Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 20–29

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

S PRAR: A novel relational association rule mining classification model applied for academic performance prediction

Gabriela Czibula[a], Andrei Mihai[a], Liana Maria Crivei[a,*]

[a]*Department of Computer Science, Babeş-Bolyai University*
1, M. Kogalniceanu Street, 400084, Cluj-Napoca, Romania

Abstract

This paper analyses the problem of predicting students' academic performance, a subject that is increasingly investigated within the Educational Data Mining literature. For a better understanding of the educational related phenomena, there is a continuous interest in applying *supervised* and *unsupervised* learning methods for obtaining additional insights into the students' learning process. The problem of predicting if a student will pass or fail at a certain academic discipline based on the students' grades received during the semester is a difficult one, highly dependent on various conditions such as the course, the number of examinations during the semester, the instructors and their exigences. We propose a new classification model, *S PRAR* (*Students Performance prediction using Relational Association Rules*) for predicting the final result of a student at a certain academic discipline using *relational association rules* (RARs). RARs extend the classical association rules for expressing various relationships between data attributes. Experiments are performed on three real academic data sets collected from Babeş-Bolyai University from Romania. The performance of the *S PRAR* classifier on the considered case studies is compared against existing related work, being superior to previously proposed students' performance predictors.

Keywords: Educational data mining; Students' performance prediction; Supervised learning; Relational association rules
2000 MSC: 68T05; 68P15

1. Introduction

Educational data mining (EDM) is an attractive research domain which brings the data mining perspective into educational environments. The major goal of EDM topics is to offer a better comprehension of the educational related phenomena by uncovering relevant hidden patterns from educational data sets. Within the EDM field, a problem intensively investigated is that of students' performance prediction.

* Corresponding author. Tel.: +4-264-405-327; fax: +4-264-591-906.
* Corresponding author. Tel.: +4-264-405-327; fax: +4-264-591-906.
E-mail address: liana.crivei@cs.ubbcluj.ro

10.1016/j.procs.2019.09.156

Detecting meaningful patterns from the educational processes is of great interest in academic environments, since it would be useful for understanding the students learning activities, as well as improving the learning outcomes. There is a considerable attention from the research community for EDM, as extracting knowledge from educational data is of major concern for the academic institutions and also useful for improving their didactic methodologies and principles [17].

Relational Association Rules (RARs) [21] were introduced in the data mining literature as an extension of classical association rules with the goal of capturing different kinds of relationships between the attributes characterizing a data set. Both ordinal and relational association rule mining have been successfully applied to address problems ranging from data mining tasks (e.g. detecting software design defects [10] or data cleaning [5]) to supervised classification (e.g. software defect prediction [16]).

This paper approaches the problem of predicting if a student will pass or fail at a certain academic discipline based on the students' grades received during the semester. The problem has a major practical relevance in educational environments, since it could provide relevant feedback to the students' which are likely to fail at a certain academic course. Having such an advice during the semester, the students will have the possibility to prevent a possible academic unsuccess. Despite its obvious pragmatic importance, the problem is a complex and difficult one, highly dependent on various conditions such as the course, the number of examinations during the semester, the instructors and their exigences. We propose a new classification model, *S PRAR* (*Students Performance prediction using Relational Association Rules*) for predicting the final result of a student at a certain academic discipline using *relational association rules* (RARs). We have to note *S PRAR* classifier's generality, that is not specific to the students' performance prediction task. *S PRAR* can be easily adapted to other classification problems.

The classification model introduced in the paper is a binary one (i.e. there are only two classes to predict: *pass* or *fail*), but the proposed model is a general one, it can be extended for a multi-class classification problem (i.e. to predict the final grade of the student). From a supervised learning perspective, the problem of predicting the successful completion of a course during an academic semester is a hard problem, particularly due to the imbalanced nature of the training data (i.e. the number of students which *passed* the exam is generally much higher than the number of students which *failed* the exam). Experiments conducted on three real data sets from Babeş-Bolyai University, Romania highlight the effectiveness of the classification using *S PRAR*. The literature regarding students' performance prediction reveals that the use of RARs in predicting the academic performance of students is a novel approach.

In summary, the aim of the research conducted in the paper is that of proposing a classification model *S PARAR* based on RAR mining for predicting the students' academic performance, as a proof of concept. With this aim the proof of concept considers only three medium sized data sets for highlighting that *S PRAR* is suitable for the approached problem. If this stands, the study of applying *S PRAR* for students' performance prediction can be further extended on a larger scale.

The rest of the paper is organized as follows. Section 2 presents a literature review on using supervised learning methods for predicting the academic performance of students. The main background concepts on RAR mining are presented in Section 3. The methodology we propose for building *S PRAR* classifier is introduced in Section 4, while the experimental results and comparison to related work are provided in Section 5. The conclusions of our paper and directions to further improve our work are outlined in Section 6.

2. Literature review

For evaluating students performance in educational settings, we outline in the following, several recent machine learning research studies.

A. K. Pal and S. Pal [18] attempt to identify the highest performing classifier for predicting students achievements on a Computer Application final examination. The authors use three known data mining algorithms in their study: ID3 (Iterative Dichotomiser 3), C4.5 and Bootstrap aggregating. The results are effective for establishing which students require special counseling for improving their final assessment outcome. Hajizadeh and Ahmadzadeh [13] attempt to identify the factors that affect the students achievement or failure which further determine students to not re-register for a particular course. The data mining techniques used in [13] are association rule discovery and classification.

