A Study of the Influence of Rule Measures in Classifiers Induced by Evolutionary Algorithms

Claudia Regina Milaré, Gustavo E.A.P.A. Batista, André C.P.L.F. de Carvalho

***Abstract*—The Pittsburgh representation is a well-known encoding for symbolic classifiers in evolutionary algorithms, where each individual represents one symbolic classifier, and each symbolic classifier is composed by a rule set. These rule sets can be interpreted as *ordered* or *unordered* sets. The major difference between these two approaches is whether rule ordering defines a rule precedence relationship or not. Although ordered rule sets are simple to implement in a computer system, the rule set is difficult to be interpreted by human domain experts, since rules are not independent from each other. In contrast, unordered rule sets are more flexible regarding their interpretation. Rules are independent from each other and can be individually presented to a human domain expert. However, the algorithm to decide a classification of a given example is more complex. As rules have no precedence, an example should be presented to all rules at once and some criteria should be established to decide the final classification based on all fired rules. A simple approach to decide which rule should provide the final classification is to select the rule that has the best rating according to a chosen quality measure. Dozens of measures were proposed in literature; however, it is not clear whether any of them would provide a better classification performance. This work performs a comparative study of rule performance measures for unordered symbolic classifiers induced by evolutionary algorithms. We compare 9 rule quality measures in 10 data sets. Our experiments point out that confidence (also known as precision) presented the best mean results, although most of the rule quality measures presented approximated classification performance assessed with the area under the ROC curve (AUC).**

***Index Terms*—Symbolic classification, evolutionary algorithm, rule quality measures.**

I. Introduction

Evolutionary Algorithms (EAs) have been successfully applied to solve problems in a large number of domains. One of their most prominent features is to perform a global search using multiple candidate solutions, and therefore increasing the possibilities of finding an optimal solution [1]. In contrast, induction of symbolic classifiers can be seen as a search problem in which some performance measure should be optimized, such as accuracy or coverage. Conventional symbolic inducers, for instance decision tree inducers, use a simple greedy search, and EAs present an attractive alternative to better search the hypothesis space.

The Pittsburgh representation [2] is a well-known encoding for symbolic classifiers in EAs, where each individual represents one symbolic classifier, and each symbolic classifier is composed by a rule set. These rule sets can be interpreted as *ordered* or *unordered* sets. In the case of an ordered rule set, rule ordering defines a precedence relationship. For instance, when an example to be classified is presented to an ordered rule set, rules must be analyzed regarding their position in the set. The final classification is given by the class predicted by the *first* rule that covers the example. Although this approach is very simple to implement in a computer system, the rule set is difficult to be interpreted by human domain experts, since rules are not independent from each other. The knowledge expressed in a rule only holds if all preceding rules were not fired.

In contrast, unordered rule sets are more flexible regarding their interpretation. Rules are independent from each other and can be individually presented to a human domain expert. However, the algorithm to decide the classification of a given example is more complex. As rules have no precedence, an example should be presented to all rules at once and some criteria should be established to decide the final classification based on all fired rules. A simple approach to decide which rule should provide the final classification is to select the rule that has the best rating according to a chosen quality measure. Dozens of these measures were proposed in literature. However, it is not clear whether any of them would provide a better classification performance.

This work performs a comparative study of rule performance measures for unordered symbolic classifiers induced by EAs. We compare 9 rule quality measures in 10 data sets. In our experiments, we use the area under the ROC curve (AUC) [3] as the main measure to assess our results. AUC has several advantages over other conventional measure such as error rate and accuracy [4], for instance, AUC is independent from class prior probabilities. Our experiments point out that confidence (also known as precision) presented the best mean results, although most of the rule quality measures presented approximated classification performance.

This paper is organized as follows: Section II describes our EA; Section III presents the rule quality measures used in our experiments; Section IV empirically compares the measures on 10 application domains; and finally, Section V concludes this paper and presents some directions for future work.

II. Our Evolutionary Algorithm

There are two major approaches to represent decision rules as individuals in an EA. These approaches are namely Michigan and Pittsburgh [5]. In short, in Michigan [6] approach each individual codifies only one rule, and in Pittsburgh [2]

C. R. Milaré is with Centro Universitário das Faculdades Associadas de Ensino, São João da Boa Vista, Brazil, e-mail: cmilare@gmail.com.

