IRBASIR-B: Rule Induction from Similarity Relations, a Bayesian Approach

Lenniet Coello[1], Yaima Filiberto[2(✉)], Rafael Bello[3], Mabel Frias[4], and Rafael Falcon[5]

[1] Ryder, Miami, USA
Lenniet_m_coello@ryder.com
[2] AMV Soluciones, Avenida de Madrid 40, Oficina 14, Vigo, Spain
yaima.filiberto@amvsoluciones.com
[3] Central University of Las Villas Carretera Camajuaní km 5.5, Santa Clara, Cuba
rbellop@uclv.edu.cu
[4] University of Camaguey Carretera de Circunvalación Norte entre Camino Viejo de Nuevitas y Ave Ignacio Agramonte, Camagüey, Cuba
mabel.frias@reduc.edu.cu
[5] University of Ottawa, Ottawa, Canada
rfalcon@uottawa.ca

Abstract. IRBASIR is a recently proposed algorithm that, inspired on one of the extensions of classical Rough Set Theory, employs similarity relations to learn classification rules. By using similarity relations as its underlying building blocks, IRBASIR is able to process datasets with both nominal and numerical features. In this paper we propose IRBASIR-Bayes, a modification to the IRBASIR method that relies on Bayesian Networks to construct the reference vector used to generate the rules. This scheme has demonstrated satisfactory performance compared to other rule induction algorithms.

Keywords: IRBASIR · Bayesian approach · Similarity relations

1 Introduction

Technological innovation continues to grow at a fast pace, thus giving rise to modern computational tools that make capturing, storing and processing a myriad of data a much straightforward endeavor than in previous generations; this fact, together with the overwhelming volume of data being generated in the last two decades, has sparked a renewed interest in different scientific branches like Machine Learning [12].

Machine Learning is a very active research field within Artificial Intelligence that seeks to develop automated techniques capable of learning from the vast oceans of data available nowadays; in other words, the goal is the automatic extraction of meaningful and representative knowledge from the raw data streams in a particular domain of consideration [15].

J. C. Figueroa-García et al. (Eds.): WEA 2020, CCIS 1274, pp. 24–34, 2020.
https://doi.org/10.1007/978-3-030-61834-6_3

Inducing classification rules from data is a classical Machine Learning problem. Most approaches follow a sequential covering strategy in order to come up with and then refine the rule base. They do this by working on a training set consisting of objects that are characterized through a list of conditional attributes (that represent the antecedents of the ensuing rules). These methods use the conditional attributes to arrive at a conclusion regarding the decision (class/label) attribute, often expressed at the consequent of rules. On the other hand, uncertainty management as well as knowledge representation, remain pivotal issues for Artificial Intelligence. Uncertainty is very pervasive and inherent to the real world and may emerge in multiple fashions caused by: (i) inaccurate information due to wrong measurements or transmission errors; (ii) incomplete information owing to forbidden or expensive access to data sources and (iii) ambiguous concept descriptions.

Bayesian reasoning emerged in the 1980s as a probabilistic model for reasoning with uncertainty in Artificial Intelligence and in just a few years, became popular. Different successful applications of Bayesian reasoning have been reported in fields as medicine, information retrieval, computer vision, information fusion, agriculture and so on.

In this study we employ Bayesian reasoning to improve the performance of the a recently proposed rule induction algorithm, named IRBASIR, by rewriting the aggregation function from a Bayesian angle. The new method, IRBASIR-Bayes, shows superior classification accuracy in comparison to other well-known rule induction methods. The rest of the paper is structured as follows. Section 2 elaborates on the technical background while Sect. 3 focuses on the application of Bayesian concepts to the IRBASIR algorithm. The empirical analysis is reported in Sect. 4. Conclusions and further remarks are stated in Sect. 5.

2 Technical Background

In this section we briefly review some foundational concepts in this study such as Bayesian networks as well as the Naïve-Bayes and IRBASIR algorithms.