Jishan et al. [14] aim to increase the accuracy of the final grade prediction by applying several classification models including *neural networks*, *decision trees*, and *Naive Bayes*. In order to achieve better performance for predicting the students final grade on a particular course, *Optimal Equal Width Binning* is performed for data preprocessing and

Synthetic Minority Over-Sampling (SMOTE) method is applied. Shahiri et al. [22] analyze the performance of the ML models formerly specified as well as *support vector machine* and *K-Nearest neighbour*. A comparative study is achieved in order to identify the attributes that are significant for students' performance prediction. Ahmed et al. [1] evaluate several classifiers such as J48 Decision Tree, Multilayer Perceptron, Naive Bayes and Sequential Minimal Optimization while attempting to identify the highest performing algorithm. They analyze the factors that impact students' academic achievement or failure with the goal of improving the quality of the educational system.

Tran et al. [24] attempt to predict students performance in academic environments by analyzing and comparing a regression-based approach and a recommendation system-based approach. In order to improve performance in the regression scenario several course related skills are added. With the goal of enhancing the overall prediction performance, the authors also introduce a hybrid method – a linear combination of the two previously specified methods . Random forests (RF) were applied by Beaulac and Rosenthal [2] in order to predict which students could obtain an undergraduate degree based on the courses attended and completed in the first 2 semesters of an academic year. The data set contains information regarding several courses taken by undergraduate students at a university from Canada.

Verma et. al present [23] a fuzzy inference technique which is used as a data mining tool to evaluate and predict students' performance at the end of a semester. For performance prediction in the final examination several attributes are considered: Previous Semester Marks (PSM), Previous Academic Record (PAR), Attendance (ATT) and End of Semester Marks (ESM), ESM being considered an output variable, while the other three are input variables. Students scores are classified based on their achievement level in order to reduce failure rate and improve academic decisions. The research method employs fuzzy association rule mining which is divided in several stages: defining the fuzzy sets, constructing the mining data set from the original data named data transformation in fuzzy domain, fuzzy normalization and finally the generation of the frequent itemsets using an Apriori like algorithm. Table 1 summarizes several research results obtained in the EDM literature for predicting the performance of students.

Approach	**Model**	**Measure**	**Performance**
Hajizadeh and Ahmadzadeh [13]	DT	F-score	0.774
Pal and Pal [18]	DT	F-score	0.8
Tran et al. [24]	SVM	RMSE	1.705
Ahmed et al. [1]	J48	Accuracy	0.848
Beaulac and Rosenthal [2]	RF	Accuracy	0.788

Table 1: Existing results on students' performance prediction.

3. Relational association rule mining

Association rule (AR) mining represents an important data analysis and mining technique useful in multiple *machine learning* tasks for uncovering meaningful rule based patterns in data sets. *Ordinal association rules* (OARs) [6] were proposed as a particular class of ARs which express ordinal relationships between the attributes characterizing a data set. *Relational association rules* (RARs) [5, 21] have been introduced as an extension of OARs and are able to express different type of non-ordinal relations between data attributes.

We define the concept of *Relational Association Rules* (RARs) in the following. Let us consider $D = \{s_1, s_2, \ldots, s_q\}$ a set of *records* or *instances* [8]. For every instance of the data set, we consider a sequence of m attributes $A = (a_1, \ldots, a_m)$ that describes the data set D. Each attribute a_i takes values from a non-empty and non-fuzzy domain Δ_i. *Null* (*empty*) values may also be included in the domain Δ_i. The value of attribute a_i for an instance s_j is expressed by $\Psi(s_j, a_i)$.

The set of all possible relations which are not necessarily ordinal and can be defined between two domains Δ_i and Δ_j is denoted by $\mathcal{T}$.

Definition 1. *A* relational association rule [21] *is an expression* $(a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_h}) \Rightarrow (a_{i_1}\tau_1 a_{i_2}\tau_2 a_{i_3} \ldots \tau_{h-1} a_{i_h})$*, where* $\{a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_h}\} \subseteq A$, $a_{i_k} \neq a_{i_j}$, $k, j = 1, \ldots, h$, $k \neq j$ *and* $\tau_k \in \mathcal{T}$ *is a relation over* $\Delta_{i_k} \times \Delta_{i_{k+1}}$*, where* Δ_{i_k} *is considered the domain of the attribute* a_{i_k}.

a) If $a_{i_1}, a_{i_2}, \ldots, a_{i_h}$ *are non-missing in* ω *instances from the data set then we call* $Supp = \frac{\omega}{q}$ *the* support *of the rule.*

b) *If we denote by $D' \subseteq D$ the set of instances where $a_{i_1}, a_{i_2}, a_{i_3}, \ldots, a_{i_h}$ are non-missing and all the relations $\Psi(s_j, a_{i_1})\tau_1\Psi(s_j, a_{i_2})$, $\Psi(s_j, a_{i_2})\tau_2\ \Psi(s_j, a_{i_3})$, $\ldots$, $\Psi(s_j, a_{i_{h-1}})\tau_{h-1}\Psi(s_j, a_{i_h})$ hold for each instance s from D' then we call $Conf = \frac{|D'|}{q}$ the* confidence *of the rule.*

Interesting RARs [21] were defined as representing those rules that have both their *confidence* and *support* greater than or equal to specified minimum thresholds. With the purpose of mining interesting RARs an Apriori-like algorithm called *DRAR* (Discovery of Relational Association Rules) [9] was proposed as an extension of the *DOAR* algorithm introduced [6] for uncovering OARs [8].

