ELSEVIER

Artificial Intelligence in Medicine 27 (2003) 129–154

Artificial Intelligence in Medicine

www.elsevier.com/locate/artmed

Evolutionary computing for knowledge discovery in medical diagnosis

K.C. Tan[*], Q. Yu, C.M. Heng, T.H. Lee

Department of Electrical and Computer Engineering, National University of Singapore,
4 Engineering Drive 3, Singapore 117576, Singapore

Received 31 May 2002; received in revised form 17 November 2002; accepted 10 December 2002

Abstract

One of the major challenges in medical domain is the extraction of comprehensible knowledge from medical diagnosis data. In this paper, a two-phase hybrid evolutionary classification technique is proposed to extract classification rules that can be used in clinical practice for better understanding and prevention of unwanted medical events. In the first phase, a hybrid evolutionary algorithm (EA) is utilized to confine the search space by evolving a pool of good candidate rules, e.g. genetic programming (GP) is applied to evolve nominal attributes for free structured rules and genetic algorithm (GA) is used to optimize the numeric attributes for concise classification rules without the need of discretization. These candidate rules are then used in the second phase to optimize the order and number of rules in the evolution for forming accurate and comprehensible rule sets. The proposed evolutionary classifier (EvoC) is validated upon hepatitis and breast cancer datasets obtained from the UCI machine-learning repository. Simulation results show that the evolutionary classifier produces comprehensible rules and good classification accuracy for the medical datasets. Results obtained from *t*-tests further justify its robustness and invariance to random partition of datasets.

Keywords: Medical diagnosis; Knowledge discovery; Data mining; Evolutionary computing

1. Introduction

Clinical medicine is facing a challenge of knowledge discovery from the growing volume of data. Nowadays enormous amounts of information are collected continuously by monitoring physiological parameters of patients. The growing amounts of data has made

[*] Corresponding author. Tel.: +65-6874-2127; fax: +65-6779-1103.
E-mail address: eletankc@nus.edu.sg (K.C. Tan).

0933-3657/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0933-3657(03)00002-2

manual analysis by medical experts a tedious task and sometimes impossible. Many hidden and potentially useful relationships may not be recognized by the analyst. The explosive growth of data requires an automated way to extract useful knowledge. One of the possible approaches to this problem is by means of data mining or knowledge discovery from databases (KDD) [1,3]. Through data mining, interesting knowledge and regularities can be extracted and the discovered knowledge can be applied in the corresponding field to increase the working efficiency and to improve the quality of decision making.

An important task in knowledge discovery is to extract comprehensible classification rules from the data. Classification rules are typically useful for medical problems which have been massively applied particularly in the area of medical diagnosis [10,27]. Such rules can be verified by medical experts and may provide better understanding of the problem in-hand. Numerous techniques have been applied to classification in data mining over the past few decades, such as expert systems, artificial neural networks, linear programming, database systems, and evolutionary algorithms [4,6,21,36,40,42]. Among these approaches, evolutionary algorithms have been emerged as a promising technique in dealing with the increasing challenge of data mining in medical domain [27]. The evolutionary algorithm (EA) is a class of computational techniques inspired by the natural evolution process that imitates the mechanism of natural selection and survival-of-the-fittest in solving real life problems [24,35].

The genetic programming (GP) [20] and genetic algorithms [12] are two popular approaches in evolutionary algorithms. Although, GP and GA are based on the same evolution principle, they often adopt a different chromosome representation, e.g. GA uses a fixed-length chromosome structure while GP applies a tree-based chromosome representation. Recently, GA has been utilized at different stages of knowledge discovery process in medical data mining applications. Kim and Han [18] and Liu et al. [22] applied GA at the pre-processing stage to reduce the dimension/difficulty of the problem and to increase the learning efficiency in data mining. Komosinski and Krawiec [19] proposed an evolutionary algorithm for feature weighting that gives quantitative information about the relative importance of the features. Hruschka and Ebecken [14] and Meesad and Yen [23] used GA at the post-processing stage to extract rules from a neural network. Other GA approaches for generating classification rules in data mining include [7,8,38].

Brameier and Banzhaf [3] proposed a linear genetic programming (LGP) classification approach for data mining in medical domain, but the issue of comprehensibility of the classification rules has not been addressed. Wong and Leung [41] proposed a grammar-based GP for constructing the classification rules. However, the grammar is domain specific such that a new grammar has to be provided for each new problem, and the use of grammar also reduces the autonomy of GP in discovering novel knowledge. To address the issue of comprehensibility of classification rules, Bojarczuk et al. [2] proposed a non-standard tree structure GP where functions are constructed via Boolean operators and terminal sets are chosen based on Booleanized attributes. However, the numeric attributes in this approach need to be discretized into nominal boundaries a priori in order to use the Booleanized attributes. This restricts the search capability of GP, i.e. the classification accuracy depends a lot on how well the boundaries were defined.

One possible approach of handling both nominal and numeric attributes in evolutionary data classification is through the hybridization of GA and GP. Howard and D'Angelo [13] proposed a hybrid GA and GP called genetic algorithm-program (GA-P), which has been applied to evolve expressions for symbolic regression problems. In their approach, GP was used to construct expression tree and GA was applied to find numeric constant and coefficient of nominal attributes in the expression. Although the GA-P utilized the concept of GA and GP hybridization, it was designed for the regression application that is different from the problem of data classification addressed in this paper. Unlike GA-P, a two-phase evolutionary process is adopted in our approach, i.e. the hybrid evolutionary algorithm is applied to generate good rules in the first phase, which are then used to evolve comprehensible rule sets in the second phase.

Besides evolutionary algorithm-based approaches, a number of algorithms based on artificial neural networks have also been applied to solve the classification problem in medical diagnosis. However, one common problem of artificial neural networks is that they are essentially a ‘‘black-box’’ system. Although good predictive accuracy can often be achieved, the user is prevented from knowing what is going on inside the ‘‘black-box’’. Setiono [31,32] proposed the approach of NeuralRule for rule extraction from artificial neural networks, which attempts to extract comprehensible information while preserves high accuracy of the network. Although the rule extraction algorithm is capable of obtaining compact rule sets, it often needs an independent process of network training/ pruning and rules extraction in data mining. Taha and Gosh [34] proposed a BIO-RE algorithm for extracting rules from artificial neural networks, but the approach can only be applied to data with binary attributes. Peña-Reyes and Sipper [26] proposed the method of fuzzy-genetic approach (GA) by applying genetic algorithm to generate fuzzy classification rules, which has reported good results on Wisconsin diagnostic breast cancer (WDBC) dataset [33].

This paper proposes a two-phase hybrid evolutionary rule extraction algorithm that incorporates both GA and GP to discover comprehensible classification rules for data mining in medical applications. The paper is organized as follows: Section 2 gives an overview of classification rule-learning in medical diagnostic problem. Section 3 describes the proposed two-phase hybrid evolutionary classifier (EvoC) in detailed. The hepatitis and breast cancer datasets are described in Section 4, and the classification results of the proposed evolutionary classifier are compared with existing approaches. Conclusions are drawn in Section 5.

