Improving classification with IF-THEN rules for multidimensional datasets

M. Muntean *, H. Vălean, L. Căbulea***

**"1 Decembrie 1918" University of Alba Iulia, Romania*
(Tel: +40-0258-806130; e-mail: mmuntean@uab.ro, cabuleal@uab.ro).
***Technical University of Cluj Napoca, Romania*
(Tel: +40 264 202367 e-mail:
Honoriu.Valean@aut.utcluj.ro)

Abstract: The multidimensional datasets are becoming widespread in both scientific and business computing. Dealing efficiently with high-dimensional data is a challenge for researchers in the database field.

This paper proposes BIMA, a new classification method which uses the discovered rules in RIPPER classification in order to select the boundary instances of multidimensional datasets and to multiply them in the training phase of the next evaluation. In the testing phase, the instances were kept unchanged. In the experimental part it was demonstrated that the BIMA is a promising algorithm for improving the IF-THEN rules classification accuracy and also for improving the TP value of the multidimensional datasets classes. The efficiency of the proposed algorithm is proved by using the UAB graduates' responses datasets.

Keywords: Machine learning, Classification, Accuracy, Algorithms, Boundary element method.

1. INTRODUCTION

The large amounts of data are collected and persistent stored in databases, increasing the need for efficient and effective analysis methods in order to use the information data. There could be a lot of patterns in a huge multidimensional database, and a lot of efficient data mining methods had been proposed to discover these models.

Subramanian and Ramaraj (Subramanian and Ramaraj, 2011) propose new reachability based outlier detection algorithm for multidimensional databases. The proposed problem is broken down into sub phases. The first phase calculates the reachability of each object. The second phase finds outlier from the databases.

The framework presented in (Bizzi et al., 2009) is suitable to analyze the influences of stream habitats on a full range of environmental objectives. This approach gives the possibility to develop optimum habitat classifications able to meet management requirements and to minimize the number of habitat classes identified using a growing hierarchical self-organizing map.

In (Traina et al., 2001) the authors focused on the problem of finding patterns across large, multidimensional datasets. They proposed a new tool, the tri-plot, and its generalization, the pq-plot, which classify the considered datasets. They provided a set of rules on how to interpret a tri-plot, and they applied these rules on synthetic and real datasets. The authors also showed how to use their tool for classification, when traditional methods (nearest neighbor, classification trees) may fail.

In (Charrier et al., 2012) a machine learning expert, providing a quality score is proposed. This quality measure is based on a learned classification process in order to respect human observers.

In this paper RIPPER (Cohen, 1995), (Liu et al., 2011) is used. RIPPER is a sequential covering algorithm which grows rules by adding a test of an attribute to a rule as long as using the current attribute will lead to a more accurate separation of the training data. RIPPER algorithm model can be represented in the form of IF-THEN rules, which are suitable for knowledge updating of multidimensional datasets.

The proposed Boundary Instances Multiplier Algorithm (BIMA) selects the boundary instances after the RIPPER classification in order to multiply them in the training phase of the next evaluation. In the experimental part, it was demonstrated that the BIMA can help RIPPER classifier to better recognize the class instances and it was also showed that the proposed method is suitable for multidimensional datasets.

2. RELATED WORK

Boundary instances are treated differently in many classification algorithms.

In the paper (Rotaru and Litman, 2003) the authors investigate a new topic by looking into whether exceptionality measures can be used to characterize the performance of the RIPPER rule-based learner. This paper shows that some exceptionality measures can be used as means to improve the prediction accuracy on the tasks by combining the prediction of the learner based on measures of instance exceptionality.

The reference (Panda et al., 2006) proposes an algorithm to select boundary instances as training data to substantially reduce n from $O(n^2)$ training cost, where n denotes the number of training instances, in Support Vector Machines classification. The algorithm eliminates instances that are likely to be non-support vectors.

In reference (Guo et al., 2010) the authors present a new efficient support vector selection method based on ensemble margin, a key concept in ensemble classifiers. This algorithm exploits a new version of the margin of an ensemble-based classification and selects the smallest margin instances as support vectors.

3. RULE INDUCTION USING SEQUENTIAL COVERING ALGORITHMS

By using a sequential covering algorithm (Fidelis et al., 2000), extraction of IF-THEN rules directly from the training dataset is possible. The notion of "sequential" comes from the fact that the algorithm learns the rules sequentially (one at a time), where each rule for a given class will ideally cover many of the instances of that class and hopefully none of the instances of other classes.

Unlike indirect methods of extracting rules (e.g. C.4.5 algorithm which extracts rules from decision trees), RIPPER generates rules directly from data and parses the discovered IF-THEN rules into the antecedent and consequent form in order to perform the classification process. Each antecedent is structured in more attribute tests that is, more IF sub-conditions are all gathered in a big IF condition. Each attribute test can be considered as a little antecedent. Every attribute can be a nominal (categorical) attribute or a numerical (continuous) attribute. Like the attributes, the attribute tests can be nominal or numerical. Thus, each rule can have an antecedent with mixed attribute tests. The antecedent and consequents provide a better and more structured way of working with the rule (Han and Kamber, 2006), (Jiang'hong and Xiao'li, 2009). If the condition, that is, all of the attribute tests in a rule antecedent, holds true for a given instance, then the rule covers the instance.