The *DRAR* algorithm [9] consists of length-level generation of RARs, starting with the rules of length 2. The set of 2-length rules is filtered for determining the interesting rules (i.e those rules which verify the minimum support and confidence requirements). At a given step, for determining the RARs of a certain length l, we start from the set of interesting RARs of length $l-1$ generated at the previous step. This set is used to generate by join new possible interesting RARs, called *candidate rules* which will be filtered to preserve only the rules that are interesting. After the set of l-length interesting RARs is generated, the iterative process continues with generating the rules of length $l+1$. The process stops when no new interesting RARs are discovered [8].

3.1. Example

For exemplifying the concept of RARs, we use a synthetically generated data set with five instances and three attributes denoted by a_1, a_2 and a_3. The sample data set and the discovered maximal interesting RARs are given in Figure 1. For mining the interesting RARs from the data set, *DRAR* algorithm was applied with a minimum support threshold at $Supp_{min} = 1$ and the minimum confidence threshold at $Conf_{min} = 0.4$. Since all the attributes in our experiment have integer values, two possible binary relations between integer valued attributes were used: $\geq$ and $<$.

No.	a_1	a_2	a_3
1.	7.30	5.00	6.50
2.	10.00	9.90	9.00
3.	7.95	8.30	7.00
4.	8.50	10.00	9.20
5.	10.00	9.50	9.00

Length	**Rule**	**Confidence**
2	$a_1 \geq a_3$	0.8
3	$a_1 < a_2 \geq a_3$	0.4
3	$a_1 \geq a_2 \geq a_3$	0.4

Fig. 1: Sample data set (left) and interesting maximal RARs mined for $Supp_{min} = 1$ and $Conf_{min} = 0.4$.

Each line from Table 1 describes a RAR of a certain length (depicted in the first column), which has the confidence illustrated in the third column. For example, the first line in Table 1 refers to the RAR $a_1 \geq a_3$ of length **2** (i.e. the rule contains two attributes) having a confidence of **0.8**. This rule has the following interpretation: the value of the attribute a_1 is greater than or equal to the value of the attribute a_3 in 80% of instances from the analyzed data, i.e. in 4 out of 5 instances.

4. Methodology

The classification problem we are approaching in this paper is a binary classification problem. There are two possible classes, denoted in the following by "+" and "-". By "+" we denote the class corresponding to students that *pass* a certain course and the instances belonging to the "+" class will be referred to as *positive* instances. By "-" we denote the class corresponding to students that *fail* a particular course and the instances that belong to the "-" class will be referred to as *negative* instances.

In a supervised learning scenario for predicting if students will *pass* or *fail* a certain course, a training data set D is provided. The following theoretical model will be considered. The training data set D consists of a set of instances (students), $D = \{s_1, s_2, \ldots, s_q\}$ and each instance s_i describes the performance of a student, during the academic semester, at a given course. Accordingly, each instance s_i is characterized by a set of *attributes* $A = \{a_1, a_2, \ldots, a_m\}$ representing the grades obtained by the student during the semester evaluations. Thus, each s_i is visualized as an m-dimensional vector $s_i = (s_{i1}, s_{i2}, \ldots, s_{im})$, s_{ij} representing the grade received by student s_i at the j-th semester evaluation.

A flowchart of *SPRAR* approach is given in Figure 2.

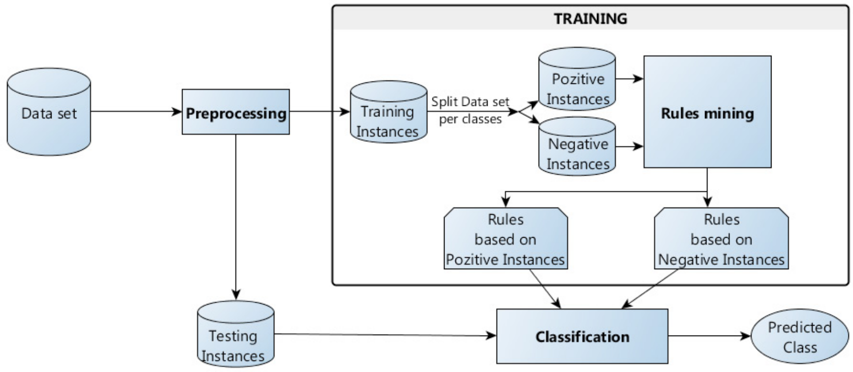


Fig. 2: Flowchart of *SPRAR* approach.

