A Multiobjective Genetic Algorithm for Attribute Selection

Gisele L. Pappa [1] **Alex A. Freitas** [2] **Celso A.A. Kaestner** [1]

[1] Postgraduate Program in Applied Computer Science
Pontificia Universidade Catolica do Parana
Rua Imaculada Conceicao, 1155
Curitiba - PR 80215-901 Brazil
{gilpappa, kaestner}@ppgia.pucpr.br

[2] Computing Laboratory
University of Kent at Canterbury
Canterbury, CT2 7NF. UK
A.A.Freitas@ukc.ac.uk
http://www.cs.ukc.ac.uk/people/staff/aaf

Abstract*: The problem of feature selection in data mining is an important real-world problem that involves multiple objectives to be simultaneously optimized. In order to tackle this problem this work proposes a multiobjective genetic algorithm for feature selection based on the wrapper approach. The algorithm's main goal is to find the best subset of features that minimizes both the error rate and the size of the tree discovered by a classification algorithm, namely C4.5, using the Pareto dominance concept.*

Keywords*: attribute selection, genetic algorithms, multiobjective optimization, data mining*

1. Introduction

Data mining emerged from the need for converting records stored in large databases into useful, interesting and comprehensible knowledge. Data mining is typically performed on real-world databases that had been created for purposes other than learning [7]. Many of these databases have irrelevant attributes that may be harmful for the data mining process, and so they should be removed. After all, no matter how "intelligent" a data mining algorithm is, it will fail to discover high-quality knowledge if it is applied to low-quality data [6].

Attribute selection is an important preprocessing step of the knowledge discovery process, and its main goal is to discover a subset of attributes that are relevant for the target data mining task. In this paper we address the task of classification, where the goal is to predict the class of an example (a record) based on the values of the predictor attributes for that example. In this context, attribute selection involves two important goals: minimize both the error rate of the classification algorithm and the complexity of the knowledge discovered by that algorithm. Note that these objectives often conflict with one another, and they normally are non-commensurable – i.e., they measure different aspects of the target problem. Hence, we propose a multi-objective GA for attribute selection, where both objectives are "simultaneously" minimized, by using the concept of Pareto dominance [2].

The basic idea of multi-objective optimization is to return to the user, as the result of the problem-solving algorithm, a set of optimal solutions (rather than a single solution) by taking both objectives into account, without a priori assigning greater priority to one objective or the other. The ultimate choice about which solution should be used in practice is left to the user, which can use his/her background knowledge and experience to choose the "best" solution for his/her needs a posteriori, among all the returned optimal solutions. In other words, in a multi-objective optimization framework the user has the advantage of being able to choose the solution representing the best trade-off between conflicting objectives a posteriori, after examining a set of high-quality solutions returned by the multi-objective problem-solving algorithm. Intuitively, this is better than forcing the user to choose a trade-off between conflicting goals a priori, which is what is done when a multiobjective optimization problem is transformed into a single-objective one.

2. Attribute Selection

Attribute selection can be cast as a search where ideally the algorithm has to find the smallest subset of features with the best classification accuracy considering a target database. At a high level of abstraction any algorithm for feature selection can be divided into two parts: the search method and the evaluation function used to measure the quality of attribute subsets.

There are at least three groups of search methods: exponential, randomized and sequential and. Exhaustive search belongs to the first group (generating an exponential number of candidate solutions), and genetic algorithms (GA) to the second one. Forward Sequential Selection (FSS) is in the third group, and it is a very popular algorithm due to its simplicity. FSS starts with an empty set of selected attributes and adds one attribute at a time to that set until a stopping criterion is met – e.g., until the quality of the current set of selected attributes cannot be improved by adding another attribute to that set.

Considering the evaluation function, attribute selection algorithms may be based on two approaches: the filter or the wrapper approach. This classification is independent of the search strategy used by the attribute selection method. It depends on whether or not the evaluation function uses the target data mining algorithm (which will be eventually applied to the ultimate set of selected attributes) to evaluate the quality of a candidate attribute subset. In the filter approach the attribute selection method does not use the data mining algorithm, whereas in the wrapper approach the attribute selection method uses the data mining algorithm to evaluate the quality of a candidate attribute subset. Note that in the wrapper approach the data mining algorithm is used as a black box.