2. Decision rules in classification

Given a set of labeled instances, the objective of classification is to discover the hidden relations or regulations between attributes and classes. The classification rules are extracted in the hope that they can be used to automate classification of future instances. In the classification task, the discovered knowledge is usually represented in the form of decision trees or IF–THEN classification rules, which has the advantage of being a high-level and symbolic knowledge representation that contributes to the comprehensibility of the discovered knowledge. In this paper, the knowledge is presented as multiple IF–THEN rules in a decision rule list or rule set. Such rules state that the presence of one or more items (antecedents) implies or predicts the presence of other items (consequences). A typical rule has the form of

$$\text{rule}: \ \text{IF}\, \boldsymbol{X}_1 \text{ and } \boldsymbol{X}_2 \text{ and } \ldots\ \boldsymbol{X}_n \text{ THEN } \boldsymbol{Y}$$

where $\boldsymbol{X}_i$, $\forall i \in \{1, 2, \ldots, n\}$ is the antecedent that leads to a prediction of $\boldsymbol{Y}$, the consequence. Each of the IF–THEN rules can be viewed as an independent piece of knowledge. New rules can be added to an existing rule set without disturbing those already there, and multiple rules can be combined together to form a set of decision rules. The basic structure of the decision rule list could be built as follows:

IF antecedent$_1$ THEN class$_1$
ELSE IF antecedent$_2$ THEN class$_2$
⋮
ELSE class$_{\text{default}}$

Such format of rule list plays an important role in medical applications, especially for medical diagnosis where the classification task is to determine the type of disease in the diagnostic problem for a given set of attributes of patients. When the rule list is evaluated or used to classify a new instance, the first rule (topmost) will be considered first. If the rule does not match the instance (i.e. not able to classify the instance), the next rule will be considered. The matching process is repeated until a corresponding rule is found. In the case where none of the rules in the rule list matches the new instance, the new instance will be classified as the default class, which is usually the largest class in the dataset. In our approach, the default class is evolved concurrently with the rule sets to promote the flexibility and possibility of producing good rule sets.

3. A two-phase hybrid evolutionary classifier

In this paper, the classification task is formulated as a complex search optimization problem, where hidden relationships of the attributes to class are targeted knowledge to be discovered. The candidate solution that is in the form of a comprehensible Boolean rule set is obtained through a two-phase evolution mechanism as shown in Fig. 1. The first phase searches for a pool of good candidate rules using Michigan coding approach [24], while the second phase finds the best Boolean rule set by evolving and forming rule sets from the pool of rules. Since rule sets with different number of rules are targeted in the second phase, Pittsburgh coding approach [24] is used to encode the individuals. In the proposed evolutionary classifier, the optimal number of rules in a rule set is decided automatically in the second phase, which is advantageous to many approaches where the number of rules in a rule set often needs to be determined a priori [17,26,28].

The proposed two-phase evolutionary classifier also confines the usually large classification search space and consequently requires a smaller population and generation size. In this way, the inherent problem of Pittsburgh's coding method in finding the usually large combination of classification rules is greatly reduced, e.g. it is relatively easy to find good

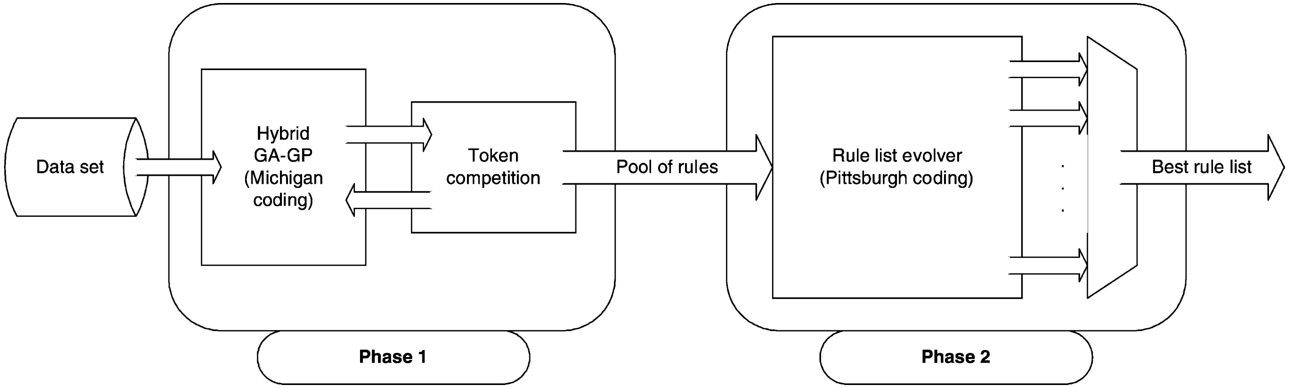
Data set
Hybrid GA-GP (Michigan coding)
Token competition
Pool of rules
Rule list evolver (Pittsburgh coding)
Best rule list
Phase 1
Phase 2

Fig. 1. Overview of the two-phase hybrid evolutionary classifier.

combination of rules in the second phase since the number of rules obtained in the first phase is confined and essential to the problem. For example, the population and generation size was set as 200 and 2500, respectively, in [26]. In our approach, the population size is set as 100 in the first phase and 50 in the second phase, and the generation size is set as 100 and 50 in the first and second phase, respectively.

3.1. Phase 1: the hybrid GA–GP

The evolutionary classifier, namely EvoC, has been implemented by the authors and integrated into the Java-based public domain data mining package 'WEKA' [11,40]. Fig. 2 shows the phase 1 of the program flowchart of EvoC, where the initial population is created from the training set. The attributes in the training set are built into nominal and numeric table for GP and GA, respectively. Each individual encodes a single rule and the population is structured such that all individuals are associated to the same class. This structure avoids the need of encoding the class values, i.e. the THEN part of the rules is encoded implicitly in the individuals.

The tournament selection scheme [1] with a tournament size of 2 is implemented in EvoC. When a new population is formed, the token competition [41] is applied as a covering algorithm to penalize redundant individuals as well as to retain individuals that cover the problem space well [15]. The winners in the token competition will be added to a pool of candidate rules and the pool is maintained such that no redundant rules may exist and the previously encountered good rules are kept for subsequent competitions. All individuals in the pool including the current population are participated in the token competition in order to ensure that no redundant rules exist in the pool.

The evolutionary process in phase 1 is run for every class of dataset. For an *n* class problem, there will be *n* evolutionary iterations. The pool of rules for every class is combined into a global pool, which will be presented as the input to the second phase. Each individual in the first phase contains two different chromosome structures, which are treated separately in the evolution and assigned to handle the nominal and numeric attributes. The chromosome structure, genetic operations and handling techniques of EvoC in phase 1 are described in the following sections.

3.1.1. Chromosome structure and genetic operations

The weather dataset shown in Table 1 is used as an example to show how the proposed hybrid GA–GP works on different types of attributes. The objective of the dataset is to learn whether a specific game can be played on a given weather. Each of the columns in Table 1 represents an attribute. The last column is the class attribute to be learned. Each of the rows represents an instance, and the collection of instances forms the dataset. For the weather dataset, the "Outlook" and "Windy" are nominal attributes, while the "Temperature" and "Humidity" are numeric attributes.

The genetic programming tree-based chromosome representation has been used to encode nominal attributes that are Booleanized, and many logical operators have been applied to evolve highly flexible solutions in classification problems [2,36]. However, the difficulty of integrating general arithmetic operators with Booleanized attributes in GP limits the flexibility of handling real-world data that often consists of both the nominal and

