Discovering Comprehensible Classification Rules with a Genetic Algorithm

M. V. Fidelis
UEPG, CPD
Praça Santos Andrade, s/n
Ponta Grossa, PR, Brasil - 80010-330

CEFET-PR, UNED-PG
Av. Monteiro Lobato Km 4, s/n
Ponta Grossa, PR, Brasil - 84000-000
fidelis@cpgei.cefetpr.br

H. S. Lopes
CEFET-PR, CPGEI
Av. Sete de Setembro, 3165
Curitiba, PR, Brasil - 80230-901
hslopes@cpgei.cefetpr.br

A. A. Freitas
PUC-PR, PPGIA
Rua Imaculada Conceição, 1155
Curitiba, PR, Brasil - 80215-901
alex@ppgia.pucpr.br

Abstract- This work presents a classification algorithm based on genetic algorithms (GAs) that discovers comprehensible IF-THEN rules, in the spirit of data mining. The proposed GA has a flexible chromosome encoding where each chromosome corresponds to a classification rule. Although the number of genes (genotype) is fixed, the number of rule conditions (phenotype) is variable. The GA also has specific mutation operators for this chromosome encoding. The algorithm was evaluated on two public domain, real-world data sets (on the medical domains of dermatology and breast cancer).

1 Introduction

This work presents a system based on genetic algorithms (GAs) to perform the task of classification. The system is evaluated in two medical domains: diagnosis of dermatological diseases and prediction of recurrence of breast cancer. The use of GAs in classification is an attempt to effectively exploit the large search space usually associated with classification tasks. The GA presented here was designed according to some concepts of data mining and knowledge discovery, where the goal is to find not only accurate knowledge but also comprehensible knowledge [6, 8]. Hence, the GA's individuals (or chromosomes) encode IF-THEN classification rules, similarly (in form) to the rules discovered by data mining algorithms based on the rule induction paradigm.

This paper is organized as follows. Section 2 briefly reviews the basic characteristics of genetic algorithms and the classification task (from a data mining viewpoint). Section 3 describes in detail our proposed system. Section 4 briefly describes the data sets used in the experiments. Section 5 discusses the results of the experiments. Section 6 discusses related work. Finally, section 7 concludes the paper.

2 An Overview of Classification and Genetic Algorithms

The classification task is one of the most studied in data mining. In essence, the problem consists of assigning records to one out of a small set of pre-defined classes, by discovering some relationship between attributes. Each record (henceforth an example) consists of a set of predicting attributes and a goal attribute to be predicted [11, 8]. A data-mining algorithm is applied to a set of training examples, with a known class, to discover rules detecting some relationship between the predicting attributes and the goal attribute. This relationship is then used to predict the class (the value of the goal attribute) of examples whose class is unknown.

The discovered knowledge is usually represented in the form of IF-THEN prediction rules, which have the advantage of being a high-level, symbolic knowledge representation, contributing to the comprehensibility of the discovered knowledge. The discovered rules can be evaluated according to several criteria, such as the degree of confidence in the prediction, classification accuracy rate on unknown-class examples, comprehensibility, etc. We emphasize that this latter is a crucial criterion in the context of data mining.

Genetic Algorithms (GAs) are a search method that has been widely used in applications where the size of the search space is very large. In essence, GAs are "search algorithms based on the mechanics of natural selection and natural genetics" [9]. GAs are inspired on the principle of survival of the fittest, where the fittest individuals are selected to produce offspring for the next generation. In the context of search, individuals are candidate solutions to a given search problem. Hence, reproduction of the fittest individuals means reproduction of the best current candidate solutions. Genetic operators such as selection, crossover and mutation generate offspring from the fittest individuals. One of the advantages of GAs over "traditional" search methods is that the former performs a kind of global search using a population of individuals, rather than performing a local, hill-climbing search. Global search methods are less likely to get trapped into local maxima, in comparison with local search methods.