The wrapper approach tends to obtain a predictive accuracy better than the filter approach, since it finds an attribute subset "customized" for a given data mining algorithm. However, the wrapper approach is considerably more computationally expensive than the filter approach, since the former requires many runs of the data mining algorithm.

3. Multiobjective Optimization

Real world problems often involve multiple objectives that should be simultaneously optimized. Unlike simple optimization, multiobjetive optimization considers that there is no single solution that is optimum with respect to all objectives. So it generates a set of solutions with different trade-offs among the objectives. This set of solutions is found using the Pareto dominance concept. The basic idea is that a given solution x_1 dominates another solution x_2 if and only if [2]:

1. Solution x_1 is not worse than solution x_2 in any of the objectives;
2. Solution x_1 is strictly better than solution x_2 in at least one of the objectives.

Figure 1 shows an example of possible solutions found for a multiobjective attribute selection problem. The solutions that are not dominated by any other are considered Pareto-optimal solutions, and they are represented by the dotted line in Figure 1. Note that solution A has a small tree size but a large error rate. Solution B has a large tree size but a small error rate. Assuming that minimizing both objectives is important, one cannot say that solution A is better than B, nor vice-versa. On the other hand, solution C is clearly not a good solution, since it is dominated, for instance, by B.

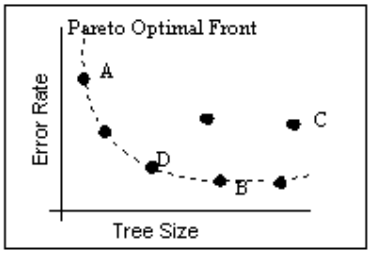

Fig. 1: Example of Pareto dominance in a two-objective problem [2]

4. The Proposed Multiobjective GA for Attribute Selection

A genetic algorithm (GA) is a search algorithm inspired by the principle of natural selection. The basic idea is to evolve a population of individuals, where each individual is a candidate solution to a given problem. Each individual is evaluated by a fitness function, which measures the quality of its corresponding solution. At each generation (iteration) the fittest (the best) individuals of the current population survive and produce offspring resembling them, so that the population gradually contains fitter and fitter individuals – i.e., better and better candidate solutions to the underlying problem. For a comprehensive review of GAs in general the reader is referred to [5]. For a comprehensive review of GAs applied to data mining the reader is referred to [6].

This work proposes a multiobjective genetic algorithm (GA) for attribute selection. Our motivation for developing a GA for attribute selection, in a multiobjective optimization framework, was that: (a) GAs are a robust search method, capable of effectively exploring the large search spaces often associated with attribute selection problems; (b) GAs perform a global search [5], [4], so that they tend to cope better with attribute interaction than greedy search methods [6], which is also an important advantage in attribute selection; and (c) GAs already work with a population of candidate solutions, which makes them naturally suitable for multiobjective problem solving, where the search algorithm is required to consider a set of optimal solutions at each iteration.

The goal of the proposed GA is to find a subset of relevant attributes that leads to a reduction in both the classification error rate and the complexity (size) of the rule set discovered by a data mining algorithm (improving the comprehensibility of discovered knowledge). In this paper the data mining algorithm is C4.5 [8], a very well-known decision tree algorithm. The proposed GA follows the wrapper approach, evaluating the quality of a candidate attribute subset by using the target classification algorithm (C4.5). Hence, the fitness function of the GA is based on the error rate and on the size of the decision tree built by C4.5. These two criteria (objectives) are to be minimized according to the concept of Pareto dominance.

4.1 Individual Encoding and Fitness Function

In the proposed GA, each individual represents a candidate subset of selected attributes, out of all original attributes. Each individual consists of M genes, where M is the number of original attributes in the data being mined. Each gene can take on the value 1 or 0, indicating that the corresponding attribute occurs or not (respectively) in the candidate subset of selected attributes.

