework, data are represented by a two-dimensional table. Each row represents a student and each column represents an attribute in the student profile. These attributes are called condition attributes. To facilitate the student classification process, we define a decision attribute named "*performance*" to indicate whether a student was successful (he/she received a passing score of A, B or C) or unsuccessful in the

Table 1. Attributes. There are 8 condition attributes in each student profile. The table defines the code name, the description, and the value range for each attribute

Attribute	Description	Attribute range
Age	The age of the student	<21: Less than 21 years old 22–26: Between 22 and 26 years old >26: greater than 26 years old
Child	The student has children	Yes No
Crhr	Number of credit hours the student is taking during the semester	1–5 6–11 >12
Wrhr	Number of hours/week a student spend working outside the school	0–10 11–20 21–30 >30
Trnsf	The student has been transferred from another institution	Yes No
Crch	The student has made a career change	Yes No
Calc	Number of semesters elapsed since taking a prerequisite course	<4 >4
GPA	Overall GPA of a student (On a scale of 0.0 to 4.0. However, no students with GPA<2.0 were in the courses.)	2.0–2.5 2.5–3.0 3.0–3.5 3.5–4.0

Table 2. Decision Table. The decision table presents the relationship between condition attributes and the corresponding decision attribute. Here, the decision attribute, performance, is used to classify a student as failing or passing the course

	The condition attributes								**Decision attribute**
Student	**Age**	**Child**	**Crhr**	**Wrhr**	**Trnsf**	**Crch**	**Calc**	**GPA**	***Performance***
1	<21	NO	>12	0–10	NO	NO	<4	2.5–3.0	Failing
2	>26	YES	>12	>30	YES	YES	<4	3.5–4.0	Failing
3	22–26	NO	>12	0–10	YES	NO	<4	3.5–4.0	Passing
4	<21	NO	>12	11–20	NO	NO	<4	3.5–4.0	Passing

class (he/she receive a failing grade (D, F) or dropped the course). Table 2 is a *decision table* which shows an instance of the dataset used in this study including the decision attribute.

4. Knowledge discovery process

The objective of the knowledge discovery process is to identify meaningful relationships between condition and decision attributes. A comprehensive description of the RST-based knowledge discovery process is outlined in Figure 1. The main steps involved can be categorized in three phases: preprocessing, data mining, and post-processing. The rest of this section describes in detail each of these phases.

4.1 Data preprocessing

The first step in the knowledge discovery process is to identify and resolve missing values in the dataset. Several methodologies have been described in the literature [25–27] for imputing missing values such as bootstrapping, pattern analysis, deletion, mean substitution, and maximum likelihood estimation. In this study, all but one of the student profiles collected were complete. Therefore, we proceeded with deletion of the particular profile.

Next step in the knowledge discovery process is to split the entire dataset into two distinct datasets. One of the datasets will be used as the training set and the other as the testing set. In this study, we used the 2007–2008 student profiles as the training set and the 2009–2010 student profiles as the testing set. Table 3 shows the distribution of student performance in these two sets.

The RST-based knowledge discovery process continues with the discretization step which involves the representation of data using intervals and ranges in lieu of exact observations to define a coarser and more qualitative rather than quantitative representation of the data. The data discretization problem has been extensively studied and various heuristic search algorithms have been proposed [28–31]. In this work, all attributes in the student profiles are categorical as shown in Table 1; therefore the discretization step is not required.

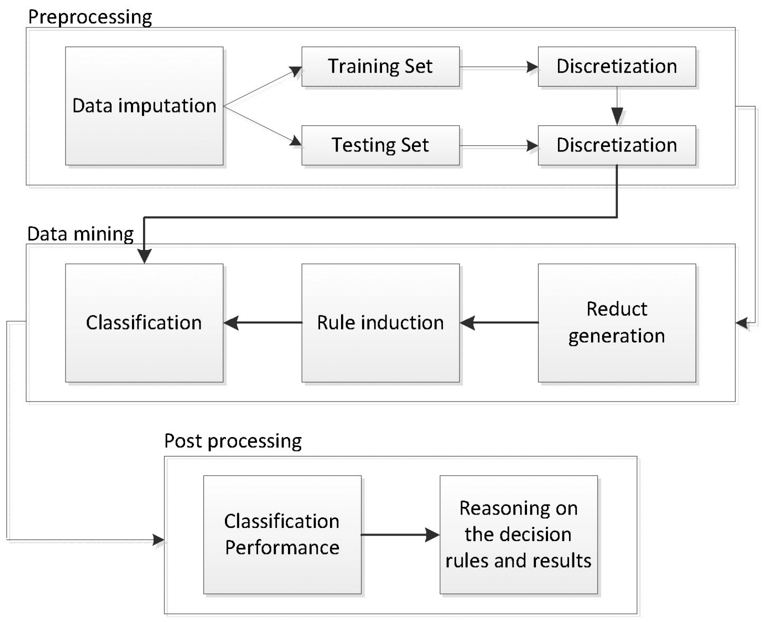

Fig. 1. Knowledge discovery methodology. There are three phases in the knowledge discovery process: data preprocessing, data mining, and data post processing.

