A Comparison of Methods for Rule Subset Selection Applied to Associative Classification

Gustavo E. A. P. A. Batista, Claudia R. Milaré, Ronaldo C. Prati, Maria C. Monard

Pontifícia Universidade Católica de Campinas
Rodovia D. Pedro I, Km 136, CEP 13086-900
Campinas-SP, Brasil
gbatista at puc-campinas edu br

Centro Universitário das Faculdades Associadas de Ensino
Caixa Postal 96, CEP 13870-377
São João da Boa Vista-SP, Brasil
cmilare at fae br

Depto. de Ciência da Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - Campus de São Carlos
Caixa Posta 668, CEP 13560-970
São Carlos-SP, Brasil
{prati, mcmonard} at icmc usp br

Abstract

This paper presents Garss, a new algorithm for rule subset selection based on genetic algorithms, which uses the area under the ROC curve – AUC – as fitness function. Garss is a post-processing method that can be applied to any rule learning algorithm. In this work, Garss is analysed in the context of associative classification, where an association rule algorithm generates a set rules to be used as a classifier. An experimental evaluation was performed in order to analyse the behaviour of the proposed method. Results are compared with Roccer, a recently proposed algorithm for rule subset selection based on ROC analysis.

Keywords: Rule Subset Selection, Associative Classification, Genetic Algorithms, Machine Learning.

1 Introduction

Learning rules from data is a relevant task in Knowledge Discovery from Databases – KDD. Rules have several properties that make them appropriate for KDD. One of the most important properties is that rules can be easily comprehended or interpreted by non-experts. However, there are situations that might lead to the induction of excessively large and complex rule sets. For instance, large rule sets may occur when rule learning algorithms are directly applied to large data sets. Another example is the generation of association rules that frequently produces large rule sets.

Several techniques have been proposed in order to

Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No 32 (2006), pp. 29-35.
ISSN: 1137-3601. ©AEPIA (http://www.aepia.org/revista)

overcome this problem. A very direct approach to reduce the number of rules is to postprocess the rule set, selecting a subset of the induced rules, a method known as *rule subset selection.* Rule subsets can be constructed aiming to maximize several important rule set quality criteria, such as comprehensibility, interestingness, and classification performance.

This work presents a new approach to rule subset selection based on genetic algorithms – GAs, namely Genetic Algorithm for Rule Subset Selection – Garss. This approach differs from previous ones by searching for a rule subset that maximizes the area under the ROC curve – AUC, a classification performance metric that has several advantages over accuracy and error rate. In order to analyze the performance of Garss, we carried out an experimental evaluation and the obtained results are compared with a recently proposed algorithm, Roccer [17], which looks for a rule subset that creates a convex hull in the ROC space.

Our experiments are performed using an *associative classifier*, *i.e.*, a classifier generated by an associative rule learning algorithm. An associative classifier is composed by all rules where the consequent is the class-attribute, known as *class association rules – CARs.* Usually, an associative classifier is composed by a large rule set. The number of rules frequently outnumbers the number of examples, implying serious restrictions for knowledge deployment. Garss objective is to considerably reduce the number of rules of these classifiers, and still be able to achieve similar classification performance in terms of AUC.

This work is organized as follows. Section 2 reviews a body of related work. Section 3 briefly describes ROC analysis in general, and the area under the ROC curve in particular, as a way to evaluate classifiers. Section 4 describes Garss and Roccer, the two rule subset selection algorithms which use ROC based methods to search for the best rule subset. Section 5 discusses our experimental results comparing the proposed methods with C4.5 [20] and the classifier composed by all class association rules. Finally, Section 6 presents our concluding remarks and directions for future work.

2 Related Work

In general, rule learning algorithms perform a greedy search for a set of rules that predicts a data sample. In a general way, these algorithms can be divided into two large families: separate-and-conquer such as CN2 [6] and Ripper [7], and divide-and-conquer such as C4.5 [20] and CART [5]. Greedy search imposes a limitation in algorithms of both families: if a bad rule has been introduced into the rule set, there is no chance of finding a better rule [8].

On the other hand, association rule algorithms perform a global search to find all rules that satisfy some constraints such as minimum support and confidence. Although association rule algorithms are primarily used for describing data, *i.e.*, they belong to the non supervised learning approach, it is still possible to use association rule algorithms for prediction, as proposed in [14], known as associative classification. In other words, associative classification consists of set of rules found by an association rule algorithms using only one consequent, the class attribute.

