DOI 10.1007/s10115-003-0143-4
Springer-Verlag London Ltd. © 2004
Knowledge and Information Systems (2005) 7: 179–201

Knowledge and
Information Systems

An evolutionary approach for automatically extracting intelligible classification rules

I. De Falco[1], A. Della Cioppa[2], A. Iazzetta[3], E. Tarantino[1]

[1]ICAR, National Research Council of Italy, Naples, Italy
[2]Department of Computer Science and Electrical Engineering, University of Salerno, Fisciano (SA), Italy
[3]IM, National Research Council of Italy, Naples, Italy

Abstract. The process of automatically extracting novel, useful and ultimately comprehensible information from large databases, known as data mining, has become of great importance due to the ever-increasing amounts of data collected by large organizations. In particular, the emphasis is devoted to heuristic search methods able to discover patterns that are hard or impossible to detect using standard query mechanisms and classical statistical techniques. In this paper an evolutionary system capable of extracting explicit classification rules is presented. Special interest is dedicated to find easily interpretable rules that may be used to make crucial decisions. A comparison with the findings achieved by other methods on a real problem, the breast cancer diagnosis, is performed.

Keywords: Data mining; Classification; Evolutionary Algorithms; Breast cancer diagnosis

1. Introduction

During the last decade, we have seen an explosive growth in our capabilities to collect data, thanks to the availability of cheap and effective storage devices. The advances in data collection have generated an urgent need for techniques that can intelligently and automatically analyse and mine knowledge from huge amounts of data. The progress in knowledge discovery brings together the latest research in statistics, databases, machine learning and artificial intelligence that are part of the exciting and rapidly growing field of data mining (Fayyad et al. 1996).

The term data mining is normally used to refer to the process of searching through a large volume of data to discover interesting and useful information. The core of this process is the application of machine learning-based algorithms to databases. There are two basic ways of performing data mining and data analysis: the supervised and

Received 14 February 2003
Revised 21 May 2003
Accepted 19 June 2003
Published online 8 April 2004

the unsupervised learning. Supervised learning exploits known cases that show or imply well-defined patterns to find new patterns by means of which generalizations are formed. In unsupervised learning, data patterns are found starting from some characterization of the regularities in a set of data.

Classification is perhaps the most commonly applied data mining technique. It employs a set of preclassified examples to develop a model, which generates a set of grouping rules by means of which a new object may be categorized. There are different classification techniques used to extract relevant relationships in the data, ranging from symbolic learning implementation (Quinlan 1986) to neural networks (Rumelhart et al. 1986). Though these classification tools are algorithmically strong, they require significant expertise to work effectively and do not provide intelligible rules.

The classification problem becomes very hard when the number of possible different combinations of parameters is so high that techniques based on exhaustive searches of the parameter space rapidly become computationally infeasible. Packard has shown in Breeden and Packard (1992) how learning and optimization algorithms can be used to produce optimal modeling of experimental data in the absence of previous theoretical explanations. Moreover, the self-adaptability of Evolutionary Algorithms is extremely appealing for information-retrieval applications. Thus, it is natural to devote attention to a heuristic approach to find a good-enough solution to the classification problem. In this paper, the objective is to exploit the capability of Evolutionary Algorithms to search easily comprehensible classification rules.

The paper is organized as follows: in Sect. 2, a brief review of the state of the art of classification methods is illustrated. In Sect. 3, an automatic classification system based on an evolutionary algorithm is presented together with implementation details. Section 4 describes the real problem faced, the breast cancer diagnosis, while Sect. 5 contains the performance of our system compared with that achieved by other methods. In the last section, final remarks and future work are outlined.

2. State of the art

Information mining and knowledge discovery from large databases have been recognized as a key research topic in database systems and machine learning. Since the late 1980s, knowledge-based techniques have been used extensively by information science researchers. These techniques have attempted to capture searchers' and information specialists' domain knowledge and classification scheme knowledge, effective search strategies and query refinement heuristics in document retrieval systems design (Chen and Dhar 1991). Despite their usefulness, systems of this type are considered performance systems – they only perform what they were programmed to do (i.e., they are without learning ability). Significant efforts are often required to acquire knowledge from domain experts and to maintain and update the knowledge base.

A newer paradigm, generally considered to be the machine learning approach, has attracted attention of researchers in artificial intelligence, computer science, and other functional disciplines such as engineering, medicine and business (Michalski 1983; Carbonell et al. 1993; Weiss and Kulikowski 1991). In contrast with performance systems, which acquire knowledge from human experts, machine learning systems acquire knowledge automatically from examples, i.e., from source data. The most frequently used techniques include symbolic, inductive learning algorithms such as

ID3 (Quinlan 1986), which uses a fixed number of generalization values, multiple-layered feedforward neural networks such as Backpropagation networks (Rumelhart et al. 1986) that can, in principle, produce many more interpolation values not present in the training cases, and Genetic Algorithms (GAs) (Holland 1975; Goldberg 1989). Many information science researchers have started to experiment with these evolutionary techniques as well (Gordon 1988; Belew 1989; Chen and Lynch 1992; Chen et al. 1993). A classification of the data mining techniques and a comparative study of such techniques can be found in (Holsheimer and Siebes 1994; Chen et al. 1996).

Data classification represents an important theme in data mining (Fayyad et al. 1996) and it has been studied in statistics, machine learning, neural networks and expert systems (Weiss and Kulikowski 1991). Several classification methods have been proposed. Those based on decision trees (Quinlan 1986, 1993) operate performing a successive partitioning of cases until all subsets belong to a single class. This operating way is impracticable except for trivial data sets. Other data classification techniques include statistical and rough sets approaches (Fayyad et al. 1996; Ziarko 1994) and neural networks (Lu et al. 1995; Hung et al. 2001). Most data mining related GAs proposed in the literature address the task of rule extraction in propositional and first-order logics (Giordana et al. 1994; Augier et al. 1995; Neri and Giordana 1995; De La Iglesia et al. 1996; Anglano et al. 1997; Noda et al. 1999). A further interesting GA–based method for choosing an appropriate set of fuzzy if–then rules for classification problems can be found in Ishibuchi et al. (1995), while in Salim and Yao (2002), an innovative evolutionary algorithm to knowledge discovery in databases by evolving SQL queries has been presented. Hybrid classification learning systems involve a combination of artificial neural networks with evolutionary techniques (Yao and Liu 1997) and with linear discriminant models (Fogel et al. 1998), and an integration of rule induction and lazy learning (Lee and Shin 1999). Furthermore, Genetic Programming (Koza 1992) frameworks for discovering comprehensible classification rules have been investigated (Freitas 1997; Ngan et al. 1998; Bojarczuk et al. 1999; Brameier and Banzhaf 2001).

3. The evolutionary approach

Our aim is the implementation of an evolutionary system able to acquire information from databases and extract intelligible classification rules for each available class, given the values of some attributes, called predicting attributes. Each rule is constituted by conditions on the predicting attributes. These conditions determine a class description which can be used to construct the classification rule.

Given a number of attributes for each object and its related domain, it is easily understandable that, for complex classification problems, the number of possible descriptions is enormous. An exhaustive search by enumerating all the possible descriptions is computationally impracticable. Hence, we appeal to heuristic search techniques. In our case, evolutionary approaches based on variants of GAs and Breeder Genetic Algorithms (BGAs) (Mühlenbein and Schlierkamp-Voosen 1993) have been used.