The structure of each attribute test in an antecedent contains a name, a relational operator and a value field (Figure 1).

Name	Relational Operator	Value

Fig. 1. The structure of an attribute test.

The value is generated randomly from the range of attribute values. The relational operator's task is to verify if the corresponding instance is covered by the rule. It is also generated randomly from the list of possible relational operator's values. The name represents the name of the attribute and it is extracted from the *arff* ("attribute-relation file format") input file.

An *arff* file represents a standard way of representing datasets that consist of independent, unordered instances and do not involve relationships among instances.

First of all, the RIPPER classifier starts by loading the dataset, and by finding the IF-THEN rules. Then, the discovered rules are parsed (Figure 2):

(1) load the dataset to be processed;

(2) parse the IF-THEN rule into the antecedent and consequent form;

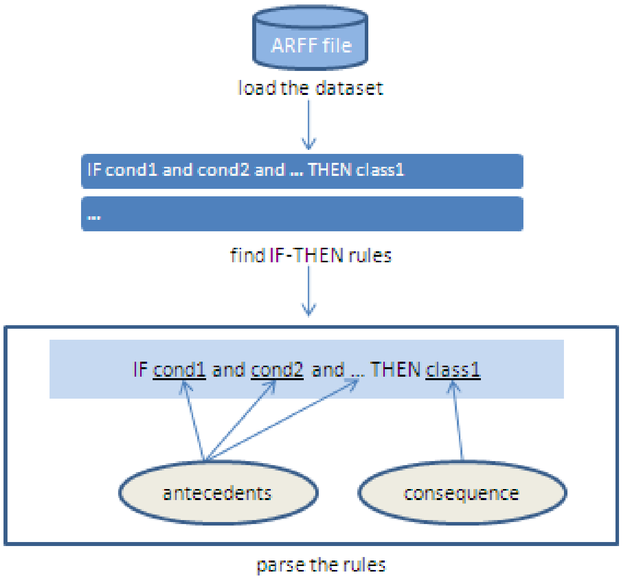

Fig. 2. The discovery and computation of rules.

The algorithm continues with the computation of the distribution of a rule. After this, the distribution for each instance in the data set is determined, by simply checking the class and setting a flag. This represents the actual distribution of the instance. In the same for loop, the predicted distribution of each instance is calculated (Figure 3) using the following procedure (Muntean et al., 2010):

(3) compute the class distribution for the rule;

(4) compare the class distribution given by the rule and the class distribution given by the instance;

(5) refer to instances as true and false positives and negatives;

(6) provide the sensitivity and specificity measures;

(7) determine the fitness by multiplying these measures.

The sensitivity and specificity measures are computed using true positives (TP), true negatives (TN), false positive (FP) and false negative (FN) measures. The TP and TN are correct classifications. A FP occurs when the outcome is incorrectly predicted as belonging to the positive class, when it actually belongs to the negative one (considering the two class classification problem). A FN occurs when the outcome is incorrectly predicted as negative when it is actually positive.

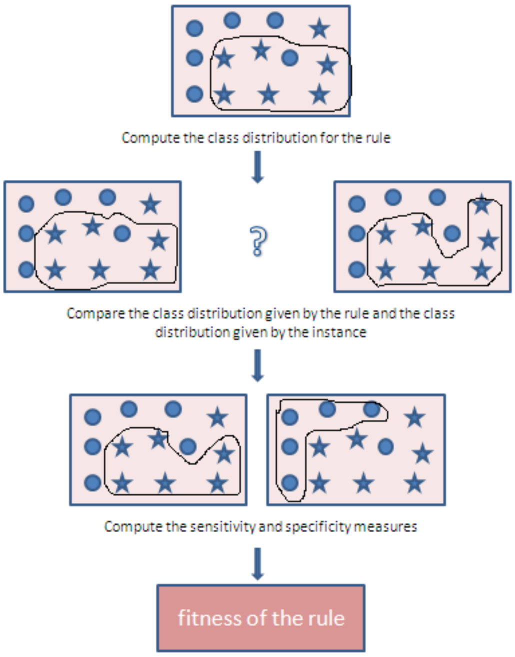

Fig. 3. The fitness measure computation.

The Confusion Matrix for a two class classification problem is shown in Table I.

Table 1. Confusion Matrix for a two class problem.