The underlying idea behind our approach is briefly described in the following. The training data set D is splitted into a subset of *positive* instances, denoted by D_+ and a subset of *negative* instances denoted by D_-. These sets will be used for training the *SPRAR* classifier. During training, the interesting RARs from the sets D_+ and D_- are uncovered using the *DRAR* algorithm [9]. We detect in the training data sets all the interesting RARs, with respect to the user-provided support and confidence thresholds: RAR_+ the set of interesting RARs discovered in D_+ and RAR_- the set of interesting RARs discovered in D_-. After the training was completed, when a new student (instance) s has to be classified (as "+" or "-"), two probabilities p_+ and p_- (expressing the probabilities that the query instance belongs to the *positive* and *negative* classes, respectively) will be computed.

For classifying using *SPRAR* an instance as belonging to the *positive* or *negative* class, the following steps will be performed.

First, a **data pre-processing** step is applied. The pre-processing step is highly dependent on the data and the approached classification problem. For the problem approached in this paper, with the goal of enlarging the set of potential RARs which may be uncovered in the training data, the set of attributes A is extended with four additional attributes: $a_{m+1} = 5$, $a_{m+2} = 6$, $a_{m+3} = 7$ and $a_{m+4} = 8$. Since *DRAR* algorithm considers only binary relations between attributes values, the newly added attributes $(a_{m+1}, \ldots, a_{m+4})$ will allow the discovery of unary relations between attributes values, as well. This way, more expressive and meaningful RARs will be identified. The second step consists of **training** the *SPRAR* classifier. The training step consists of mining from the training data set D two sets of interesting RARs: RAR_+ and RAR_-. RAR_+ consists of the interesting RARs discovered from the subset of D representing students that *pass* the course and RAR_- consists of the interesting RARs discovered from the subset of D representing students that *fail* to pass the course. After *SPRAR* model has been built during the training step, it will be tested in order to evaluate its predictive performance. The *classification* methodology of *SPRAR* will be introduced in Section 4.1.

4.1. Classification using *SPRAR*

At the classification stage, when a new query instance s has to be classified, we compute the probabilities $p_+(s)$ and $p_-(s)$ that the query instance belongs to the *positive* and *negative* classes, respectively. Our intuition behind defining these probabilities was that the similarity of an instance s to a particular class c is very likely to be influenced by the confidences of the rules from RAR_c that are verified in the instance s, but also by the rules from $RAR_{\neg c}$ that are not verified in the instance s. By $\neg c$ we denoted a class which is the opposite of c. Starting from this intuition, $p_c(s)$ will measure not only how "close" the instance s is to the class c, but also how "far" it is from the opposite class $\neg c$.

Let us denote by n_+ the sum of confidences of the RARs from RAR_+ and by n_- the sum of confidences of the RARs from RAR_-. For the query instance s and each rule r (from RAR_+ and RAR_+) we aim to compute the degree to which r is verified in the instance s. Thus, we express by $v^+(s)$ the sum of confidences of the rules from RAR_+ which are verified in s and by $v^-(s)$ the sum of confidences of the rules from RAR_- which are verified in s. Intuitively, if all rules from RAR_+ are verified in s, then $v^+(s) = n_+$. Similarly, by $nv^+(s)$ we denote the sum of confidences of the rules from RAR_+ which are not verified in s and by $nv^-(s)$ the sum of confidences of the rules from RAR_- which are not verified in s. Obviously, $nv_+(s) = n_+ - v_+(s)$ and $nv_-(s) = n_- - v_-(s)$. When computing the similarity of instance s to a particular class (e.g. "+") we consider not only the "similarity" of s to the class, but also the "dissimilarity" of s to the opposite class ("-" in our example). $p_+(s)$ and $p_-(s)$ are computed as described in Formulas (1). It can be observed that $p_+(s)$ and $p_-(s)$ are probabilities, i.e. $0 \leq p_+(s), p_-(s) \leq 1$ and $p_+(s) + p_-(s) = 1$.

$$p_+(s) = \frac{v^+(s) + nv^-(s)}{n_+ + n_-}, \qquad p_-(s) = \frac{v^-(s) + nv^+(s)}{n_+ + n_-} \tag{1}$$

When instance s has to be classified, if $p_+ \geq 0.5$ then instance s will be classified as a *positive* one, otherwise it will be classified as *negative*.

4.2. Evaluation measures

For evaluating the performance of the $SPRAR$ model on a testing data set, the confusion matrix for the two possible classes (*pass* and *fail*) will be computed on a testing set, considering that the *pass* class is the *positive* one and the *fail* class is the *negative* one. The confusion matrix consists of the following values: TP (*true positives*), FP (*false positives*), TN (*true negatives*) and FN (*false negatives*). Using the values from the confusion matrix, evaluation measures which are usually used for binary classification are determined [12]: *precision* (for the *positive* class, $Prec = \frac{TP}{TP+FP}$), *sensitivity* (or *recall*, $Sens = \frac{TP}{TP+FN}$), *negative predictive value* NPV (the precision for the *negative* class, $NPV = \frac{TN}{TN+FN}$), *specificity* (or *true negative rate*, $Spec = \frac{TN}{TN+FP}$), *F-score*$^+$ (*F-score* for the positive class, *F-score*$^+ = \frac{1}{\frac{1}{Prec}+\frac{1}{Sens}}$), *F-score*$^-$ (*F-score* for the negative class, *F-score*$^- = \frac{1}{\frac{1}{NPV}+\frac{1}{Spec}}$) and *Area Under the ROC Curve* (AUC). All these measures range from 0 to 1 and larger values indicate better classifiers.