The basic idea is to consider a population composed by individuals each representing a single candidate rule, and to gradually improve the quality of these rules by constructing new fitter rules until either rules of sufficient quality are found or no further improvements occur. The major steps of this evolutionary system can be formalized as follows:

1. Generate at random an initial population of rules representing potential solutions to the classification problem.
2. Evaluate each rule on the basis of an appropriate fitness function.
3. Select the rules to undergo the mechanism of reproduction.
4. Apply the genetic operators, such as recombination and mutation, to generate new rules.
5. Reinsert these offspring to create the new current population.
6. Repeat steps 2 to 5 until either correct (see Sect. 3.3) classification rules are found or a fixed maximum number of generations has been reached.

To construct the classification model, data is partitioned into two sets: the training and the test sets. The training set contains the known objects used during the evolution process to find one explicit classification rule able to separate an instance of a class from instances of all other classes, while the test set is used to evaluate the generalization ability of the rule found. It should be observed that, for a multiple-class problem, the system needs as many rules as the number of classes, say c. Thus, the training phase consists in running c times the system in order to find these rules, each of which establishes the related membership class. The found rules are used to predict the class of the examples in the test set. If for an example only one rule is applicable, i.e., all its conditions are satisfied, the example is assigned to the class predicted by the rule. Instead, if more or no rules are applicable, the example is classified as indeterminate by our system.

3.1. Encoding

A single rule is defined by a genetic encoding, in which each genotype codes for the different attributes. The phenotype is the classification rule itself. This rule is constituted by a number of conditional clauses, in which conditions on some attributes are set, and by a predictive clause representing the class. A class together with its description forms a classification rule 'if <description> then <class>'. The conditional part of the rule is formed by the conjunction (logical AND) of all the active conditional clauses. This choice is a limitation to the expressive power of the rule and it is due to the chosen encoding. Actually, this limitation could be overcome by letting the conjunctions evolve within a set containing AND ($\wedge$), OR ($\vee$) and NOT ($\neg$). However this would make the chromosome handling much more troublesome. In fact, the use of further connectives would require the introduction of delimiting symbols such as parentheses in order to ensure rule consistency. Moreover, this would imply variable-sized chromosomes.

It is easily comprehensible that the optimal search of a classification rule includes two tightly coupled subproblems: the search of the more discriminating attributes and the search of the variation interval within the domain of these attributes. Then it is necessary to provide an encoding able to represent a rule with conditions on any number of available attributes and to specify which types of conditions we can establish on a generic attribute A_i. In our case, the domains can vary in an integer or a real range according to the chosen database. We have considered four types of possible conditions:

$$
\begin{array}{ll}
A_i \in [k_{i1}, k_{i2}] & (COND1) \\
A_i \leq k_{i1} & (COND2) \\
A_i \geq k_{i2} & (COND3) \\
(A_i \leq k_{i1}) \vee (A_i \geq k_{i2}) & (COND4)
\end{array}
$$

Table 1. An example of the interval and the condition vectors

Interval vector									
4.2	8.7	2.1	0.8	12	89	67	6.5	7.5	61.7

Condition vector					
1	3	0	2	4	2

where k_{i1} and k_{i2} are numerical constants related to attribute A_i. This means that we have made reference to 0 order logic. Also this is a limitation due to the evolutionary algorithm and to the chosen encoding.

The encoding must consider the absence or the presence of a condition on an attribute and, in the latter case, the condition type is to be specified. The genotype of each individual in the population is represented by using two vectors. The first vector, called *interval vector*, is constituted by a number of loci which is twice the number of attributes. They contain in sequence for each attribute A_i pairs of numerical values v_{i1} and v_{i2} representing the current extremes of the variation interval. The second vector, named *condition vector*, has a number of loci equal to the number of attributes. Each allele of this vector can take on five values ($0 \div 4$), indicating five possible variants on the corresponding attribute condition. Namely, with reference to the aforementioned condition types, if the value in the *i*th locus is 0, there is absence of the condition for the *i*th attribute A_i; if it is 1, it means that there is a condition of type ($COND1$) and so on. The values k_{i1} and k_{i2} indicated in the conditions are tied to the values v_{i1} and v_{i2} of the first vector by means of the following relationships: $k_{i1} = min\{v_{i1}, v_{i2}\}$ and $k_{i2} = max\{v_{i1}, v_{i2}\}$. Finally, in the last position, the condition vector contains a further element representing the class. Supposing there are only five attributes, indicated with $A_1, \dots, A_5$, and the interval and the condition vectors are as in Table 1, the classification rule can be interpreted as follows:

$$\text{if } (A_1 \in [4.2, 8.7]) \wedge (A_2 \geq 2.1) \wedge (A_4 \leq 6.5) \wedge ((A_5 \leq 7.5) \vee (A_5 \geq 61.7))$$
$$\text{then } C_2$$

where C_2 is the class labelled with the value 2.

3.2. Genetic operators

As concerns the genetic operators, apart from the crossover and mutation extended to other representation languages with m-ary rather than binary alphabets, recombination and mutation operators able to directly deal with real variables have been taken into account.

These last operators are those typical of BGAs (Mühlenbein and Schlierkamp-Voosen 1993). In particular, as far as the recombination operator is concerned, the Discrete Recombination (DR), the Extended Intermediate Recombination (EIR) and the Extended Line Recombination (ELR) have been investigated. For the mutation operator, the Discrete Mutation (DM) and the Continuous Mutation (CM) have been considered. A detailed description of how these operators work can be found in Mühlenbein and Schlierkamp-Voosen (1993, 1994).

3.3. Fitness function

We are looking for classification rules and different criteria can be used to evaluate the fitness of a rule. However, in an evolutionary search, this fitness must encapsulate as much as possible the desired features. Each individual in the population is a possible class description, that is to say, a set of conditions on attributes of the objects to classify. Denoting with D the set of all possible descriptions for a given class, to each description d in D corresponds a subset of the training set S, denoted with $\sigma_D(S)$, i.e., the set of points where the conditions of a rule are satisfied, and a size of the class C representing the points where the prediction of the rule is true. Intuitively, a description is correct if it covers all positive and none of the negative examples. During the iterative process, the search system will encounter many incorrect descriptions, yet useful as components for new, and hopefully better, descriptions. The concept of correctness needs to be extended to be able to select the most promising descriptions out of a set of incorrect ones. Thus, for each description d for a class C, we recall the definition of the *accuracy* ϕ as:

$$\phi \triangleq \frac{\sigma_D(S) \cap C}{\sigma_D(S)} \tag{1}$$

and *coverage* γ as:

$$\gamma \triangleq \frac{\sigma_D(S) \cap C}{C} \tag{2}$$

The *accuracy* of a description represents the probability that an object covered by the description belongs to the class while the *coverage* is the probability that an object belonging to class C is covered by the description D. Moreover, on the basis of these values, the following kinds of rules can be distinguished:

- *Complete rules*: the rule is complete if γ is equal to 1, that means any object belonging to the class is covered by the description for this class, i.e., $C \subseteq \sigma_D(S)$.
- *Consistent rules*: the rule is consistent if ϕ is equal to 1, that is, any object covered by the description belongs to the class, i.e., $C \supseteq \sigma_D(S)$.
- *Correct rules*: the rule is correct if both the classification accuracy and the coverage are equal to 1, i.e., if $\sigma_D(S) = C$.