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	TP	FN
	Class = 0	FP	TN

In order to assess how well the model can classify the instances, the three mentioned measures (sensitivity, specificity and fitness) were used:

$$sensitivity = \frac{TP}{TP + FN} \quad (1)$$

$$specificity = \frac{TN}{TN + FP} \quad (2)$$

$$fitness = sensitivity * specificity \quad (3)$$

In addition, the accuracy measure was defined. It represents the ratio between correctly classified instances and the sum of all instances classified, both correct and incorrect ones:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

The basic sequential covering algorithm is presented bellow (Han and Kamber, 2006):

(1) Rule set = {}; *// initial set of rules learned is empty*
(2) *for* each class c *do*
(3) *repeat*
(4) Rule = Learn One Rule(D, Att_vals, c); *// where D is a dataset class-labeled tuples and Att_vals represent the set of all attributes and their possible values.*
(5) remove tuples covered by Rule from D;
(6) *until* terminating condition;
(7) Rule set = Rule set +Rule; *// add new rule to rule set*
(8) *end for*
(9) *return* Rule Set.

The process of learning rules continues until the terminating condition is met, such as when there are no more training tuples or the quality of a rule returned is below a user-specified threshold. *The Learn One Rule* technique finds the best rule for the current class, given the current set of training tuples.

4. BOUNDARY INSTANCES MULTIPLIER ALGORITHM

After finding the best classification rules for a specific dataset using a sequential covering algorithm, it was applied the proposed Boundary Instances Multiplier Algorithm (BIMA) in order to improve the accuracy of classification.

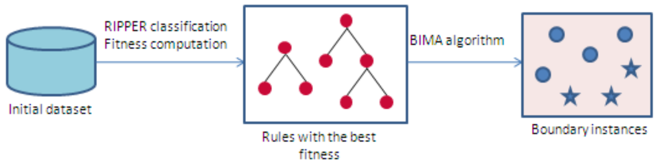

Fig. 4. The selection of boundary instances.

The BIMA works as follows (Muntean et al., 2012):

(1) determine the rules with the best fitness for each class
(2) *repeat*
(3) *read* an IF-THEN rule from the rule set obtained at step (1);
(4) *for* each IF sub-condition *do*
(5) *if* the attribute is numeric
(6) *then if* the relational operator is equal to "=" or it is equal to "<>"
(7) *then* do not change the attribute value
(8) *else if* relational operator is equal to "<"
(9) *then* search the instances with the attribute value within the interval $(g\_value - \Delta value/10, g\_value)$
(10) *else if* relational operator is equal to "<="
(11) *then* search the instances with the attribute value within the interval $[g\_value - \Delta value/10, g\_value]$
(12) *else if* relational operator is equal to ">"
(13) *then* search the instances with the attribute value within the interval $(g\_value, g\_value + \Delta value/10)$
(14) *else* search the instances with the attribute value within the interval $[g\_value, g\_value + \Delta value/10]$
(15) *else if* the attribute is nominal

(16) *then* keep its value unchanged
(17) *end for*
(18) *until* there are no more discovered rules
(19) extract from the initial dataset the labelled instances found at the previous steps
(20) multiply the extracted instances in the training classification phase
(21) evaluate the dataset

In the presented above algorithm, g_value represents the generated value, i.e. the value generated for the current attribute test in RIPPER learning process, $\Delta value = value_{\max} - value_{\min}$ represents the difference between the maximum value from the dataset for the current attribute and the minimum value from the dataset for the current attribute.

After some experiments, the measure $\Delta value / 10$ was established as being the proper one for finding the boundary instances using RIPPER classifier model.

Clear results were obtained by choosing a 66% split percentage, which means that about 34% records were used as test data. The boundary instances discovered with BIMA were multiplied in the training phase with the condition of keeping the instances unchanged in the testing phase of classification (Figure 5).

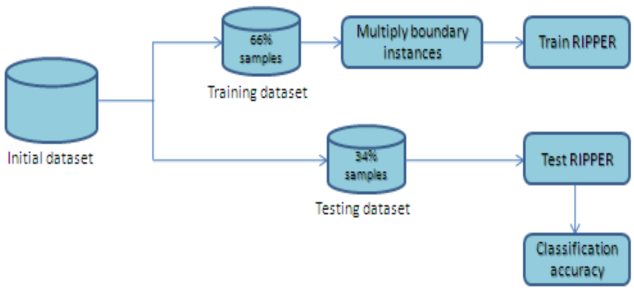

Fig. 5. The percentage split evaluation.

The proposed algorithm helps RIPPER better recognize the instances situated close to the separation margins of the classes.

5. EXPERIMENTAL RESULTS

5.1 The datasets description

The UAB graduates' responses datasets contains the answers of 593, respectively 141 graduates of 1 Decembrie 1918 University of Alba Iulia to a questionnaire with 91 questions. The purpose of this study was to evaluate if knowledge, competencies and skills obtained during the studies were sufficient to enable university graduates, promotion 2008-2009, respectively promotion 2005-2006, to engage in the labour market or to continue their studies.