It is interesting to note that, overall, the knowledge discovery paradigm most used in data mining is still rule induction. Most of the algorithms in this paradigm perform a kind of local search.

3 The Genetic Algorithm

The GA used in this work was developed based in the GALOPPS 3.2 system [10]. This is a public-domain tool that incorporates several features proposed by GA researchers and is very portable. The next subsections describe several aspects of the proposed algorithm, namely individual encoding, genetic operators, and fitness function.

3.1 Individual Encoding

A chromosome is divided into n genes, where each gene corresponds to a condition involving one attribute, and n is the number of predicting attributes in the data being mined. The genes are positional, i.e. the first gene represents the first attribute, the second gene represents the second attribute, and so on. Each i-th gene, $i=1...n$, is subdivided into three fields: weight (W_i), operator (O_i) and value (V_i), as shown in figure 1. Each gene corresponds to one condition in the IF part of a rule, and the entire chromosome (individual) corresponds to the entire IF part of the rule. The THEN part does not need to be coded into the chromosome, as will be explained later. Therefore, henceforth we will refer to the IF part of the rule encoded into a chromosome simply as the rule.

Gene $_1$			…	…	Gene$_n$		
W_1	O_1	V_1	…	…	W_n	O_n	V_n

Fig. 1. Representation of a chromosome (individual)

The field weight (W_i) is a real-valued variable taking values in the range [0..1]. This variable indicates whether or not the corresponding attribute is present in the rule. More precisely, when W_i is smaller than a user-defined threshold (called *Limit*) the i-th condition is effectively removed from the rule. Therefore, the greater the value of the threshold *Limit*, the smaller the probability that the corresponding condition will be present in the rule. We used a *Limit* value of 0.3, so that conditions with a weight smaller than or equal to 0.3 were effectively removed from the rule. Note that mutations in the field weight can cause the corresponding attribute to be removed or re-inserted into the rule.

The field operator (O_i) is a variable that indicates the relational operator employed in the i-th condition. If attribute A_i is categorical (nominal) this field can contain the operators "=" and "≠". If attribute A_i is continuous, this field can contain the operators "≥" and "<".

The field value (V_i) contains one of the values belonging to the domain of attribute A_i. The value V_i is coded into a binary string, which is properly decoded for purposes of fitness evaluation. The number of bits used to code V_i is proportional to the number of values in the domain of attribute A_i.

Note that the above encoding is quite flexible with respect to the length of the rules. A "traditional" GA is very limited in this aspect, since it can only cope with fixed-length rules. In our approach, although each chromosome has a fixed length, the genes are "interpreted" (based the value of the weight W_i) in such a way that the individual phenotype (the rule) has a variable length. Hence, different individuals correspond to rules with different number of conditions.

3.2 Genetic Operators

We used conventional genetic operators of selection and crossover. More precisely, we used stochastic tournament selection with tournament size 3 and two-point crossover, with crossover probability = 100%. We also used an elitist reproduction strategy, where the best individual of each generation was passed unaltered to the next generation.

We developed three mutation operators tailored for our genome representation, namely weight mutation, relational-operator mutation and value mutation. Each of these operators acts on a different field of a gene - see the previous subsection. We used mutation rates of 30% for each kind of mutation.

The weight mutation was developed to modify the weight of a rule condition. This operator randomly generates a small real-valued number that is then added to or subtracted from the current weight of the condition. Hence, conditions can be removed or inserted in a rule, as the value of the weight field gets smaller or greater than the threshold *Limit*.

The relational-operator mutation modifies the relational operator currently being used in a condition of the rule, by replacing it with another one, randomly generated among the valid relational operators (depending on whether the attribute is categorical or continuous).

The value mutation modifies the contents of the value field, by replacing the current value with another one randomly generated. There are two possible cases here. First, if the attribute is categorical, this mutation simply replaces the current value field with another value belonging to the domain of the attribute. Second, if the attribute is continuous, the mutation produces a small number that is then added to or subtracted from the current contents of the value field. This is implemented in such a way that the lower and upper bounds of the domain of the attribute are never exceeded.

