The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Obtaining Accurate Classifiers with Pareto-optimal and Near Pareto-optimal Rules

Isao Kuwajima, Yusuke Nojima, and Hisao Ishibuchi

Gakuen-cho 1-1, Naka-ku, Sakai, Osaka 599-8531, Japan
(Tel : 81-072-254-9198; Fax : 81-072-254-9915)
(Email {kuwajima@ci., nojima@, hisaoi@}cs.osakafu-u.ac.jp)

Abstract: In the field of data mining, confidence and support are often used to evaluate a rule. Pareto-optimality of rules can be defined using these criteria. In this paper, we examine the effectiveness of designing classifiers from Pareto-optimal and near Pareto-optimal rules. We compare the Pareto-optimal and near Pareto-optimal rules with the rules obtained by different rule evaluation criteria. Through computational experiments, we show that classifiers obtained from Pareto-optimal rules have higher accuracy than those from rules extracted by other criteria.

Keywords: Rule-based classifier, evolutionary multiobjective optimization, knowledge extraction.

I. INTRODUCTION

Data mining is a very active and rapidly growing research area in the field of computer science. The task of data mining is to extract useful knowledge for users from a database. Association rule mining is one of the most well-known data mining techniques. In its basic form, all association rules satisfying some constraints are extracted from a database. A number of rule evaluation criteria which quantify the interestingness or goodness of a rule have been proposed.

Rule evaluation criteria are often used to define the constraints on rule extraction. In the field of data mining, two rule evaluation criteria called confidence and support are widely used. In addition to confidence and support, a number of rule evaluation criteria have also been proposed. Among them are gain, variance, chi-squared value, entropy gain, gini, laplace, lift, and conviction. It is shown in [1] that the best rule according to any of the above-mentioned criteria is included in the Pareto-optimal rules with respect to the maximization of confidence and support.

Evolutionary multiobjective rule selection is an approach to the search for accurate and interpretable rule-based classifiers. It tries to find Pareto-optimal classifiers in terms of the maximization of accuracy and interpretability. Through the analysis of our experimental results, we found that Pareto-optimal classifiers obtained by the evolutionary multiobjective rule selection contain a number of Pareto-optimal rules and near Pareto-optimal rules. From this observation, we have proposed designing classifiers from Pareto-optimal and near Pareto-optimal rules [2].

In this paper, we examine the effectiveness of designing classifiers from Pareto-optimal and near Pareto-optimal rules by comparing them with those from rules extracted by other criteria. Through computational experiments, we show that classifiers obtained from Pareto-optimal rules have higher accuracy than those from rules extracted by other criteria.

II. RULE EVALUATION CRITERIA

Let $\mathbf{x} = (x_1, ..., x_n)$ be an n-dimensional pattern vector. For an n-dimensional classification problem, we use the rules of the following form:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and } ... \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q , \quad (1)$$

where $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ is an antecedent interval, C_q is a consequent class, and CF_q is a rule weight (i.e., certainty factor). We denote the rule R_q in (1) as $\mathbf{A}_q \Rightarrow C_q$. Each antecedent condition "x_i is A_{qi}" in (1) means the inclusion relation "$x_i \in A_{qi}$." It should be noted that R_q in (1) does not always have n antecedent conditions. Some conditions can be *don't care*. Some rules have only a few conditions while others may have many conditions.

In the field of data mining, confidence and support are often used to examine the quality of a rule. Let us denote the support count of the rule $\mathbf{A}_q \Rightarrow C_q$ by $SUP(\mathbf{A}_q \Rightarrow C_q)$, which is equal to the number of patterns covered by antecedent set $\mathbf{A}_q$ and whose class is the same as the consequent class C_q. $SUP(\mathbf{A}_q)$ and $SUP(C_q)$ are also defined in the same manner. Let us assume that we have m training patterns from M classes. The confidence of the rule $\mathbf{A}_q \Rightarrow C_q$ is defined as

©ISAROB 2008

$$c(\mathbf{A}_q \Rightarrow C_q) = \frac{SUP(\mathbf{A}_q \Rightarrow C_q)}{SUP(\mathbf{A}_q)}. \quad (2)$$

On the other hand, the support of $\mathbf{A}_q \Rightarrow C_q$ is defined as

$$s(\mathbf{A}_q \Rightarrow C_q) = \frac{SUP(\mathbf{A}_q \Rightarrow C_q)}{m}. \quad (3)$$

A number of rule evaluation criteria other than confidence and support have been proposed. In this paper, we examine the following five criteria: the difference in confidence, slave, cover, laplace and lift.