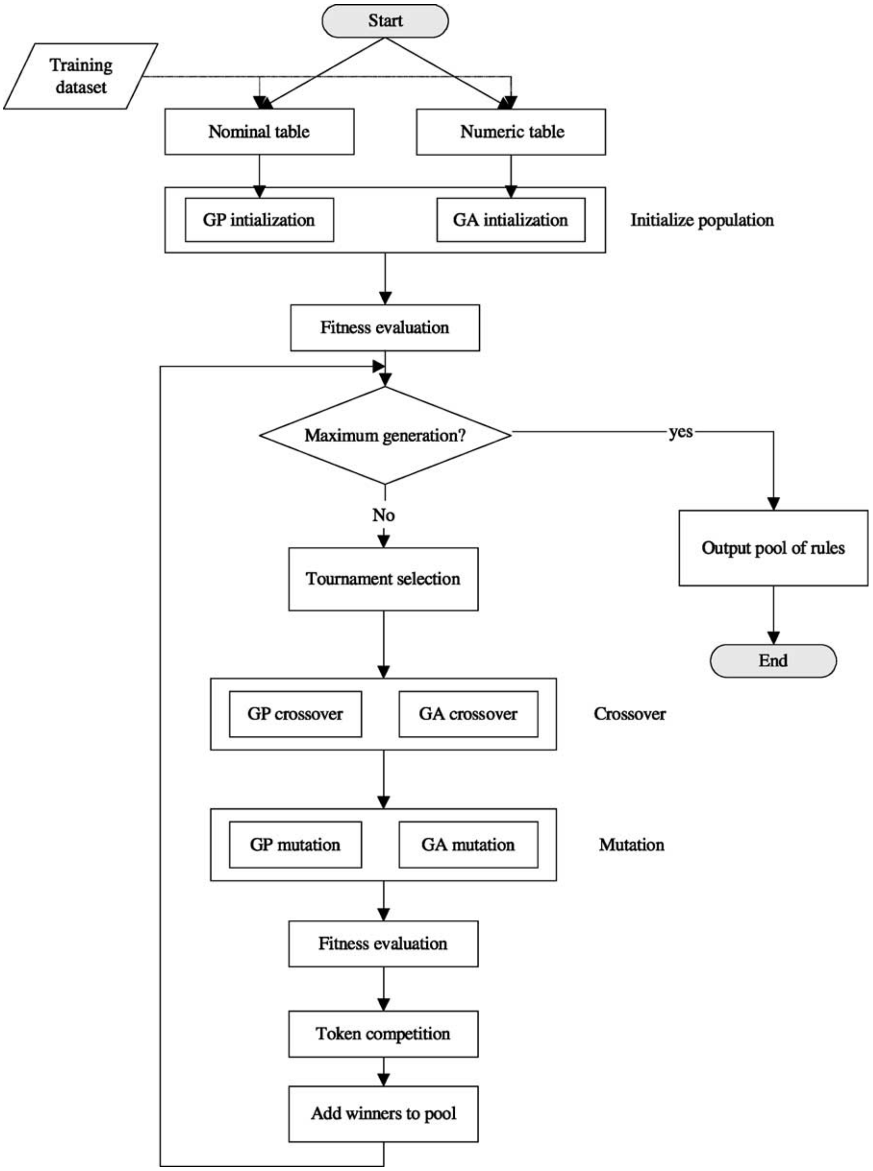

Fig. 2. The program flowchart for the phase 1 of EvoC.

numeric attributes [15]. One passive approach is to discretize the numeric attributes into boundaries at the expense of lower classification accuracy for the rules found. To address this problem, the approach of having independent chromosomes to handle numeric attributes in data classification is adopted in this paper. Since the representation of fixed-length chromosome with numeric genes in genetic algorithms is well-suited for

Table 1
The weather dataset

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	70	96	False	Yes
Rainy	68	80	False	Yes
Rainy	65	70	True	No
Overcast	64	65	True	Yes
Sunny	72	95	False	No
Sunny	69	70	False	Yes
Rainy	75	80	False	Yes
Sunny	75	70	True	Yes
Overcast	72	90	True	Yes
Overcast	81	75	False	Yes
Rainy	71	91	True	No

numerical optimization [12], it is used in EvoC to deal with the numeric attributes in the classification.

(A) *GP chromosome structure*: The selection of functions and terminals is the preparatory step in genetic programming [20]. In EvoC, two Boolean operators are adopted as functions, i.e. ‘AND’ and ‘NOT’. These two functions are sufficient to build a basic classification rule in the form of “IF antecedent$_1$ AND (NOT antecedent$_2$) AND . . . THEN consequence.” The classification rules that are built of ‘AND’ and ‘NOT’ can then be combined to form the decision rule set (the ‘OR’ effect). The terminal set contains all possible *attribute–value* pairs for a given dataset. For example, the possible attribute–value pairs could be outlook—overcast, outlook—sunny, outlook—rainy, windy—TRUE or windy—FALSE for the weather dataset. To avoid redundant or conflicting nodes exist in the same tree, these terminals are built into a table and only one *attribute–value* pair can be selected from each attribute entry for a tree structure. The initial population in GP is created with the approach of ‘*ramped-half-and-half*’ [20].

(B) *GA chromosome structure*: The fixed-length real-coding chromosome structure is adopted in GA [12]. The range of each numeric attribute is represented by two real-coded genes: one encodes the upper bound and the other the lower bound. As depicted in Fig. 3, the *M*th $(M \leq N)$ and $(M + N)$th genes encode the range of the *M*th numeric attribute. In the initial population, the lower and upper bound of each attribute is initialized as the corresponding minimum and maximum, respectively. For example, in the weather problem, the minimum and maximum of the two numeric attributes “Temperature” and “Humidity” is (64, 85) and (70, 96), respectively. Therefore, the initialization of chromosomes is given as (64, 70, 85, 96). Obviously, such an approach starts the evolution with generality and subsequently searches for specificity. Since nominal attributes consist of a finite number of values, their hidden relationships are often easier to be discovered. Based on this assumption,

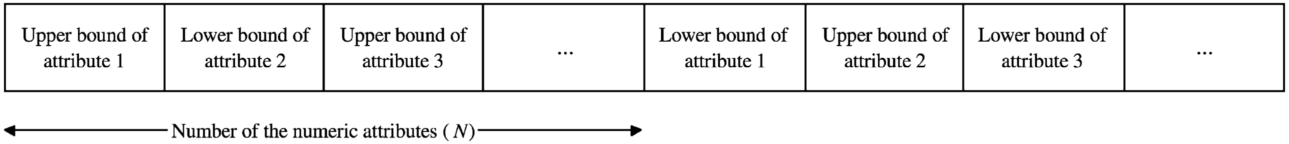
Upper bound of attribute 1
Lower bound of attribute 2
Upper bound of attribute 3
...
Lower bound of attribute 1
Upper bound of attribute 2
Lower bound of attribute 3
...
Number of the numeric attributes (N)

Fig. 3. The chromosome structure in GA.

the above initialization is adopted to give nominal attributes a higher priority. As the evolution process, the range of certain numeric attributes will shrink and the corresponding nominal parts will improve accordingly to produce better classification accuracy.

(C) *Mutation and crossover*: Since the GP in EvoC only deals with nominal attributes, standard tree-based crossover and mutation operators are employed in the GP [20]. However, a specialized mutation operator is used in GA in order to avoid annoyance rules such as age ≤ 45.23. The values of every numeric attributes of the dataset to be learned are stored in a table, and the mutation is performed by fetching a random corresponding value from the table and replaces the value of the attribute in the chromosome. Standard single-point crossover where two parents exchange their genes from a random position to reproduce the offspring is adopted in GA [12]. It should be noted that these genetic operators utilized data in the available datasets for the lower and upper bounds of the attributes, which not only makes the best use of information in the datasets, but also guarantees the meaningfulness of the final rules produced by EvoC.

3.1.2. Automatic attribute selection

Although a dataset often contains many attributes, it is common that only a fraction of the attributes will appear in a single rule. For example, a rule for the weather dataset may be in the form of "IF outlook = sunny and humidity ≥ 83, THEN play = no," where only two out of the five attributes are considered in this rule. This characteristic of rules seems to be counterintuitive to the fixed-length chromosome structure of GA, where all numeric attributes are considered in the evolution. If chromosomes in the GA are converted to rules directly, all the numeric attributes will be included in the rules which may result in the redundant rules. Such contradiction, however, could be overcome by studying the characteristic of chromosomes in GA for succinct presentation of rules.