G. E. A. P. A. Batista and A. C. P. L. F. de Carvalho are with Instituto de Ciências Matemáticas e de Computação – ICMC, Universidade de São Paulo – USP, São Carlos - SP, Brazil, e-mail: {gbatista, andre}@icmc.usp.br.

approach each individual codifies a classifiers, i.e. a rule set. This difference is more than a simple technical detail. Michigan approach is used when we are interested in a single rule with a determined propriety, such as accuracy. Even though the final population has several rules, those rules usually do not have a collective property, such as complementary coverage. Therefore, Michigan approach is frequently used to induce descriptive rules.

In contrast, as Pittsburgh representation codifies a rule set in each individual, a search is performed to optimize some collective property. Thus, Pittsburgh representation is commonly used to induce predictive classifiers, since such classifiers have to combine rules that are individually predictive and collectively complementary, in a way that a large number of examples is covered and correctly classified.

As previously stated, we use the Pittsburgh representation in our EA. One possible criticism regarding this representation is that no search is performed at rule level, i.e., rules are not improved by the search procedure. One possible strategy is to combine Michigan and Pittsburgh in a hybrid representation [7], [8]. However, this approach increases considerably the search space and doubles the number parameters. As consequence, this approach is more computationally intensive, its results are more difficult to analyze (due the larger number of parameters that should be tuned), and more important, the larger search space increases considerably changes of overfitting training data.

In order to use the Pittsburgh representation over a set of predictive rules, we use the Ripper rule induction algorithm [9] to generate an initial rule set. However, for most data sets, the rule set induced by Ripper is usually of restricted number of rules. Thus, we use a bootstrapping sampling strategy to generate multiple training sets. This sampling strategy allow us to increase the number and diversity of the rules.

In more details, in our experiments we use the k-fold stratified cross-validation resampling method for generating k training set $\mathcal{T}_1, \mathcal{T}_2, \ldots \mathcal{T}_k$ and their correspondent test sets from each data set. Next, n bootstrapping samples *with replacement* $\mathcal{T}_{i1}, \mathcal{T}_{i2}, \ldots \mathcal{T}_{in}$ are created from each training set $\mathcal{T}_i, 1 \leq i \leq k$. Each bootstrapped sample has the same number of examples as its corresponding training set, i.e., $|\mathcal{T}_{ij}| = |\mathcal{T}_i|$, $1 \leq j \leq n$.

Each bootstrapped training set $\mathcal{T}_{ij}$ is given as input to Ripper algorithm and n rule sets are induced $\mathcal{R}_{i1}, \mathcal{R}_{i2}, \ldots \mathcal{R}_{in}$. All rule sets are integrated into a unique pool of rules, and the repeated rules are discarded. Figure 1 illustrates this sampling approach.

In a second step, all rules are given as input to our EA. Internally, each rule is associated with a unique identifier. An individual, i.e., a rule set is represented as a set of rule identifiers. Finally, the population is a table that contain all sets of individuals. This representation scheme is very convenient, since conventional evolutionary operations, such as mutation and crossover, can be readily implemented as simple manipulations of the population table. Figure 2 illustrates this representation scheme.

In this work, we analyze how different rule quality measures might influence the classification performance of a rule set

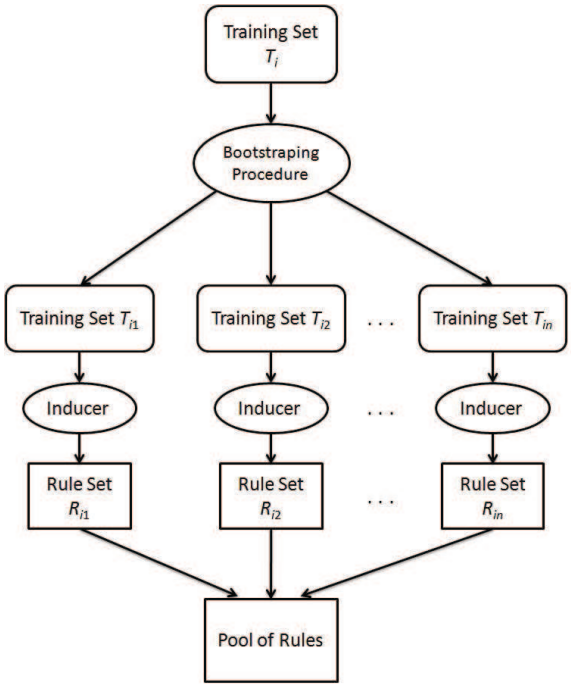

Fig. 1. Approach used to generate multiple rule sets using bootstrapping samples.