The answers to this monitoring questionnaire were stored into two databases and then pre-processed and saved as an *.arff* (Attribute Relation File Format) file (Figure 6). The "I don't know" answer was codified with the value -9, and the "I don't answer" affirmation was codified with -7.

```
2439,130000,2009,2,-7,-7,-7,8,2005,669,713,717,725,-9,-9,-9,-9,1,1,13
2545,130000,2009,1,10000,11000,11003,3,2006,681,713,-9,-9,-9,-9,-9,-9
2562,130000,2009,1,10000,11000,11009,3,2005,-9,-9,-9,-9,669,713,-9,-9
2577,130000,2005,1,10000,11000,11003,2,2001,621,665,669,685,693,709,-
2682,130000,2009,1,10000,11000,11002,8,1999,681,713,-9,-9,-9,-9,-9,-9
2724,130000,2009,1,10000,12000,12036,1,2005,681,713,717,9999,-9,-9,-9
2736,130000,2009,1,10000,11000,11003,4,1994,681,713,729,9999,-9,-9,-9
2778,130000,2005,1,10000,11000,11007,1,2001,620,665,693,713,-8,-9,-9,

130000,130300,130301,2,8.66,1,2,390000,390200,390201,2,9.9,-7,-
-9,1,1,130000,130300,130304,1,9.5,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7
-9,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,4,1,130000,130200,
,-9,-9,1,1,130000,130200,130204,1,8.74,1,2,130000,130200,130203
-9,4,1,130000,130200,130205,3,7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-
-9,-9,1,1,130000,130300,130303,1,9.17,3,2,130000,130200,130206,
-9,-9,1,1,130000,130300,130306,1,9.11,3,2,130000,130300,130306,

8.66,1,2,390000,390200,390201,2,9.9,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7
0304,1,9.5,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,
,-7,-7,-7,-7,-7,-7,-7,4,1,130000,130200,130207,1,9.3,-7,-7,-7,-7,
0,130204,1,8.74,1,2,130000,130200,130203,3,9.53,1,2,130000,130200
0205,3,7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7
,130303,1,9.17,3,2,130000,130200,130206,3,9.17,-7,-7,-7,-7,-7,-7,
,130306,1,9.11,3,2,130000,130300,130306,3,-9,-7,-7,-7,-7,-7,-7,-7
130101,1,8.91,1,2,130000,130100,130101,1,9.41,3,3,200000,200800,-
```

Fig. 6. The pre-processed answers of the graduate students.

A question can have multiple choices, so the corresponding attributes have similar names (for instance STU1_1, STU1_2). A sample of attribute declaration in the *.arff* file is presented in Figure 7.

```
@relation absolventi_uab
@attribute ID numeric
@attribute HEI numeric
@attribute GY numeric
@attribute R1 numeric
@attribute R1_J numeric
@attribute R1_T numeric
@attribute R1_L numeric
@attribute R2 numeric
@attribute R3 numeric
@attribute STU1_1 numeric
@attribute STU1_2 numeric
@attribute STU2_1 numeric
@attribute STU2_2 numeric
@attribute STU3_1 numeric
@attribute STU3_2 numeric
@attribute STU4_1 numeric
@attribute STU4_2 numeric
@attribute R6_1 numeric
@attribute R7_1 numeric
@attribute R8_1_I numeric
@attribute R8_1_F numeric
@attribute R8_1_P numeric
@attribute R10_1 numeric
@attribute R11_1 numeric
@attribute R6_2 numeric
@attribute R7_2 numeric
@attribute R8_2_I numeric
@attribute R8_2_F numeric
@attribute R8_2_P numeric
@attribute R10_2 numeric
@attribute R11_2 numeric
@attribute R6_3 numeric
@attribute R7_3 numeric
@attribute R8_3_I numeric
```

Fig. 7. A sample of attribute declaration in the *.arff* file.

The datasets used for the experiments have 256 attributes and 593, respectively 141 instances, both numeric and nominal ones.

Table 2. The datasets used in the experiments.

Dataset	Dimension of the dataset	No. of instances	Attributes types
dataset1, promotion 2008-2009	256 + 1	593	Num, Nom
dataset2, promotion 2005-2006	256 + 1	141	Num, Nom

In the pre-processing step, k-means data analysis algorithm (Witten et al., 2011) was trained and tested, in order to discover the percentage of graduate students that had success in their career, and in order to label this category of students with class 1 in each of the two datasets. Also the graduates that don't have a job or that don't have a success career were labelled (class 2).

According k-means algorithm, there were arbitrarily chosen two objects as the two initial cluster centers. Each object was distributed to a cluster based on the cluster center to which it is the nearest. Next, the cluster centers were updated. Then, the mean value of each cluster was recalculated based on the current objects in the cluster. Using the new cluster centers, the objects were redistributed to the clusters based on which cluster center is the nearest. The process of iteratively reassigning objects to clusters to improve the partitioning is referred to as iterative relocation. Eventually, no redistribution of the objects in any cluster occurs, and so the process terminates. The resulting clusters are returned by the clustering process, meaning the two categories of students. The algorithm labelled these categories with the proposed labels.