3.3 Fitness Function

The fitness function evaluates the quality of each rule (individual). This work uses the fitness function employed by [15]. Before we can define the fitness function, it is necessary to recall a few basic concepts on classification-rule evaluation. When using a rule for classifying an example, depending on the class predicted by a rule and on the true class of the example, four different types of results can be observed for the prediction, as follows:

- true positive (tp) - the rule predicts that the patient has a given disease and the patient really have that disease;
- false positive (fp) – the rule predicts that the patient has a given disease but the patient does not have it;
- true negative (tn) - the rule predicts that the patient does not have a given disease, and indeed the patient does not have it;
- false negative (fn) - the rule predicts that the patient does not have a given disease but the patient does have it.

Our fitness function combines two indicators commonly used in medical domains, namely the sensitivity (*Se*) and the specificity (*Sp*), defined as follows:

$$Se = tp \,/\, (tp + fn) \qquad (1)$$

$$Sp = tn \,/\, (tn + fp) \qquad (2)$$

Finally, the fitness function used by our system is defined as the product of these two indicators, i.e.:

$$fitness = Se * Sp \qquad (3)$$

Therefore, the goal of our system is to maximize both the *Se* and the *Sp* at the same time, and the product shown in equation (3) provides a good gradient for the function.

Each run of our GA solves a two-class classification problem, where the goal is to predict whether or not the patient has a given disease. Therefore, the GA is run at least once for each class (value of the goal attribute). For instance, supposing that the application domain has 6 classes, we need to run the GA at least 6 times. In the first run the GA would search for rules predicting class 1; in the second run it would search for rules predicting class 2, and so on. When the GA is searching for rules predicting a given class, all other classes are effectively merged into a large class, which can be conceptually thought of as meaning that the patient does not have the disease predicted by the rule. Hence, the above formulas for *Se* and *Sp* can be applied to problems with any number of classes.

This characteristic of our GA also explains why it is not necessary to encode the class predicted by a rule into the chromosome representation, as mentioned in section 3.1. In effect, in a given run of the GA all individuals are searching for rules predicting the same class.

4 Data Sets Used in the Experiments

We did some experiments with two public domain data sets, in the medical domains of dermatology and breast cancer. These data sets were obtained from the UCI (University of California at Irvine) - Machine Learning Repository [17]. These data sets have been used extensively for classification tasks using different paradigms, see, for instance [3] and [5]. The main characteristics of each of these domains are described in the next two subsections.

4.1 Dermatology Data Set

The differential diagnosis of the disease erythemato-squamous is an important problem in dermatology. There are six different diagnoses (six classes), and all of them share some clinical characteristics of erythema and scaling, with few differences. The six classes are: psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis and pityriasis rubra pilaris. Some characteristics are more incident in certain diseases, but they can also appear in some stages of development of other diseases, making more difficult the diagnosis. Some characteristics of the diseases are discussed in [5].

This data set contains 366 records, each one with 34 attributes. All attributes had their values mapped to a four-valued graded scale in the range [0...3], where 0 indicates the lack of the corresponding characteristic and 3 indicates a great incidence of that characteristic. Two exceptions are the attribute age, which remains with its values expressed in years, and the attribute family history, which takes on the value 1 when the disease has been observed in the family of the patient and 0 otherwise.

4.2 Breast Cancer Data Set

Breast cancer re-occurs in up to 30% of the patients that undergo a breast cancer surgery [3]. This data set contains 286 records, each with 9 attributes, and the goal is to determine the patients for whom the cancer will re-occur. Hence, there are only two classes, namely no-recurrence-events and recurrence-events. All attributes are categorical. More detailed information about this data set can be found in [3].

5 Computational Results

Each GA run consisted of a population of 50 individuals evolving during 50 generations. The set of parameters used for the genetic operators (crossover and mutation rates, etc.) were as defined in section 3.