Table 3. Performance distribution in training and testing sets

Dataset	**Failing**	**Passing**
Training set	58.33%	41.67%
Testing set	37.68%	62.32%

4.2 Reduct generation

The reduct generation step is utilized in an effort to reduce the dimensionality of the dataset by removing redundant information and consequently decreasing the complexity of the mining process. Formally, a reduct is the minimal set of attributes that enable the same classification as the complete set of attributes without loss of information. There are many algorithms for computing reducts. As will be shown later in this paper, the effect of the reduct generation algorithm to the classification performance is critical. Therefore, the optimal algorithm is identified as the one producing the best classification results. However, since the computational complexity of the reduct generation problem is NP-hard [28, 32], various suboptimal techniques have been proposed. The technique most appropriate to the problem is the one that generates better classification accuracy in the testing dataset. In this work, two techniques are used for reduct generation: genetic algorithms and dynamic reducts. The rest of this section describes these techniques.

4.2.1 Computing reducts using genetic algorithms

The computational cost for reduct computation is exponential with respect to the size of the decision table. Genetic algorithms, operating based on the principle of survival of the fittest, can be used to reduce the computational complexity [32–34]. Given a function $f : S_{+}$, the goal of a genetic algorithm is to find an $x_0 \in S$ for which $f(x_0) = \max(f(x) : x \in S)$. Elements of S are called *individuals* and the function f is the *fitness function*. The values of function $f(x)$ correspond to the ability of the individual x to survive the evolution process. The evolution process begins by creating a random initial fixed size population of individuals. In an iterative manner, the algorithm generates a new population of individuals. First, the fitness of each individual in the current population is calculated and those individuals with high fitness are selected as parents which interact based on a genetic operator (e.g. mutation and crossover) to produce the new population, child. The process is repeated until some stopping condition is achieved.

The genetic algorithm for the reduct generation uses as *individuals* the attributes in the student profile, and as fitness function the output of a heuristic algorithm that evaluates the quality of each reduct generated. The details of the genetic algorithm used for the reduct generation are presented in [32]. Using genetic algorithms, one reduct {Age, Crhr, Wrhr, Trnsf, GPA} is generated which includes 5 out of the 8 attributes.

4.2.2 Computing dynamic reducts

The main advantage of utilizing genetic algorithms for reduct generation is the reduction in computational complexity. However, the results obtained are highly dependent on the specific training dataset and therefore could change each time a different training set is selected. A strategy that generates reducts invariant to the training set is expected to generate more stable reducts. To this end, Bazan et. al. [28, 32, 35], proposed a reduct generation technique called *Dynamic Reducts*. This technique aims at obtaining the most stable sets of reducts for a given dataset by sampling within this dataset. For example, in an iterative manner different samples of the testing set are selected for which reducts are computed using a genetic algorithm. The reducts appearing more frequently in these samples are selected as the most stable.

Based on the principle of the dynamic reducts technique, we have randomly selected 100 subdivisions of the training set to use for reduct generation. The actual number of student profiles included in each subdivision of the training set varies as follows:

10 subdivisions with number of student profiles equal to 50% of the training data set
10 subdivisions with number of student profiles equal to 60% of the training data set
10 subdivisions with number of student profiles equal to 70% of the training data set
10 subdivisions with number of student profiles equal to 80% of the training data set
10 subdivisions with number of student profiles equal to 90% of the training data set

The reducts for each subdivision as well as the reduct from the complete training set are computed. The most stable reducts obtained are as follows:

{Age, Crhr, Wrhr, Trnsf, GPA}
{Age, Wrhr, Trnsf, Calc, GPA}
{Age, Wrhr, Trnsf, GPA}
{Age, Crhr, Wrhr, Calc, GPA}
{Crhr, Wrhr, Trnsf, Calc, GPA}
{Age, Crhr, Wrhr, GPA}
{Wrhr, Trnsf, Crch, Calc, GPA}
{Age, Crhr, Trnsf, Calc, GPA}
{Age, Crhr, Trnsf, Crch, GPA}
{Age, Child, Crhr, Trnsf, GPA}
{Wrhr, Trnsf, Calc, GPA}
{Crhr, Wrhr, Trnsf, GPA}
{Wrhr, Trnsf, Crch, GPA}
{Child, Wrhr, Trnsf, GPA}
{Age, Crhr, Trnsf, Crch, Calc, GPA}
{Wrhr, Trnsf, GPA}
{Age, Wrhr, GPA}
{Age, Child, Wrhr, Trnsf, Calc}