The *F-score* is used in the literature for measuring performance in case of imbalanced data sets, still it is highly dependent on the labeling of the classes (i.e. *positive* or *negative* class). Thus, in the case of imbalanced data, the evaluation measure that is recommended as relevant for representing the performance of the classifiers is the *Area Under the ROC Curve (AUC)* measure. The ROC (Receiver Operating Characteristics) curve represents a two-dimensional plot of (1-specificity, sensitivity). For the classifiers that are directly returning the class, the point (1-$Spec$, $Sens$) is linked to the points at (0,0) and (1,1), and the area under the resulting ROC curve (AUC) will be computed as $\frac{(Spec+Sens)}{2}$.

We notice that for the problem of predicting if a student will *pass* or *fail*, the main focus is to increase the *true negative rate* (*specificity*) (i.e. to maximize the number of students that were correctly classified as *fail*), or, equivalently to decrease the *false positive rate*. For our problem, having *false positives* is a more serious problem than having *false negatives*, since predicting a false success (*pass* label) for a student would be a more serious error than predicting a false failure (class *fail*).

5. Results and discussion

This section presents the experimental results obtained by evaluating the performance of $SPRAR$ classifier, as well as a discussion on the results and a comparison to related work.

5.1. Data sets

Three real data sets collected from Babeş-Bolyai University, Romania are used in our experiments. The data sets contain the grades obtained by students at CS undergraduate courses offered in the *first*, *second* and *third* semesters at Babeş-Bolyai University. We depict in Table 2 the description of the data sets used in our case studies. The complete data sets are available at [7]. For each data set, the number of *positive* and *negative* instances is illustrated and the *difficulty* of the data set is computed. For measuring the difficulty of the classification task, the *difficulty* for a data set is provided. The *difficulty* was introduced by Boetticher [3] and is computed as the percentage of instances for which the nearest neighbor (ignoring the class label of the instances) has a different class label. When computing the difficulty of our data sets we considered only the percentage of entities from the minority class (*negative*) for which the nearest neighbor is from the majority class (*positive*).

Data set	# of negative instances	# of positive instances	Difficulty
D1	49 (12.76%)	335 (87.24%)	51.02%
D2	152 (17.61%)	711 (82.39%)	61.84%
D3	310 (26.86%)	844 (73.14 %)	32.58%

Table 2: Description of the used data sets.

Data set	# of training instances		# of testing instances	
	negative	positive	negative	positive
D1	34	260	15	75
D2	112	571	40	140
D3	240	528	70	316

Table 3: Description of training and testing sets.

a_1	a_2	a_3	a_4	Final grade	Class
9.625	10	10	9.25	9	"+"
10	10	10	10	10	
5.5	8.69	10	1	7	
9.625	8.63	9.25	10	6	
2.50	6	5.70	5	5	
2.50	9.50	10	10	4	"-"
2.50	5.71	10	5	4	
2.50	9.64	7	9	4	
5.00	7.96	9.10	8.50	4	
7.3	4.8	4.6	10	4	

Rule	Confidence
$a_2 \geq a_4$	0.8
$a_2 \geq 8$	0.8
$a_3 \geq a_4$	0.8
$a_3 \geq 8$	0.8
$a_1 \geq 5$	0.8
$a_2 \geq 5$	1
$a_2 \geq 6$	1
$a_2 \geq 7$	0.8
$a_3 \geq 5$	1
$a_3 \geq 6$	0.8
$a_3 \geq 7$	0.8
$a_4 \geq 5$	0.8

Rule	Confidence
$a_1 < 8$	1
$a_1 < 6$	0.8
$a_1 < 7$	0.8
$a_1 < a_4 \geq 8$	0.8
$a_1 < a_2 \geq 5$	0.8
$a_1 < a_3 \geq 5$	0.8
$a_1 < a_3 \geq 6$	0.8
$a_1 < a_3 \geq 7$	0.8
$a_1 < a_4 \geq 5$	1.0
$a_1 < a_4 \geq 6$	0.8
$a_1 < a_4 \geq 7$	0.8

Fig. 3: Sample data (left), the sets RAR_+ (middle) and RAR_- (right) mined for a minimum confidence threshold of 0.8.

Test instance	Actual class	v^+	v^-	nv^+	nv^-	n^+	n^-	p_+	p_-	Predicted class
$(a_1 = 9.53, a_2 = 6.07, a_3 = 10, a_4 = 9.05)$	"+"	7.8	2.4	2.4	6.8	10.2	9.2	0.75	0.25	"+"
$(a_1 = 5.0, a_2 = 1.0, a_3 = 10.0, a_4 = 7.0)$	"-"	5.8	7.6	4.4	1.6	10.2	9.2	0.38	0.62	"-"

Table 4: The classification of $SPRAR$ trained on the data set from Figure 3.

From Table 2 one can observe that all data sets are imbalanced, with all number of *fail* students smaller than the number of *pass* students. The most imbalanced data set is D1, followed by D2 and D3. Moreover, we observe the difficulty of the prediction task depicted in the last column from Table 2. From the classification viewpoint, the most difficult data set seem to be D2, then D1 and D3.