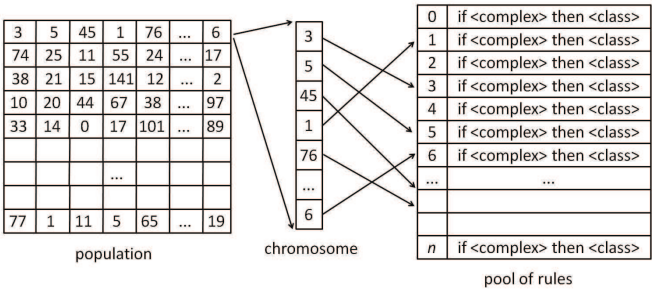

Fig. 2. An entire population represented as a table where each individual is a row (*left*). Each individual is a set of rule identifiers (*middle*). Each rule identifier corresponds to an unique rule in the pool (*right*).

searched by an EA. Given a classifier $\mathcal{C}$, such as the one represented by the individual in Figure 2-*middle*, and an example e to be classified, several rules of $\mathcal{C}$ might cover the example e. Although different classes may be predicted by the fired rules, we want to choose one rule that will provide the final classification. In the next section, we review some popular rule quality measures. These measures are used to decide which rule will provide the class of e.

III. Rule Measures

A *classification rule* is an intelligible representation of a piece of knowledge. A rule R is given in the form

$$B \rightarrow H$$

where B, called *body*, is a conjunction of conditions and H, called *head*, is the class value predicted by the rule.

Given a rule R and an example e, R *covers* e if all conditions of B are verified true in e. A rule *correctly* covers

TABLE I
CONTINGENCY MATRIX.

	H	$\overline{H}$	
B	f_{bh}	$f_{b\bar{h}}$	f_b
$\overline{B}$	$f_{\bar{b}h}$	$f_{\bar{b}\bar{h}}$	$f_{\overline{b}}$
	f_h	$f_{\overline{h}}$	1

an example if the rule covers the example and correctly predicts its class.

When a rule is evaluated against a data set, the examples may be distributed along four sets, B, $\bar{B}$, H and $\bar{H}$. Examples covered by a rule belong to B, while examples having the same class as predicted by the rule belong to H. Their complements, $\bar{B}$ and $\bar{H}$ contain the examples *not covered* and the examples *incorrectly* predicted by the rule. The corollary intersections contain examples correctly covered, incorrectly covered, not covered but correctly predicted, and finally, not covered and incorrectly predicted. These sets are important to construct the rule's *contingency matrix* (see Table I), which is the basis of the Lavrač framework [10].

We use the notation f_{xy} to denote the empirical frequency of an event $x \in \{B, \bar{B}\}$ and an event $y \in \{H, \bar{H}\}$. Therefore, f_{xy} in an empirical estimate of the probability $p(x, y)$. For sake of completeness, we describe all empirical frequencies as follows:

- f_{bh} is the percentage of examples covered and correctly classified by a rule R;
- $f_{b\bar{h}}$ is the percentage of examples covered and incorrectly classified by a rule R;
- $f_{\bar{b}h}$ is the percentage of examples not covered by R, but the class predicted by R is the same class of the example;
- $f_{\bar{b}\bar{h}}$ is the percentage of examples not covered by R, and the class predicted by R is different from the class of the example.

The marginal frequencies f_b, $f_{\bar{b}}$, f_h, $f_{\bar{h}}$ are defined as:

- f_b is the percentage of examples covered by R;
- $f_{\bar{b}}$ is the percentage of examples not covered by R;
- f_h is the percentage of examples correctly classified by R, independently if R covers or not the examples. It is also the prior probability estimate of the class predicted by the rule;
- $f_{\bar{h}}$ is the percentage of examples incorrectly classified by R, independently if R covers or not the examples.