Difference in confidence (CF): The difference between confidences is defined as

$$CF_q = c(\mathbf{A}_q \Rightarrow C_q) - \sum_{h \neq C_q} c(\mathbf{A}_q \Rightarrow \text{Class } h), \quad (4)$$

which is also used as the rule weight CF in this paper. It is shown that this value is effective as the rule weight for multi-class classification problems with more than two classes [3].

Slave: Slave is defined as

$$Slave(\mathbf{A}_q \Rightarrow C_q) = s(\mathbf{A}_q \Rightarrow C_q) - \sum_{h \neq C_q} s(\mathbf{A}_q \Rightarrow \text{Class } h), \quad (5)$$

which is the same as the difference in support. This criterion can be viewed as a simplified version of a rule evaluation criterion used in an iterative fuzzy genetics-based machine learning algorithm called SLAVE [4].

Cover: Cover is defined as

$$Cover(\mathbf{A}_q \Rightarrow C_q) = \frac{SUP(\mathbf{A}_q)}{m}, \quad (6)$$

which indicates the percentage of patterns covered by antecedent set $\mathbf{A}_q$. It is also known as the antecedent support.

Laplace: Laplace is defined as

$$Laplace(\mathbf{A}_q \Rightarrow C_q) = \frac{m \cdot SUP(\mathbf{A}_q \Rightarrow C_q) + 1}{SUP(\mathbf{A}_q) + k}, \quad (7)$$

where k is an integer greater than 1 (usually set to the number of classes M). This criterion is commonly used to rank rules for classification purposes.

Lift: Lift, a well-known statistical measure, is defined as

$$Lift(\mathbf{A}_q \Rightarrow C_q) = \frac{c(\mathbf{A}_q \Rightarrow C_q)}{SUP(\mathbf{A}_q)/m}. \quad (8)$$

Lift is a value that gives us information about the increase in probability of the consequent part given the antecedent part.

Pareto-optimal Rules

We call the rules that are Pareto-optimal in terms of the maximization of the confidence and support *Pareto-optimal rules*. Figure 1 shows the distribution of the Pareto-optimal rules in confidence-support space.

In order to handle not only Pareto-optimal rules but also near Pareto-optimal rules, we define ε-Pareto-optimal rules using a dominance margin ε. A rule R_i is said to be ε-dominated by another rule R_j when at least one of the two following conditions are satisfied.

$$s(R_i) + \varepsilon < s(R_j) \text{ and } c(R_i) + \varepsilon \leq c(R_j), \quad (9)$$

$$c(R_i) + \varepsilon < c(R_j) \text{ and } s(R_i) + \varepsilon \leq s(R_j). \quad (10)$$

When a rule R_i is not dominated by any other rules in the sense of ε-dominance in (9) and (10), we call R_i *ε-Pareto-optimal rules*.

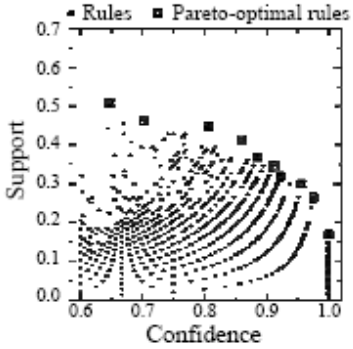


Fig. 1. The distribution of Pareto-optimal rules.

III. EVOLUTIONARY MULTIOBJECTIVE RULE SELECTION

In order to obtain simple and accurate classifiers, we use the evolutionary multiobjective rule selection. Let us assume that we have already extracted N rules by an association rule mining technique. A subset S of the N candidate rules is represented by a binary string of length N as

$$S = s_1 s_2 \ldots s_N, \quad (11)$$

where $s_j = 1$ and $s_j = 0$ mean that the j-th candidate rule is included in S and excluded from S, respectively.

Since any subsets of the N rules are represented by binary strings in (11), we can use almost all evolution-

ary multiobjective optimization algorithms. In this paper, we use the NSGA-II algorithm [5] because it is a well-known high-performance EMO algorithm.