Suppose a candidate individual in the solution produces a rule in the form of "IF outlook = sunny and $64 \leq$ temperature ≤ 85 and $83 \leq$ humidity ≤ 96, THEN play = no." Since '64' and '85' are the lower and upper limits of the temperature, it casts no restrictions on all data samples and thus the temperature condition will be discarded from the rule. This observation is also applicable to the humidity attribute. Since '96' is the upper limit of humidity, all the instances whose humidity are higher than '83' will satisfy this condition. Hence, '96' is unnecessary and will also be excluded from the rule. After these operations, the final concise rule becomes "IF outlook = sunny and humidity ≥ 83, THEN play = no."

3.1.3. Fitness function

When a rule or individual is used to classify a given training instance, one of the four possible concepts can be observed: true positive (tp), false positive (fp), true negative (tn) and false negative (fn). The true positive and true negative are correct classifications, while false positive and false negative are incorrect classifications. For a two-class case, with class 'yes' and 'no', the four concepts can be easily understood with the following descriptions:

- true positive: the rule predicts that the class is 'yes' (positive) and the class of the given instance is indeed 'yes' (true);

- false positive: the rule predicts that the class is ‘yes’ (positive) but the class of the given instance is in fact ‘no’ (false);
- true negative: the rule predicts that the class is ‘no’ (negative) and the class of the given instance is indeed ‘no’ (false);
- false negative: the rule predicts that the class is ‘no’ (negative) but the class of the given instance is in fact ‘yes’ (true).

Using these concepts, the fitness function used in the first phase of EvoC is defined as

$$\text{fitness} = w \times \frac{\text{tp}}{\text{tp} + \text{fn}} \times \left(1 + \frac{\text{tn}}{\text{tn} + \text{fp}}\right) \qquad (1)$$

with

$$w = \frac{N}{N + \text{fp}} \qquad (2)$$

where N is the total number of instances in the training set and w a penalty factor. The value of the fitness function is in the range of 0–2. The fitness value is 2 (the fittest) when all instances are correctly classified by the rule, i.e. when fp and fn are 0. A penalty factor w that tends to minimize fp is included in the fitness function to evaluate the quality of the combined individuals in the rule set. This is because Boolean sequential rule list (where rules are considered one after another) is very sensitive and tends to have a large number of false positives (fp) due to the virtual ‘OR’ connections among the rules, e.g. when a rule with large fp is considered first in a rule list, many of the instances will be classified incorrectly.

3.1.4. The covering algorithm

The covering algorithm employs the token competition [41] to promote the diversity and to evolve multiple rules in the first phase of EvoC. Multiple rules that cover the same instances in the training set often increase the tendency of premature convergence in the evolution. In most cases, only a few of these multiple rules are useful and cover most of the instances while others are redundant. To achieve the optimal performance for the rule list evolver in the second phase, all rules that are able to cover at least one instance in phase 1 will be retained in the pool of candidate rules, which are maintained by the covering algorithm.

3.2. Phase 2: the rule set evolver

In the phase 1 of EvoC, the hybrid GA–GP approach is applied to find good classification rules in a usually complex search space. The approach is a Michigan-style algorithm where classification performance of the rule set is not needed for fitness evaluations. Although the token competition can serve as a rule selection mechanism, e.g. rules that fail to seize any token (a token represents an example in the dataset) will be eliminated, a Pittsburgh-like approach is required in the second phase in order to find the optimal order and number of rules in a rule set from the pool of candidate rules evolved in phase 1. To determine the optimal number of rules in a rule set, the population in phase 2 is divided into several sub-populations, where each sub-population is dedicated to optimize the order of rules with a given rule number. For example, if the number of rules in the candidate pool is n, then there will be n sub-populations and the ith sub-population will be evolved to optimize the rule set containing i rules.

After the initialization, each sub-population will be evolved independently and there is no interaction among the sub-populations. At the end of the evolution, each sub-population outputs its 'best' candidate rule set, which will compete (based on the classification accuracy) with the 'best' rule sets generated by other sub-populations to obtain the final optimal rule set. In this approach, the order and number of rules in the rule sets can be optimized and determined simultaneously. To retain concise rule sets in the classification, a shorter rule set is preferable to a longer one even if both achieved the same classification accuracy. All rules obtained in the first phase of EvoC are given an index and these rules will be selected randomly to build up the rule sets. Fig. 4 depicts the initialization of two chromosomes having three rules set and six rules set, respectively. Similar to the GA in phase 1 of EvoC, standard single-point crossover and tournament selection schemes are adopted in phase 2. The mutation operation is performed by randomly selecting a rule from the pool of candidate rules to replace the rule for mutation. In phase 2, the fitness function considers the classification accuracy on the training set as given by,

$$\text{fitness} = \frac{\text{tp} + \text{tn}}{N} \tag{3}$$

where N is the total number of instances in the training set.

4. Evolutionary knowledge discovery in medical diagnosis

4.1. The medical diagnosis datasets

The medical diagnosis datasets used in this study are the hepatitis dataset and breast cancer diagnosis databases obtained from University of California, Irvine (UCI) machine-learning

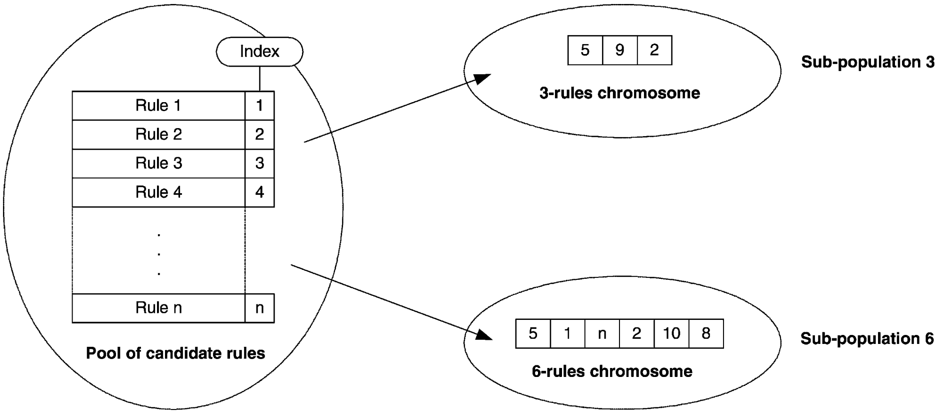

Fig. 4. Example of chromosomes initialization in the phase 2 of EvoC.

Table 2
Summary of the HEPA dataset

Attribute	Possible values
Age	Integer 1–80
Sex	Male, female
Steroid	No, yes
Antivirals	No, yes
Fatigue	No, yes
Malaise	No, yes
Anorexia	No, yes
Liver big	No, yes
Liver firm	No, yes
Spleen palpable	No, yes
Spiders	No, yes
Ascites	No, yes
Varices	No, yes
Bilirubin	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
Alk phosphate	33, 80, 120, 160, 200, 250
SGOT	13, 100, 200, 300, 400, 500
Albumin	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
Protime	10, 20, 30, 40, 50, 60, 70, 80, 90
Histology	No, yes
Class	DIE (20.65%), LIVE (79.35%)

repository at http://www.ics.uci.edu/~mlearn/MLRepository.html. The hepatitis dataset was collected at Carnegie-Mellon University [5] and donated to UCI ML repository in 1988. The two breast cancer diagnosis datasets, i.e. Wisconsin breast cancer database (WBCD) and Wisconsin diagnostic breast cancer (WDBC), were collected at different periods of time with different attributes recorded [33]. The former was donated to UCI ML repository in 1991, while the latter was in 1995 for public access. For both breast cancer datasets, the classification task is to determine case of benign or malignant from the physical attributes of cell given in the datasets. The characteristics of these datasets are briefly described as follows:

(A) *The hepatitis dataset* (*HEPA*): The hepatitis dataset is summarized in Table 2, which consists of 155 instances. Each instance consists of 19 attributes, namely age, sex, steroid, antivirals, fatigue, malaise, anorexia, liver big, liver firm, spleen palpable, spiders, ascites, varices, bilirubin, alk phosphate, SGOT, albumin, protime and histology. This problem includes both nominal and numeric attributes, which is particularly suitable for verifying the performance of EvoC. The HEPA is a complex and noisy dataset since it contains a large number of missing data. The class is distributed with 32 (20.65%) DIE samples and 123 (79.35%) LIVE samples. The classification task is to predict whether a patient with hepatitis will live or die.