The Lavrač framework allows to define different rule measures under a same organization. In this work, we analyze the influence of 9 rule measures listed in Table II. We briefly describe each measure as follows:

1) **Confidence**, also known as *precision* or *strength*, is the probability that a rule R will provide a correct prediction given that it covered the example. In practice, is probability might be very high for rules that cover a restricted number of examples;
2) **Laplace** is the confidence measure with Laplace correction. Laplace correction is frequently used to improve probabilities estimates when data are scarse. This implementation of Laplace approximates the estimated probability to 0.5 as fewer examples are covered by the rule;
3) **Lift** measures the confidence of R relative to the prior probability of the class predicted by R. This measure is based on the idea that an useful rule should have a confidence higher than a default rule that always predicts the same class;
4) **Conviction** is similar to lift since it also relates confidence with class prior probability. However, conviction is very sensitive to the confidence of a rule. Rules with a confidence value of 1, which it is not rare for low coverage rules, will have an infinite conviction;
5) **Leverage**, also known as *Piatetsky-Shapiro's* measure, is derived from the concept of statistical independence. If two events x and y are independent, then $p(x, y) = p(x) \times p(y)$. Leverage measures how much f_{bh} deviates from $f_b \times f_h$, i.e., the probability estimate assuming the events b and h independent. It is expected that an useful rule has a confidence higher than the prior probability of the class that it predicts, i.e., $\frac{f_{bh}}{f_b} > f_h$. Therefore, we should look for rules that $f_{bh} > f_b \times f_h$;
6) $\mathcal{X}^2$ is a well-known statistical test of independence. It is used to measure the independence between the rule antecedent and consequent. It is closely related to ϕ-**coefficient**, but it takes in account the number of instances in the data set (N);
7) **Jaccard** is a measure of overlapping between the number of cases covered by the rule and the number of cases that belongs to the predicted class. This measure has maximum value $f_b = f_h = f_{hb}$, i.e., when the rule covers all examples of the predicted class and none example from other classes. In contrast, its minimum value is given when $f_{bh} = 0$, i.e., when the rule misclassify every case it predicts;
8) **Cosine** is frequently used in text mining to measure the similarity between two vectors of attributes. For scalar values, cosine is similar to Jaccard measure and assume its minimal and maximal values in the same conditions;
9) ϕ-**coefficient** is a statistical measure of association between two binary variables. This measure is related to the Pearson correlation coefficient and also to the $\mathcal{X}^2$ measure. Since $\phi = \frac{\mathcal{X}^2}{N}$ [12], where N is the number of data instances, the ϕ-coefficient is independent from the data set size.

IV. EXPERIMENTAL EVALUATION

We carried out a number of experiments to evaluate the influence of each rule quality measure in the performance of symbolic classifiers searched by the evolutionary algorithm. The experiments were performed using 10 benchmark data sets, collected from the UCI repository [13]. In addition, we used AUC as the main measure to assess our results. Table III summarizes the main features of these data sets, which are: Identifier – identification of the data set used in the text; #Examples – the total number of examples; #Attributes (quanti., quali.) – the total number of attributes, as well as the number of quantitative and qualitative attributes; Classes (min.,

TABLE II
SUMMARY OF RULE QUALITY MEASURES (ADAPTED FROM [11]).

Measure	Definition	Range
Confidence	$conf = \frac{f_{bh}}{f_b}$	0...1
Laplace	$lapl = \frac{f_{bh}+1}{f_b+2}$	0...1
Lift	$lift = \frac{conf}{f_h}$	$0...+\infty$
Conviction	$conv = \frac{1-f_h}{1-conf}$	$0.5...1...+\infty$
Leverage	$leve = f_{bh} - (f_b \times f_h)$	$-0.25...0...0.25$
$\mathcal{X}^2$	$\mathcal{X}^2 = N \times \sum_{x\in\{b,\bar{b}\},y\in\{h,\bar{h}\}} \frac{(f_{xy} - f_x \times f_y)^2}{f_x \times f_y}$	$0...+\infty$
Jaccard	$jacc = \frac{f_{bh}}{f_b + f_h - f_{bh}}$	0...1
Cosine	$cos = \frac{f_{bh}}{\sqrt{f_b \times f_h}}$	$0...\sqrt{f_{bh}}...1$
ϕ-coefficient	$\phi - coeff = \frac{leve}{\sqrt{f_b \times f_h \times f_{\bar{b}} \times f_{\bar{h}}}}$	$-1...0...1$

TABLE III
DATA SETS DESCRIPTION.