2.1 Bayesian Networks

An increasing number of studies have to cope with a large number of variables that exhibit complex relationships among them. Bayesian Networks, (BNs) are a class of probabilistic networks that model precisely these interactions. Probabilistic networks are graphical representations of a problems variables and their relationships [14]. Simply put, BNs have a topological/structural component and a parametric/numerical component. The former is realized via a Directed Acyclic Graph (DAG) that encodes the qualitative knowledge of the model through probabilistic relationships of conditional dependence and independence. This knowledge is articulated in defining these dependence/independence relationships among the model variables. These relationships range from complete independence to a functional dependence. The fact that the model is specified in a

graph-based fashion makes BNs really attractive among other peer knowledge representation formalisms. BNs not only qualitatively model the domain knowledge but also quantitatively express the "strength" of the relationships among the variables by means of probability distributions as the extent of the belief we hold on the underlying relations among the model variables.

A Directed Acyclic Graph is composed of nodes denoting the problem variables, which could be descriptive features or attributes. Each pair of nodes is connected by directed edges. These edges represent probabilistic dependencies between the variables. In terms of probabilities, linking X to Y means there is a conditional dependence of Y with respect to X, i.e., $Y's$ probability distribution is different from that of Y given X. These variables will be linked to the decision class that would act as the root variable. The parametric component of a BN take the form of Conditional Probability Tables (CPTs) established from the information encoded in the DAG. The Bayes theorem is given by Eq. 1,

$$P(h|O) = \frac{P(O|h)P(h)}{P(O)} \tag{1}$$

where h is a hypothesis, O is an observation (or set of observations) and $P(hO)$, $P(Oh)$ are conditional probabilities. The latter is the *likelihood* that hypothesis h may have produced the set of observations $O.P(h)$ is called the *prior* probability and represents the initial degree of belief in h. For a classification problem with the class variable C and set of input attributes $A = \{A_1, A_2, \cdots, A_n\}$ the Bayes theorem adopts the form in Eq. 2,

$$P(c|A) = \frac{P(A|c)P(c)}{P(A)} \tag{2}$$

What we want is to identify the most plausible decision class c_* in the set $C = \{c_1, c_2, \cdots, c_k\}$ for an observation O that needs to be classified. In the Bayesian framework, the most plausible hypothesis is the one with the maximum a posteriori (MAP) probability. In other words, the decision class c_* to be assigned to O is computed by means of Eq. 3,

$$c_* = \arg\max_{c \in C} \{P(A|c)P(c)\} \tag{3}$$

Notice that the denominator in Eq. 2 has not been included in Eq. 3 for simplification purposes. Bayes' theorem hence provides a neat and interpretable way to solve a classification problem.

2.2 Naïve-Bayes

A Bayesian classifier that is usually quite accurate despite its simplicity is known as Naïve-Bayes (NB) [11]. NB is actually one of the most widely used methods among the BN family of classification techniques.

The core of the NB classifier is the underlying assumption that all attributes are independent given the value of the class variable. NB's name stems from the

assumption that predictive variables are conditionally independent given the decision variable; this assumption gives rise to a simplified form of Eq. 2, hence we will only need to learn the conditional probabilities of the attributes given the class values [15]. The simplified expression used by NB for classification purposes is given in Eq. 4,

$$c_* = \arg\max_{c \in C} \prod_{i=1}^{n} P(A|c) \tag{4}$$

This procedure is known as *maximum likelihood estimation*. Unfortunately, it requires a large sample size and tends to overfit the data. More complex estimators like Laplace succession are applied to counter this limitation.

NB deals with the numerical data assuming all attributes are generated from different normal or Gaussian probability distributions. The mean and standard deviation for each decision class and numeric attribute is then calculated [15]. The probability density function for a normal distribution with mean μ and standard deviation σ is given by Eq. 5,

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5}$$

NB is one of the strongest and most popular classifiers. Several studies confirm that its classification results are competitive with regards to, or even surpass, those produced by other models such as decision trees, neural networks, etc.