We can use the correctness-criterion as a fitness function f_c. A value 1 is assigned to f_c if the description is correct, while its value for any incorrect rule is smaller than 1. Piatesky-Shapiro (1991) proposes principles for the construction of f_c which assigns a numerical value indicating the correctness of any description d in the description space D. The correctness depends on the size of $\sigma_D(S)$, covered by the description, the size of the class C and the size of their overlapping region $\sigma_D(S) \cap C$.

The simplest function to evaluate the fitness of a rule is:

$$f_c = |\sigma_D(S) \cap C| - \frac{|\sigma_D(S)||C|}{|S|} \tag{3}$$

This function can be intuitively understood as the difference between the actual number of examples for which the rule classifies properly and the expected number if C were independent of D. It assumes its maximum value when the examples belonging to C are all and only those that satisfy the condition of D, that is to say, when $\sigma_D(S) = C$. In this case:

$$f_{c_{max}} = |C| - \frac{|C||C|}{|S|} \tag{4}$$

Another possible fitness function is that proposed by Radcliffe and Surry (1994). It takes into account a correction term that measures how statistically meaningful a rule is, as suggested in Holsheimer and Siebes (1994). The model proposed is:

$$f_c^* = (log(1 + |\sigma_D(S) \cap C|) + log(1 + |\sigma_{D'}(S) \cap C'|)) \cdot \left(\phi - \frac{|\sigma_{D'}(S) \cap C|}{|\sigma_{D'}(S)|}\right) \tag{5}$$

where $\sigma_D{}'(S)$ is the set of the points in the database in which the conditions are not satisfied while C' is the set of the points in which the prediction of the rule is false. Equation (5) assumes its maximum value when $\phi = 1$ and $\sigma_{D'} \cap C = 0$:

$$f_{c_{max}}^* = log(1 + |\sigma_D(S) \cap C|) + log(1 + |\sigma_{D'}(S) \cap C'|) \tag{6}$$

By looking at the fitness functions reported both in (3) and in (5), it is clear that they increase with coverage and accuracy. Consequently, this also guarantees an improvement in terms of completeness and consistency.

Apart from these statistical considerations, the quality function could also take some other factors into account. Keeping in mind that most data mining systems rely on Ockham's razor (Derkse 1993) ("the simpler a description, the more likely it is that it describes some really existing relationships in the database"), we have decided to add further terms to yield a more discriminating fitness function. In particular, we have considered two quantities that take into account in some way the simplicity and the compactness of the description.

The concept of simplicity is incorporated in the function f_1 and it is related to the number of conditions. Namely:

$$f_1 = 1 - \frac{n}{n_{max}} \tag{7}$$

where n is the number of the conditions active in the current description and n_{max} is the maximum number of conditions that, in our encoding, corresponds to the number of database attributes. Its goal is to prefer the rules with a lower number of conditions.

The compactness is considered in the function f_2. For each condition active in the current rule, the ratio between the range of the corresponding attribute and the range of the attribute domain is evaluated. The function f_2 contains the sum of these n ratios divided by their number. This factor varies in [0.0, 1.0] and gives an indication on the width of the intervals for the conditions present in the rule. The function f_2 can be formalized as follows:

$$f_2 = 1 - \frac{1}{n}\sum_{i=1}^{n}\frac{\delta_i}{\Delta_i}$$

where $\Delta_i = (max_i - min_i)$ is the range of the domain of the ith attribute and δ_i is given by:

$$\delta_i = \begin{cases} k_{i2} - k_{i1} & \text{if the condition is of type} \quad (COND1) \\ k_{i1} - min_i & \text{if the condition is of type} \quad (COND2) \\ max_i - k_{i2} & \text{if the condition is of type} \quad (COND3) \\ \Delta_i - (k_{i2} - k_{i1}) & \text{if the condition is of type} \quad (COND4) \end{cases}$$

where k_{i1} and k_{i2} are the same as in Sect. 3.1. Its aim is to favour the rules with more restrictive conditions.

The total fitness function f_{tot} considered during the training phase is then the sum of three terms:

$$f_{tot} = \frac{1}{k}\,(f_{stat} + p_1 f_1 + p_2 f_2) \tag{8}$$

with

$$f_{stat} = \begin{cases} \dfrac{f_v}{f_{v_{max}}} & \text{if} \quad f_v > 0 \\ 0 & \text{if} \quad f_v \leq 0 \end{cases}$$

where f_v corresponds to (3) or (5) for the linear and the logarithmic fitness functions, respectively, $f_{v_{max}}$ represents the best value that f_v can assume in the ideal case, while $k = \frac{1}{1+p_1+p_2}$ represents a normalization factor. The weights p_1 and p_2 must assume values much lower than 1 which is the assigned weight for f_{stat}. This is in order not to affect too much the evaluation of the description which must take into account the correctness above all. The function f_{stat} is normalized in [0.0, 1.0]. With these choices, the problem becomes a maximisation task. It should be noted that the chosen evaluation mechanism does not guarantee to find the single best rule describing the class under consideration. This is why it is based on some subjective criteria, and even if a perfect evaluation mechanism could be devised, a selection of rules could be necessary for representing different instances of patterns within the database.

4. The problem

In order to exploit the evolutionary approach ability to face a classification task, an evolutionary system has been implemented and applied to one of the most important real problems in the medical domain, i.e., the breast cancer problem. The purpose is to find intelligible rules to classify a tumour as either benign or malignant.

Breast cancer data sets were originally obtained from W.H. Wolberg at the University of Wisconsin Hospitals, Madison. We have considered two data sets. The first contains 10 integer-valued attributes, of which the first is the diagnosis class, while the other nine attributes are related to cell descriptions gathered by microscopic examination (Wolberg and Mangasarian 1990). All these attributes have values in the set $\{1, 2, \ldots, 10\}$. The data set is constituted by 699 examples, of which 458 are benign examples and 241 are malignant examples. In the following, this database will be denoted as CANCER1a. It should be noted that this database contains 16 missing attribute values. If we omit the examples with missing attributes, the total number of instances becomes 683, of which 444 are benign and 239 are malignant. This database without missing values will be called CANCER1b.

The second data set contains 569 instances, of which 357 are diagnosed as benign and the remaining 212 are known to be malignant. These data have been obtained by means of an image analysis system developed at the University of Wisconsin. First, a fine-needle aspirate (FNA) (Mangasarian et al. 1995) is taken from a lump in a patient's breast. Then the fluid from the FNA is placed onto a glass slide to highlight the nuclei of the cells. An area of the slide is considered to generate a digitized image. Ten real-valued features are computed for each cell nucleus. The mean, standard error and worst or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features in addition to the diagnosis class. This database will be called CANCER2.

Note that we consider a two-class problem. Nonetheless, this is not restrictive because each multiple-class classification problem can be reduced to a two-class problem. In fact, in the case of multiple classes, during the search of the rules predicting a given class, all the other classes can be conceptually thought of as merged into a larger class containing the examples that do not belong to the class predicted. The breast cancer problem is intended as a test to evaluate the effectiveness of the approach proposed.