Some research papers have shown that associative classifiers are competitive with classifiers induced by separate-and-conquer and divide-and-conquer algorithms [23]. However, the number of association rules generated can easily surpass the human capacity. Thus, the use of these sorts of algorithms in KDD requires a technique for selecting the most promising rules.

In this way, some extensions have been proposed to the original association rule Apriori algorithm [1] in order to create more comprehensible association rule classifiers. One of them is the Classification Based on Association – CBA – algorithm [14] which first evaluates the association rules to decide which ones will be included in the associative classifier. This evaluation is based on rules' confidence and support information. Another extension was implemented in the Classification Based on Multiple Association Rule – CMAR – algorithm [13], which uses a 3-stage pruning strategy. Rules are pruned based on their support and confidence, as well as the correlation with other rules and number of covered training examples. A similar idea has been explored in the AprioriC and AprioriSD algorithms [11, 12], where a filtering step is used to remove some of the redundant rules. However, AprioriC still tends to build large rule sets and

AprioriSD has been developed mainly for subgroup discovery. Other extensions were implemented, such as the Classification Based on Predictive Association Rule – CPAR –algorithm [23], which uses a greedy search to generate a reduced number of rules, and a method proposed in [22] to select rules based on their misclassification cost instead of error rate.

Our proposed method differs from the aforementioned research work by using AUC as a major metric to select rules and using a search based on genetic algorithms. AUC and ROC curves are briefly described in the next section.

3 ROC analysis

As a rule, error rate (or accuracy) considers misclassification of examples equally important. However, in most real-world applications this is an unrealistic scenario since certain types of misclassification are likely to be more serious than others. Unfortunately, misclassification costs are often difficult to estimate. Moreover, when prior class probabilities are very different, the use of error rate or accuracy might lead to misleading conclusions, since there is a strong bias to favour the majority class. For instance, it is straightforward to create a classifier having an error rate of 1% (or 99% of accuracy) in a domain where the majority class holds 99% of the examples, by simply forecasting every new example as belonging to the majority class. Another point that should be considered when studying the effect of class distribution on learning systems is that misclassification costs and class distribution may not be static.

In scenarios where the target class priors and/or misclassification costs are unknown or are likely to change, the use of error rate as a basic performance measure may lead to misleading conclusions. This is due to the fact that the error rate strongly depends on class distribution and misclassification costs. Furthermore, the use of the error rate in such conditions does not allow the direct comparison/evaluation of how learning algorithms would perform in different scenarios. Receiver Operating Characteristic (ROC) analysis [21] provides a way of assessing a classifier performance independently of misclassification costs or changes in the class distribution. ROC based methods provide a fundamental tool for analyzing and assessing classifiers performance in imprecise environments. The basic idea is to decouple relative error rate (percentage of false positives or false positive rate – *fpr*) from hit rate (percentage of true positives or true positive rate – *tpr*) by using each of them as axis in a bi-dimensional space. Thus, in ROC analysis a classifier is represented by a pair of values instead of a single error rate value. Furthermore, spreading the classifier criterion over all possible trades off of hits and errors, a curve that works as an index that reflects the subjective probabilities and utilities that determine all possible criteria is obtained.

For instance, consider a classifier that provides the probabilities of an example belonging to each class, such as the Naive Bayes classifier [16], we can use these probabilities as a threshold parameter biasing the final class selection. Then, for each threshold, we plot the percentage of hits against the percentage of errors. The result is a bowed curve, rising from the lower left corner (0,0), where both percentages are zero, to the upper right corner (1,1), where both percentages are 100%. The more sharply the curve bends, the greater the ability of coping with different class proportions and misclassification costs, since the number of hits relative to the number of false alarms is higher. By doing so, it is possible to consider what might happen if a particular score is selected as a classification threshold, allowing selecting the most suitable threshold given a specific situation.

In situations where neither the target cost distribution nor the class distribution are known, an alternative metric to compare models through ROC analysis is the area under the ROC curve (AUC). The AUC represents the probability that a randomly chosen positive example will be rated higher than a negative one [19], and in this sense it is equivalent to the Wilcoxon test of ranks. However, it should be kept in mind that given a specific target condition, the classifier with the maximum AUC may not be the classifier with the lowest error rate.