The class distribution of the datasets is illustrated below (Table 3):

Table 3. The distribution of instances in the two classes

Dataset	dataset1		dataset2	
Class	class 0	class 1	class 0	class 1
No. of instances	466 (79%)	127 (21%)	108 (77%)	33 (23%)

5.2 Experimental results with different classifiers

The most adequate classifiers for multidimensional datasets were used to perform the classification task of data mining process in order to conclude which one classifies better the considered datasets. The classifiers trained and tested were: Naïve Bayes, Support Vector Machines, k-Nearest Neighbour, IF-THEN rules and Decision Trees.

In order to implement these experiments the Weka Data Mining Software was used. Originally proposed for didactic purposes, Weka is a framework for the implementation and deployment of data mining methods. It is also open-source software developed in Java, released under the GNU General Public License (GPL), being currently available to Windows, MAC OS and Linux platforms (Weka, 2013).

Weka contains tools for classification, regression, clustering, association rules, data visualization and works with *.arff* files (Attribute Relation File Format) and also with files in *.csv* format (Comma Separated Values).

Clear results were obtained by choosing a 66% split percentage, which means that about 34% records were used as test data in the pre-implemented training process before classification (Wang et al., 2011).

The classifiers were evaluated on how well they predicted the percentage of the data held out for testing.

Table 4 presents the classification results of the two datasets with different learning models.

Table 4. The classification results

	dataset1	dataset2
Classifier	Classification accuracy (%)	
Naïve Bayes (NB)	98.81	97.16
Stochastic Gradient Descent (SGD)	98.31	98.58
Logistic	98.65	97.16
Instance-based learner (KStar)	91.90	90.07
Decision tree (J48)	98.31	97.87
Decision tree (REPTree)	96.79	96.45
IF-THEN rules classifier (ZeroR)	78.58	76.59
RIPPER classifier (JRip)	98.31	94.32

It can be seen that Naïve Bayes best classified the first dataset (98.81%) and Stochastic Gradient Descent was the most suitable classifier for the second dataset (98.58%), but also RIPPER learning method has high accuracy rates (98.31% and 94.32%, respectively) (Figure 8 and Figure 9).

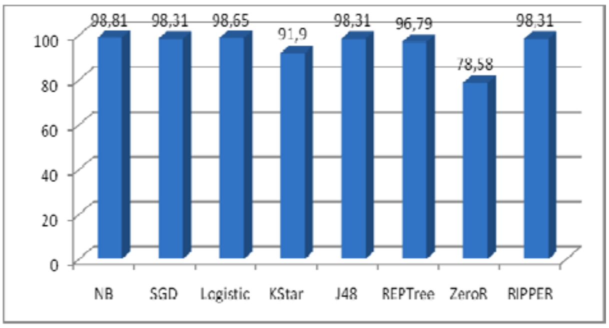

Fig. 8. The dataset1 classification results.

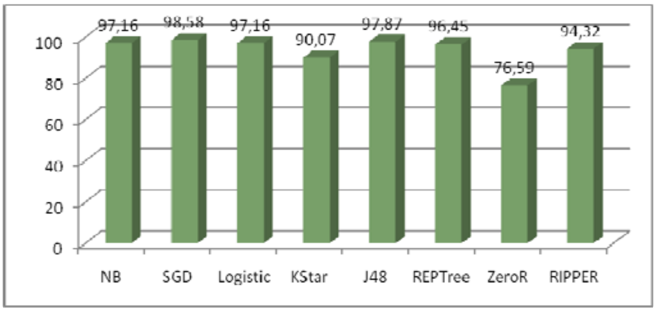

Fig. 9. The dataset2 classification results.

5.3 Improving the overall accuracy with RIPPER classifier

In WEKA, a cloned RIPPER algorithm called JRip is designed to execute classification of datasets while simulating the process of sequential covering algorithm.
The JRip discovered rules in the evaluation of the *dataset1* were the following ones:

IF ((R32_9 <= -4) and (R31_17 <= -4)) THEN cluster1 ELSE cluster0
IF ((R30_1 <= -9) and (R45_1 <=-8)) THEN cluster1 ELSE cluster0

In the case of *dataset2*, the discovered model consists of the following rule:

IF (R32_5 <= -4) THEN cluster1 ELSE cluster0

Some of the test attributes like *R31_17* and *R45_1* refer to the importance of the graduated study domain in the development of the personal career of the graduate students. Other attributes like *R30_1* and *R32_5* evaluate the importance of the practical activities undertaken within the graduate study program. In other words, RIPPER selected from the set of 256 attributes the most important ones in order to perform a high accuracy classification.