4.1. Related work

The breast cancer problem has been faced by means of different techniques. As concerns the CANCER1 data set, initially the classification was performed by linear programming methods (Mangasarian et al. 1990; Bennett and Mangasarian 1992). Prechelt (1994) showed the results obtained with manually constructed artificial neural networks and Setiono and Hui (1995) used a new neural algorithm called FNNCA. A comparison with these results is effected by Yao and Liu (1997) who present a new evolutionary system, i.e., EP-Net, for evolving artificial neural networks and compare their results with those attained in Prechelt (1994); Setiono and Hui (1995). These approaches have the disadvantage of lacking explicit rules. In Sherrah et al. (1997) the authors proposed a system that can perform both feature selection and feature construction, but they still do not focus on the discovery of comprehensible rules. Taha and Ghosh, in Taha and Ghosh (1997), have exploited rule extraction techniques from trained feedforward neural networks while Peña–Reyes and Sipper, in Peña and Sipper (1999), have combined fuzzy systems and Evolutionary Algorithms to provide comprehensible classification rules.

Linear programming techniques (Mangasarian et al. 1995; Fung and Mangasarian 1999) and machine learning methods (Hung et al. 2001; Wolberg et al. 1995) have been applied to breast cancer diagnosis and prognosis using the real-valued CANCER2 data set.

5. Experimental results

The evolutionary system works on the training set only. At the end of the training phase the best rules found are evaluated on the test set. The system allows attaining two rules covering the benign and the malignant cases. To achieve these two rules, the evolutionary algorithm is run twice. In practice, we analyse one class at a time.

The training sets must be reasonably sized to ensure adequate population coverage. Moreover, as indicated by Prechelt (1994, 1995), it is insufficient to indicate the number of the examples in each of the partitioned sets because the results may vary significantly for different partitions even when the number of examples in each set is unchanged.

5.1. Genetic parameter setup

The evolutionary classification system requires that some control parameters be specified. Preliminary trials have been performed for an appropriate tuning of these parameters, which vary as a function of the problem chosen. For both the problems, the selection mechanism and the fitness function chosen have been the same. The tournament selection with a tournament size $\mu = 20\%$ has been used. It should be noted that the results remain similar if the parameter μ is within 15% and 25% of the population. This selection scheme has outperformed the proportional and the truncation selections. Furthermore a 1-elitism mechanism has been applied. The fitness function chosen has been (8) where p_1 and p_2 have been derived empirically equal to 0.05. Moreover, it should be pointed out that a linear normalization in [0.0, 1.0] has been applied to all the values in the databases to avoid some attribute being more significant than others.

The values of the other parameters depend on the problem. For the database CANCER1, the population size is equal to 200. Because we have nine attributes plus the class, on the basis of the fixed encoding, each chromosome is composed of 28 genes. The single-point crossover has been used for both the condition vector and the interval vector, as we are dealing with integer values. This operator has resulted in being more efficient with respect to the uniform crossover. In the interval vector, the mutation operator randomly transforms with uniform probability the value of an attribute into another value belonging to the domain of that attribute. The mutation rate used was 0.7. For the condition vector the mutation changes the condition related to a single attribute. Its application probability was 0.3. This last value is not restrictive. For example, the goodness of the results remains about the same if the mutation probability on the condition vector varies in the range $[0.2, 0.3]$. The difference in the mutation rates is due to the fact that the operator used for the condition vector may introduce or destroy new conditions so as to introduce significant variations, while the mutation on the interval vector changes the range of the attribute only and thus its probability can be higher without risking the loss of basic information. The evolution process terminates after at most 100 generations if a correct rule is not found before.

As concerns the database CANCER2, the population size is 300, the search space being larger than in the previous case. Because we deal with 30 attributes plus 1 for the class, the chromosome on the basis of the chosen encoding is constituted by 91 genes.

For the integer-valued condition vector, we have used the single-point crossover while, for the real-valued interval vector, EIR has resulted in being more efficient than ELR and DR. On the basis of their definitions in Mühlenbein and Schlierkamp-Voosen (1994), for EIR $d = 0.3$, so that the scalar parameter α_i is distributed in the range $[-0.3, 1.3]$. For the interval vector, DM has had worse performance than CM (Mühlenbein and Schlierkamp-Voosen 1993). Hence, CM with $range_i = 0.5$, $s = 8$ and $\beta \in [0.0, 1.0]$ has been considered. The mutation operator on the condition vector and the mutation rates as well have been the same as in the previous problem. The finding of one correct rule or a number of at most 200 generations has been fixed as termination criterion.

5.2. Performance measures

In order to determine the validity of our system, let us formulate some definitions. For each class, we indicate with:

- T^+ the number of true positive examples, i.e., the number of the examples correctly classified as belonging to the class
- T^- the number of true negative examples, that is to say, the number of examples correctly classified as not belonging to the class
- F^+ the number of false positive examples that are the examples classified incorrectly as belonging to the class
- F^- the false negative examples, i.e., those examples that are incorrectly classified as not belonging to the class

Based on these definitions, in the medical domain, there are two indicators, namely the sensitivity S_e and the specificity S_p defined as follows:

$$S_e = \frac{T^+}{T^+ + F^-} \qquad S_p = \frac{T^-}{T^- + F^+}$$

which indicate the rule's ability to classify correctly examples as belonging or not belonging to the predicted class, respectively.

As our system is constituted, we are concerned with two classification rules. We will denote with I_1 and I_2 the indeterminate cases, which include examples satisfying both the rules or no rule, respectively. Moreover, we indicate with CC and UC the total number of examples correctly and incorrectly classified, respectively. Finally, we denote with $\%Ac$ the percentage of classification accuracy, with $\%C$ and $\%U$ the percentage of cases correctly and incorrectly classified, respectively, and at the end, with $\%I$ the percentage of indeterminate examples. These last values are computed by means of the following formulas:

$$\%Ac = \frac{CC}{CC + UC}100 \qquad \%C = \frac{CC}{N_V}100$$

$$\%U = \frac{UC}{N_V}100 \qquad \%I = \frac{I_1 + I_2}{N_V}100$$

where N_V is the number of the examples in the test set. These parameters are tied by the formula:

$$N_V = CC + UC + I_1 + I_2.$$

5.3. First set of experiments

Several experiments have been performed on a SUN workstation for the database CANCER1, varying the size of the training and the test sets. Moreover, both the linear and the logarithmic fitness functions proposed in Sect. 3.3 have been tested. The execution of this algorithm requires about 6 minutes if a correct rule is not found before.

Table 2. The results of the system averaged over 10 runs

	I_1	I_2	$\%Ac$	$\%C$	$\%U$	$\%I$
A_v	3.5	5.1	99.58	94.69	0.4	4.91
StdDev	0.71	2.81	0.29	0.97	0.28	1.21

Table 3. The results for the best malignant rule

T^+	F^+	F^-	T^-
38	5	1	131

Table 4. The results obtained by the best benign rule

T^+	F^+	F^-	T^-
133	0	3	39

5.3.1. Results on the CANCER1*b database*

Because the database contains some missing values, we have initially decided to merely remove instances with the missing attributes, with the awareness that this approach may lead to serious biases (Little and Rubin 1987). The available 683 instances of the database CANCER1b have been subdivided into 508 examples for the training set and 175 for the test set. The test set remains unchanged and contains the same 136 benign and 39 malignant examples. The results achieved over 10 runs by using the linear fitness function (3) in (8) are reported in Table 2 in terms of average values A_v and standard deviations *StdDev*.