4 Rule Selecting Methods Based on ROC

This section describes the two algorithms to select rules considered in this work. Both of them select rules aiming to maximise the overall AUC.

We start by describing Garss, our proposed algorithm. Afterwards, we briefly describe Roccer, which uses the ROC convex hull to select the best subset of rules.

4.1 Garss

The proposed Genetic Algorithm for Rule Subset Selection – Garss – is a rule selection algorithm that uses genetic algorithms to select rules aiming to maximize the AUC metric. Genetic algorithms are search algorithms based on natural selection and natural genetics [10]. They consist of successive sets of potential solutions (population), encoded as a sequence of bits or numbers (chromosomes), which are generated by applying a set of transformations (the most used ones are the genetic operations crossover and mutation), and by evaluating the quality (fitness) of the chromosomes as solutions to the particular problem.

In this work we use a rule database composed by all rules generated by the Apriori algorithm [2]. A primary key composed by a positive integer is associated to each rule in the database. Therefore, each rule can be accessed independently by its key. An array of keys (integers) is used to represent a chromosome, *i.*e. a rule set to be interpreted as a classifier (potential solution). In our implementation, the initial population is randomly composed by 30 chromosomes of 30 elements (rules) each. The evaluation function used is the AUC metric. The selection method is the fitness-proportionate selection, in which the number of times a chromosome is expected to reproduce is proportional to its fitness. A two-point crossover operator was applied with probability 0.6. In a two-point crossover, two parent chromosomes are selected from the population and two (potentially different) positions are randomly chosen. Each position is used to separate a parent chromosomes in two parts, and these parts are exchanged to generate two offsprings. With a two-point crossover, the offsprings are likely to have different sizes (number of rules) if compared with the parent chromosomes. Therefore, Garss can converge to rule sets of different sizes than those stipulated in the initial population. The mutation operator alters randomly selected positions of chromosomes, *i.e.*, this operator exchanges a random selected rule by other rule. Mutation was applied with a probability of 0.10. Mutation and cross-over probabilities were chosen based on our previous experience with genetic algorithms [15]. Finally, an elitist method of population replacement was used. According to this method, the best chromosome from each population is preserved by copying it to the next generation.

4.2 Roccer

Roughly speaking, a ROC graph is a plot of the fraction of positive examples misclassified — false positive rate (*fpr*) — on the x axis against the fraction of positive examples correctly classified — true positive rate (*tpr*) — on the y axis. It is possible to plot either a single rule, a classifier (composed by a rule set, or not) or even a partial classifier (composed by a subset of a rule set, for instance) in a ROC graph.

In [9] it is shown that rule learning using a set covering approach can be seen as tracing a curve in the ROC space. To see why, assume we have an empty rule list, represented by the point $(0, 0)$ in the ROC space. Adding a new rule R_j to the rule list implies a shift to the point (fpr_j, tpr_j), where fpr_j and tpr_j is the *tpr* and *fpr* of the partial rule list (interpreted as a decision list) containing all rules already learnt including R_j. A curve can be traced by plotting all partial rule lists (fpr_j, tpr_j), for j varying from 0 to the total number n of rules in the final rule list in the order they are learnt. A final default rule that always predicts the positive class can be added at the end, connecting the point (fpr_n, tpr_n) to the point $(1, 1)$.

Roccer [17] is a rule selection algorithm based on the ROC graph. Rules come from an external larger set of rules and the algorithm performs a selection step based on the current convex hull in the ROC graph. This approach is motivated by the observation that optimum classifiers under different misclassification costs lie on the upper convex hull in the ROC space [18]. The basic idea is to only insert a rule in the rule list if the insertion of that rule leads to a point outside the current ROC convex hull (the current ROC convex hull is the upper convex hull of the rules that are already in the rule list). Otherwise the rule is discarded.