(B) *The Wisconsin diagnostic breast cancer* (*WDBC*): The WDBC dataset is summarized in Table 3 and consists of 569 instances. Each instance consists of 10 real-valued attributes of the nuclear for the cancer cell, namely radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal

Table 3
Summary of the WDBC dataset

Attribute	Possible values	Description
Radius	Real	Mean of distances from center to points on perimeter
Texture	Real	Standard deviation of gray-scale values
Perimeter	Real	–
Area	Real	–
Smoothness	Real	Local variation in radius lengths
Compactness	Real	Perimeter^2/area $-$ 1.0
Concavity	Real	Severity of concave portions of the contour
Concave points	Real	Number of concave portions of the contour
Symmetry	Real	–
Fractal dimension	Real	"Coastline approximation" $-$ 1.0
Diagnosis	Benign (62.7%), malignant (37.3%)	–

dimension. These attributes are modeled such that higher values are typically associated with malignancy. The mean, worst (mean of the three largest values), and standard error of each attribute were computed for the original dataset, resulting in a total of 30 attributes. In this study, however, only the mean values were considered in the rule extraction process. Detailed description of these 10 attributes is available from [33]. All the instances have been properly recorded and there is no missing value in this dataset. The diagnosis class is distributed with 357 (62.7%) benign samples and 212 (37.3%) malignant samples.

(C) *The Wisconsin breast cancer database* (*WBCD*): The WBCD dataset is summarized in Table 4 and consists of 699 instances taken from fine needle aspirates (FNA) of human breast tissue. Each instance consists of nine measurements (without considering the sample's code number), namely clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. The measurements are assigned an integer value between 1 and 10, with 1 being the closest to benign and 10

Table 4
Summary of the WBCD dataset

Attribute	Possible values
Clump thickness	Integer 1–10
Uniformity of cell size	Integer 1–10
Uniformity of cell shape	Integer 1–10
Marginal adhesion	Integer 1–10
Single epithelial cell size	Integer 1–10
Bare nuclei	Integer 1–10
Bland chromatin	Integer 1–10
Normal nucleoli	Integer 1–10
Mitoses	Integer 1–10
Class	Benign (65.5%), malignant (34.5%)

the most anaplastic. Associated with each sample is its class label, which is either benign or malignant. This dataset contains 16 instances with missing attributes' values. Since many classification algorithms have discarded these data samples, for the ease of comparison, the same way is followed and the remaining 683 samples are taken for use. Therefore, the class is distributed with 444 (65.0%) benign samples and 239 (35.0%) malignant samples.

4.2. Simulation settings

The EvoC was implemented in Java programming based on the Java Developers Kit (JDK 1.3.1) from Sun Microsystems. The simulations were performed using an Intel Pentium III 933 MHz processor with 512 MB SDRAM. To ensure the validity and replicability of the results, all experiments were designed carefully and all datasets used by the EvoC were partitioned into two sets: a training set and a testing set (or validation set). As indicated by Prechelt [29], the fuzzy specification of the partitioning of training versus testing data is a big obstacle to reproduce or compare published machine-learning results. It is insufficient to only indicate the number of examples for each set in the partition since the experimental results may vary significantly for different partitions even if the numbers in each set are the same [43]. In this work, a total of 100 simulation runs were performed for each of the three medical datasets, and a random seed[1] that is similar to the number of runs (i.e. the 50th simulation run uses a random seed of 50) was used to randomize the orders of data in the datasets. Each randomized dataset was then partitioned into 66% of training data and 34% of testing data as follows:

- for the hepatitis dataset, the first 102 examples are used for the training set and the remaining 53 examples for the testing set;
- for the WDBC dataset, the first 376 examples are used for the training set and the remaining 193 examples for the testing set;
- for the WBCD dataset, the first 451 examples are used for the training set and the remaining 232 examples for the testing set.

Table 5 lists the parameter settings of EvoC used in the simulations. The maximum initial depth and maximum crossover depth are GP specified control parameters, which are used to control the complexity of GP trees during the evolution. The parameter settings in Table 5 were applied to all experiments in this work, which should not be taken as the optimal set of parameters for each problem, but rather a generalized one for which the EvoC performs well over a number of different datasets.

4.3. Simulation results

Table 6 summarizes the classification results produced by EvoC over the 100 independent simulation runs for both the training and testing datasets. To obtain a better

[1] The random number generator used in the experiments is provided by Sun's JDK 1.3.1 and the dataset randomizer used is provided by WEKA. Different partitioning of datasets might be resulted under different programming environments.

Table 5
The setting of parameters in EvoC

Parameters	Parameter description	Phase 1	Phase 2
MaxInitDepth	The permitted depth of GP tree in initialization	6	–
MaxCrossoverDepth	The permitted depth of GP tree after crossover	17	–
ReproductionProb	The probability of an individual that will be copied to the next generation without changes	0.1	–
MutationProb	The probability of mutation	0.5	0.1
CrossoverProb	The probability of crossover	0.9	0.8
MaxGeneration	The generation number for the evolution	100	50
PopulationSize	The population size for the evolution	100	50

understanding of the classification performances for the different simulations, the histograms that summarize the experiment results of the three datasets are shown in Figs. 5–7. For all the histograms, the classification performance axis indicates the classification accuracy achieved by the different number of rule sets obtained over the 100 independent simulation runs.

Tables 7–9 list the classification rules having the highest predictive accuracy (i.e. the classification accuracy on the testing dataset) for the three medical datasets. Besides the fitness value, support factor and confidence factor are also provided to measure the performance of each rule. The support factor measures the coverage of a rule, which is the ratio of the number of instances covered by the rule to the total number of instances. The confidence factor measures the accuracy of a rule. For a rule “IF X THEN Y” and a training set of N instances, the support factor and confidence factor are given as

$$\text{support} = \frac{\text{number of instances with both } X \text{ and } Y}{N} \tag{4}$$

$$\text{confidence} = \frac{\text{number of instances with both } X \text{ and } Y}{\text{number of instances with } X} \tag{5}$$

Table 6
Summary of the results in EvoC over the 100 independent simulation runs

Classification accuracy	HEPA	WDBC	WBCD
Training			
Maximum (%)	90.20	96.28	99.33
Minimum (%)	79.41	91.22	96.23
Mean (%)	85.04	94.36	97.80
S.D. (%)	1.76	0.91	0.51
Testing			
Maximum (%)	94.34	96.37	99.13
Minimum (%)	75.47	88.60	95.26
Mean (%)	83.92	93.04	97.57
S.D. (%)	4.03	1.47	0.85
Average number of rules	2.93	9.74	5.99

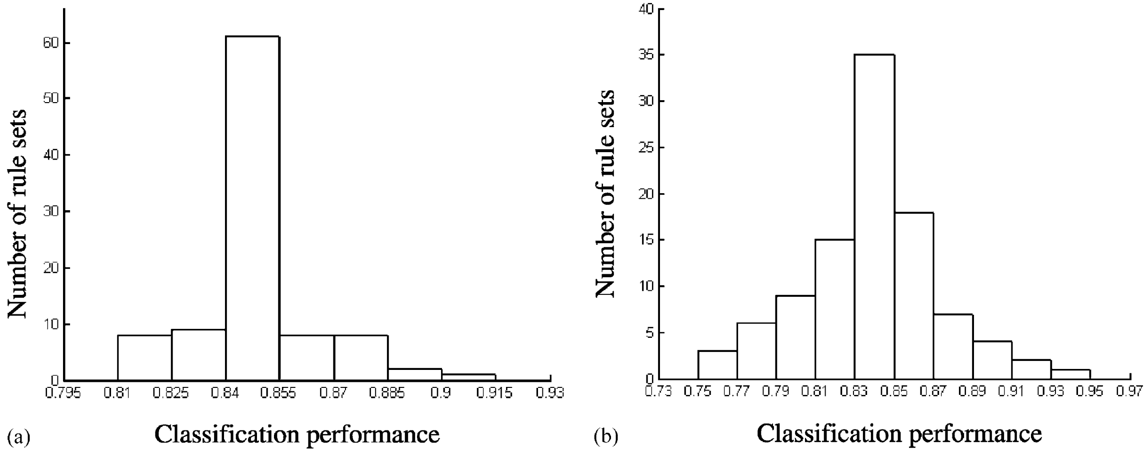

Fig. 5. The performance of EvoC for the HEPA problem: (a) training; (b) testing.