As can be observed by the reported values, the system shows an average percentage for accuracy equal to 99.58%, with a standard deviation equal to 0.29%. This means that over 100 examples for which the system has been able to classify on average more than 99 examples are correctly catalogued. Nevertheless, it is possible to note from the table that there is 4.91% of indeterminate examples. It should be observed that, in many cases, it is better that the system does not classify rather than performs an incorrect classification. However, the system has correctly classified 94.69% of examples, with an error classification equal to 0.4%.

The best rule found by the system for the malignant cases presents the following conditions:

$$(A_2 \geq 2) \wedge (A_3 \geq 3)$$

This rule classifies the examples in the test set as shown in Table 3.

The best rule found for the benign cases is:

$$(A_2 \leq 3) \wedge (A_6 \leq 5) \wedge (A_8 \leq 3)$$

This rule classifies the examples in the test set as in Table 4.

In our case for the malignant rule we have $S_e = 0.97$ and $S_p = 0.96$, so that we correctly classify 97% of individuals having the disease and 96% of those truly

Table 5. The results achieved by using the two best rules

Classification	Benign	Malignant
Benign	129	0
Indeterminate	6	1
Malignant	1	38

Table 6. The results of the system averaged over 10 runs

	I_1	I_2	$\%Ac$	$\%C$	$\%U$	$\%I$
A_v	2.7	4.8	99.35	95.09	0.63	4.28
StdDev	2.06	2.57	0.52	1.4	0.5	1.62

without disease. For the benign rule, $S_e = 0.98$ and $S_p = 1$, and thus this rule correctly classifies 98% of benign and 100% of malignant cases.

The results obtained by using both the rules are reported in the global Table 5. The system with these two rules has $\%Ac = 99.40$, $\%C = 95.43$, $\%U = 0.57$ and $\%I = 4$.

The connection between the tables reporting the results of the application of the two rules separately and the global table can be understood observing that the number F^+ in Table 3 increases either the number of cases satisfying both the rules or the number of examples incorrectly classified, while the number F^- in the same table increases either the number of examples that satisfy no rule or the number of cases incorrectly classified. The same observations are possible for Table 4.

From the analysis of the rules, it is possible to find out which attributes are more discriminant for the diagnosis. For example, during the trials effected, it has been observed that the attributes A_2 and A_3 for the malignant classification rules and A_2, A_6 and A_8 for the benign classification rules are almost always present. Moreover, the conditions on these attributes are often very similar.

The fitness (8) with the logarithmic function (5) has been tested on the same database. The results achieved over 10 executions are shown in Table 6.

The best rules found for the malignant and benign cases and the global system behaviour as well are the same as those obtained during the previous test.

As can be observed by the reported values, the system shows an accuracy percentage on average equal to 99.35%, with a standard deviation equal to 0.52%. Nevertheless, it is possible to note by comparing Table 6 with Table 2 that there is a greater percentage of correctly classified cases, a lower percentage of indeterminate examples but a greater number of incorrectly classified examples. Furthermore, the standard deviations are higher except for one parameter.

In Taha and Ghosh (1997), the authors divided randomly the available 683 instances into a training set of size 341 and a test set of size 342. Three rule extraction techniques from trained feedforward networks were applied. Furthermore, a method of integrating the output decisions of both the extracted rule-based system and the corresponding trained network is proposed. The rule evaluation is based on performance measures, among which are the soundness (T^+) and the false alarms (F^+). The dimensionality of the breast-cancer input space is reduced from 9 to 6 inputs. Different from Taha and Ghosh, we have used the complete set of attributes without performing any kind of data preprocessing. As regards the performance measures, our single-rule classification system is able to achieve better results in terms of soundness but this is detrimental to the number of false alarms. In fact, in Taha and Ghosh (1997), considering the single best rule for the malignant and the benign case, we have an overall classification rate of 92.83 with 21 false alarms. By performing randomly their same subdivision of the instances, the best overall classification rate found by our system over 10 runs is 96.35 with 33 false alarms. However, Taha and Ghosh obtained better results than ours for their five-rule system. In particular, their best overall classification rate is 96.63. A simple explanation of all of our above reported results is that this multiple-rule approach is conceived taking in mind that the classification system will be constituted by the conjunction in OR of more rules. In this way, the aim is to control the number of true positive cases to make the global system more reliable, but this is also detrimental to simple interpretability of the results. Our system provides two easily interpretable rules with good performance. Moreover, it can be noted that it is difficult to try describing complex phenomena by means of single rules able to generalize over the whole data set.

Better results are obtained by Peña–Reyes and Sipper, who present a fuzzy-genetic system (Peña and Sipper 1999). They presented very good results for multiple-rule systems, e.g., the overall classification rate for their best system is 97.8%, but the same authors admit that, for this system, there are 39 cases for which they are "somewhat less confident about the output." Their best fuzzy one-rule system presents an overall performance of 97.07% but no information is given about the diagnostic confidence. Besides, their threshold system is based on the knowledge of the problem at hand, while our results have been obtained without assigning any external value.

5.3.2. Results on the CANCER1*a database*

The second experiment involved all the 699 instances: the 16 missing attributes have to be replaced. Little and Rubin (1987) describe several approaches to estimate the missing values, but all of the proposed methods are biased because they treat the replacement value as the actual missing value. Another replacement strategy based on a Monte Carlo simulation technique called multiple imputation (Shafer 1997) allows the generation of multiple values for each missing datum. These values are analysed by standard complete-data methods and integrated into a single model. From a practical standpoint, a single replacement value must be chosen for each missing datum and this reintroduces bias. A further method, rather than trying to estimate the unknown attribute, treats as unknown a new possible value for each attribute and deals with it as other values (Lee and Shin 1999). We have chosen to replace the missing data with random values within the variation interval of the single missing attribute.

The first three quarters of the data (524 patterns) have been used for the training set and the last 175 for the test set, of which 137 are benign and 38 malignant patterns. We have run the evolutionary system 10 times, each with a different starting random population and using the (3) within the total fitness function (8). The average results are outlined in Table 7. The average percentage for accuracy is equal to 98.52%, with a standard deviation equal to 0.8%.

The best rule found for the malignant cases is the following:

$$(A_2 \geq 3) \wedge (A_7 \geq 2)$$

Table 7. The results of the system averaged over 10 runs

	I_1	I_2	$\%Ac$	$\%C$	$\%U$	$\%I$
A_v	2.5	3.5	98.52	95.14	1.43	3.43
StdDev	1.18	1.65	0.8	0.86	0.77	0.81

Table 8. The findings of the best malignant rule

T^+	F^+	F^-	T^-
38	5	0	132

Table 9. The results for the best benign rule

T^+	F^+	F^-	T^-
134	0	3	38

Table 10. The results achieved by using the two best rules

Classification	Benign	Malignant
Benign	131	0
Indeterminate	4	0
Malignant	2	38

This rule classifies the examples in the test set as shown in Table 8. For this rule, we have $S_e = 1$ and $S_p = 0.96$.

The best rule found for the benign cases is:

$$(A_1 \leq 6) \wedge (A_3 \leq 4) \wedge (A_5 \leq 4) \wedge (A_8 \leq 3)$$

This rule classifies the examples in the test set as in Table 9 with $S_e = 0.98$ and $S_p = 1$.