2.3 IRBASIR

IRBASIR (Induction of Rules BAsed on SImilarity Relations) is an algorithm [4, 5, 7] for the automatic generation of classification rules in decision systems that may contain both nominal and numerical attributes. The algorithm leans upon the learning of similarity relations [3] for building similarity classes of objects, which is accomplished by extending the canonical Rough Set Theory (RST). The overall similarity relation learned from data encompasses attribute-wise similarity functions for both nominal and numerical descriptors. Algorithm 1 depicts the operational workflow of the IRBASIR algorithm introduced in [7].

Algorithm 1. The IRBASIR Algorithm

Input : a decision system $D = (U; A \cup \{d\})$ with $|U| = m \neq 0$ objects and $|A| = n \neq 0$ attributes
Output : a set of classification rules

1: Define the set of attribute-wise similarity functions
2: Build the similarity relation R
3: Generate classification rules and their associated confidence values by using $GenRulesRST()$

In step 1, we need to specify the set of functions $\delta_i(x, y)$ that determine the similarity between two objects x and y based on their values for the attribute A_i.

Equations 6 show the expressions used in [7] to deal with numerical and nominal attributes, respectively,

$$\partial_i(x,y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are real and } |x_i - y_i| \leq \varepsilon \\ & \text{or } x \text{ and } y \text{ are discrete and } x = y \\ 0 & \text{otherside} \end{cases} \tag{6}$$

To build the similarity relation R Eq. 7 are employed,

$$xRy \text{ if and only if } F_1(x,y) \geq \varepsilon \text{ and } F_2(x,y) = 1 \tag{7}$$

where ε denotes a threshold. For the experiment the value of $\varepsilon = 0.85$ was used for all datasets suggested by [8]. The functions F_1 and F_2 are defined by Eqs. 8 and 9:

$$F_1(x,y) = \sum_{i=1}^{n} w_i * \partial_i(x,y) \tag{8}$$

$$F_2(x,y) = \begin{cases} 1 & \text{if } class(x) = class(y) \\ 0 & \text{otherside} \end{cases} \tag{9}$$

where n is the number of features, w_i is the weight of feature i calculated according to the method proposed in [5,6] and ∂_i is a features comparison function which calculates the similarity between the values of objects x and y with respect to the feature instances i, is defined by expression 10,

$$\partial_i(x_i,y_i) = \begin{cases} 1 - \dfrac{|x_i - y_i|}{\max(D_i) - \min(D_i)} & \text{if } i \text{ is continuous} \\ 1 & \text{if } i \text{ is discrete and } x_i = y_i \\ 0 & \text{if } i \text{ is discrete and } x_i \neq y_i \end{cases} \tag{10}$$

where D_i is the domain of feature i.

The third step is the automatic learning of classification rules based on Algorithm 2. This procedure is to generate classification rules and it's *certidumbre value.*

In the pseudo code of Algorithm 2, $[x_i]_R$ returns the set of decision classes of all objects in a set passed as input argument. Upon calculation of the similarity class $[x_i]_R$ for an object x_i, if the objects in the similarity class all share the same value of the decision class, a rule for that decision class is created; otherwise, the most frequent class in that object set is taken and a rule is created for it.

Then the objects in $[x_i]_R$ are all flagged as used and the rule generation process carries on. The *GenRulSim()* procedure is given in Algorithm 3.