When dealing with multiple sets of reducts, the most significant attributes of the dataset can be identified. These attributes are called *core attributes* and appear in every reduct. Omitting core attributes from the classification process considerably affects the classification accuracy. In the aforementioned list of reducts, there is no attribute common among all the reducts. Therefore, the set of core attributes is empty. However, the attribute GPA appears in 17 out of the 18 reducts indicating that GPA can be considered as a significant attribute in classifying student performance. Similarly, the attributes *Trnsf* and *Wrhr* appear in 15 and 14 reducts, respectively and are considered critical to the classification model.

4.3 Rule induction

The ultimate goal of the RST-based knowledge discovery methodology is to generate decision rules which will be used in classifying each student as failing or passing. A decision rule has the form *if A then B* ($A \rightarrow B$), where A is called the condition and B the decision of the rule. Decision rules can be thought of as a formal language for drawing conclusions from data.

A decision rule is generated using the attributes in a student profile that are included in a reduct. For example, consider the decision table shown in Table 2 and the reduct {Age, Crhr, Wrhr, Trnsf, GPA} obtained using genetic algorithms. Since the reduct includes only five attributes, the decision table can be represented by Table 4. From the Reduced Decision Table in Table 4 we can define four decision rules as follows:

If the student is younger than 21 years old, takes more than 12 credit hours in a semester, works for less than 10 hours, is not a transfer student and has GPA between 2.5 and 3.0, he/she will fail the class.
If the student is older than 26 years old, takes more than 12 credit hours in a semester, works for more than 30 hours, is a transfer student and has GPA between 3.5 and 4.0, he/she will fail the class.
If the student is between 22 and 26 years old, takes more than 12 credit hours in a semester, works for less than 10 hours, is a transfer student and has GPA between 3.5 and 4.0, he/she will pass the class.
If the student is younger than 21 years old, takes more than 12 credit hours in a semester, works for 11 to 20 hours, is not a transfer students and has GPA between 3.5 and 4.0, he/she will fail the class.

Considering the attributes in the reduct {Age, Crhr, Wrhr, Trnsf, GPA} and the complete training set, we can create 43 decisions rules. A portion of these rules with the highest LHS Support are listed in Table 5. The *LHS Support* indicates the number of students satisfying the condition of the rule while the *RHS Support* indicates the number of students satisfying the decision of the rule.

Table 4. Decision table and reduced decision table. The reduced decision table is used to generate the decision rules for the classification model. Here, the reduced decision table has three attributes fewer than the original decision table

Original Decision Table

	Condition attributes								**Decision attribute**
Student	**Age**	**Child**	**Crhr**	**Wrhr**	**Trnsf**	**Crch**	**Calc**	**GPA**	**Performance**
1	<21	NO	>12	0–10	NO	NO	<4	2.5–3.0	Failing
2	>26	YES	>12	>30	YES	YES	<4	3.5–4.0	Failing
3	22–26	NO	>12	0–10	YES	NO	<4	3.5–4.0	Passing
4	<21	NO	>12	11–20	NO	NO	<4	3.5–4.0	Passing

Reduced Decision Table based on reduct {Age, Crhr, Wrhr, Trnsf, GPA}

	Condition attributes								**Decision attribute**
Student	**Age**	**Child**	**Crhr**	**Wrhr**	**Trnsf**	**Crch**	**Calc**	**GPA**	**Performance**
1	<21		>12	0–10	NO			2.5–3.0	Failing
2	>26		>12	>30	YES			3.5–4.0	Failing
3	22–26		>12	0–10	YES			3.5–4.0	Passing
4	<21		>12	11–20	NO			3.5–4.0	Passing

4.4 Classification process

Based on the set of rules generated, we can classify students as passing or failing. However, as seen in Table 5, not all rules are conclusive. Consider rules 1 and 3 in Table 5. Students with profiles identical to the conditions of the rules are not decisively classified as passing or failing. In addition, there are situations of contradictory rules, e.g. one or more rules classify a student as passing and some other rules classify the same student as failing. To overcome these problems, a *standard voting* algorithm [28] is used which allows all rules to participate in the decision process and classify a student based on majority voting.

