Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
DAAAM Symposium

IMPROVING IF-THEN RULES WITH GENETIC ALGORITHM

MUNTEAN, M[aria]; ILEANA, I[oan]; JOLDES, R[emus]; OLTEANU, E[mil] & KADAR, M[anuella]

Abstract: *This paper aims to challenge the problem of finding accurate and relevant rules for the task of classification. The scope is to improve the accuracy, or at least to provide a comparable accuracy measure, for classification algorithms implemented so far. Because the task of classification must be as accurate as possible, the paper proposes a method based on genetic algorithms to enhance the speed and quality of classification. Thus, by using a genetic approach, there is a chance that the classification process will execute faster. A known fact is that genetic algorithms are well suited for the increase of performance.*

Key words: *classification; IF-THEN rules; genetic algorithm*

1. INTRODUCTION

Classification, one of the primary tasks in data mining, consists in assigning objects to one of several predefined categories [Nadal et al., 1990]. Many approaches have been proposed aiming at the construction of an accurate classifier. Among them, the following approaches are well-known in the literature: decision-tree, rule-induction, association-based, and hybrid ones. AQ [Michalski, 1969] and CN2 [Clark & Niblett, 1989] are two methods based on rule-induction approach where the methods induce one rule at a time and all the data examples covered by that rule are removed from the search space. The more recent rule induction using a sequential covering algorithm, RIPPER [Cohen, 1995], [Huang et al., 2007], grows rules by adding a test of an attribute to that rule if using that attribute will result in a more accurate separation of the training data. Since the IF-THEN rules can be easily represented by bit strings of fixed length, we applied the Genetic Algorithms (GAs) in rule induction [Fidelis et al., 2000], [Wan & Zhao, 2009], in order to increase the classifier's performance.

2. COST-SENSITIVE CLASSIFICATION

The percentage of instances on a given test set instances that are correctly classified by the classifier represents the accuracy of a classifier. A very useful tool, used for analyzing how well a classifier can recognize instances of different classes, is the confusion matrix. A confusion matrix for two classes is shown in Table 1.

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

Tab. 1. Confusion matrix for a two-class problem

The true positives (TP) and true negatives (TN) are correct classifications. A false positive (FP) occurs when the outcome is incorrectly predicted as 1 (or positive) when it is actually 0 (negative). A false negative (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive. In order to access how well the model can classify the instances, defined two new measures, sensitivity and specificity:

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

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

3. DESCRIPTION OF THE GA USED

Genetic algorithms have been used for classification as well as other optimization problems. In data mining, they may be used to evaluate the fitness of other algorithms. In the context of data mining, the population represents the solution search space, where a solution is an IF-THEN rule. A chromosome is an IF-THEN rule and a gene represents a sub-condition in the IF-THEN rule. Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules in the current population, as well as offspring of these rules. Typically, the fitness of a rule is assessed by its classification accuracy on a set of training samples.

In order to cover a considerable percent of the instances in the test set, we have used a sequential covering algorithm. First, we have created rules in a genetic manner. Second, to compute the fitness measure of the rules, we developed a fitness method. Then, in the fitness method the distribution of the learned rule was computed. For each instance, the real and predicted distribution was calculated. The predicted and real distributions are compared and the instances are classified as TP and FN. Then, the sensitivity and specificity were determined using the true and false positives and negatives measures calculated beforehand. The fitness value was computed by simply multiplying the sensitivity and specificity.

This step represented a preprocessing stage before the rule was learned and its fitness evaluated. Thus, we made a correspondence between the genetic data types and the instances' data types.

In the genetic component, the instances were parsed to chromosomes, which were ranked, recombined and evaluated. The classifier component provided the means of evaluation and test, and measured how good the classifier performed. Our gene and chromosome had a special representation (Fig. 1 and Fig. 2). It was divided in 3 fields: weight, relational operator and value. The weight was used in order to determine whether the corresponding gene was part of the rule. The genes that had below a certain threshold were not considered anymore as part of the chromosome. The relational operator's task was to verify if the weight was below that threshold. The value represented the value of the gene.

Fig. 1. The representation of the gene as a three value field

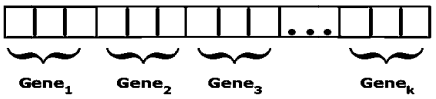

Fig. 2. The representation of a chromosome

4. EXPERIMENTAL RESULTS

For the evaluation, we used 4 data sets: Australian, Bupa, Cleveland and Pima, obtained from the online UCI Machine Learning Repository. The majority of these data sets have accuracy levels which are not very high, so improvement is possible.