Algorithm 2. The GenRulesRST Algorithm

Input : a decision system $D = (U; A \cup \{d\})$ with $|U| = m \neq 0$ objects and $|A| = n \neq 0$ attributes
Output : a set of classification rules (ruleBase)

1: objectUsed[j]← false $\forall j \in \{1, \ldots, m\}$;
2: $ruleBase \leftarrow \emptyset$;
3: i ← index of the first unused object;
4: **if** i==0 **then**
5: stop; return ruleBase
6: **else**
7: objectUsed[i]← true;
8: compute similarity class $[x_i]_R$;
9: rule ← $GenRulSim(k, [x_i]_R, C)$;
10: ruleBase ← ruleBase ∪ rule;
11: objectUsed[j]← true $\forall j : x_j \in C \bigwedge d(x_j) == k$
12: Go to line 3;

Algorithm 3. The GenRulesSim Algorithm

Input : decision class k, similarity class object set O_s, objects in O_s with class O_s^k
Output : a classification rule as well as its accuracy and coverage

1: create a vector p for the objects in O_s^k, $p(i) = f(V_i[O_s^k]); \forall i \in \{1, \ldots, n\}$
2: generate the rule from the vector p, rule ← IF $w_1\partial_1(x_1, p_1) + \cdots + w_n\delta_n(x_n, p_n) \geq \varepsilon$ THEN $d \leftarrow k$;
3: compute the rule accuracy $acc(rule) \leftarrow \dfrac{|A(rule) \cap O_s|}{|A(rule)|}$;
4: compute the rule coverage $cov(rule) \leftarrow \dfrac{|A(rule) \cap O_s|}{|O_s|}$;
5: return $\langle rule, acc(rule), cov(rule)\rangle$;

The term $V_i[O_s^k]$ in Step 1 denotes the set of values of the attribute V_i for those objects belonging to the set O_s^k. The function $f(V_i[O_s^k])$ aggregates the set of values in $V_i[O_s^k]$ depending on the type of the attribute V_i, e.g., the mode in case of V_i being a nominal attribute or the mean/median if it is a numerical attribute.

Hence, the vector p could be thought of as a prototype (or centroid) for all objects in O_s^k. Additionally, the weight vector $(w_1, \ldots, w_n)$, the similarity threshold ε and the attribute-wise similarity functions δ_i and come from Eqs. 7 and 8.

3 The IRBASIR-Bayes Algorithm

In order to improve the performance of the IRBASIR algorithm, a modification in Step 1 of the *GenRulSim()* method is introduced. The idea is to vary the aggregation function $f(\cdot)$ to find the vector p with n components for the set of reference objects in O_s^k, from which the rules are generated.

The main idea of the proposed IRBASIR-Bayes approach is to rewrite the function $f(\cdot)$ by switching from a purely statistical aggregation method, such as the mean or the mode of a set, to a probabilistic scheme like the one used in the NaïveBayes classifier.

Each component $p(i), i = 1, \cdots, n$, of the reference (central tendency) vector p is now calculated as shown in Eq. 11 if V_i is a nominal attribute and by means of Eq. 11 if V_i is a numerical attribute,

$$p_{(i)} = \arg \max_{v \in V_i[O_s^k]} \left\{ \frac{|x \in O_s^k : x_i == v|}{O_s^k} \right\} \tag{11}$$

$$p_{(i)} = \arg \max_{v \in V_i[O_s^k]} \left\{ \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right\} \tag{12}$$

In the nominal/discrete case portrayed in Eq. 11, the estimation of the conditional probability is based on the frequencies with which the different values for the attribute V_i appear in the objects of the set O_s^k. The reader may notice that this approach is equivalent to the calculation of the mode of V_i over O_s^k. Semantically, this means that the vector component $p(i)$ will take the most frequently occurring value in that attribute for the objects under consideration. Therefore, the reference vector p is made up by the most frequent values of all nominal attributes in the decision system.

When it comes to numerical attributes, however, the NB method assumes they follow an underlying Gaussian probability distribution; hence, the first step is the calculation of the sample mean μ_i and the sample standard deviation σ_i for the objects in the similarity class O_s^k. Then, Eq. 12 applies to each numeric value x of the attribute V_i over the objects in O_s^k. From a semantic standpoint, this means that the representative value for that attribute captured by the vector component $p(i)$ will be the one with the highest probability distribution value among all other values of V_i for the objects under consideration.