Identifier	#Examples	#Attributes (quanti., quali.)	Classes (min., maj.)
Blood	748	4 (4, 0)	(1, 0) (24.00%, 76.00%)
Breast	699	10 (10, 0)	(benign, malignant) (34.99%, 65.01%)
Bupa	345	6 (6, 0)	(1, 2) (42.02%, 57.98%)
CMC	1473	9 (2, 7)	(1, remaining) (42.73%, 57.27%)
Flare	1066	10 (2, 8)	(C-class, remaining) (17.07%, 82.93%)
Haberman	306	3 (3, 0)	(2, 1) (26.47%, 73.53%)
New-Thyroid	215	5 (5, 0)	(remaining, 1-normal) (30.23%, 69.77%)
Pima	768	8 (8, 0)	(1, 0) (34.89%, 65.11%)
Vehicle	946	18 (18, 0)	(van, remaining) (23.89%, 76.11%)
Yeast	1484	8 (8, 0)	(NUC, remaining) (28.90%, 71.10%)

maj.) % (min., maj.) – the label of the minority and majority classes and the percentage of minority and majority classes. In order to measure the performance of the classifiers using AUC, data sets with more than two classes were transformed in binary classification problems by selecting one of the classes as minority/majority class (as indicated in column Classes) and assigning the examples from the other classes to the majority/minority class.

As previously described, we used the 10-fold stratified cross-validation resampling method for generating training and their correspondent test sets. In addition, 30 bootstrapping samples with replacement were created for each training set. We empirically chose the number of 30 bootstrapping samples since it allowed to create a diverse pool of rules. Increasing this number did not improve our results, but increased the training times.

Each bootstrapped training set was given as input to the Machine Learning algorithm Ripper. The rules from all rule sets were integrated into a unique pool of rules, and the repeated rules were discarded. Next, the pool of rules was given as input to the evolutionary algorithm that outputted a final rule set (a classifier). Finally, AUC was measured over the test set.

TABLE IV
CHROMOSOME SIZE FOR EACH DATA SET BASED ON THE MEAN NUMBER OF RULES PROVIDED BY RIPPER.

Data set	Chromosome size
Blood	6
Breast	6
Bupa	8
CMC	12
Flare	6
Haberman	4
New-Thyroid	4
Pima	10
Vehicle	6
Yeast	8

Our evolutionary algorithm was set to use 40 chromosomes in all experiments. The chromosome size, i.e., the number of rules of an individual classifier was defined according to the average size of the classifiers generated by Ripper in each data set. In the Pittsburgh approach, it is a commonsense to allow variable-sized chromosomes. Therefore, the evolutionary algorithm is free to search for rule sets with different number of rules using a two-point cross-over operator. In our case, we noticed that a search with variable-sized chromosomes resulted in very large rule sets for most domains. These large rule sets had a poor performance in the test set, indicating overfitting. Therefore, we opted to keep all chromosomes with fixed sizes. The chromosome size chosen for each data set is the mean number of rules induced by Ripper in the same data set. Table IV lists the chromosome sizes for each data set.

The fitness function used is the AUC metric measured over the training examples. The selection method is the fitness-proportionate selection. The crossover operator was applied with probability 0.4 and the mutation operator was applied

with probability 0.1. Mutation and crossover rates were chosen based on our previous experience with EAs [14], [15]. The number of generations was limited to 20. Our implementation uses an elitism operator to ensure that the best classifier is kept in the next population. Finally, since evolutionary algorithms perform a stochastic search that might provide different results in each execution, we repeated each experiment 10 times and averaged the results.

Table V presents results obtained. All results represent the mean AUC values calculated over the 10 pairs of training and test sets and averaged for 10 repeated executions. The standard deviations are also showed between parentheses. The second column shows the results obtained by Ripper for all data sets. The next columns show the results obtained by the evolutionary algorithm for each rule quality measure. The best AUC for each data set is emphasized in boldface. We can note that the EA search has improved considerably the AUC when compared with the results obtained by Ripper. However, the best AUC values are scattered throughout the table, indicating that no single measure systematically provides the best results.