It should be noted that the rules found are different from those achieved in the previous test. This may be due both to the insertion in the database of the previously discarded examples and to the fact that the evolutionary system provides a number of solutions that are nearly suboptimal from the performance point of view. This does not imply that the provided solutions be genotypically similar. The availability of different rules could represent an assistance for a human expert to make a decision.

The results obtained by using both the rules are reported in the global Table 10. The system has $\%Ac = 98.83$, $\%C = 96.57$, $\%U = 1.14$ and $\%I = 2.29$.

The (8) with the logarithmic function (5) has been tested on the same database and with the same subdivision for the training and the test set. The average results achieved over 10 executions are shown in Table 11.

The best rule found for the malignant cases is the following:

$$(A_2 \geq 4)$$

Table 11. The results of the system averaged over 10 runs

	I_1	I_2	$\%Ac$	$\%C$	$\%U$	$\%I$
A_v	3.8	3.2	99.47	95.49	0.51	4
$StdDev$	2.62	0.63	0.59	1.19	0.57	1.48

Table 12. The findings of the best malignant rule

T^+	F^+	F^-	T^-
36	2	2	135

Table 13. The results for the best benign rule

T^+	F^+	F^-	T^-
135	0	2	38

Table 14. The results obtained by using the two best rules

Classification	Benign	Malignant
Benign	133	0
Indeterminate	4	2
Malignant	0	36

This rule classifies the examples in the test set as presented in Table 12. For this rule we have $S_e = 0.95$ and $S_p = 0.99$.

The best rule found for the benign cases is:

$$(A_1 \leq 6) \wedge (A_2 \leq 4) \wedge (A_6 \leq 6) \wedge (A_8 \leq 8)$$

This rule classifies the examples in the test set as in Table 13. For this rule, we have $S_e = 0.99$ and $S_p = 1$.

The results obtained by using both the rules are reported in the global Table 14. The system has $\%Ac = 100$, $\%C = 96.57$, $\%U = 0$ and $\%I = 3.43$.

Apart from the best results, it should be noted that the performance of the fitness with the linear function is more robust, the standard deviations of the different parameters being lower on average.

In Yao and Liu (1997), the same problem is faced by taking into account the missing attributes. Unfortunately, they do not give any information on how the missing attributes are treated. They have presented a new evolutionary system for evolving feedforward ANNs applied to this medical diagnosis problem. Their results show a lower percentage of wrong classifications. No indeterminate cases are provided. Though the percentage of the wrong classifications obtained by our classification system is higher with respect to that attained by Yao and Liu, it is important to emphasize that we provide intelligible rules, unlike they do. This, in our opinion, counterbalances the worse results. However, our system includes indeterminate cases

Table 15. The results of a cross-validation method

	I_1	I_2	$\%Ac$	$\%C$	$\%U$	$\%I$
A_v	6.1	6.4	97.30	90.35	2.51	7.14
StdDev	0.74	2.46	1.00	1.49	0.94	1.58

Table 16. The results averaged over 10 runs

	I_1	I_2	$\%Ac$	$\%C$	$\%U$	$\%I$
A_v	1.4	10.7	96.71	86.41	2.98	10.61
StdDev	1.58	4.11	2.23	2.38	2.08	3.35

but it is easily explicable by the fact that it is sufficient that one single condition be not satisfied to make the examples classified as indeterminate. Moreover, the limited expressive power of the chosen encoding plays an important role.

An interesting characteristic of a stochastic classification system is that it can provide, when run more times, rules with different features. For example in our case, apart from the best rules above reported, we have at our disposal several more:

if $(A_2 \geq 2) \wedge (A_3 \geq 3)$ then malignant
if $(A_2 \geq 4)$ then malignant
if $(A_1 \leq 6) \wedge (A_2 \leq 4) \wedge (A_6 \leq 6) \wedge (A_8 \leq 3)$ then benign
if $(A_1 \leq 6) \wedge (A_3 \leq 4) \wedge (A_8 \leq 3)$ then benign

Note that some of these rules have been already found previously. Among them, the first and the third have together an accuracy equal to 100% while the second and the fourth present together the highest percentage of correctly classified examples, i.e., 96.57%.

To evaluate the effectiveness of our automatic classification system, a cross-validation method has been applied. Considering the database with 699 examples, 524 of which are in the training set and the remaining in the test set, the examples in the two sets are randomly varied over 10 runs. The results averaged over the runs using the linear function in (8) are reported in Table 15.

It is simple to note that the accuracy is lower and the percentage of indeterminate cases is higher with respect to the database with the first examples in the training set and the others in the test set. The increase in the indeterminate cases could be ascribed to the presence of the anomalous examples in the test set.

5.4. Second set of experiments

The system has also been tested on the database CANCER2. It should be considered that this problem is more complex because the search space of the descriptions is much larger. Several experiments have been performed in Hung et al. (2001) considering a training set composed by 455 examples and a test set of 114 examples. The test set is randomly varied but it always includes 76 benign and 38 malignant examples. We have carried out 10 runs on a SUN workstation with this subdivision

Table 17. The findings of the best malignant rule

T^+	F^+	F^-	T^-
30	1	8	75

Table 18. The results of the best benign rule

T^+	F^+	F^-	T^-
74	4	2	34

Table 19. The results achieved by applying the two best rules

Classification	Benign	Malignant
Benign	74	3
Indeterminate	1	6
Malignant	1	29

and with the linear function in (8). The execution of this algorithm requires about 40 minutes if a correct rule is not found before. Our classification system has produced the results shown in Table 16.

The best rule for the malignant cases is:

$$(A_2 \in [16.1443, 33.5886]) \wedge (A_{17} < 0.1441) \wedge (A_{20} < 0.01284) \\ \wedge (A_{21} > 16.7940)$$

This rule produces the results in Table 17 with $S_e = 0.79$ and $S_p = 0.99$.

The rule for the benign cases is:

$$(A_{14} < 54.4786) \wedge (A_{23} < 116.0951) \wedge (A_{24} < 950.2699) \wedge (A_{28} < 0.1604)$$

This rule determines the results in Table 18. In this case, $S_e = 0.97$ and $S_p = 0.89$.

In Table 19, the results obtained by applying both the rules are shown.

As an example of evolution, the values related to these two best rules during the training phase in terms of the best and average fitness and standard deviation are shown in Fig. 1. The values within the range [0.7, 0.9] have not been reported, while the values in the range [0.9, 1.0] have been magnified to make evident the small variations for the best fitness value.

As can be noted, the increase in fitness values shows an initial remarkable quasi-linear phase, then this increase gets slower. During the final phase, no further fitness improvements are obtained until the end of the run. Because a correct rule is not achieved, the evolution terminates when the fixed maximum number of generations is reached.