4 Experimental Results

For the empirical analysis, we relied on 13 UCI Machine Learning Repository data sets[1] in .ARFF format. They were used to gauge the efficacy and efficiency of the algorithms under consideration. Table 1 lists the data sets.

We will empirically compare the classification accuracy of the following classifiers: C4.5 [13], EXPLORER [10], MODLEM [10], LEM2 [9], IRBASIR [4] and IRBASIRBayes for each of the datasets using 10-fold cross validation. Table 2 reports these results. To calculate the measurement accuracy there was used the measure that is typically used in these cases, as shown in expression 13:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

where: TP (True Positives), TN (True Negatives), FP (False Positives) and FN (False Negatives).

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html.

Table 1. Description of the datasets used in the experiments.

Datasets	#Instances	#Attributes
ecoli	336	7
iris	150	4
pima	768	8
heart-statlog	270	13
balance-scale	625	4
haberman	306	3
biomed	194	8
breast-w	699	9
wisconsin	699	9
optdigits	5620	64
waveform	5000	21
diabetes	768	8
vehicle	845	18

Table 2. Classification accuracy results for all rule induction algorithms

Datasets	IRBASIR-Bayes	IRBASIR	C4.5	LEM2	EXPLORE	MODLEM
ecoli	**80.14**	80.09	79.47	52.97	0.6	75.7
iris	**96.06**	96	94.67	94	88	94
pima	**75.29**	75.26	75.26	65.89	60.29	74.23
heart-statlog	**82.04**	79.63	77.78	75.19	81.85	75.19
balance-scale	**85.95**	84.95	77.9	83.67	84.78	80.47
haberman	**73.25**	72.87	70.97	71.48	67.91	70.27
biomed	**88.14**	87.16	86.53	66.5	66.63	85
breast-w	**84.7**	80.46	75.1	59.82	0	70.26
wisconsin	**96.25**	95.15	94.57	95.16	89.13	93.99
optdigits	**96.51**	96.3	95.56	95.75	89.41	94.28
waveform	93.88	**93.98**	90	79.59	0	78.72
diabetes	75.19	**75.66**	74.62	64.83	59.9	75.27
vehicle	**72.01**	71.78	71.81	49.76	58.27	53.66

To statistically validate the results, we employed non-parametric multiple-comparison tests in order to identify the best algorithm. The Friedman test is the non-parametric standard test for comparison among several classifiers. This test works by assigning ranks in ascending order, i.e., the first rank for the best obtained results, the second rank for the second best result and so on [2].

The null hypothesis states that all algorithms behave similarly, so the ranks should be similar. In the experimental study conducted in this investigation, the p-value found using the Friedman statistic was zero, while the critical point for a Chi-square distribution with five degrees of freedom was 44.241758. As the value obtained by the Friedman statistic is much lower than this latter value, there is enough evidence to reject the null hypothesis and conclude that there are indeed statistically significant differences among the groups. Table 3 displays the ranks produced by the Friedman test upon the set of algorithms under comparison, with IRBASIS-Bayes topping the list.

Table 3. Friedman ranks for each algorithm

Algorithms	Ranks
IRBASIR-Bayes	1.2308
IRBASIR	2.1154
C4.5	3.5
LEM2	4.4615
MODLEM	4.4615
EXPLORER	5.2308

Table 4. Holm post-hoc procedure results with IRBASIR-Bayes as the control method

Algorithm	$z = (R_0 - R_i)/SE$	p	Holm	Hypothesis
EXPLORER	5,451,081	0	0.01	Rejected
LEM2	4,402,796	0.000011	0.0125	Rejected
MODLEM	4,402,796	0.000011	0.016667	Rejected
C45	309,244	0.001985	0.025	Rejected
IRBASIR	1,205,528	0.228	0.05	Rejected

Table 4 displays the results of the Holm post-hoc procedure used to compare the control algorithm (IRBASIR-BAYES) with the rest at a 5% significance level. The test statistic for comparing the ith algorithm and jth algorithm, z, depends on the main nonparametric procedure used; where R_0 and R_i are the average rankings by the Friedman test of the algorithms compared [1]. Holm test rejects those hypotheses having a p-value ≤ 0.05. The test rejects all cases in favor of the best-ranked algorithm; hence we can claim that the proposed algorithm is statistically superior to the other ones in terms of classification accuracy.