Since no single measure provided the best results, we decided to rank the measures considering their mean AUC values. Table VI shows the results for this ranking. The second-to-last column of table shows the sum of ranks obtained by each measure for all the data sets. The last column shows a score based on the sum of ranks for each measure, in a way that the measure that has the lowest sum of ranks scores 1. The measure with lowest score is $confidence$. $Confidence$ obtained the better AUC values for Vehicle and Yeast data sets; the second better AUC values to Breast, CMC, Flare and New-Thyroid data sets; the third better AUC value to Bupa and Pima data set; the fourth better AUC value to Blood; and, the fifth better AUC value to Haberman data set.

In order to analyze whether there is a statistically significant difference among the compared measures, we ran the Friedman test[1]. The Friedman test was run with the null-hypotheses that the performance of all rule measures is comparable. When the null-hypothesis is rejected by the Friedman test, at 95% confidence level, we can proceed with a post-hoc test to detect which differences among the methods are significant. For such, we ran the Bonferroni-Dunn multiple comparisons with a control test.

The null-hypothesis was rejected by the Friedman test at 95% confidence level. So, we ran the Bonferroni-Dunn test using the measure $confidence$ as control. The Bonferroni-Dunn test indicate that the EA allied the measure $confidence$ outperforms Ripper with 95% confidence level. However, there are no statistically significant differences among the rule quality measures.

Our results differ from previously published results. To the best of our knowledge, the most similar work in literature is [11] which compares rule quality measures in the context of association rule classification. Their results indicate that $conviction$ presented the best results. Our experiments indicate that $confidence$ and $lift$ performed slightly better than $conviction$, but with no statistical difference. This difference between the results presented might be motivated by the use of different performance measures, since error rate was used in [11].

[1]The Friedman test is a nonparametric equivalent of the repeated-measures ANOVA. See [16] for a thorough discussion regarding statistical tests in Machine Learning research.

V. Conclusion

In this work, we compared 9 different rule quality measures in 10 different benchmark data sets. The rule measures were used to decide which rule should provide the final classification in an unordered rule set. Our results indicate that the use of different rule measures have a marginal effect over the classification performance assessed by the area under the ROC curve. The $confidence$ measure presented the best mean results, but with no statistical difference to the other rule measures.

As future work, we plan to investigate the use of different rule quality measures as a weighting factor in a voting approach in which all fired rules contribute to the final classification.

Acknowledgment

The authors would like to thank CNPq, CAPES and FAPESP, Brazilian funding agencies, for the financial support.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[2] S. F. Smith, "A learning system based on genetic adaptive algorithms," Ph.D. dissertation, Pittsburgh, PA, USA, 1980.

[3] R. D. J. A. Swets and J. Monahan, "Better decisions through science," *Scientific American*, 2000.

[4] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," HP Labs, Tech. Rep. HPL-2003-4, 2004.

[5] A. A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag, 2002.

[6] J. H. Holland, *Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems*, ser. Machine learning: An artificial intelligence approach. Morgan Kaufmann, 1986, vol. 2.

[7] D. P. Greene and S. F. Smith, "Competition-based induction of decision models from examples," *Machine Learning*, vol. 13, no. 2–3, pp. 229–257, 1993.

[8] A. Giordana and F. Neri, "Search-intensive concept induction," *Evolutionary Computation*, vol. 3, no. 4, pp. 375–416, 1995.

[9] W. Cohen, "Fast effective rule induction," in *International Conference on Machine Learning*, 1995, pp. 115–123.

[10] N. Lavrač, P. Flach, and R. Zupan, "Rule evaluation measures: A unifying view," in *Proceedings of the Ninth International Workshop on Inductive Logic Programming (ILP-99)*, vol. 1634. Springer-Verlag, 1999, pp. 74–185.

[11] P. J. Azevedo and A. M. Jorge, "Comparing rule measures for predictive association rules," in *18th European Conference on Machine Learning*, 2007, pp. 510–517.

[12] P.-N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," *Information Systems*, vol. 29, no. 4, pp. 293–313, 2004.

[13] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html

[14] C. R. Milaré, G. E. A. P. A. Batista, A. C. P. L. F. Carvalho, and M. C. Monard, "Applying genetic and symbolic learning algorithms to extract rules from artificial neural neworks," in *Proc. Mexican International Conference on Artificial Intelligence*, ser. LNAI, vol. 2972. Springer-Verlag, 2004, pp. 833–843.