Let RUL denote the set of all decision rules obtained from the training set. When a student with student profile x from the testing set is presented for classification, the standard voting algorithm operates as follows:

1. Assume that a student with profile x = {*age <21, Crhr >12, Wrhr = 0-10, Trnsf = NO, GPA = 3.5-4.0*} is to be classified. Let $RUL(x) \subseteq RUL$ denote the set of firing rules (those with the same conditions as student profile x).
   - If $RUL(x)$ is empty, then no classification can be made and x is declared undefined.
   - If $RUL(x)$ is not empty, an election process is performed among the rules in $RUL(x)$ as follows: Compute the number of votes each rule contributes to student profile x. Each rule $r \in RUL(x)$, casts a number of votes in favor of the decision class the rule indicates. Typically the number of votes is related to the RHS support of the rule. For example, consider the 1st rule presented in Table 7 with $RHS\ Support = 1; 6$. Then $votes(1^{st} rule, Failing) = 1$ and $votes(1^{st} rule, Passing) = 6$.
2. Compute the *normalization factor* associated with the student profile x and the number of rules fired: A normalization factor $norm(x)$ is computed for each student profile as the sum of all votes from all rules fired to serve as a scaling factor. In our example, since only the first rule fired for x, $norm(x) = 7$.
3. Calculate the *certainty coefficient* associated with each decision class as follows:
   - $Certainty(x, Failing) = \sum_i \frac{votes(r_{x,i}, Failing)}{norm(x)}$, with $r_{x,i}$ denoting all rules fired for student x.

Table 5. A subset of decision rules based on genetic algorithm. The table presents a subset of rules generated using the reduct {Age, Crhr, Wrhr, Trnsf, GPA}. LHS support and RHS support correspond to the number of students satisfying the condition of the rule and the number of students satisfying the decision of the rule respectively. For rules with dual decision (e.g. rule 1) there are two values for RHS Support corresponding to each decision

Rule	**Description**	**LHS Support**	**RHS Support**
1	Age(<21) AND Crhr(>12) AND Wrhr(0–10) AND Trnsf(NO) AND GPA(3.5–4.0) Then Performance(Fail) OR Performance(Success)	7	1; 6
2	Age(<21) AND Crhr(>12) AND Wrhr(0–10) AND Trnsf(NO) AND GPA(2.5–3.0) Then Performance(Pass)	4	4
3	Age(<21) AND Crhr(>12) AND Wrhr(11–20) AND Trnsf(NO) AND GPA(3.0–3.5) Then Performance(Fail) OR Performance(Pass)	3	2; 1

- $Certainty(x, Passing) = \sum_i \frac{votes(r_{x,i}, Passing)}{norm(x)}$.
  For our example, $Certainty(x, Failing) = \frac{1}{7}$ and $Certainty(x, Passing) = \frac{6}{7}$.

4. Finally, classify the student with profile x in the decision class for which the certainty factor is greater than a *threshold value* (τ) which is typically fixed at 0.5. In this example, the student with profile x is classified as *Passing*.

5. Results

This section compares the performance of the classification processes based on the decision rules generated using the reduct generation techniques described in sections 4.2.1–4.2.2. At this stage of the knowledge discovery methodology, the objects (student profiles) in training dataset are classified as passing, failing or undefined based on the induced rules and the classification process described. The results are presented in a confusion matrix form. The confusion matrix for each model includes the numbers of True Positive (*TP*), True Negative (*TN*), False Positive (*FP*) and False Negative (*FN*) results. Our perspective on positive and negative results relates to the necessitation for action for failing students. Specifically, we define:

TP: the number of students classified as failing the course, when in fact failed the course (shown in the top left cell of the confusion matrix).

FP: the number of students classified as failing the course, when in fact passed the course (shown in the bottom left cell of the confusion matrix).

TN: the number of students classified as passing the course, when in fact passed the course (shown in the bottom right cell of the confusion matrix).

FN: the number of students classified as passing the course, when in fact failed the course (shown in the top right cell of the confusion matrix).

Using these values we can compute the measures of specificity and sensitivity as:

Sensitivity: The fraction of failing students correctly classified by the classification algorithm.

$$Sensitivity = \frac{TP}{TP + FN} \tag{1}$$

Specificity: The fraction of passing students correctly classified by the classification algorithm.

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

The accuracy of each classification model is reported in terms of Area under the Receiver Operating Characteristic (ROC) curve (AUC). The ROC curve graphs the sensitivity of the classification algorithm in terms of (1-specificity). The best possible classification is achieved when AUC is equal to 1, while no classification ability exists when AUC is equal to 0.5.

5.1 Performance of the classification algorithm using reducts generated by genetic algorithms

Table 6 presents the confusion matrix for the classification model based on reducts generated using genetic algorithms. The classifier consists of 43 rules. With sensitivity equal to 80%, the classifier demonstrates an ability to correctly identify the failing students, however, the specificity score is much lower (20%), which implies that the classifier fails to correctly identify passing students. The term *undefined* in Table 6 refers to 59 students (almost 85.5% of students in the testing sample) for whom the classification algorithm was unable to classify either as passing or failing. The coverage of the classifier (defined by the ratio of objects classified to the total number of objects in the testing set) is 14.5% since we are able to classify 10 students from the 70 in the training set. Overall, the AUC score is equal to 0.5 indicating classification inability.