5 Results

In order to empirically evaluate Garss and Roccer, we carried out an experimental evaluation using four data sets from UCI [3]. We only used

Data set	#Attrs	#Examples	Maj. Class %
Breast	10	683	65.00
Bupa	7	345	57.98
German	21	1000	70.00
Heart	14	270	55.55

Table 1: UCI Data sets used in experiments.

data sets without missing values as Apriori cannot handle them. Table 1 summarizes the data sets used in this study. For each data set, it shows the data set name, number of attributes (#Attrs), number of examples (#Examples), and percentage of examples in the majority class. Although both algorithms (Garss and Roccer) can handle more than two classes, we restricted our experiments to two-class problems in order to calculate AUC values. For data sets having more than two classes, we chose the class with fewer examples as the positive class, and collapsed the remainder classes as the negative class.

In the experiments, the AUC values were estimated using stratified 10-fold cross-validation. The experiments were paired, *i.e.*, all inducers were given the same training and test sets. Altogether, four algorithms were compared: Garss and Roccer (described in Section 4); C4.5 [20] a well-known divide-and-conquer learning system (used as a first basis to compare the results); and All that is a classifier composed by all class association rules generated by Apriori (used as a second basis to compare the results). Our main objectives are to create a classifier that has a similar or better classification performance than C4.5 and that have a rule set with much fewer rules than All.

The Apriori implementation of [4] was used to generate class association rules. Parameters were set to 50% of confidence and 1/3 of the percentage of minority class as support.

The performance results are presented in Table 2. Results reported are the mean AUC values averaged over the 10 test sets and their respective standard deviations in brackets. In general, Garss and Roccer present a classification performance that is similar or better than C4.5. For all data sets except Heart, Garss and Roccer present average AUC values that are greater than the average values obtained by C4.5. Only for the Heart data set, Garss presents an average AUC value that is smaller than the one for C4.5. The average AUC values obtained over all data sets (last line in Table 2) show that Roccer has an average performance slightly better than Garss, and both methods performed better than C4.5. In addition, the classifier composed by all class association rules generated by Apriori performs very well. However, these classifiers are usually composed by a large number of rules that might hinder classifier comprehensibility.

Table 3 presents the average number of induced rules for each algorithm. As stated before, classifiers such as the one created by All, which are composed by all class association rules present large rule sets. Observe that for German and Heart data sets, the average number of rules is even greater than the number of examples. On the other hand, Garss and Roccer considerably reduced the number of rules in the constructed classifiers. For Garss, the average number of rules for Breast, Bupa, German and Heart data sets represents 9.16%, 20.87%, 1.21% and 2.34% of all generated rules for each data set, respectively. For Roccer, they represent 9.64%, 1.33%, 0.82% and 3.64% of all generated rules for each data set, respectively. Although the average number of rules has a high variation for each data set, comparing the mean number of rules for all data sets (last line in Table 3, Roccer and C4.5 show similar results, while Garss has constructed rule sets that are a little larger.

The best results for Garss is for the German dataset, where the classifier constructed by the selected rules presents the highest AUC value, along with a rule set with a size slightly larger than Roccer and of about one half the size of the classifier induced by C4.5. The best result for Roccer was in the Bupa data set, with an AUC similar to the AUC values using all rules although only with 4 rules in the rule set. The best result for the classifier constructed with all rules is for the Heart dataset. In this dataset, using all class association rules the results present an AUC of 90.72%, about 5% higher than the other algorithms. However, this classifier is composed by more than 1800 rules, a number about 7 times larger than the number of examples.

In general, Garss and Roccer provided aver-

Data sets	Garss	Roccer	C4.5	All
Breast	99.06(0.46)	98.63(1.88)	97.76(1.51)	99.07(0.87)
Bupa	64.65(3.96)	65.30(7.93)	62.14(9.91)	65.38(10.63)
German	74.16(1.60)	72.08(6.02)	71.43(5.89)	73.37(4.84)
Heart	82.86(3.36)	85.78(8.43)	84.81(6.57)	90.72(6.28)
Avg	80.18	80.45	79.04	82.14

Table 2: Average AUC values and standard deviations estimated with stratified 10-fold cross-validation.