TABLE V
AVERAGE AUC VALUE OBTAINED BY RIPPER AND EVALUATED MEASURES.

	Ripper	$conf$	$lapl$	$lift$	$conv$	$leve$	$\mathcal{X}^2$	$jacc$	cos	ϕ-coeff
Blood	63.34(3.69)	67.42(6.38)	66.89(6.44)	**67.74(6.25)**	67.67(6.16)	66.83(6.66)	66.78(6.30)	67.34(6.28)	67.22(6.32)	67.49(6.27)
Breast	97.33(2.26)	97.60(1.74)	97.06(1.72)	**97.84(1.28)**	97.32(1.34)	97.00(1.44)	96.89(1.70)	96.79(1.81)	96.95(1.91)	96.75(1.90)
Bupa	67.13(6.19)	67.34(4.90)	66.06(6.37)	**69.06(5.20)**	67.25(4.78)	67.19(6.07)	66.49(5.51)	66.21(5.26)	66.77(6.30)	68.42(6.09)
CMC	68.64(2.27)	69.58(3.45)	69.13(3.35)	**69.63(3.18)**	69.30(3.43)	69.11(3.15)	69.32(3.34)	68.87(3.37)	69.08(3.83)	68.96(3.49)
Flare	56.94(2.25)	63.13(4.55)	62.20(4.76)	62.84(4.84)	62.94(4.69)	**63.43(4.85)**	62.26(5.10)	62.56(4.66)	62.45(4.56)	62.30(4.86)
Haberman	60.94(11.31)	63.53(7.95)	63.65(9.05)	62.83(8.08)	63.97(8.17)	62.60(7.95)	63.73(7.97)	**64.09(8.02)**	62.63(7.67)	63.07(7.85)
New-Thyroid	92.50(7.92)	95.06(4.92)	94.86(5.70)	93.95(5.94)	93.36(6.82)	**95.18(5.35)**	94.31(6.08)	94.26(6.23)	94.43(5.85)	94.47(5.40)
Pima	69.98(2.21)	74.12(3.51)	72.28(4.34)	74.19(3.58)	**74.54(2.97)**	71.92(4.28)	72.27(3.84)	72.50(3.36)	72.37(4.22)	72.22(4.05)
Vehicle	92.21(2.55)	**94.42(1.90)**	94.06(1.89)	93.90(1.86)	93.52(2.41)	92.98(2.22)	93.67(2.09)	93.82(1.77)	93.52(2.21)	93.25(1.94)
Yeast	65.99(2.13)	**69.49(3.15)**	68.41(2.81)	69.01(2.72)	69.02(3.37)	68.30(3.32)	68.69(3.33)	67.78(2.66)	68.49(2.61)	68.80(2.85)

TABLE VI
RANKING OF AUC VALUES OBTAINED BY RIPPER AND EVALUATED MEASURES.

Data Set	Blood	Breast	Bupa	CMC	Flare	Haberman	New-Thyroid	Pima	Vehicle	Yeast	Sum	Score
Ripper	10	3	6	10	10	10	10	10	10	10	89	8
$conf$	4	2	3	2	2	5	2	3	1	1	25	1
$lapl$	7	5	10	5	9	4	3	6	2	7	58	4
$lift$	1	1	1	1	4	7	8	2	3	3	31	2
$conv$	2	4	4	4	3	2	9	1	7	2	38	3
$leve$	8	6	5	6	1	9	1	9	9	8	62	6
$\mathcal{X}^2$	9	8	8	3	8	3	6	7	5	5	62	6
$jacc$	5	9	9	9	5	1	7	4	4	9	62	6
cos	6	7	7	7	6	8	5	5	6	6	63	7
$\phi - coeff$	3	10	2	8	7	6	4	8	8	4	60	5

[15] C. R. Milaré, G. E. A. P. A. Batista, and A. C. P. L. F. Carvalho, "A hybrid approach to learn with imbalanced classes using evolutionary algorithms," in *Proc. 9th International Conference Computational and Mathematical Methods in Science and Engineering (CMMSE)*, vol. II, 2009, pp. 701–710.

[16] J. Demšar, "Statistical comparisons of classifiers over multiple data sets." *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.