Each test included 10 runs, with an 80/20 percentage split and with a seed from 1 to 10 for each run. For the tests, the genetic parameters were tuned and different behaviors for different parameters were observed. The tests were validated against existing rule learners: JRip and PART, Weka implementation of RIPPER algorithm. Parameters used for default test were: population size (50 individuals), parent selection technique (tournament technique), crossover (in 2 points, random) and mutation (0.3 mutation rate).

A. The first test set

The first of the test series was tested on all the four data sets by varying the seed of the randomization process (from 1 to 10) and keeping the other parameters constant.

Algorithm	Genetic Classifier (%)		
Data set	*aed acc*	*std dev*	*best acc*
Australian	74,49	8,01	86,95
Bupa	61,73	5,16	68,11
Cleveland	49,67	5,01	60,65
Pima	69,35	4,41	75,97

Tab. 2. The results given by the first test set

The table above (Table 2) shows that the best results were attained for the Bupa dataset. Compared to the JRip algorithm, the genetic classifier had a better median accuracy and also a better standard deviation. Although, the average accuracy for the PART algorithm was better on the Cleveland data set than in the case of the new classifier.

B. The second test set

In this test, the number of randomly generated crossover points was changed from to 2 to 3 and for each of the two cases the seed was again changed. All other parameters are kept constant. The average accuracies were obtained for the Australian and Cleveland data sets (Table 3) when the number of crossover points is set to 3 and for the Bupa and Pima data set when the number of crossover points was equal to 2. The average accuracy of the Bupa data set seems to be careless to the parametric changes. Comparing the values obtained for this data set with the values in the first test set, it can be observed that Australian and Cleveland perform 1% better. The deviations calculated in the table are rather high, except for the Pima data set, which obtains 4.3 %, when the number of crossover points is 3.

No of crossover points	**2**		**3**	
Data set	***avg acc***	***std dev***	***avg acc***	***std dev***
Australian	74.78	9.94	52.62	9.81
Bupa	61.01	6.67	60	7.91
Cleveland	48.36	13.16	75.21	5.21
Pima	70.06	6.99	68.89	4.31

Tab. 3. The values for the second test set

C. The third test set

In the last test set, we varied the number of evaluation cycles starting from 100 to 1000 and ending up to see the results of the classifier in the case when we have 10000 evaluation cycles. Again, the seed for each setting of the number of evaluation cycles was varied (Table 4).

No of cycles	100		1000		10000	
Data set	*avg acc*	*std dev*	*avg acc*	*std dev*	*avg acc*	*std dev*
Australian	72.39	6.53	74.13	6.57	72.31	7.47
Bupa	61.73	9.49	60.28	8.17	60.43	7.03
Cleveland	51.96	7.16	50	7.73	49.67	12.62
Pima	69.87	8.22	72.07	5.62	69.09	5.59

Tab. 4. The values resulted from the third experiment

5. CONCLUSIONS

The experiments proved that learning classification rules using genetic algorithms can have similar accuracy measures in comparison with the results obtained from the JRip and PART algorithms (Table 5).

	Bupa (%)	Austra-lian(%)	Cleve-land(%)	Pima (%)
JRip	59.42	75.52	53.3	74.53
PART	60.72	79.53	52.45	75.04
Genetic Classifier	64.05	79.03	54.59	73.96

Tab. 5. Comparative study between JRip, PART and the Genetic Classifier

The Bupa data set behaved well throughout all the test sets. Its accuracy remained constant when evaluated within all the three experiments.

The experiments performed tried to obtain a confirmation of the fact that by combining a genetic component with a heuristic component, high accuracy levels can be obtained in comparison to other classification algorithms.

6. REFERENCES

Clark, P. & Niblett, T. (1989). The CN2 induction algorithm, *Machine Learning*, Vol.3(4), pp.261-283, ISSN 0885-6125

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

Fidelis, M. V., Lopes, H. S. & Freitas A. A. (2000). Discovering Comprehensible Classification Rules with a Genetic Algorithm, *Proceedings of the 2000 Congress on Evolutionary Comp.*, Vol.1, pp.805-810, La Jolla, USA

Huang, J., Cai, Y. & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information, *Pattern Recognition Letters*, Vol. 28(13), pp.1825-1844

Michalski, R. S. (1969). On the quasi-minimal solution of the general covering problem, *Proceedings of the V International Symposium on Information Processing*, Switching Circuits, Vol.A3, Yugoslavia, Bled, pp.125-128

Nadal, C.; Legault, R. & Suen, C. Y. (1990). Complementary algorithms for the recognition of totally uncontrained handwritten numerals, *Proceedings of the 10th Int. Conf. on Pattern Recognition*, Vol.A, pp.434–449, Atlantic City

Wan, L. & Zhao, C. (2009). Rule Acquisition with an Entropy-based Hybrid Genetic Algorithm, *2009 International Conference on Networking and Digital Society*, pp.275-278