In Hung et al. (2001), the classification has been performed by using feedforward neural networks. The network is used to estimate the posterior probabilities of the observations in the test set. According to Mangasarian et al. (1995), a case with

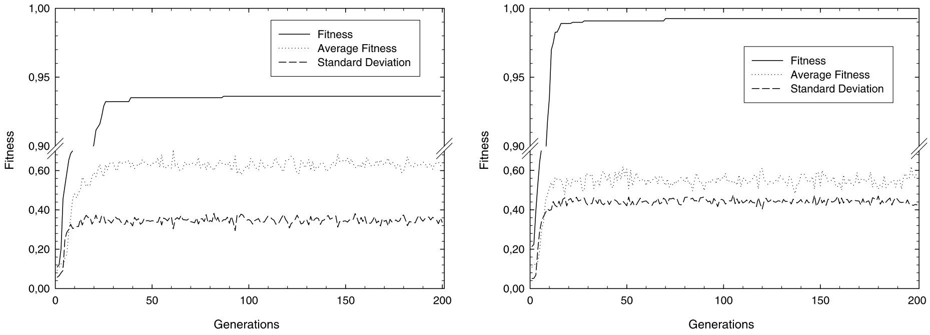


Fig. 1. Best and average fitness value and standard deviation related to the best run for the malignant (left) and for the benign rule (right)

malignancy probability between 0.30 and 0.70 is classified as indeterminate, while for values lower than 0.3 as benign and finally malignant for values higher than 0.7. The paper illustrates several neural network models for classification. The value of the posterior probability is obtained by considering the mean of the outputs of 200 trained networks. Each model allows attaining high correct classification rates, but it is to be pointed out that the best results are obtained by applying a feature selection procedure which results in a model dealing with only 9 variables instead of 30. This reduces the corresponding search space, while we have left out of consideration any kind of preprocessing and postprocessing activity in the database construction. The best results in Hung et al. (2001) outperform those achieved by our system but their classification technique has the disadvantage of lacking in comprehensible rules. The availability of explicit rules is of noticeable importance because it provides human experts with a further investigation tool.

6. Conclusions and future works

In this paper we have presented an evolutionary classification system for automatically extracting explicit rules. The system has been evaluated on two two-class problems in the medical domain, both related to breast cancer diagnosis. It should be pointed out that this test problem has been chosen only to evaluate the ability of an evolutionary technique in designing an automatic classification system. Naturally, the conceived system is easily applicable to any other kind of database and generalizable to multiple-class problems. We have compared our system with other methods. Experimental results have demonstrated the effectiveness of the approach proposed in providing the user with comprehensible classification rules.

Future work will include the investigation of other evolutionary techniques and their application to different real-world data sets in order to further improve the promising results reported in the present paper. In particular, a Genetic Programming approach will be investigated to enhance the expressive power of the extracted rules. This will allow us both to easily introduce a wider set of conjunctions ($\wedge$, $\vee$ and $\neg$) and to use higher order logics, i.e., to create clauses containing two attributes. Furthermore, niching methods (Goldberg and Richardson 1987; Smith et al. 1992) will be exploited with the aim of finding rule sets.

Another interesting task to face will be unsupervised data mining, in which the goal is to discover rules that predict a value of a goal attribute which, unlike classification, is not chosen a priori.

References

Anglano C, Giordana A, Lo Bello G et al (1997) A network genetic algorithm for concept learning. In: Proceedings of the 7th international conference on genetic algorithms. Kaufmann, San Francisco, CA, pp 434–441

Augier S, Venturini G, Kodratoff Y (1995) Learning first order logic rules with a genetic algorithm. In: Proceedings of the 1st international conference on knowledge discovery and data mining. AAAI, Menlo Park, CA, pp 21–26

Belew RK (1989) Adaptive information retrieval. In: Proceedings of the 12th annual international ACM/SIGIR conference on research and development in information retrieval, Cambridge, MA, 25–28 June, pp 11–20

Bennett KP, Mangasarian OL (1992) Robust linear programming discrimination of two linearly inseparable sets. Optim Methods Softw 1:23–34

Bojarczuk CC, Lopes HS, Freitas AA (1999) Discovering comprehensible classification rules using genetic programming: a case study in a medical domain. In: Proceedings of the genetic and evolutionary computation conference, Orlando, Florida, 14–17 July, pp 953–958

Brameier M, Banzhaf W (2001) A comparison of linear genetic programming and neural networks. IEEE Trans Evol Comput 5(1):17–26

Breeden JL, Packard NH (1992) A learning algorithm for optimal representations of experimental data. Tech Rep CCSR-92-11, University of Illinois Urbana–Champaign

Carbonell JG, Michalski RS, Mitchell TM (1993) An overview of machine learning. In: Carbonell JG, Michalski RS, Mitchell TM (eds) Machine learning, an artificial intelligence approach. Tioga, Palo Alto, CA, pp 3–23

Chen H, Dhar V (1991) Cognitive process as a basis for intelligent retrieval systems design. Inf Process Manage 27(5):405–432

Chen H, Lynch KJ (1992) Automatic construction of networks of concepts characterizing document databases. IEEE Trans Syst Man Cybernet 22(5):885–902

Chen H, Lynch KJ, Basu K et al (1993) Generating, integrating, and activating thesauri for concept-based document retrieval. IEEE EXPERT 8(2):25–34

Chen M, Han J, Yu PS (1996) Data mining: an overview from database perspective. IEEE Trans Knowl Data Eng 8(6):866–883

De La Iglesia B, Debuse JCW, Rayward-Smith VJ (1996) Discovering knowledge in commercial databases using modern heuristic techniques. In: Proceedings of the 2nd international conference on knowledge discovery and data mining. AAAI, Menlo Park, CA, pp 44–49

Derkse W (1993) On simplicity and elegance. Delft, Eburon

Fayyad UM, Piatetsky-Shapiro G, Smith P (1996) From data mining to knowledge discovery: an overview. In: Fayyad UM et al (eds) Advances in knowledge discovery and data mining. AAAI/MIT, pp 1–34

Fogel DB, Wasson EC, Boughton EM et al (1998) Linear and neural models for classifying breast masses. IEEE Trans Med Imag 17(3):485–488

Freitas AA (1997) A genetic programming framework for two data mining tasks: classification and generalized rule induction. In: Genetic programming 1997: proceedings of the 2nd annual conference. Kaufmann, San Francisco, CA, pp 96–101

Fung G, Mangasarian OL (1999) Semi-supervised support vector machines for unlabeled data classification. Tech Rep, Computer Sciences Department, University of Wisconsin

Giordana A, Saitta L, Zini F (1994) Learning disjunctive concepts by means of genetic algorithms. In: Proceedings of the 11th international conference on machine learning, pp 96–104

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA

Goldberg DE, Richardson J (1987) Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette JJ (ed) Genetic algorithms and their applications. Erlbaum, Hillsdale, NJ, pp 41–49

Gordon M (1988) Probabilistic and genetic algorithms for document retrieval. Commun ACM 31(10):1208–1218

Holland JH (1975) Adaptation in natural and artificial systems. MIT Press, Cambridge, MA

Holsheimer M, and Siebes A (1994) Data mining: the search for knowledge in databases. Tech Rep CS-R9406, CWI, Amsterdam

Hung MS, Shanker M, Hu M (2001) Estimating breast cancer risks using neural networks. Eur J Oper Res Soc 52:1–10

Ishibuchi H, Nozaki K, Yamamoto N et al (1995) Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans Fuzzy Syst 3(3):260–270

Koza JR (1992) Genetic programming: on programming computers by means of natural selection and genetics. MIT Press, Cambridge, MA

Lee CH, Shin DG (1999) A multistrategy approach to classification learning in databases. Data Knowl Eng 31:67–93