5.2. Example of classification using $SPRAR$

An example illustrating how $SPRAR$ classifier works is further considered. The left side table from Figure 3 contains a sample data set (extracted from data set D2) in which each instance is characterized by four attributes. For each instance, the final examination grade and the class ("+" or "-") are also provided. The middle and rightmost tables from Figure 3 illustrate the sets of RARs mined from the positive and negative instances from the sample data for a minimum confidence threshold of 0.8. Table 4 depicts the classification process of $SPRAR$ for two test instances, together with their actual classes and the classes predicted by SPRAR using the computed probabilities.

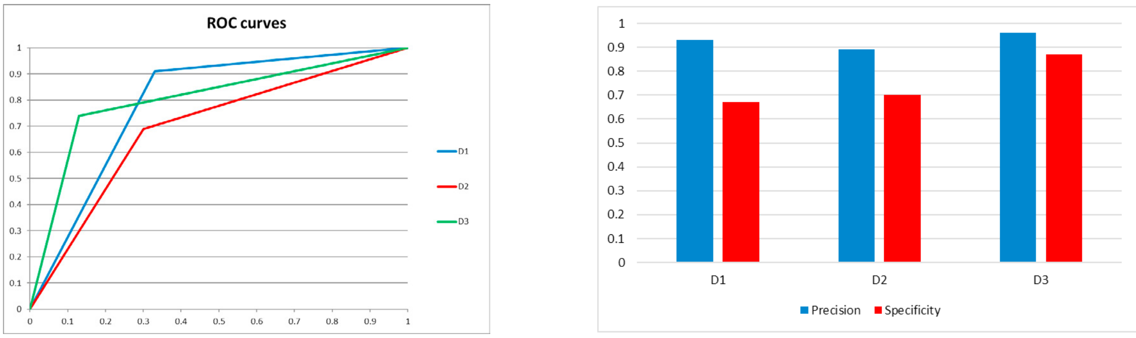
5.3. Results and discussion

Before building the $SPRAR$ classifier, we randomly selected a training subset from each data set. The set of instances which were not selected for training, will be subsequently used for testing. Table 3 describes the structure of the training/testing subsets from data sets D1, D2 and D3. For training $SPRAR$ classifier and generating the sets RAR_+ and RAR_- the following parameters were used: 1 for the minimum support threshold, 0.6 for the minimum confidence threshold and two possible binary relations between the attributes ($<$ and $\geq$). After the $SPRAR$ classifier has been built using the methodology introduced in Section 4, it is applied on the case studies described in Section 5.1 for evaluating its performance. The evaluation measures described in Section 4.2 are computed both on the training and testing subsets and the obtained confusion matrices are presented in Table 5. As a performance indicator for $SPRAR$, the last column from Table 5 contains the AUC value with a 95% confidence interval (CI) [4]. The values for all evaluation measures are given in Table 6. The results from Table 5 do not indicate an overfitting tendency of $SPRAR$ classifier, for none of the data sets.

Data set	TP		TN		FP		FN		AUC	
	training	**testing**	**training**	**testing**	**training**	**testing**	**training**	**testing**	**training**	**testing**
D1	228	68	24	10	10	5	32	7	0.79 ± 0.05	0.79 ± 0.08
D2	451	97	76	28	36	12	120	43	0.73 ± 0.03	0.70 ± 0.07
D3	351	234	190	61	50	9	177	82	0.73 ± 0.03	0.81 ± 0.04

Table 5: Performance of *SPRAR* classifier on the considered case studies. 95% CIs are used for the results.

Figure 4 illustrates the ROC curves (leftside image) and a plot for the positive predictive value (*precision*) vs. true negative rate (*specificity*) (rightside image) for the *SPRAR* classifier on the considered case studies (data sets D1, D2 and D3). From Table 6 and Figure 4 we observe, for all data sets, a precision above 0.89, a specificity above 0.67 and AUC values ranging from 0.7 to 0.81. All these values express a very good predictive performance of *SPRAR* classifier, both for the *positive* and *negative* class, considering the imbalanced nature of the data sets.



(a) (b)

Fig. 4: Experimental results: ROC curves (left) and positive predictive value vs. true negative rate (right) obtained by *SPRAR* on all data sets.

We observe from Table 5 that there is a perfect correlation between the difficulty of the data sets (Table 2) based on the AUC values. An analysis of the obtained results also reveals that the problem that presented the most difficulty to the *SPRAR* classifier was differentiating the students with grades between 4 and 5. While, on average, passing students would fit more rules from RAR_+ and failing students would fit more rules from RAR_-, students with grade 5 would fit similar number of rules from both sets, and would even fit more RAR_- rules than students with grade 4. Interestingly, this is not simply because the classifier has problems with instances at the limit. When raising the threshold for the positive class from 5 to 6, the performance increases significantly. For the D2 data set the AUC increased from **0.70** to **0.81** and for the D3 data set it increased from **0.81** to **0.92**. For the D1 data set a slightly decrease in AUC (from **0.79** to **0.70**) has been observed, but this is due to the small number of instances (only seven instances with the final grade 5 were found in the entire data set). The difficulty of classifying the instances with the final grade 5 is not necessarily a limitation of *SPRAR*, but is a specificity of the academic data sets, as other well known classifiers such ANNs, DTs, SVMs encounter the same problem (Table 6). A possible explanation for the complexity to discriminate the students with the final grade 4 and 5 may be the fact that deciding which students pass or fail is a real-life limit which may be more susceptible to be affected by external events (e.g. the willingness of the instructor to artificially give a passing grade to students that were really close to passing).