The fitness function measures the quality of a candidate attribute subset represented by an individual. The fitness of an individual consists of two quality measures: (a) the error rate of C4.5; and (b) the size of the decision tree built by C4.5. Both (a) and (b) are computed by running C4.5 with the individual's attribute subset only, and by using a hold-out method to estimate C4.5's error rate, as follows. First, the training data is partitioned into two mutually-exclusive data subsets, the building subset and the validation subset. Then we run C4.5 using as its training set only the examples (records) in the building subset. Once the decision tree has been built, it is used to classify examples in the validation set. The two components of the fitness vector are then the error rate in the validation set and the size (number of nodes) of the tree built by C4.5.

4.2 Selection Method and Genetic Operators

At each generation, selection is performed as follows. First the GA selects all the non-dominated individuals (the Pareto front) of the current population. These non-dominated individuals are passed unaltered to the next generation by elitism [1]. Let N be the population size, and let N_{elit} be the number of individuals reproduced by elitism, where $N_{elit} \leq (N / 2)$.. Then the other $N - N_{elit}$ individuals to reproduce are chosen by performing $N - N_{elit}$ times a tournament selection procedure, as follows.

First, the GA randomly picks k individuals from the current population, where k is the tournament size, a user-specified parameter which was set to 2 in all our experiments. Then the GA compares the fitness values of the two individuals playing the tournament and selects as the winner the one with the best fitness values. The selection of the best individual is based on the concept of Pareto dominance, taking into account the two objectives to be minimized (error rate and decision tree size). Given two individuals I_1 and I_2 playing a tournament, there are two possible situations. The first one is that one of the individuals dominates the other. In this case the former is selected as the winner of the tournament.

The second situation is that none of the individuals dominates the other. In this case, as a tie-breaking criterion, we compute an additional measure of quality for each individual by taking both objectives into account. Following the principle of Pareto dominance, care must be taken to avoid that this tie-breaking criterion assigns greater priority to any of the objectives. Hence, we propose the following tie-breaking criterion. For each of the two individuals I_i, i=1,2, playing a tournament, the GA computes X_i as the number of individuals in the current population that are dominated by I_i, and Y_i as the number of individuals in the current population that dominate I_i. Then the GA selects as the winner of the tournament the individual I_i with the largest value of the formula: $X_i - Y_i$. Finally, if I_1 and I_2 have the same value of the formula $X_i - Y_i$ (which is rarely the case), the tournament winner is chosen at random. This tie-breaking criterion is also used to reduce the elitist set size when $N_{elit} > (N / 2)$.

Individuals selected by tournament selection undergo crossover and mutation, in order to create new offspring [4]. In essence, crossover consists of swapping bits between two individuals, whereas mutation consists simply of flipping the value of a bit. In all our experiments the probabilities of crossover and mutation were set to 80% and 1%, respectively. The population size N was set to 100 individuals, which evolve for 50 generations. These values were used in all our experiments.

After the individuals of the last generation have been evaluated, the set of non-dominated individuals to be returned (as the answer of the GA) is refined by performing an "internal 10-fold cross-validation", i.e., a cross-validation based on the training set only (merging the building and validation sets). Only the individuals that are non-dominated in all folds of the cross-validation procedure are returned as possible solutions for the attribute selection problem. In some cases, there is no individual that is non-dominated in all folds of this internal cross-validation. In this case, the system computes the average error rate and tree size of all individuals in all folds and return the non-dominated individuals, according to this average result. Note that this internal cross-validation is computationally expensive. That is why it is performed only after the last generation, rather than at every generation.

5. Computational Results

We have performed experiments with six public-domain, real-world data sets obtained from the UCI (University of California at Irvine)'s data set repository [9]. The number of examples, attributes and classes of these data sets is shown in Table 1.