The discovered rules were used as input for the BIMA algorithm in order to determine the boundary instances and to multiply them in the training phase of the next evaluation. After evaluating the two multidimensional datasets with different values for the multiplication rate, it could be seen that the accuracy of classification reached two peaks of maxima in the case of dataset1, while in the case of dataset2 the accuracy was maintained at high rates (98.95%), (Figure 10 and Figure 11).

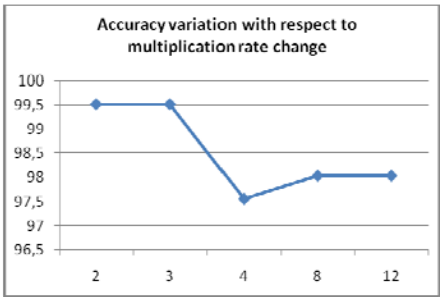

Fig. 10. Accuracy variation with respect to multiplication rate change (dataset1).

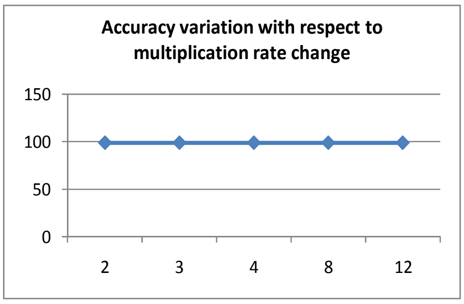

Fig. 11. Accuracy variation with respect to multiplication rate change (dataset2).

These experiments show that for certain values given to the multiplication rate, the classification accuracy was better than the one found in JRip evaluation before applying BIMA algorithm (Figure 10 and Figure 11) in the both datasets. In the case of dataset1 the accuracy found with JRip classification was 98.31 while the one obtained with BIMA algorithm was equal to 99.51. It can be also observed an improvement in the classification of the dataset2 from 94.32 (found with JRip) to 98.95 (after applying BIMA method).

The accuracy was highest even than Naïve Bayes classification in the case of the first dataset.

Figure 12 presents the comparison between initial accuracy and the accuracy obtained after applying the best multiplication rate obtained at the previous step, for the first dataset.

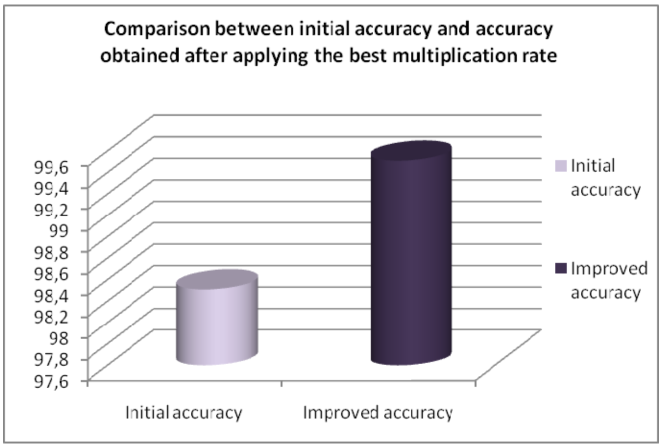

Fig. 12. Comparison between initial accuracy and the accuracy obtained after applying the best multiplication rate (dataset1).

In the training phase of the first dataset, the RIPPER classifier improved with BIMA algorithm needed 0.1 seconds more time in order to build the model (0.66 seconds compared with 0.56 seconds in the case of RIPPER classification, Figure 13).

An important aspect is that the time taken to test model on training split was the same in the two classifications: 0.03 seconds.

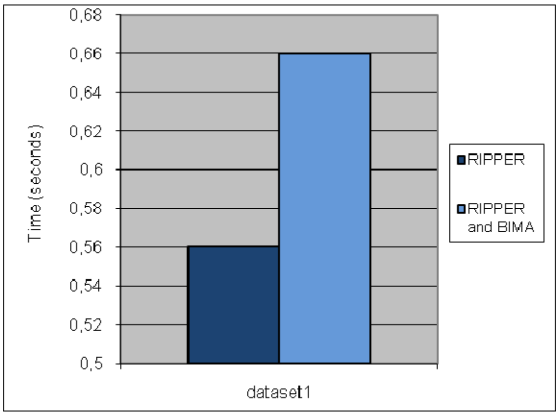

Fig. 13. Comparison between the time (in seconds) necessary to build the model with RIPPER classifier and with RIPPER improved with BIMA in the case of dataset1.

The accuracy was highest even than Stochastic Gradient Descent accuracy for the second dataset, meaning a better value than the best one found in the previous experiments (Figure 14). The multiplication rate was set to 2, the accuracy being constant with respect to the rate change.