Little RJA, Rubin DB (1987) Statistical analysis with missing data. Wiley, New York

Lu H, Setiono R, Liu H (1995) NeuroRule: a connectionist approach to data mining. In: Proceedings of the 21st international conference on very large data bases, pp 478–489

Mangasarian OL, Setiono R, Wolberg WH (1990) Pattern recognition via linear programming: theory and applications to medical diagnosis. In: Coleman TF et al (eds) Large-scale numerical optimization. SIAM, Philadelphia, pp 22–30

Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via linear programming. Oper Res 43(4):570–577

Michalski RS (1983) A theory and methodology of inductive learning. In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine learning, an artificial intelligence approach. Tioga, Palo Alto, CA, pp 83–134

Mühlenbein H, Schlierkamp-Voosen D (1993) Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization. Evol Comput 1(1):2–49

Mühlenbein H, Schlierkamp-Voosen D (1994) Strategy adaptation by competing subpopulations, In: Proceedings of the international conference on parallel problem solving from nature. Springer, Berlin Heidelberg New York, pp 199–208

Neri F, Giordana A (1995) A parallel genetic algorithm for concept learning. In: Proceedings of the 6th international conference on genetic algorithms. Kaufmann, San Mateo, CA, pp 436–443

Ngan PS, Wong ML, Leung KS (1998) Using grammar based genetic programming for data mining of medical knowledge. In: Genetic programming 1998: proceedings of the 3rd annual conference. Kaufmann, San Francisco, CA, pp 304–312

Noda E, Freitas AA, Lopes HS (1999) Discovering interesting prediction rules with a genetic algorithm. In: Proceedings of the congress on evolutionary computation, Washington, DC, 6–9 July, pp 1322–1329

Peña CA, Sipper M (1999) Designing breast cancer diagnosis systems via a hybrid fuzzy-genetic methodology. In: Proceedings of the IEEE international fuzzy systems conference, vol 1, pp 135–139

Piatesky-Shapiro G (1991) Discovery, analysis and presentation of strong rules. In: Piatesky-Shapiro G, Frawley W (eds) Knowledge discovery in databases. AAAI, Menlo Park, CA, pp 229–248

Prechelt L (1994) Proben1—a set of neural network benchmark problems and benchmarking rules. Tech Rep 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany

Prechelt L (1995) Some notes on neural learning algorithm benchmarking. Neurocomputing 9(3):343–347

Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106

Quinlan JR (1993) C4.5: programs for machine learning. Kaufmann, San Mateo, CA

Radcliffe NJ, Surry PD (1994) Co-operation through hierarchical competition in genetic data mining. Tech Rep 94-09, Edinburgh Parallel Computing Centre, University of Edinburgh, Scotland

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, the PDP Res Group (eds) Parallel distributed processing. MIT Press, Cambridge, MA, pp 318–362

Salim M, Yao X (2002) Evolving SQL queries for data mining. In: Yin H, Allinson N, Freeman R, Keane J, Hubbard S (eds) Proceedings of the 3rd international conference on intelligent data engineering and automated learning (IDEAL'02). Lecture notes in computer science, vol 2412. Springer, Berlin Heidelberg New York, pp 62–67

Setiono R, Hui LCK (1995) Use of a quasi-Newton method in a feedforward neural networks construction algorithm. IEEE Trans Neural Net 6(1):273–277

Shafer J (1997) Analysis of incomplete multivariate data. Chapman and Hall, New York

Sherrah JR, Bogner RE, Bouzerdoum A (1997) The evolutionary pre-processor: automatic feature extraction for supervised classification using genetic programming. In: Proceedings of the 2nd annual genetic programming conference. Kaufmann, Stanford University, 13–16 July, pp 304–312

Smith RE, Forrest S, Perelson AS (1992) Searching for diverse, cooperative populations with genetic algorithms. Evol Comput 1(2):127–149

Taha I, Ghosh J (1997) Evaluation and ordering of rules extracted from feedforward networks. In: Proceedings of the IEEE international conference on neural networks, Houston, TX, pp 221–226

Weiss SM, Kulikowski CA (1991) Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. Kaufmann, San Mateo, CA

Wolberg WH, Mangasarian OL (1990) Multisurface method of pattern separation for medical diagnosis applied to breast cancer cytology. Proc Natl Acad Sci 87:9193–9196

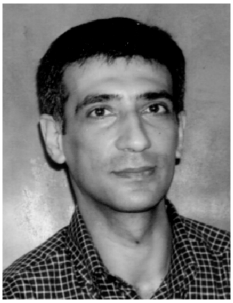
Wolberg WH, Street WN, Mangasarian OL (1995) Image analysis and machine learning applied to breast cancer diagnosis and prognosis. Anal Quant Cytol Histol 17(2):77–87

Yao X, Liu Y (1997) A new evolutionary system for evolving artificial neural networks. IEEE Trans Neural Net 8(3):694–713

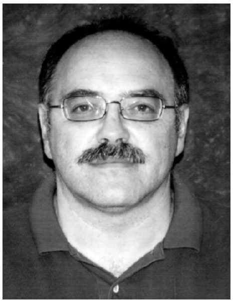
Ziarko W (1994) Rough sets, fuzzy sets and knowledge discovery. Springer, Berlin Heidelberg New York

Author biographies

Ivanoe De Falco was born in Naples (Italy) on 2 March 1961. He received his Laurea degree in Electrical Engineering cum laude in 1987 at University of Naples “Federico II.” Since then, he has been involved in research and is currently a researcher at National Research Council of Italy (CNR). His main fields of interest include evolutionary algorithms, soft computing and parallel computing. He is a member of the European Network of Excellence in Evolutionary Computing (EvoNet), of the World Federation on Soft Computing (WFSC) and of the Machine Learning Network (MLnet).



Antonio Della Cioppa was born in Bellona (Italy) on 13 June 1964. He received his Laurea degree in physics and a Ph.D. in computer science, both from University of Naples “Federico II,” Italy, in 1993 and 1999, respectively. From 2000, he has been a postdoctoral fellow at the Department of Computer Science and Electrical Engineering, University of Salerno, Italy. He is a member of EvoNet, ECCAI, ECCAI–Italian Chapter. His research interests are in the fields of complexity, evolutionary computation and artificial life.



Aniello Iazzetta (1953) received his Laurea degree in computer science from the University of Salerno, Italy. He is a researcher at the Istituto Motori of National Research Council of Italy (CNR), where he conducts research related to tools development for intelligent transport systems. From 1985 until 2001, he was a researcher at the Istituto per la Ricerca sui Sistemi Informatici Paralleli of CNR. He has been working on parallel architectures, neural networks and evolutionary systems.

Ernesto Tarantino was born in S. Angelo a Cupolo, Italy, in 1961. He received the Laurea degree in electrical engineering in 1988 from the University of Naples, Italy. He is currently a researcher at the National Research Council of Italy. After completing his studies, he conducted research in parallel and distributed computing. During the past decade, his research interests have been in the areas of theory and application of evolutionary techniques and related sectors of computational intelligence. He has served on several program committees of conferences in the area of evolutionary computation.

Correspondence and offprint requests to: Ernesto Tarantino, ICAR—National Research Council, Via P. Castellino 111, 80131, Naples, Italy. Email: ernesto.tarantino@na.icar.cnr.it