Each data set was randomly partitioned into two parts, with 2/3 of the records used for training and 1/3 of the record used for testing the quality of the discovered rules. As usual in the literature, this partition was done in such a way that the proportion of examples belonging to each class (the relative frequency of the class) was kept the same in both the training and the test set. Since the current version of our algorithm cannot cope with missing values, in each data set the few records that contained missing values were simply removed, for the purposes of the experiments reported below.

5.1 Results for the Dermatology Data Set

Table 1 presents the final 6 rules discovered by the GA - one rule for each class. For each class, the GA was run three times, varying the random seed used to generate the initial population. The best rule of the three runs, according to its fitness value measured on the training set, was selected as the rule predicting that class (this is the rule shown in Table 1). Once the 6 rules were selected, they were evaluated on a separate test set, as mentioned above. Note that the set of rules used for classification also includes a "default" rule, i.e. a rule with no conditions which is automatically applied when no other rule has its conditions satisfied by the example to be classified. This rule simply predicts class C1, which is the "majority" class - i.e. the most frequent class in the training set.

Table 1. Discovered Rule Set for the Dermatology Data Set

C	Rule	*Fitness*
1	IF (*clubbing of the rete ridges*)≥1 AND (*perifollicular parakeratosis*)=0	0,973-0,973
2	IF (*koebner phenomenon*)=0 AND (*vacuolisation and damage of basal layer*)<1 AND (*spongiosis*)≥2	0,855-0,855
3	IF (*band-like infiltrate*)≥2	1,000-0,979
4	IF(*knee and elbow involvement*)=0 AND (*family history*)=0 AND (*acanthosis*)<3 AND (*focal hypergranulosis*)<2 AND (*spongiosis*)≥1 AND (*inflammatory monoluclear infiltrate*)≥1	0,860-0,783
5	IF (*melanin incontinence*)<1 AND (*fibrosis of the papillary dermis*)≠0 AND (*munro microabcess*)=0	1,000-1,000
6	IF (*follicular papules*)≥1 AND (*perifollicular parakeratosis*)≥1	1,000-1,000

For each rule in Table 1 the third column shows two values, namely the fitness of the rule - computed by equation (3) - in the training set and in the test set, respectively. One can see that all the rules discovered from the training data generalize well for examples in the test set. In most cases the fitness in the test set is nearly equal to the fitness in the training set. The only exception is the rule for class C4, where the fitness of 86% in the training set dropped to a fitness of 78.3% in the test set.

The fitness values reported in Table 1 are useful for evaluating the performance of each rule separately. However, it is also important to evaluate the performance of the rule set as a whole. As usual in the literature, this evaluation was done by measuring the accuracy rate on the test set, i.e. the ratio of the number of examples correctly classified over the total number of examples in the test set. The accuracy rate of the rule set was 95% (out of 119 examples, 113 were correctly classified).

We also observed that in 5 out of the 6 classes the same rule conditions were discovered by the three GA runs for that class. Since the different random seeds could lead to very different results across the three runs, we believe that the algorithm has reliably identified the main predicting attributes for each class. However, this conclusion needs to be validated by more experiments.

5.2 Results for the Breast Cancer Data Set

Table 2 shows the final two rules discovered by the GA - one rule for each class. We followed the same procedure as described in section 5.1, running the GA three times for each class.

Table 2. Discovered Rule Set for the Breast Cancer Data Set

C	Rule	*Fitness*
1	IF (*inv-nodes*)<(9-11) AND (*deg-malig*)<2 AND (*irradiat*)=(no)	0,564-0,365
2	IF (*menopause*)≥(ge40) AND (*tumor-size*)≠(45-49) AND (*deg-malig*)≥2	0,497-0,393

For each rule in Table 2 it is shown two values, namely the fitness of the rule - computed by equation 3 - in the training set and in the test set, respectively. This time, however, the rules discovered from the training set did not generalize so well for examples of the test set. This seems to be due to the fact that this is a considerably more difficult classification problem, in comparison with the dermatology data set, and the data is quite noisy.