5.2 Performance of the classification algorithm using dynamic reducts

Table 7 shows the confusion matrix for the classification model based on dynamic reducts. There are 593 decision rules. The classifier's ability to correctly identify failing and passing students is 0.68 and 0.675, respectively. The overall classification performance as indicated by the AUC is equal to 0.8, considerably better compared to the genetic algorithm classifier. In addition, the number of undefined cases has been decreased to four student profiles and the coverage of the classifier is 96%. Using dynamic reducts instead of genetic algo-

Table 6. Confusion matrix. The classifier presents AUC equal to 0.5 indicating classification inability

	Predicted		
	Failing	**Passing**	**Undefined**
Actual			
Failing	4	1	21
Passing	4	1	38
Sensitivity: 0.8, Specificity: 0.2, AUC: 0.5			

Table 7. Confusion matrix. The classifier presents AUC equal to 0.8 indicates good classification ability

	Predicted		
	Failing	**Passing**	**Undefined**
Actual			
Failing	17	8	1
Passing	13	27	3
Sensitivity: 0.68, Specificity: 0.675, AUC: 0.8			

Table 8. Comparison of classifiers. A classifier has been created based on each reduct generation technique described in sections 4.2.1–4.2.2

	Strategy	
Performance measures	**Genetic Algorithms**	**Dynamic reducts**
Sensitivity	0.8	0.68
Specificity	0.2	0.675
AUC	0.5	0.8
Coverage	14.5%	94%
# of reducts	1	18
# of decision rules	43	593

rithms for reduct generation improved the overall classification performance.

Table 8 summarizes our findings regarding the performance of each classifier in predicting student performance.

6. Discussion

The *threshold value* (τ) in the classification process described in Section 4.4 has a significant impact on the accuracy as well as the usability of the classification process, especially in this application of student performance prediction. To better understand the role of this threshold value, consider the definitions of sensitivity, the fraction of failing students correctly classified, and specificity, the fraction of passing students correctly classified by the classification algorithm. In our particular application of predicting student performance in a course, to engage the potentially unsuccessful students early on and to improve their performance, the "cost" of misclassifying a failing student (as passing) is much higher than that of misclassifying a passing student (as failing). After all, if a potentially weak/unsuccessful student is misclassified as passing, the opportunity to engage this student early is lost. On the other hand, if a passing student is misclassified as failing and is enrolled in activities to improve his/her performance, he/she may actually end up with an improved grade. Therefore, especially in this particular application, it is significantly more important to ensure that the sensitivity value is closer to 1 than the specificity value.

The threshold value is the parameter that establishes the relation between sensitivity and specificity in the classification process. A higher threshold value would require a higher certainty coefficient value (making it more difficult) for a student to be classified as failing, decreasing the sensitivity and increasing specificity. In the same manner, a lower threshold value would increase sensitivity and reduce specificity, which is the more desirable condition in this application.

The ROC curve describes the predictive behavior of a classifier for varying values of the threshold ($0 \leq \tau \leq 1$), in terms of sensitivity, specificity and classifier accuracy. Figure 2 shows the ROC curve generated from the classification model based on dynamic reducts. The area under the ROC curve characterizes the overall accuracy of the classifier. Each point on the curve corresponds to a different pair of sensitivity and specificity values based on varying the value of the threshold (τ).

Table 9 shows some selected points on the ROC curve and the associated threshold value used during the classification process. For example, the default value of $\tau = 0.5$ leads to the sensitivity and specificity values reported in Table 7. The conditional maximum values of both sensitivity and specificity are obtained when the threshold values