Table 1. Main characteristics of the data sets used in the experiments

Data Set	# examples	# attributes	# classes
Arrhythmia	452	269	16
Dermatology	366	34	6
Vehicle	846	18	4
Promoters	106	57	2
Ionosphere	351	34	2
Crx	690	15	2

All the experiments were performed with a well-known stratified 10-fold cross-validation procedure. For each iteration of the cross-validation procedure, once the GA run is over we compare the performance of C4.5 using all the original attributes with the performance of C4.5 using only the attributes selected by the GA. In both runs of C4.5, the decision tree is built using the entire training set (9 partitions), and then we measure C4.5's error rate in the test set. Therefore, the GA can be considered successful to the extent that the attributes subsets selected by it lead to a reduction in the error rate and size of the tree built by C4.5, by comparison with the use of all original attributes.

There is a final point concerning the evaluation of the solutions returned by the GA. As explained before, the solution for a multiobjective optimization problem consists of all non-dominated solutions (the Pareto front). Hence, each run of the GA outputs the set of all non-dominated solutions (attribute subsets) present in the last generation's population. In a real-world application, it would be left to the user the final choice of the solution to be used in practice. However, in our research-oriented work, involving public-domain data sets, no user was available. Hence, in order to evaluate the quality of the non-dominated attribute subsets found by the GA in an automatic, data-driven manner – as usual in the majority of the data mining and machine learning literature – we measure the error rate and the size of the decision tree built by C4.5 using each of the non-dominated attribute subsets returned by the GA.

In order to compare the proposed multiobjective GA with another multiobjective feature selection method, we propose a multiobjective version of forward sequential selection (MOFSS). The method follows the same principles of the simple FSS. It starts with an empty set of attributes and at the first iteration the solutions with one attribute are evaluated. However, instead of selecting just one best solution, a set of non-dominated solutions are selected considering the Pareto dominance principle and inserted into a list. The solutions to be evaluated in the next iteration are generated combining each of the solutions in the list with each of the remainder attributes. Again, the set of non-dominated solutions is selected and inserted into a list, and so on, until no more solutions are included in the list.

Table 2 shows results comparing the proposed multiobjective GA with C4.5 alone. Table 3 shows results comparing the proposed multiobjective FSS with the conventional FSS method. The fourth column of Table 2 shows the *total* number of non-dominated solutions found by the GA. Once these solutions (hereafter called GA-found solutions) are obtained, they are compared with the baseline solution, i.e., the set of all attributes, corresponding to the use of C4.5 alone. Hence, the last three columns of Table 2 show, respectively, the number of GA-found solutions which *dominate* the baseline solution, the number of GA-found solutions which are *dominated by* the baseline solution and the number of GA-found solutions that neither dominate nor are dominated by the baseline solution, hereafter referred to as the number of *neutral* solutions. The columns of Table 3 have an analogous meaning. The figures are an average over the 10 iterations of the cross-validation procedure (measuring error rate in the test set). The values after the "±" symbol represent the standard deviations.

Table 2. Results comparing the multiobjective GA with C4.5 alone

Data Set	C4.5 alone		MOGA solutions			
	Tree size	Error rate	Total	Dominate	Dominated	Neutral
Arrhythmia	80.2 ± 2.1	32.93 ± 3.11	3.9 ± 0.54	0.8 ± 0.38	1.3 ± 0.68	1.8 ± 0.44
Dermatology	29.0 ± 3.65	15.95 ± 1.43	1.11 ± 0.11	0.88 ± 0.17	0	0.22 ± 0.11
Vehicle	134.0 ± 6.17	26.03 ± 1.78	6.1 ± 0.76	1.5 ± 0.43	1.1 ± 0.46	3.5 ± 0.82
Promoters	23.8 ± 1.04	16.83 ± 2.55	1.5 ± 0.16	0.5 ± 0.22	0	1.0 ±0.26
Ionosphere	26.2 ± 1.74	10.2 ± 1.25	1.14 ± 0.14	0.42 ± 0.2	0.14 ± 0.14	0.57 ± 0.3
Crx	29.0 ± 3.65	15.95 ± 1.43	4.55 ± 0.67	2.55 ± 0.69	0.22 ± 0.15	1.77± 0.77

The results in Table 2 show that the number of GA-found solutions that dominate the baseline solution (5th column) is larger than the number of GA-found solutions that are dominated by the baseline solution (6th column) in 5 out of the 6 data sets. The only exception is the Arrhythmia data set, but the difference is not significant, taking into account the standard deviations. The better performance of the GA-found solutions, by comparison with the baseline solution, is particularly significant in the Dermatology and Crx data sets.