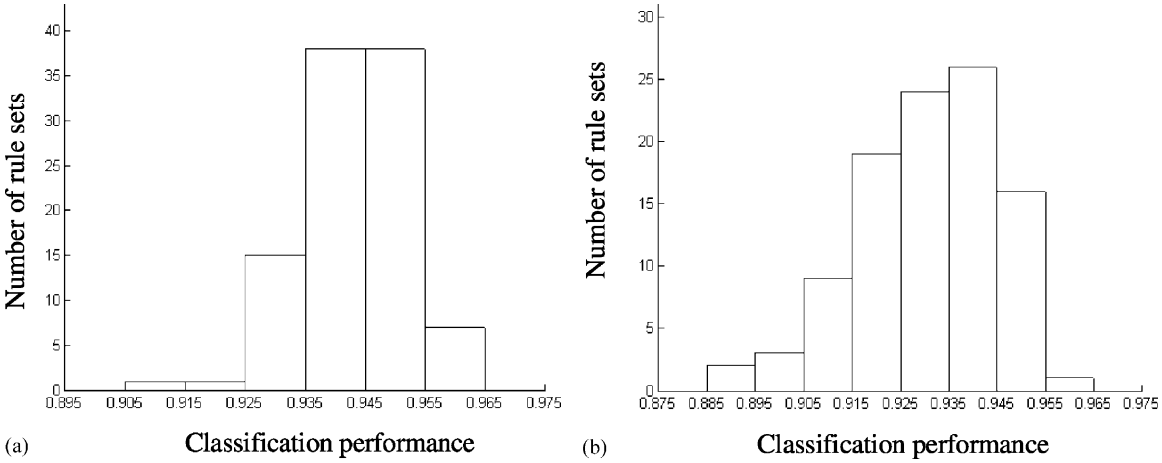

Fig. 6. The performance of EvoC for the WDBC problem: (a) training; (b) testing.

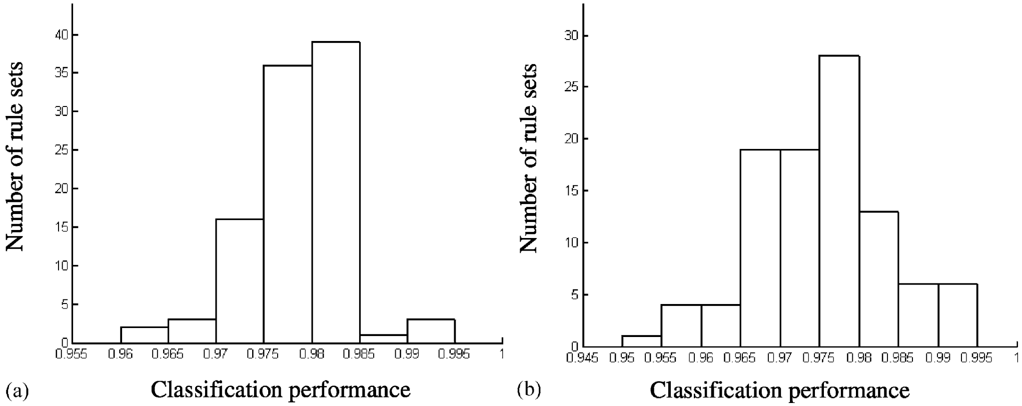

Fig. 7. The performance of EvoC for the WBCD problem: (a) training; (b) testing.

Table 7
The best rule set of HEPA with an accuracy of 94.34%

No.	Rule	Fitness	Support factor	Confidence factor
1	IF Fatigue = yes AND Age $\geq$ 30.0 AND ALK phosphate $\leq$ 280.0 AND Albumin $\leq$ 4.3 AND Protime $\leq$ 46.0 THEN Class = DIE	1.2338	0.1961	0.5128

2	IF Anorexia = no
	AND Bilirubin $\leq$ 1.8
	AND SGOT $\leq$ 420.0
	THEN Class = LIVE
		1.0912	0.5588	0.8636

3	IF Spiders = yes AND Age $\geq$ 30.0 AND 62.0 $\leq$ ALK phosphate $\leq$ 175.0 AND Albumin $\leq$ 4.3 AND Protime $\leq$ 85.0 THEN Class = DIE	1.2989	0.1765	0.6667
4	ELSE Class = LIVE			

Table 8
The best rule set of WDBC with an accuracy of 96.37%

No.	Rule	Fitness	Support factor	Confidence factor
1	IF Radius $\leq$ 14.95 AND Perimeter $\leq$ 116.1 AND Concavity $\leq$ 0.313 AND Concave points $\leq$ 0.04908 THEN Diagnosis = benign	1.6774	0.5479	0.9763
2	IF Radius $\geq$ 13.0 AND Texture $\geq$ 15.76 AND Perimeter $\geq$ 74.72 AND Area $\geq$ 572.6 AND Concavity $\geq$ 0.03885 AND Concave points $\geq$ 0.02402 THEN Diagnosis = malignant	1.5315	0.3032	0.8769
3	IF Radius $\leq$ 17.01 AND Perimeter $\leq$ 116.1 AND Concavity $\leq$ 0.1122 AND Concave points $\leq$ 0.1265 THEN Diagnosis = benign	1.5876	0.5931	0.8956
4	ELSE Diagnosis = malignant			

Table 9
The best rule set of WBCD with an accuracy of 99.13%

No.	Rule	Fitness	Support factor	Confidence factor
1	IF Clump thickness $\leq$ 8.0 AND Uniformity of cell shape $\leq$ 8.0 AND Marginal adhesion $\leq$ 3.0 AND Bare nuclei $\leq$ 5.0 AND Bland chromatin $\leq$ 7.0 AND Normal nucleoli $\leq$ 8.0 THEN Class = benign	1.8702	0.6186	0.9789
2	IF Uniformity of cell shape $\geq$ 3.0 AND Single epi cell size $\geq$ 2.0 AND Bland chromatin $\geq$ 2.0 THEN Class = malignant	1.739	0.3459	0.8571
3	IF Clump thickness $\leq$ 8.0 AND Uniformity of cell size $\leq$ 4.0 AND Bland chromatin $\leq$ 3.0 AND Normal nucleoli $\leq$ 9.0 AND Mitoses $\leq$ 1.0 THEN Class = benign	1.783	0.5898	0.9779
4	ELSE Class = malignant			

A careful examination of the relationship between the predictive accuracy of a rule set and its number of rules reveals an interesting finding. The rule sets with a large number of rules will not necessarily lead to high predictive accuracy, although they generally provide good performances on the training sets. It can also be observed that the first few rules in a rule set often cover a large portion of the samples and left relatively few samples for the remaining rules. Therefore when the dataset is not noise-free, a large number of rules may cause over-fitting and leads to poor generalization. For example, in the WBCD problem, all of the best six rule sets that achieve a predictive accuracy of above 99% only contain an average of four rules. However, the four largest rule sets (all of which contain more than 15 rules) only produce an average accuracy of 97.75% on the testing samples.