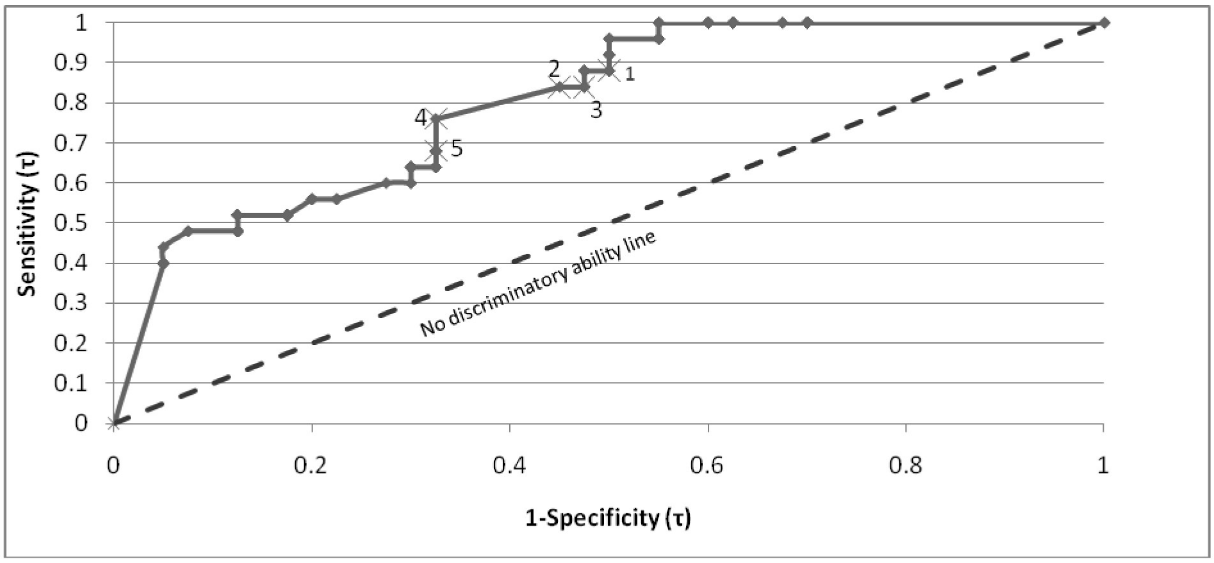

Fig. 2. ROC Curve for the classification model based on dynamic reducts. The specificity and sensitivity are controlled by the threshold value.

Table 9. Sensitivity and specificity values for varying threshold values

Point #	Sensitivity	Specificity	Threshold(t)
1	0.88	0.500	0.330
2	0.88	0.525	0.380
3	0.84	0.525	0.400
4	0.76	0.675	0.416
5	0.68	0.675	0.500

Table 10. Confusion Matrix using t=0.416. The smaller threshold value results in higher sensitivity and lower specificity values. Compared to Table 7, 2 more students have been correctly identified as failing

	Predicted		
	Failing	Passing	Undefined
Actual			
Failing	19	6	1
Passing	13	27	3
Sensitivity: 0.76, Specificity: 0.675, AUC: 0.8			

Table 11. Confusion Matrix using t=0.38. Compared to Table 10 more students have been correctly identified as failing while 6 students have been incorrectly identified as not passing the course

	Predicted		
	Failing	Passing	Undefined
Actual			
Failing	22	3	1
Passing	19	21	3
Sensitivity: 0.88, Specificity: 0.525, AUC: 0.8			

is 0.416. The confusion matrix for this threshold value is shown in Table 10.

Considering the nature of this particular application where the intent may lean towards maximizing sensitivity, point 2 in Fig. 2 results in possibly the most effective classification where only three failing students were misclassified and 22 were correctly classified. On the other hand, nearly half of the passing students were classified as failing greatly increasing the total number of students classified as failing. The decision of which threshold value to use for classification is a subjective matter depending on the cost and capacity of the available programs and activities to improve student performance. For example, if the planned activity to help potentially unsuccessful students is a web-based activity such as endless quizzes [36] where questions and grading are done automatically by the computer, then the additional number of students may not be prohibitive.

Point 5 in Fig. 2 corresponds to threshold equal to 0.5 which results in sensitivity 0.68 and specificity 0.675 (Table 7).

As the value of threshold decreases, the sensitivity of the classification model increases in the expense of specificity. For the student performance application, an increased sensitivity is a desirable outcome.

7. Conclusions

The presented work is significant in the sense that, to our knowledge, it is the first example of applying an RST-based knowledge discovery process for predicting student success in a single course in academic settings. Most relevant research associated with the use of soft computing approaches focuses exclusively on the development and evaluation of the data mining techniques neglecting pre and post mining phases crucial to the effective use of the data mining results. The work presented addresses all stages of the knowledge discovery process and describes how the classification methodology can be tailored to varying levels of sensitivity and specificity, and provide effective decision support depending on the cost and capacity of the available programs and activities to improve student performance.

Another important distinctive feature of the work presented is that the training and testing sets are distinct sets of students. Many of the proposed methodologies in the field of educational performance prediction do not validate their findings in different student populations and may often suffer from over-fitting, which has been proven to cause poor prediction performance when applied to different datasets.

In the prediction model presented, the condition attributes are general and limited to data that can be collected by administering a brief in-class survey at the beginning of the course. We note that the accuracy of this baseline prediction model may be further improved by incorporating more cognitive factors such as attributes related to metacognitive skills and self-efficacy. A discipline-neutral prediction model may further be focused by incorporating attributes related to the discipline-specific skills. For example, analytical and math skills would be likely candidates for engineering courses. The degree of complexity of the predictive model and the effort required for data collection should be carefully evaluated in accordance with the objectives and scope of the predictive model.