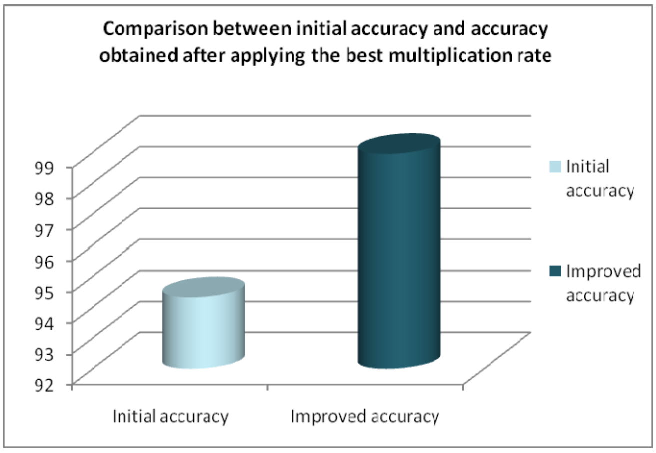

Fig. 14. Comparison between initial accuracy and the accuracy obtained after applying the best multiplication rate (dataset2).

The timespan for building model of the second dataset with RIPPER and BIMA classifier was 0.52 seconds, comparing to 0.41 seconds consumed in the training phase by RIPPER classifier (Figure 15). The difference is approximately the same as in the case of the first dataset classification, 0.1 seconds.

In the split percentage testing phase for the second dataset, the time spent by both classifiers (RIPPER and RIPPER improved with BIMA) was the same, meaning 0.02 seconds.

Considering that in the testing phase the model used 202 instances from de first dataset and 48 instances belonging to the second dataset, the computing time of the classifier was very good. These results are probably due more to an accurate pre-process of data by storing all graduates' answers as numbers.

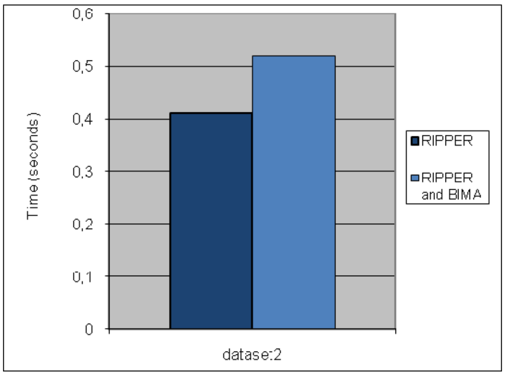

Fig. 15. Comparison between the time (in seconds) necessary to build the model with RIPPER classifier and with RIPPER improved with BIMA in the case of dataset2.

5.4 Improving the TP of the weak represented class

The considered datasets have unbalanced data distribution because the class 2 of data has few training examples compared to class 1. The JRip and BIMA proposed method classified the instances of some classes of interest better than the classification of the JRip algorithm (Figure 16 and Figure 17).

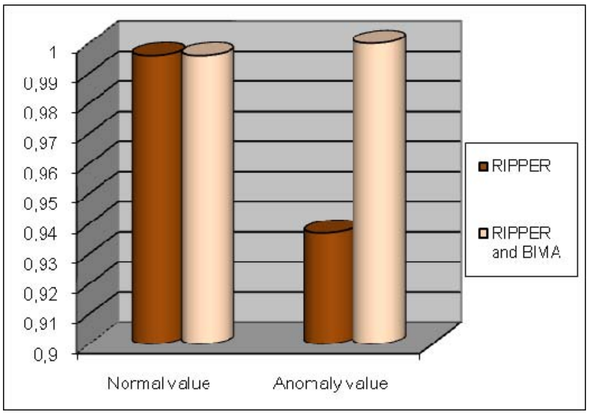

Fig. 16. Comparison between the TP of the classes resulting JRip Evaluation and JRip and BIMA Evaluation (dataset1).

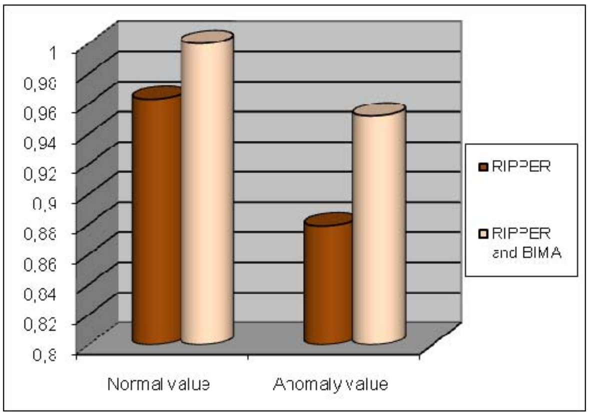

Fig. 17. Comparison between the TP of the classes resulting JRip Evaluation and JRip and BIMA Evaluation (dataset2).

The BIMA algorithm improved the classification of the weakly represented class of the multidimensional datasets, while also improving the general accuracy. Finding the instances from the separating class margins and helping the classifier to recognize better these instances proved to be a promising method, after performing the experiments.