4.4. *Performance comparisons*

This section compares the performance of EvoC with three popular machine-learning algorithms, i.e. *C4.5*, *PART* and *Naïve Bayes*. These algorithms are briefly described below:

- The C4.5 proposed by Quinlan [30] is a landmark decision tree program that has been widely used in practice.
- The PART is a rule-learning scheme capable of generating classification rules [9].
- The Naïve Bayes utilizes the Bayesian techniques, which has been studied by many machine-learning researchers [16].

Table 10
The P-values of the paired t-tests against C4.5, PART and Naïve Bayes

Algorithm	HEPA	WDBC	WBCD
C4.5 [30]	1.362×10^{-12}	3.545×10^{-2}	1.607×10^{-32}
PART [9]	5.565×10^{-10}	2.364×10^{-3}	2.990×10^{-30}
Naïve Bayes [16]	0.313	2.644×10^{-9}	1.302×10^{-17}

Table 11
The comparison results for the HEPA dataset

Algorithm	Number of rules	Time (s)	Average accuracy (%)	Best accuracy (%)	S.D. (%)
EvoC	2.93	4.84×10^5	83.92	94.34	4.03
C4.5 [30]	5.85	<1	78.94	90.57	4.84
PART [9]	6.64	<1	80.02	94.34	4.98
Naïve Bayes [16]	–	<1	83.62	94.34	4.90
GA-based FKIF [39]	–	–	–	92.9	–

Besides comparing the average results and standard deviations of the algorithms over the 100 simulation runs, a paired t-test [25] has also been performed, since the comparison results may be affected by the distribution, noise or partition of datasets. In addition, the best results for the three datasets available in the literature[2] according to the authors' best knowledge are also provided in the comparisons. Table 10 lists the P-values of the paired t-tests against the algorithms of C4.5, PART and Naïve Bayes for the three datasets. As can be seen, the P-values are rather small showing that the EvoC has outperformed the approaches of C4.5, PART and Naïve Bayes with a great confidence.

(A) *Comparison results for the HEPA dataset*: Wang et al. [39] proposed an evolutionary rule-learning algorithm, called GA-based fuzzy knowledge integration framework (GA-based FKIF), which utilized genetic algorithms to generate an optimal or near-optimal set of fuzzy rules and membership functions from the initial population of knowledge. As shown in Table 11, only the best result produced by this algorithm is compared with EvoC since the average performance of GA-based FKIF was not provided in [39]. The P-values of the paired t-tests on HEPA dataset as listed in Table 10 (EvoC versus C4.5: $P = 1.36 \times 10^{-12}$; EvoC versus PART: $P = 5.56 \times 10^{-10}$; EvoC versus Naïve Bayes: $P = 0.31$) show that the EvoC outperforms C4.5 and PART, and is comparable to Naïve Bayes based on the average results over the 100 simulation runs when the level of significance α is set as 0.005.

(B) *Comparison results for the WDBC dataset*: Table 12 compares the results from EvoC, C4.5, PART, and Naïve Bayes for the WDBC dataset. It can be seen that the

[2] Recently, the WBCD dataset is widely adopted by many machine-learning algorithms in the medical domain. Therefore comparisons between different algorithms based on this dataset are relatively more comprehensive than the other two datasets studied in this paper.

Table 12
The comparison results for the WDBC dataset

Algorithm	Number of rules	Time (s)	Average accuracy (%)	Best accuracy (%)	S.D. (%)
EvoC	9.74	4.50×10^5	93.04	96.37	1.47
C4.5 [30]	10.06	<1	92.61	97.93	1.98
PART [9]	6.23	<1	92.35	97.41	1.65
Naïve Bayes [16]	–	<1	91.56	95.37	2.01

EvoC produces competitive classification accuracies, besides giving the smallest standard deviation among all methods. In addition, the P-values of the paired t-tests on WDBC dataset as listed in Table 10 (EvoC versus C4.5: $P = 3.54 \times 10^{-2}$; EvoC versus PART: $P = 2.36 \times 10^{-3}$; EvoC versus Naïve Bayes: $P = 2.64 \times 10^{-9}$) show that the EvoC outperforms the algorithms of C4.5, PART and Naïve Bayes based on the average results over the 100 simulation runs when the level of significance α is set as 0.05.

(C) *Comparison results for the WBCD dataset*: Peña-Reyes and Sipper [26] proposed a fuzzy-genetic approach by combining fuzzy logic and evolutionary algorithms to form a diagnostic system. In the total of 120 evolutionary runs [26], 78 runs led to fuzzy systems with accuracies exceed 96.5% and 8 runs with accuracies exceed 97.5%. As shown in Table 13, only the three best performances of fuzzy-genetic approach are comparable to the average results of 97.57% by EvoC. Moreover, the average performance of fuzzy-genetic approach over the 120 runs is 96.02%, which is only slightly better than the worst rule sets (with a predictive accuracy of 95.26%) generated by EvoC. If only the best results are considered, a four-rule fuzzy system achieves the predictive accuracy of 98.24%, which is lower than the best predictive accuracy of 99.13% by EvoC.

The EvoC has also been compared with the NeuralRule approach proposed by Setiono [32], which is capable of extracting classification rules from trained neural networks. Setiono [32] trained 200 neural networks in total and after pruning the network to 95 and 98% accuracies on the training set, an accuracy of 95.44 and 96.66% was achieved on the testing set, respectively. In terms of the best results produced by the pruned networks,

Table 13
The comparison results for the WBCD dataset

Algorithm	Number of rules	Time (s)	Average accuracy (%)	Best accuracy (%)	S.D. (%)
EvoC	5.99	3.35×10^5	97.57	99.13	0.51
C4.5 [30]	8.99	<1	95.09	97.84	1.16
PART [9]	9.03	<1	95.33	98.28	1.16
Naïve Bayes [16]	–	<1	96.37	98.28	0.89
NeuroRule—rule 3 [32]	5	–	–	98.24	–
Fuzzy-GA4 [26]	4	–	96.02	98.24	–

NeuralRule achieves an accuracy of 98.25% on the testing set, which is lower than the best predictive accuracy of 99.13% by EvoC.

Although the EvoC is capable of evolving comprehensible classification rules with good generalization performance, it often requires extensive computational effort as compared to existing approaches. The EvoC is generally developed for off-line data classification, which could be useful for many applications where the training time is less important than the generalization in classification. To reduce the computational effort significantly, the EvoC is currently being integrated into the 'Paladin-DEC' distributed evolutionary computing framework [37], where multiple inter-communicating subpopulations will be implemented to share and distribute the classification workload among multiple computers over the Internet.

5. Conclusions

A two-phase hybrid evolutionary classifier capable of extracting comprehensible classification rules with good accuracy in medical diagnosis has been proposed in this paper. In the first phase, genetic programming has been applied to evolve nominal attributes for free structured rules while genetic algorithms have been used to optimize the numeric attributes for concise classification rules without the need of discretization. The second phase then formulates accurate rule sets by optimizing the order and number of rules in the evolution based upon the pool of confined candidate rules obtained in the phase 1. The proposed evolutionary classifier has been validated upon one hepatitis and two breast cancer datasets, which are representative real-world data collected to aid the prognosis and diagnosis of disease. Simulation results show that the EvoC produces comprehensible and good classification rules for the three medical datasets. Results obtained from the *t*-tests further justify its robustness and invariance to random partition of datasets.

Acknowledgements

The authors would like to thank Gail Gong and Dr. W.H. Wolberg for making the hepatitis and breast cancer datasets, respectively, available in public, and the WEKA Development Group for providing the source code of WEKA. The authors also wish to thank Prof. Klaus-Peter Adlassnig and the anonymous reviewers for their valuable comments and helpful suggestions, which greatly improved the paper's quality.