The long-term goal of our research is the development of a decision support system that enables both students and educators to actively participate in the development of a personalized education plan taking into consideration the needs of the individual student as well as the availability of resources to provide the personalization.

References

1. Z. Pawlak, *Rough sets: Theoretical aspects of reasoning about data*, Kluwer Academic Publishers, Norwell, MA, 1991.
2. Z. Pawlak, Rough set approach to knowledge-based decision

support, *European Journal of Operational Research,* **99**(1), 1997, pp. 48–57.

3. G. J. Klir and B. Yuan, *Fuzzy sets and fuzzy logic: Theory and Applications*, Prentice Hall PTR, New Jersey, 1995.
4. S. Glenn, *A mathematical theory of evidence*, Princeton University Press, New Jersey, 1976.
5. L. M. Tho, Self-efficacy and Student Performance in an Accounting Course, *Journal of Financial Reporting and Accounting,* **4**, 2006, pp. 129–146.
6. K. Eunhee, F. B. Newton, R. G. Downey and S. L. Benton, Personal Factors Impacting College Student Success: Constructing College Learning Effectiveness Inventory (Clei), *College Student Journal,* **44**(1), 2010, pp. 112–125.
7. I. D. Cherney and R. R. Cooney, Predicting Student Performance in a Statistics Course using The Mathematics and Statistics Perception Scale (MPSP), *Transactions of the Nebraska Academy of Sciences and Affiliated Societies,* **30**, 2005, pp. 1–8.
8. V. Garcia, J. Alvarado, and A. Jimenez, Predicting Academic Achievement: Linear Regression versus Logistic Regression, *Psicothema,* **12**(2), 2000, pp. 248–252.
9. A. Luuk and K. Luuk, Predicting Students' Academic Performance in Aviation College from their Admission Test Results, in *European Association for Aviation Psychology (EEAP)*, 2008.
10. C. M. Cornwell, D. B. Mustard and J. van Parys, How Does the New SAT Predict Academic Achievement in College?, Georgia Tech2008.
11. N. Carupatanapong, W. C. McCormick and K. L. Rascati, Predicting Academic Performance of Pharmacy Students: Demographic Comparisons, *American Journal of Pharmacy Education,* **58**(3), 1994, pp. 262–268.
12. S. D. Ridgell and J. W. Lounsbury, Predicting Academic Success: General Intelligence, "Big Five" Personality Traits, And Work Drive, *College Student Journal,* **38**(4), 2004, pp. 607–618.
13. M. A. Geiger and E. A. Cooper, Predicting Academic Performance: The Impact of Expectancy and Needs Theory, *The Journal of Experimental Education,* **63**(3), 1995, pp. 251–262.
14. M. Potgieter, M. Ackermann, and L. Fletcher, Inaccuracy of Self-Evaluation as Additional Variable for Prediction of Students at Risk of Failing First-Year Chemistry, *Chemistry Education Research and Practice,* **11**(17–24), 2010.
15. B. Friedman and R. Mandel, The Prediction of College Student Academic Performance and Retention: Application of Expectancy and Goal Setting Theories, *Journal of College Student Retention: Research, Theory and Practice,* **11**(2), 2009, pp. 227–246.
16. H. Guruler, A. Istanbullu and M. Karahasan, A new student performance analysing system using knowledge discovery in higher educational databases, *Computers & Education,* **55**(1), 2010, pp. 247–254.
17. G. Mendez, T. Buskirk, S. Lohr, and S. Haag, Factors associated with persistence in science and engineering majors: An exploratory study using classification trees and random forests, *Journal of Engineering Education,* **97**(1), 2008, p. 57.
18. P. Ramasubramanian, V. Suresnkumar, P. Iyakutti and P. Thangavelu, Mining Analysis of SIS Database Using Rough Set Theory, in *IEEE International Conference on Computational Intelligence and Multimedia Applications*, 2008, pp. 81–87.
19. A. Salazar, J. Gosalbez, I. Bosch, R. Miralles and L. Vergara, A case study of knowledge discovery on academic achievement, student desertion and student retention, in *IEEE International Conference on Information Technology: Research and Education*, 2005, pp. 150–154.
20. W. Hämäläinen and M. Vinni, Comparison of Machine Learning Methods for Intelligent Tutoring Systems, in *Intelligent Tutoring Systems*, Springer Berlin/Heidelberg, 2006.
21. J. P. Vandamme, N. Meskens and J. F. Superby, "Predicting Academic Performance by Data Mining Methods," vol. 15, ed: Routledge, 2007, pp. 405—419.
22. N. Fang and J. Lu, A decision tree approach to predictive modeling of student performance in engineering dynamics, *International J of Engineering Education,* **36**, 2010, pp. 87–95.
23. L. Fan and T. Matsuyama, Rough Set Approach to Analysis of Students Academic Performance in Web-based Learning Support System, in *Proceedings of the 15th International Workshops on Conceptual Structures*, Sheffield, UK, 2007.
24. P.-F. Pai, Y.-J. Lyu and Y.-M. Wang, Analyzing academic achievement of junior high school students by an improved rough set model, *Computers & Education,* **54**(4), 2010, pp. 889–900.
25. P. D. Allison, *Missing Data*, Sage Publications, Thousand Oaks, CA, 2001.
26. R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, John Wiley & Sons, New York, NY, 1987.
27. J. L. Schafer, *Analysis of incomplete multivariate data*, London, 1997.
28. J. Bazan, H. Nguyen, S. Nguyen, P. Synak, J. Wroblewski, L. Polkowski, S. Tsumoto and T. Lin, Rough Set Algorithms in Classification Problem, in *Rough set methods and applications: new developments in knowledge discovery in information systems*, Physica-Verlag, 2000.
29. H. S. Nguyen, S. H. Nguyen and A. Skowron, Searching for features defined by hyperplanes, in *Foundations of Intelligent Systems*, Springer Berlin / Heidelberg, 1996.
30. S. H. Nguyen and H. S. Nguyen, Some efficient algorithms for rough set methods, in *Information Processing and Management of Uncertainty on Knowledge Based Systems*, Granada, Spain, 1996, pp. 1451–1456.
31. S. H. Nguyen, A. Skowron, P. Synak and J. Wroblewski, Knowledge discovery in databases: Rough set approach, in *Second Joint Annual Conference on Information Sciences*, Wrightsville Beach, North Carolina, 1997, pp. 34–37.
32. J. Bazan, A. Skowron, and P. Synak, Dynamic reducts as a tool for extracting laws from decisions tables, in *Methodologies for Intelligent Systems*, Springer Berlin / Heidelberg, 1994.
33. D. E. Goldberg, *GA in search, optimisation, and machine learning*, Addison-Wesley 1989.
34. J. H. Holland, *Adaptation in natural and artificial systems*, The MIT Press, Cambridge, 1992.
35. J. Bazan, Dynamic Reducts and Statistical Inference, in *Sixth International Conference on Information Procesing and Management of Uncertainty in Knowledge-Based Systems*, Granada, Spain, 1996, pp. 1147–1152.
36. G. Lee-Thomas, A. Kaw, and A. Yalcin, Using Online Endless Quizzes as Graded Homework, in *2011 Annual ASEE Conference and Exposition*, Vancouver, Canada, 2011.