Anyway, it is interesting to evaluate the performance of the set of discovered rules as a whole, by measuring the accuracy rate, as done in the previous section. The accuracy rate of the discovered rule set was 67% (out of 92 examples, 62 were correctly classified).

Similarly to the results of section 5.1, for each class the same rule conditions were discovered by the three GA runs for that class, showing that the GA converged to the same best rule despite variations in the random seed.

6 Related Work

Several GAs designed for discovering some kind of comprehensible classification rules have been proposed in the literature. We briefly review some of them below.

GIL [12] uses several generalization/specialization operators proposed by [16] to extend the genetic operators of conventional GA, creating a knowledge- intensive GA for the classification task. GIL follows the Pittsburgh's approach for rule learning, where each individual of the population corresponds to a set of rules. GABIL [4] also follows the Pittsburgh's approach and also suggests a few task-specific genetic operators. Yet another project following the Pittsburgh's approach is HDPDCS [19]. This system was explicitly designed for both classification and feature extraction from high dimensionality data sets.

A major difference between our system and the above projects is that we do not follow the Pittsburgh approach. In our system, an individual corresponds to a single rule, which at least tends to be more computationally efficient. This is usually known as the Michigan approach.

REGAL learns first-order-logic (FOL) rules [18]. The discovery of FOL rules is an interesting feature shared by few GAs, but in the case of REGAL it requires that the user provides a kind of template of the logical formula to be learned. This requirement reduces the autonomy of the system, which can be considered a drawback in the context of data mining. Several parallel GAs are descendant from REGAL, e.g. the G-NET system [1].

SIAO1 [2] also learns FOL class descriptions, but it does not need a user-specified formula template. Instead, it adapts the technique of generalizing a seed example to learn FOL classification rules [16]. From a data-mining viewpoint, it seems that the main drawback of SIAO1 is that this kind of generalization technique is quite computationally expensive, making it difficult to apply SIAO1 to large databases.

Both REGAL and SIAO1 are very different from our system, since the latter discovers conventional propositional logic rules, rather than FOL ones.

A system somewhat more similar to ours is EDRL [13]. This system also requires one run of the GA for discovering a rule for each class. In addition, in this system each chromosome is represented as a conjunction of conditions of a single rule, similarly to our system. However, this system cannot cope with continuous attributes, i.e. it assumes that continuous attributes have been previously discretized as a pre-processing step. Our system does not have this limitation, i.e. it copes with both categorical and continuous attributes. It chould be noted, however, that a more recent version of EDRL copes with continuous attributes [14].

Another GA for discovering rules following the approach of associating a chromosome with a conjunction of conditions of a single rule is described in [7]. However, this system does not address the classification task. Rather, it addresses the dependence modeling (or generalized rule induction) task. The latter can be regarded as a generalization of the former, since different rules can predict different goal attributes, rather than a single goal (or class) attribute.

7 Conclusions and Future Work

The preliminary results reported in this paper are promising, and allow us to conclude that our chromosome encoding and its associated rule set representation are a good alternative for extracting a small set of comprehensible rules, which is important in the context of data mining.

The accuracy rates achieved by our GA in the dermatology data set (95%) and in the breast cancer data set (67%) are similar to the ones reported by other researchers in these data sets. However, our GA seems to be particularly effective in finding a concise set of comprehensible rules, since (by design) it discovers only a single rule for each class. Other data mining algorithms often discover several rules for a single class, which makes it difficult for the user to understand the numerous discovered rules.

Future work should consist of more experiments with other data sets, as well as more elaborated experiments to optimize several parameters of the algorithm, such as mutation rates, the Limit threshold for the weight field, etc. (Recall that the results of this paper were achieved without any serious attempt to optimize the parameters of the GA.)