5 Conclusions

The IRBASIR algorithm allows inducing classification rules for mixed data (i.e., numerical and nominal attributes). It is characterized by not requiring the discretization of the numerical attributes, neither as a preprocessing step nor during

the learning process. IRBASIR generates the rules from a reference or prototype vector that allows constructing similarity classes for the objects in the training set. This vector is simply calculated as the mean/median for each numerical attribute.

IRBASIR-Bayes is a spin-off from IRBASIR that alters the way in which prototype vectors are constructed, namely by switching from the mean/median statistical approach to a probabilistic approach through the classification functions used by Naïve-Bayes. IRABASIR-Bayes has the disadvantage of assuming normality and independence among data. Nevertheless, the results obtained with 13 publicly available Machine Learning repositories evidence a statistically verified superiority in terms of classification accuracy to other well-known rule induction algorithms (C4.5, MODLEM, EXPLORER, LEM2, and IRBASIR itself); another benefit is that the method allows processing attributes with missing values. Future work will concentrate on the parametric tuning of IRBASIR and IRBASIR-Bayes so as to obtain even higher accuracy marks.

References

1. Daniel, W.: Applied Nonparametric Statistics, 2nd edn. Duxbury Thomson Learning, Pacific Grove (2000)
2. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**(1), 3–18 (2011)
3. Fernandez, Y., Coello, L., Filiberto, Y., Bello, R., Falcon, R.: Learning similarity measures from data with fuzzy sets and particle swarms. In: 2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1–6 (2014)
4. Filiberto, Y., Bello, R., Caballero, Y., Frias, M.: Algoritmo para el aprendizaje de reglas de clasificacion basado en la teoria de los conjuntos aproximados extendida. Dyna **78**(169), 62–70 (2011)
5. Filiberto, Y., Bello, R., Caballero, Y., Frias, M.: An analysis about the measure quality of similarity and its applications in machine learning. In: Fourth International Workshop on Knowledge Discovery, Knowledge Management and Decision Support (2013)
6. Filiberto, Y., Bello, R., Caballero, Y., Larrua, R.: Using PSO and RST to predict the resistant capacity of connections in composite structures. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol. 284, pp. 359–370. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12538-6_30
7. Filiberto, Y., Caballero, Y., Larrua, R., Bello, R.: A method to build similarity relations into extended rough set theory. In: 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 1314–1319 (2010)
8. Fras, M., Filiberto, Y., Fernndez, Y., Caballero, Y., Bello, R.: Prototypes selection based on similarity relations for classification problems, October 2015
9. Grzymala-Busse, J.W.: A new version of the rule induction system LERS. Fundamenta Informaticae **31**(1), 27–39 (1997)

10. Grzymala-Busse, J.W.: Mining numerical data – a rough set approach. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets XI. LNCS, vol. 5946, pp. 1–13. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11479-3_1
11. Keerthika, G., Priya, D.S.: Feature subset evaluation and classification using Naive-Bayes classifier. J. Netw. Commun. Emerg. Technol. (JNCET), **1**(1) (2015). www.jncet.org
12. Kodratoff, Y., Michalski, R.S.: Machine Learning: An Artificial Intelligence Approach. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-12405-5
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Mateo (1993)
14. Whittaker, J.: Graphical Models in Applied Multivariate Statistics. Wiley, Chichester (2009)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, San Francisco (2005)