Let P be the current population in NSGA-II. The outline of NSGA-II can be written as follows:

1: P = Initialize(P)
2: **while** a termination condition is not satisfied **do**
3: P' = Selection(P)
4: P'' = Genetic Operations(P')
5: P = Replace($P \cup P''$)
6: **end while**
7: **return** non-dominated solutions (P)

First an initial population is generated in line 1. In line 3, parent individuals (i.e., P') are selected from the current population P. The standard binary tournament selection is usually used to choose a pair of parent individuals. In line 4, an offspring population P'' is generated from the parent population P' by genetic operations such as crossover and mutation. In line 5, the best individuals are chosen from the merged population ($P \cup P''$) to generate the next population P. For details of NSGA-II, see [5].

IV. COMPUTATIONAL EXPERIMENTS

Experiments were conducted using the following eight data sets from the UCI machine learning repository: Breast W, Car, Glass, Heart C, Iris, Soybean L, Vote, and Wine. For all of these data sets, the accuracy of the classifiers was examined by iterating the two-fold cross-validation five times (5×2CV). The domain of each attribute was divided into multiples intervals using the optimal splitting method [6] based on the class entropy measure.

We compare the ε-Pareto-optimal rules with the same number of rules obtained by other criteria. In addition to the seven criteria mentioned before (i.e., *CF*, confidence, cover, laplace, lift, slave, support), we use the *random* criteria to compare the ε-Pareto-optimal rules with rules obtained by chance.

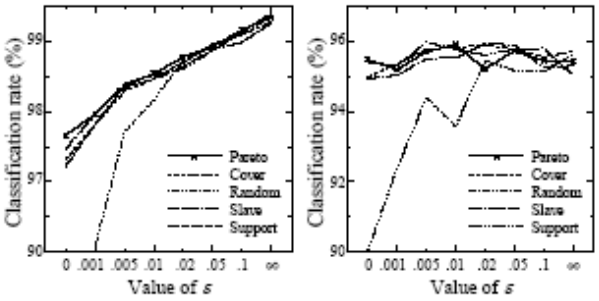
First, we extract a number of rules by specifying minimum confidence and support as 0.7 and 0.04. Then, we choose ε-Pareto-optimal rules from the extracted rules for each class. Let the total number of ε-Pareto-optimal rules be N_ε. For other criteria, we choose $\lceil N_\varepsilon / M \rceil$ rules for each class to keep the total number of rules with other criteria equal to or greater than N_ε. If extracted rules for a particular class are fewer than $\lceil N_\varepsilon / M \rceil$, we randomly choose the rest of rules from the other classes. Table 1 shows N_ε for each data set.

Results for the Pareto-optimal rules (i.e., $\varepsilon = 0$) are presented in Tables 2 and 3. Because a number of non-dominated classifiers were obtained by the evolutionary multiobjective rule selection, we can choose various classifiers with different accuracy and complexity. Due to the page limitation, we show only the results of the classifier with the highest training data accuracy. From Tables 2 and 3, we can see that the Pareto-optimal rules have the highest training and test data accuracy for three out of eight data sets. There is, however, no overall best criterion suited for all the datasets.

Figures 2 and 3 show the results for different values of ε. We show the results of the best four criteria and ε-Pareto-optimal rules. A larger value of ε means that more rules are included in the candidate rules. This is summarized in Table 1. From Figs. 2 and 3, we can see that the performance of the ε-Pareto-optimal rules becomes the same as that of rules obtained by the other criteria. In contrast, when the value of ε is small, the ε-Pareto-optimal rules have higher accuracy than rules obtained by the other criteria.

Table 1. Number of ε-Pareto-optimal rules.

	Value of ε				
Data set	0	0.001	0.01	0.1	∞
Breast W	18	34	105	897	4280
Car	2	3	4	10	425
Glass	104	194	453	2854	3314
Heart C	14	19	47	1079	3582
Iris	21	23	46	96	463
Soybean L	866	2416	3778	7762	7765
Vote	8	10	31	309	2739
Wine	26	174	194	2496	26276



(a) Training data. (b) Test data.