Table 3. Results comparing multiobjective FSS with conventional FSS

Data Set	FSS		MOFSS solutions			
	Tree size	Error rate	Total	Dominate	Dominated	Neutral
Arrhythmia	31.4 ± 5.35	37.37 ± 1.42	32.2 ± 10.82	17.4 ± 9.38	0	14.8 ± 8.68
Dermatology	21.6 ± 0.52	9.61 ± 2.14	76.5 ± 10.3	0	0	76.5 ± 10.3
Vehicle	88.8 ± 11.02	34.93 ± 2.11	3.6 ± 0.16	0.6 ± 0.16	0	3.0 ± 0.15
Promoters	5 ± 0	32.84 ± 5.88	66.6 ± 12.66	18.2 ± 11.43	0	48.4 ± 14.16
Ionosphere	10.2 ± 1.49	12.46 ± 1.6	12.9 ± 6.23	1.8 ± 1.21	0	11.1 ± 5.21
Crx	3 ± 0	14.46 ± 1.48	84.1 ± 2.05	75.0 ± 8.55	0	9.1 ± 9.1

The results in Table 3 show that, in all 6 data sets, none of the MOFSS-found solutions were dominated by the baseline solution (the set of all attributes). In general, MOFSS found more solutions than the GA, except in the Vehicle data set. In 2 data sets (Arrhytmia and Crx) the majority of solutions found by MOFSS dominate the baseline solution. However, in the other 4 data sets the majority of solutions found by MOFSS are neutral. These neutral solutions generally have small tree sizes but high error rates, by comparison with the baseline solution. In other words, they are concentrated in one part of the Pareto front. By contrast, the solutions found by the GA are more spread along the Pareto front, representing a better diversity of trade-offs between tree size and error rate, which gives the user more flexibility to chose among the solutions returned by the GA.

6. Conclusions and Future Work

This work proposed a multiobjective genetic algorithm (MOGA) for attribute selection and a multiobjective version of forward sequential selection (MOFSS). The computational results showed that, overall, the number of MOGA-found and MOFSS-found solutions that dominate the baseline solution (the set of all attributes) is larger than the number of MOGA-found and MOFSS-found solutions that are dominated by the baseline solution, respectively. Hence, both multiobjective methods can be considered good alternatives to the attribute selection problem. The MOGA had the advantage of finding solutions that are more spread along the Pareto front, by comparison with the solutions found by MOFSS (most of which had small tree sizes but high error rates).

We are currently doing experiments with more data sets, to further validate the results reported in this paper. In the future it would also be interesting to compute the dominance relations between the solutions found by MOGA and MOFSS. (In these paper the dominance relations of these solutions were compared only with the baseline solution.)

References

[1] S. Bhattacharyya, Evolutionary Algorithms in Data mining: Multi-Objective Performance Modeling for Direct Marketing, Proc. of KDD-2000, pp 465-471. ACM Press, 2000.

[2] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, 2001.

[3] H.Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer, 1998.

[4] M. V. Fidelis, H. S. Lopes, A.A. Freitas, Discovering Comprehensible Classification Rules with a Genetic Algorithm, Proc. Congress on Evolutionary Computation (CEC-2000). IEEE, 2000.

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

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

[7] M. Holsheimer, A. Siebes, Data Mining – The Search for Knowledge in Databases, Report CS-R9406, Amsterdam: CWI, 1994.

[8] J. R. Quinlan , C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[9] P.M. Murphy, D. W. Aha, UCI Repository of Machine Learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of information and Computer Science, 1994.