Data sets	Garss	Roccer	C4.5	All
Breast	46.00(2.56)	48.4(2.32)	37.8(12.62)	502.1(8.96)
Bupa	61.10(1.55)	3.9(0.99)	15(10.53)	292.8(21.57)
German	34.90(5.88)	23.7(6.75)	78.2(18.5)	2886.1(577.3)
Heart	43.90(1.89)	68.2(4.42)	13.2(4.49)	1875.6(91.9)
Avg	46.48	36.05	36.05	1389.15

Table 3: Average number of rules and respective standard deviations.

age AUC values that are similar or better than C4.5. Comparing with all class association rules, the average AUC results are similar for the data sets Breast, Bupa and German, although lower for Heart. On the other hand, Garss and Roccer provided a significant reduction in the average number of rules, thus improving the comprehensibility of the final rule sets.

6 Conclusion

This work describes Garss, a novel algorithm to select rules. Garss uses a genetic algorithm and selects rules aiming to maximize the area under the ROC curve (AUC). We compared the results obtained by Garss with a recently proposed algorithm, Roccer, with C4.5 and with the classifier composed by all class association rules generated by Apriori. The results are promising, since Garss and Roccer showed a classification performance (in AUC) similar or better than C4.5. In addition, Garss and Roccer were able to create rule sets considerably smaller than the associative classifier with all class association rules, still maintaining similar classification performance in three of the four data sets used in the experiments.

As Garss and Roccer are post-processing algorithms, they are not limited to associative classifiers. An interesting approach for future research is to use Garss and Roccer as a pruning method, where instead of feeding them with class association rules, we use as input other symbolic classifiers (such as the tree induced by C4.5). Another interesting research direction is to evaluate the performance of Garss and Roccer as methods for combining rules from different classifiers.

7 Acknowledgements

We would like to thank Fundação Parque Tecnológico Itaipu (PDTA/FPTI-Br) for financial support.

References

[1] R. Agrawal, T. ImielinskiP., and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the International Conference on Management of Data*, SIGMOD, pages 207–216, 1993.

[2] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[3] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[4] Christian Borgelt and Rudolf Kruse. Induction of association rules: A priori implementation. In *15th Conf. on Computational Statistics*, 2002.

[5] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

[6] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proc. 5th European Conf. on Machine Learning*, volume 482 of *LNAI*, pages 151–163. Springer-Verlag, 1991.

[7] W. Cohen. Fast effective rule induction. In *Proc. 12th Int. Conf. on Machine Learning*, pages 115–123, 1995.

[8] P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24(2):141–168, 1996.

[9] J. Fürnkranz and P. Flach. ROC'n'rule learning – toward a better understanding of rule covering algorithms. *Machine Learning*, 58(1):39–77, 2005.

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison Wesley, 1998.

[11] V. Javanoski and N. Lavrač. Classification rule learning with Apriori-C. In *Proc. 10th Portuguese Conf. on Artificial Intelligence*, volume 2258 of *LNAI*, pages 44–52, Porto, Portugal, 2001. Springer-Verlag.

[12] B. Kavšek, N. Lavrač, and V. Javanoski. Apriori-SD: Adapting association rule learning to subgroup discovery. In *Proc. 5th Inteligent Data Analysis*, volume 2810 of *LNCS*, pages 230–241. Springer Verlag, 2003.

[13] Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 369–376. IEEE Computer Society, 2001.

[14] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining*, pages 80–86, New York, USA, 1998.

[15] 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*, volume 2972 of *LNAI*, pages 833–843. Springer-Verlag, 2004.

[16] Tom Mitchell. *Machine Learning.* McGraw Hill, New York, NY, USA, 1997.

[17] Ronaldo C. Prati and Peter A. Flach. Roccer: An algorithm for rule learning based on ROC analysis. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'2005)*, pages 823–828, Edinburgh, Scotland, UK, 2005.

[18] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

[19] Foster Provost and Tom Fawcett. Robust Classification Systems for Imprecise Environments. In *National Conference on Artificial Intelligence (AAAI'98)*, 1998.

[20] J. R. Quinlan. *C4.5 Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.

[21] John A. Swets and Ronald M. Pickett. *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory.* Academic Press, New York, 1982.

[22] Adriano Veloso and Wagner Meira Jr. Rule generation and rule selection techniques for cost-sensitive associative classification. In *20 Simpósio Brasileiro de Bancos de Dados*, pages 295–309, 2005.

[23] Xiaoxin Yin and Jiawei Han. Cpar: Classification based on predictive association rules. In *Proceedings of the Third SIAM International Conference on Data Mining*, San Francisco, CA, 2003. SIAM.