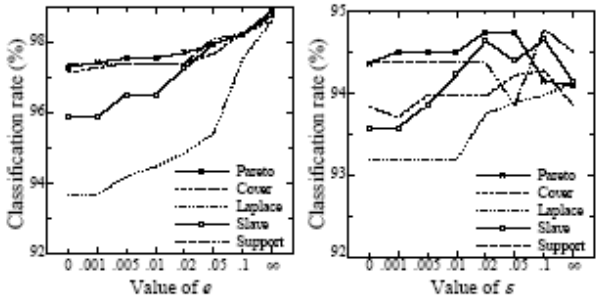
Fig. 2. Results for different values of ε (Breast W).

Table 2. Training data accuracy (%) when the number of rules is the same as Pareto-optimal rules (i.e., $\varepsilon = 0$).

	Pareto	CF	Conf	Cover	Laplace	Lift	Random	Slave	Support
Breast W	**97.7**	66.4	66.4	97.3	90.1	66.4	92.4	97.2	97.5
Car	59.8	55.7	55.7	64.1	62.7	55.7	59.5	62.7	**66.7**
Glass	72.5	71.2	71.2	74.7	69.7	71.2	**77.8**	69.6	74.0
Heart C	57.5	56.1	56.1	**59.5**	57.1	56.1	57.2	59.0	59.4
Iris	97.3	91.3	91.3	97.2	93.7	91.3	88.3	95.9	**97.4**
Soybean L	57.9	58.3	58.3	**61.0**	60.5	58.3	60.8	59.4	**61.0**
Vote	**97.0**	66.8	66.8	89.8	88.8	66.8	74.7	**97.0**	**97.0**
Wine	**98.3**	84.5	84.5	92.8	98.1	84.5	84.1	98.0	98.2

Table 3. Test data accuracy (%) when the number of rules is the same as Pareto-optimal rules (i.e., $\varepsilon = 0$).

	Pareto	CF	Conf	Cover	Laplace	Lift	Random	Slave	Support
Breast W	**95.5**	63.7	63.7	95.0	88.7	63.7	90.1	94.9	95.4
Car	58.6	55.4	55.4	63.9	62.5	55.4	59.3	62.5	**66.3**
Glass	57.2	52.0	52.0	59.4	52.3	52.0	**59.5**	53.2	58.9
Heart C	49.7	48.0	48.0	**52.4**	49.6	48.0	49.6	50.3	51.5
Iris	**94.4**	90.1	90.1	93.8	93.2	90.1	81.2	93.6	**94.4**
Soybean L	46.1	45.7	45.7	**49.3**	48.8	45.7	48.2	47.1	48.7
Vote	**97.0**	66.2	66.2	87.0	87.8	66.2	72.2	**97.0**	**97.0**
Wine	89.5	69.9	69.9	86.4	89.1	69.9	69.2	**89.9**	89.0



(a) Training data. (b) Test data.

Fig. 3. Results for different values of ε (Iris).

V. CONCLUSION

We examined the effectiveness of designing classifiers from Pareto-optimal and near Pareto-optimal rules. A number of rule evaluation criteria, in addition to support and confidence, were examined and compared with the Pareto-optimal and near Pareto-optimal rules.

Through computational experiments, we showed that classifiers obtained from Pareto-optimal rules had high accuracy. There was, however, no overall best criterion suited for all the datasets. We also showed that the performance of ε-Pareto-optimal rules gradually became the same as that of rules extracted by the other criteria.

REFERENCES

[1] Bayardo RJ Jr., Agrawal R (1999), Mining the most interesting rules, Proc. of 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 145-153.

[2] Ishibuchi H, Kuwajima I, Nojima Y (2007), Prescreening of candidate rules using association rule mining and Pareto-optimality in genetic rule selection, Proc. of 11th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems: 86-93.

[3] Ishibuchi H, Yamamoto T (2005), Rule weight specification in fuzzy rule-based classification systems, IEEE Trans. on Fuzzy Systems 13 (4): 428-435.

[4] Gonzalez A, Perez R (1999), SLAVE: A genetic leaning system based on an iterative approach, IEEE Trans. on Fuzzy Systems 7 (2): 176-191.

[5] Deb K, Pratap A, Agarwal S, Meyarivan T (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. on Evolutionary Computation 6 (2): 182-197.

[6] Elomaa T, Rousu J (1999), General and efficient multisplitting of numerical attributes, Machine Learning 36 (3): 201-244.

©ISAROB 2008