References

[1] Banzhaf W, Nordin P, Keller RE, Francone FD. Genetic programming: an introduction on the automatic evolution of computer programs and its applications. San Francisco (CA): Kaufmann (Morgan); 1998.

[2] Bojarczuk CC, Lopes HS, Freitas AA. Genetic programming for knowledge discovery in chest-pain diagnosis. IEEE Eng Med Biol Mag 2000;4(19):38–44.

[3] Brameier M, Banzhaf W. A comparison of linear genetic programming neural networks in medical data mining. IEEE Trans Evol Comput 2001;5(1):17–26.

[4] Cattral R, Oppacher F, Deugo D. Rule acquisition with a genetic algorithm. Proc IEEE Cong Evol Comput 1999;1:125–9.

[5] Cestnik G, Konenenko I, Bratko I. Assistant-86: a knowledge-elicitation tool for sophisticated users. In: Bratko I, Lavrac N, editors. Machine learning. Wilmslow: Sigma Press; 1987. p. 31–45.

[6] Chang YH, Zheng B, Wang XH, Good WF. Computer-aided diagnosis of breast cancer using artificial neural networks: comparison of backpropagation and genetic algorithms. In: Proceedings of the International Joint Conference on Neural Networks, Washington, DC, USA, vol. 5. Washington (DC): IEEE Press; 1999. p. 3674–9.

[7] Congdon CB. Classification of epidemiological data: a comparison of genetic algorithm and decision tree approaches. In: Proceedings of the IEEE Congress on Evolutionary Computation, La Jolla Marriott, San Diego, CA, USA, vol. 1. Piscataway (NJ): IEEE Press; 2000. p. 442–9.

[8] Fidelis MV, Lopes HS, Freitas AA. Discovering comprehensible classification rules with a genetic algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, La Jolla Marriott, San Diego, CA, USA, vol. 1. Piscataway (NJ): IEEE Press; 2000. p. 805–10.

[9] Frank E, Witten IH. Generating accurate rule sets without global optimization. In: Proceedings of the 15th International Conference Machine Learning (ICML'98), Madison, WI, USA. San Francisco (CA): Kaufmann (Morgan); 1998. p. 144–51.

[10] Freitas AA. A survey of evolutionary algorithms for data mining and knowledge discovery. In: Ghosh A, Tsutsui S, editors. Advances in evolutionary computation. Berlin: Springer; 2002.

[11] Garner SR. WEKA: the Waikato environment for knowledge analysis. In: Proceedings of the New Zealand Computer Science Research Students Conference, The University of Waikato, Hamilton, New Zealand, 1995. p. 57–64.

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

[13] Howard LM, D'Angelo DJ. The GA-P: a genetic algorithm and genetic programming hybrid. IEEE Expert 1995;(June):11–5.

[14] Hruschka ER, Ebecken NFF. A clustering genetic algorithm for extracting rules from supervised neural network models in data mining tasks. Int J Comput Syst Signals 2000;1(1):17–29.

[15] Hu YJ. Constructive induction covering attributes spectrum. In: Liu H, Motoda H, editors. Feature extraction construction and selection: a data mining perspective. Norwell (MA): Kluwer Academic Publishers; 1998. p. 257–69.

[16] John GH, Langley P. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, McGill University, Montreal, Que., Canada. San Mateo (CA): Kaufmann (Morgan); 1995. p. 338–45.

[17] Joshi MV, Agarwal RC, Kumar V. Mining needles in a haystack: classifying rare classes via two-phase rule induction. In: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA; 2001. p. 91–101.

[18] Kim KJ, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst Appl 2000;19(2):125–32.

[19] Komosinski M, Krawiec K. Evolutionary weighting of image features for diagnosing of CNS tumors. Artif Intell Med 2000;19:25–38 [special issue on evolutionary computation in medicine].

[20] Koza JR. Genetic programming: on the programming of computers by means of natural selection. Cambridge (MA): MIT Press; 1992.

[21] Kupinski MA, Anastasio MA. Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves. IEEE Trans Med Imaging 1999; 18(8):675–85.

[22] Liu JNK, Li BNL, Dillon TS. An improved Naïve Bayesian classifier technique coupled with a novel input solution method. IEEE Trans Syst Man Cybernetics C: Appl Rev 2001;31(2):249–56.

[23] Meesad P, Yen GG. A hybrid intelligent system for medical diagnosis. In: Proceedings of International Joint Conference on Neural Networks, vol. 4. Piscataway (NJ): IEEE Press; 2001. p. 2558–63.

[24] Michalewicz Z. Genetic algorithms + data structures = evolution programs. London: Kluwer Academic Publishers; 1994.

[25] Montgomery DC, Runger GC, Hubele NF. Engineering statistics. 2nd ed. New York: Wiley; 2001.

[26] Peña-Reyes CA, Sipper M. A fuzzy-genetic approach to breast cancer diagnosis. Artif Intell Med 1999;17(2):131–55.

[27] Peña-Reyes CA, Sipper M. Evolutionary computation in medicine: an overview. Artif Intell Med 2000;19(1):1–23.

[28] Peña-Reyes CA, Sipper M. Fuzzy CoCo: a cooperative coevolutionary approach to fuzzy modeling. IEEE Trans Fuzzy Syst 2001;9(5):727–37.

[29] Prechelt L. Some notes on neural learning algorithm benchmarking. NeuralComputing 1995;9(3):343–7.

[30] Quinlan JR. C4.5: programs for machine learning. San Mateo (CA): Kaufmann (Morgan); 1992.

[31] Setiono R. Extracting rules from pruned neural networks for breast cancer diagnosis. Artif Intell Med 1996;8(1):37–51.

[32] Setiono R. Generating concise and accurate classification rules for breast cancer diagnosis. Artif Intell Med 2000;18(3):205–19.

[33] Street WN, Wolberg WH, Mangasarian OL. Nuclear feature extraction for breast tumor diagnosis. In: IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, vol. 1905, San Jose, CA, 1993. p. 861–70.

[34] Taha I, Gosh J. Symbolic interpretation of artificial neural networks. Technical report TR-97-01-106. Austin: The Computer and Vision Research Center, University of Texas; 1996.

[35] Tan KC, Lee TH, Khoo D, Khor EF. A multi-objective evolutionary algorithm toolbox for computer-aided multi-objective optimization. IEEE Trans Syst Man Cybernetics B (Cybernetics) 2001;31(4):537–56.

[36] Tan KC, Tay A, Lee TH, Heng CM. Mining multiple comprehensible classification rules using genetic programming. In: IEEE Congress on Evolutionary Computation, Honolulu, HI, 2002. p. 1302–7.

[37] Tan KC, Khor EF, Cai J, Heng CM, Lee TH. Automating the drug scheduling of cancer chemotherapy via evolutionary computation. Artif Intell Med 2002;25:169–85.

[38] Wang CH, Hong TP, Tseng SS, Liao CM. Automatically integrating multiple rule sets in a distributed-knowledge environment. IEEE Trans Syst Man Cybernetics C: Appl Rev 1998;28(3):471–6.

[39] Wang CH, Hong TP, Tseng SS. Integrating membership functions and fuzzy rule sets from multiple knowledge sources. Fuzzy Sets Syst 2000;112(1):141–54.

[40] Witten IH, Frank E. Data mining: practical machine learning tools and techniques with Java implementations. San Francisco (CA): Kaufmann (Morgan); 1999.

[41] Wong ML, Leung KS. Data mining using grammar based genetic programming and applications. London: Kluwer Academic Publishers; 2000.

[42] Wong ML, Lam W, Leung KS, Ngan PS, Cheng JCY. Discovering knowledge from medical databases using evolutionary algorithms. IEEE Eng Med Biol Mag 2000;19(4):45–55.

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