Eleazar Gil-Herrera is a Doctoral Student in the Department of Industrial and Management System Engineering at the University of South Florida. In 2005, he received his B.S in Computer Engineering from the National University of Cusco, Peru. In 2009, Eleazar Gil-Herrera received his MS degree in Industrial Engineering from the University of Puerto Rico. His research interests are in the areas of knowledge discovery, data mining, decision support systems and prognostic models.

Athanasios Tsalatsanis is an Assistant Professor with the Center for Evidence-based Medicine, University of South Florida. In 2008, he received his Ph.D. in Industrial Engineering from the University of South Florida, in which he focused on the development of algorithms and control methodologies for autonomous robotic systems. He joined the Center for Evidence Based Medicine in 2009. Since then, his research has been concentrated on healthcare engineering, specifically in the areas of decision support systems, health information systems, prognostic models, and social network analysis.

Ali Yalcin is an Associate Professor of Industrial and Management Systems Engineering at the University of South Florida. He holds a Ph.D. in Industrial and Systems Engineering from Rutgers University. His research interests are in the areas of systems modeling and analysis, engineering education, information systems, and knowledge discovery. He has taught a variety of courses in Industrial Engineering in the areas of systems simulation, information systems, facilities design and linear systems. He also co-authored the 2006 Joint Publishers Book-of-the-Year textbook, Design of Industrial Information Systems, Elsevier.

Autar Kaw is a Professor of Mechanical Engineering and Jerome Krivanek Distinguished Teacher at the University of South Florida, USA. He holds a Ph.D. in Engineering Mechanics from Clemson University. His main scholarly interests are in engineering education, bascule bridge design, and mechanics of composite materials. With major funding from the US National Science Foundation, he is the lead developer of award-winning online resources for an undergraduate course in Numerical Methods (http://numericalmethods.eng.usf.edu). He is the recipient of the 2004 US Florida Professor of the Year Award from the Council for Advancement and Support of Education (CASE) and the Carnegie Foundation for the Advancement of Teaching (CFAT). He has authored several textbooks on subjects such as composite materials, numerical methods, matrix algebra, and computer programming.