Table 6 compares the performance of our *SPRAR* classifier with the performance of several related approaches existing in the literature. For an accurate comparison, we applied the techniques from the related work (DTs [13], ANNs [19] and SVMs [24]) on our data sets D1, D2 and D3, employing in the experiments the same data sets and experimental methodology as described in Section 4. For comparison, the evaluation measures described in Section 4.2 are used. The best and the second best classifiers considering both AUC and $Spec$ measures are highlighted.

For the ANN we used the implementation from the Keras deep learning API [15] using the Tensorflow neural networks framework, and for the DT and SVM we used the implementation from the scikit-learn [20] machine learning framework. For the ANN we used the following parameter settings [19]: 2 hidden layers with 17 and 35 neurons, respectively, using the Sigmoid activation function; one output neuron with the linear activation function, thus the output being the predicted grade of the student; the mean squared error loss and the predefined adadelta optimizer; we trained the network for 30 epochs using a batch of 1 instance. For the SVM and DT classifiers we used the predefined settings. For all the classifiers we set weights for the classes, in order to combat the class imbalance; we used a 4:1 weight ratio in favor of the negative class.

Data set	**Model**	*Prec*	*Sens*	*F-score*$^+$	*NPV*	*Spec*	*F-score*$^-$	*AUC*
D1	**Our** *SPRAR*	**0.93**	**0.91**	**0.92**	**0.59**	**0.67**	**0.63**	**0.79 ± 0.08**
	DT [13]	0.92	0.95	0.93	0.69	0.60	0.64	0.77 ± 0.09
	ANN [19]	0.87	1.0	0.93	1.0	0.27	0.42	0.63 ± 0.10
	SVM [24]	0.92	0.97	0.95	0.82	0.60	0.69	0.78 ± 0.08
D2	**Our** *SPRAR*	**0.89**	**0.69**	**0.78**	**0.39**	**0.70**	**0.50**	**0.70 ± 0.07**
	DT [13]	0.83	0.80	0.81	0.37	0.43	0.40	0.61 ± 0.07
	ANN [19]	0.86	0.81	0.84	0.45	0.53	0.48	0.67 ± 0.07
	SVM [24]	0.85	0.75	0.80	0.39	0.55	0.45	0.65 ± 0.07
D3	**Our** *SPRAR*	**0.96**	**0.74**	**0.84**	**0.43**	**0.87**	**0.57**	**0.81 ± 0.04**
	DT [13]	0.87	0.82	0.84	0.35	0.43	0.38	0.63 ± 0.05
	ANN [19]	0.97	0.87	0.91	0.59	**0.87**	0.71	**0.87 ± 0.03**
	SVM [24]	0.82	0.61	0.70	0.18	0.39	0.24	0.5 ± 0.05

Table 6: Comparison to related work ([13], [19] and [24]). The AUC values are provided with 95% CIs

From Table 6 we observe that out of 63 comparisons, *SPRAR* **wins** (i.e. it provides a higher or equal value) in 38 cases, it **looses** (i.e. it provides a smaller value) in 25 cases. This reveals that in 60% of the comparisons, *SPRAR* outperforms the related work. A very good overall performance of *SPRAR* is obtained, as both the *AUC* and *Spec* values outperform the related work in 89% of the cases (8 out of 9). We observe that for data sets D1 and D2, which are the most difficult and imbalanced (see Table 2) *SPRAR* outperforms DT, ANN and SVM classifiers, confirming the potential of the *SPRAR* to handle imbalanced data sets.

Analyzing the experimental results from Table 5 and the comparison to existing approaches (Table 6) we may conclude that the RARs discovered in the grades received by the students during the semester are effective for predicting if they will pass or fail a certain course and further improvements may increase the predictive performance. A possible explanation for a lower performance of *SPRAR* (particularly for data sets D1 and D2) is the small number of instances and attributes characterizing the instances (data set D1) which increases the difficulty of the prediction process. As a conclusion, for a more accurate prediction, the number of students' evaluations during the semester (attributes in the learning model) has to be increased. Besides, there are some external factors which are very likely to make the prediction difficult, such as: (1) the students' learning process is not continuous during the academic semester; (2) it is very likely that the instructors from the laboratory and seminars activities do not have the same evaluation standards.

6. Conclusions and further work

As a proof of concept, this paper introduced a novel classification model *SPRAR* based on *relational association rule* discovery for predicting the successful completion of an academic course, based on the grades received by students during the academic semester. Experiments conducted on three real data sets collected from Babeş-Bolyai University from Romania have shown a good performance of the *SPRAR* classifier. The obtained experimental results highlighted that our classifier is better than, or comparable to, the supervised classifiers already applied in the EDM literature for students' performance prediction.