6. CONCLUSIONS

In this paper, a new algorithm for finding patterns in multidimensional datasets is introduced. The proposed classification method uses the discovered rules in JRip classification in order to select the boundary instances of multidimensional datasets and to multiply them in the training phase of the next evaluation. The results have shown that our proposed BIMA is a viable method for improving the IF-THEN rules classification accuracy and also for improving the TP value of the classes. As a further research, we propose to run the BIMA algorithm also with other classifiers, such as: Naïve Bayes or Stochastic Gradient Descent.

ACKNOWLEDGMENT

This research was partially supported by the project 60/2.1/S/41750 POSDRU implemented by Executive Agency for Higher Education, Research, Development and Innovation Funding (UEFISCDI) and the National Council for Higher Education Funding in partnership with The International Centre for Higher Education Research (INCHER) Kassel.

REFERENCES

Bizzi, S., Harrison, R., F., Lerner, D., N. (2009). The Growing Hierarchical Self-Organizing Map (GHSOM) for analysing multi-dimensional stream habitat datasets, in Anderssen, R.S., R.D. Braddock and L.T.H. Newham (eds) *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation*, July 2009, ISBN: 978-0-9758400-7-8, pp. 734-740, http://mssanz.org.au/modsim09.

Charrier, C., Lézoray, O., Lebrun, G. (2012). Machine learning to design full-reference image quality assessment algorithm, *Signal Processing: Image Communication*, 27(3), 209-219.

Cohen, W., W. (1995). Fast Effective Rule Induction, *Machine Learning Proceedings of the 12th International Conference*, pp.115-123.

Fidelis, M., V., Lopes, H., S., and Freitas, A., A. (2000). Discovering Comprehensible Classification Rules with a Genetic Algorithm, *Proceedings of the 2000 Congress on Evolutionary Computation*, vol1, pp.805-810.

Guo, L., Boukir, S., and Chehata, N. (2010). Support Vectors selection for supervised learning using an ensemble approach, *2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, August 23-August 26, pp. 37 – 40.

Han, J., Kamber, M. (2006). *Data Mining: Concepts and Techniques*, Second Edition, Morgan Kaufmann Press, Elsevier Inc, San Francisco, ISBN 13: 978-1-55860-901-3, pp.319-325.

Jiang'hong, S. Xiao'li, X. (2009). Large Rotating Machinery Fault Diagnosis and Knowledge Rules Acquiring Based on Improved RIPPER, *2009 Second International Conference on Intelligent Computation Technology and Automation*, October, 10-11, Zhangjiajie, pp. 549-552.

Liu, S., Patel, R., Y., Daga, P., R., Liu, H., Fu, G., Doerksen, R., Chen, Y., Wilkins, D. (2011). Multi-class Joint Rule Extraction and Feature Selection for Biological Data, *IEEE International Conference on Bioinformatics and Biomedicine*, Atlanta, Georgia, USA, November, pp. 476-481.

Muntean, M., Vălean, H., Ileană, I. (2012). Improving classification with boundary instances multiplier algorithm based on IF-THEN rules, *Proceedings of 2012 IEEE International Conference on Automation, Quality and Testing, Robotics*, May 24-27, Cluj-Napoca, Romania, ISBN: 9781-1-4673-0702-4, IEEE Cat. No.: CFP12AQT-PRT, pp. 272-277.

Muntean, M., Vălean, H., Rotar, C., Ileană, I. (2010). Learning Classification Rules with Genetic Algorithm", 2010 8th International Conference on Communications, Vol. 1, Bucharest, Romania, June, pp. 213-216.

Panda, N. Chang, E. Wu, Y. G. (2006). Concept Boundary Detection for Speeding up SVMs, *ICML '06 Proceedings of the 23rd International Conference on Machine learning*, ACM New York, NY, USA, pp. 681 – 688.

Rotaru, M. and Litman, D. J. (2003). Exceptionality and Natural Language Learning, *Proceedings of CoNLL-2003*, Edmonton, Canada, pp. 63-70.

Subramanian, K., Ramaraj, Dr.E. (2011). A new reachability based algorithm for outlier detection in multidimensional dataset, *Journal of Theoretical and Applied Information Technology*, 15th March, Vol.25, No.1, ISSN: 1992-8645, pp. 1-9.

Traina, A., Traina, C., Papadimitriou, S., Faloutsos, C. (2001). Tri-Plots: Scalable Tools for Multidimensional Data Mining, *Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, San Francisco, CA, August, 19-26, 2001, pp. 1-21.

Wang, G., Zhang, C., Huang, L. (2008). A Study of Classification Algorithm for Data Mining Based on Hybrid Intelligent Systems, *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Phuket Thailand, pp. 371-375.

Weka Data Mining Software (2013). http://www.cs.waikato.ac.nz/ml/weka/

Witten, I., H., Frank, E., Hall, M., A. (2011). *Data mining. Practical Machine Learning Tools and Techniques*, Third Edition, Morgan Kaufmann, ISBN: 978-0-12-374856-0, pp. 139.