Acknowledgments

This breast cancer domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. Thanks go to M. Zwitter and M. Soklic for providing the data.

Thanks to H. A. Guvenir from Bilkent University and N. Ilter from Gazi University, Sukara, Turkey, for providing the dermatology data set.

Bibliography

[1] Anglano, C.; Giordana, A.; Lo Bello, G.; Saitta, L. A network genetic algorithm for concept learning. *Proc. 7th Int. Conf. Genetic Algorithms*, p. 434-441, Morgan

Kaufmann, 1997.

[2] Augier, S.; Venturini, G.; Kodratoff, Y. Learning first order logic rules with a genetic algorithm. *Proc. 1st Int. Conf. Knowledge Discovery & Data Mining*, p. 21-26. Montreal, Canada, 1995.

[3] Clark, P., Niblett, T. Induction in Noisy Domains. In: *Progress in Machine Learning* (from the Proceedings of the 2nd European Working Session on Learning), p. 11-30, Sigma Press, 1987.

[4] DeJong, K.A.; Spears, W.M.; Gordon, D.F. Using genetic algorithms for concept learning. *Machine Learning*, v. 13, p. 161-188, 1993.

[5] Demiroz, G.; Govenir, H.A.; Ilter N. Learning differential diagnosis of erythemato-squamous diseases using voting feature. *Artificial Intelligence in Medicine*, v. 13, p. 147-165, 1998.

[6] Fayyad, V.M.; Piatetsky-Shapiro, C.; Smyth, P. From data mining to knowledge discovery: an overview. Fayyad, V.M. et al. (Eds.) *Advances in knowledge Discovery and Data Mining*, p. 1-34, AAAI/MIT Press, 1996.

[7] Freitas, A.A. A genetic algorithm for generalized rule induction. R. Roy et al. (Eds.) *Advances in Soft Computing – Engineering Design and Manufacturing*, p. 340-353. Springer-Verlag, 1999.

[8] Freitas, A.A.; Lavington, S.H. *Mining Very Large Databases with Parallel Processing*, London: Kluwer Academic Publishers, 1998.

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

[10] Goodman, E.D. *An Introduction to GALLOPS - The Genetic Algorithms Optimized for Portability and Parallelism System*. East Lansing, USA: Genetic Algorithms Research and Applications Group (GARAGe), Department of Computer Science, Michigan State University, 1996.

[11] Hand, D. *Construction and Assessment if Classification Rules*. Chichester: John Wiley & Sons, 1997.

[12] Janikow, C.Z. A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning*, v. 13, p. 189-228, 1993.

[13] Kwedlo, W.; Kretowski, M. Discovering of decision rules from databases: an evolutionary approach. *Principles of Data Mining and Knowledge Discovery. Lecture Notes in Artificial Intelligence* 1510, p. 370-378, 1998.

[14] Kwedlo, W.; Kretowski, M. An Evolutionary algorithm using multivariate discretization for decision rule induction. *Principles of Data Mining and Knowledge Discovery - Lecture Notes in Artificial Intelligence* 1704, p. 392-397, 1999.

[15] Lopes, H.S.; Coutinho, M.S.; Lima, W.C. An evolutionary approach to simulate cognitive feedback learning in medical domain. In: Sanchez, E., Shibata, T., Zadeh, L.A. (Eds) *Genetic Algorithms and Fuzzy Logic Systems*. Singapore: World Scientific, p.193-207, 1997.

[16] Michalski, R.W. A theory and methodology of inductive learning. *Artificial Intelligence*, v. 20, p. 111-161, 1983.

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

[18] Neri, F.; Giordana, A. A parallel genetic algorithm for concept learning. *Proc. 6th Int. Conf. Genetic Algorithms*, p. 436-443, 1995.

[19] Pei, M.; Goodman, E.D.; Punch III, W.F. Pattern discovery from data using genetic algorithms. *Proc. 1st Pacific-Asia Conf. Knowledge Discovery and Data Mining*, 1997.