An advantage of *SPRAR* classifier introduced in this paper is its generality, as it is not specific to the students' performance prediction task. *SPRAR* is a generic classification model which can be applied to classification problems in which the instances from the training data set are characterized by a set of features. Any type of features may be used in the proposed model, as well as any binary relations between the features. In addition, *SPRAR* can be easily adapted to binary or multi-class classification problems, other than the students' performance task, such as software defect prediction [16]. The limitations of *SPRAR* are that it is not applicable to regression problems and that they currently handle only binary relations between the features' values.

Future work will be carried out in order to extend the experimental evaluation of *SPRAR* on other case studies. In addition, we aim to generalize the *SPRAR* binary classifier to a multi-class classifier such that to predict the students' final examination grade at a certain academic course. Regarding the relational association rules discovery process, we plan to extend our model considering *gradual relational association* rules [11]. Alternative measures for defining the probabilities p_+ and p_- will be further investigated, as well as the idea of using an ensemble of *SPRAR* classifier for improving the predictive performance. *SPRAR* may also be extended for handling n-ary relations between the attributes' values.

References

[1] Ahmed, A.M., Rizaner, A., Ulusoy, A.H., 2016. Using data mining to predict instructor performance. Procedia Computer Science 102, 137 – 142.
[2] Beaulac, C., Rosenthal, J.S., 2019. Predicting university students' academic success and major using random forests. Research in Higher Education , 1–25.
[3] Boetticher, G.D., 2007. Advances in Machine Learning Applications in Software Engineering. IGI Global. chapter Improving the Credibility of Machine Learner Models in Software Engineering.
[4] Brown, L., Cat, T., DasGupta, A., 2001. Interval estimation for a proportion. Statistical Science 16, 101–133.
[5] Câmpan, A., Şerban, G., Marcus, A., 2006. Relational association rules and error detection. Studia Univ. Babes-Bolyai Informatica LI, 31–36.
[6] Campan, A., Şerban, G., Truta, T.M., Marcus, A., 2006. An algorithm for the discovery of arbitrary length ordinal association rules, in: Proceedings of the 2006 International Conference on Data Mining, pp. 107–113.
[7] Crivei, L., 2018a. Academic data set. Http://www.cs.ubbcluj.ro/~liana.crivei/AcademicDataSets/SPRAR/.
[8] Crivei, L.M., 2018b. Incremental relational association rule mining of educational data sets. Studia Universitatis Babes-Bolyai Series Informatica 63, 102–117.
[9] Czibula, G., Bocicor, M.I., Czibula, I.G., 2012. Promoter sequences prediction using relational association rule mining. Evolutionary Bioinformatics 8, 181–196.
[10] Czibula, G., Marian, Z., Czibula, I.G., 2015. Detecting software design defects using relational association rule mining. Knowledge and Information Systems 42, 545–577.
[11] Czibula, I.G., Czibula, G., Miholca, D.L., 2018. Enhancing relational association rules with gradualness. International Journal of Innovative Computing, Information and Control 13, 289–305.
[12] Gu, Q., Zhu, L., Cai, Z., 2009. Evaluation measures of the classification performance of imbalanced data sets, in: International Symposium on Intelligence Computation and Applications (ISICA), Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 461–471.
[13] Hajizadeh, N., Ahmadzadeh, M., 2014. Analysis of factors that affect the students academic performance - data mining approach. International Journal of advanced studies in Computer Science and Engineering IJASCSE 3, 1–4.
[14] Jishan, S.T., Rashu, R.I., Haque, N., Rahman, R.M., 2015. Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique. Decision Analytics 2, 1–10.
[15] Keras, 2018. The Python Deep Learning library. Https://keras.io/.
[16] Miholca, D., Czibula, G., Czibula, I.G., 2018. A novel approach for software defect prediction through hybridizing gradual relational association rules with artificial neural networks. Inf. Sci. 441, 152–170.
[17] Mohamad, S.K., Tasir, Z., 2013. Educational data mining: A review. Procedia - Social and Behavioral Sciences 97, 320 – 324.
[18] Pal, A.K., Pal, S., 2013. Analysis and mining of educational data for predicting the performance of students. International Journal of Electronics, Communication and Computer Engineering 4, 2278–4209.
[19] Pauziah Mohd Arsad, N.B., lail Ab Manan, J., 2013. A neural network students performance prediction model (NNSPPM), in: Proc. of the IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), IEEE Computer Society. pp. 1–5.
[20] Scikit-learn, 2018. Machine learning in Python. Http://scikit-learn.org/stable/.
[21] Serban, G., Campan, A., Czibula, I.G., 2006. A programming interface for finding relational association rules. International Journal of Computers, Communications & Control I, 439–444.
[22] Shahiri, A.M., Husain, W., Rashid, N.A., 2015. A review on predicting student's performance using data mining techniques. Procedia Computer Science 72, 414 – 422.
[23] Sushil Kumar Verma, R.T., Jaloree, S., 2017. Fuzzy association rule mining based model to predict students performance. International Journal of Electrical and Computer Engineering 7, 2223–2231.
[24] Tran, T.O., Dang, H.T., Dinh, V.T., Truong, T.M.N., Vuong, T.P.T., Phan, X.H., 2017. Performance prediction for students: A multi-strategy approach. Cybernetics and Information Technologies 17, 164 – 182.