

Evolutionary Data Mining Approaches for Rule-based and Tree-based Classifiers

Thomas Weise

*Nature Inspired Computation & Applications Laboratory
University of Science and Technology of China
Hefei, Anhui, China
Email: tweise@ustc.edu.cn*

Raymond Chiong

*Faculty of Information & Communication Technologies
Swinburne University of Technology
Melbourne, Australia
Email: rchiong@swin.edu.au*

Preview

This document is a preview version and not necessarily identical with the original.

<http://www.it-weise.de/>

Reference: [1]

Abstract—Data mining is an important process, with applications found in many business, science and industrial problems. While a wide variety of algorithms have already been proposed in the literature for classification tasks in large data sets, and the majority of them have been proven to be very effective, not all of them are flexible and easily extensible. In this paper, we introduce two new approaches for synthesizing classifiers with Evolutionary Algorithms (EAs) in supervised data mining scenarios. The first method is based on encoding rule sets with bit string genomes and the second one utilizes Genetic Programming to create decision trees with arbitrary expressions attached to the nodes. Comparisons with some sophisticated standard approaches, such as C4.5 and Random-Forest, show that the performance of the evolved classifiers can be very competitive. We further demonstrate that both proposed approaches work well across different configurations of the EAs.

Keywords—data mining; evolutionary algorithms; rule-based classifiers; decision trees

I. INTRODUCTION

Data mining is the process of extracting implicit, not yet known information from data [2]. It is an important process in many science and business areas, and has been the subject of many articles in business and software magazines over the last decade. Today, data mining is finding increasing acceptance in the industry mainly due to the rapid growth in the amount of data stored in databases. This growth is occurring in several application areas including astronomy, bioinformatics, drug discovery, advertising, customer relationship management, fraud detection, health care, manufacturing, targeted marketing, financial transactions, government data, environmental monitoring and the Web, among others.

The need to analyze large sets of data in order to discover trends and hidden knowledge which would not otherwise

be found has made data mining one of the most active research fields. Numerous algorithms have been proposed in the literature for classification tasks in large data sets, and the majority of them have been proven to be very effective (e.g. [3, 4, 5, 6, 7, 8, 9, 10]). Many of the existing algorithms for supervised data mining fall into either of the two categories: *decision tree algorithms* or *rough set theory algorithms*. Among them, decision tree algorithms are the more popular ones. This kind of algorithms typically consists of two phases, i.e., a tree-building phase and a tree-pruning phase. Of the many decision tree algorithms that have been used in data mining, C4.5 [3] and Random-Forest [8, 11] are by far the most well-known.

Apart from decision tree algorithms and rough set theory algorithms, Genetic Algorithms (GAs) have also been used extensively in data mining tasks. The strength of GAs is that they require very little domain-specific knowledge of data to be classified, thus simplifying the classification tasks. There are two types of GA-based approaches: the Pittsburgh approach [12] and the Michigan approach [13]. The Pittsburgh approach resembles a traditional GA in which each individual in the population is a set of rules representing a complete solution. The Michigan approach, on the other hand, uses the entire population to represent individual rules, where each rule is a partial solution to the overall learning task. Some examples of recently proposed GA-based approaches can be found in [10, 14, 15, 16].

In this paper, we propose two approaches utilizing Evolutionary Algorithms (EAs) for supervised classification tasks: a rule-based approach and a tree-based approach. We compare their performances with some established classification methods by applying them to well-known example data sets. The results of these comparisons show that both approaches are promising and are comparable to the well-established algorithms.

The rest of this paper is organized as follows. Section II starts with some theoretical background regarding the task of classification in supervised data mining, followed by a brief description of EAs. Subsequently, the two proposed approaches are presented in section III. In section IV, we describe the experimental settings, including the data sets used and the classifiers to be compared, and discuss the

experimental results. Finally, section V concludes the study.

II. BACKGROUND

A. The Classification Task

The task of classification is to assign a class k from a set of possible classes K to vectors (data samples) $t = (t_1, t_2, \dots, t_n)$ consisting of n attribute values $t_i \in \mathbb{T}_i$. In supervised approaches, the starting point is a training set A including training samples $t \in A$ for which the corresponding classes $class(t) \in K$ are already known. The data mining algorithm is supposed to learn a relation (called *classifier*) $C : \mathbb{T}_1 \times \mathbb{T}_2 \times \dots \times \mathbb{T}_n \mapsto K$ which can map such attribute vectors t to a corresponding class $k \in K$. The training set A usually contains only a small subset of the possible attribute vectors and may even include contradicting samples ($a = b : a, b \in A \wedge class(a) \neq class(b)$).

The better the classifier C is, the more often it can correctly classify attribute vectors. An overfitted classifier learns only the exact relations provided in the training sample but is unable to classify samples not included in A with reasonable precision. In order to test whether a classifier C is overfitted, not the complete available data \mathbf{A} is used for learning. Instead, \mathbf{A} is divided into the training set A and the test set \bar{A} . If C 's precision on \bar{A} is much worse than on A , it is overfitted and should not be used in a practical application.

B. Evolutionary Algorithms

EAs, which typically include GAs, Genetic Programming (GP), Evolutionary Programming (EP) and Evolution Strategies (ES), belong to the family of nature-inspired optimization algorithms (see [17, 18]). In general, an EA can be schematized as a population-based search which is characterized by an initial creation of a set of candidate solutions and a generation cycle; the population of candidate solutions is presumed to evolve over the generation cycles utilizing some forms of natural processes such as selection and reproduction in order to refine the solutions iteratively [19, 20].

First, a set of randomly configured candidate solutions are created. The cycle itself then starts with the evaluation of the objective values of these solutions. Based on the results, a relative fitness is assigned to each candidate solution in the population. These fitness values are the criterion on which selection algorithms operate to pick the most promising individuals for further investigation while discarding the less successful ones. The candidate solutions which managed to enter the so-called *mating pool* are then reproduced, i.e., combined via crossover and/or slightly changed by mutation operations. When this is done, the cycle starts again in the next generation.

III. EVOLUTIONARY DATA MINING APPROACHES

A. Rule-Based Classifiers

Our first classification method is very similar to the Pittsburgh Learning Classifier System (LCS) [12, 21, 22]. Like the Pittsburgh LCSs, we use a GA to work on a population of classifier systems encoded as bit strings, each of which being a list of rules (the individual classifiers which together form the classifier system). A rule contains a classification part encoding a class $k \in K$ and a condition for each feature in the input data. Unlike the LCSs, the conditions of our rule-based approach are no simple ternary patterns (0, 1, and * for *don't care*) to be matched against the data samples, but encode a more complex relation.

ID	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
1	5.1	3.5	1.4	0.2	iris setosa
2	4.9	3.0	1.4	0.2	iris setosa
3	7.0	3.2	4.7	1.4	iris versicolor
4	6.3	3.3	6.0	2.5	iris virginica
...
n	6.4	3.2	4.5	1.4	iris versicolor

Rule 1	1	7.42	7.18	0	4.08	3.92	1	1.39	3.36	1	0.1	2.18	0
Rule 2	0	5.98	4.54	0	2.48	3.12	0	5.33	5.33	0	0.42	2.34	1
Rule 3	3	5.5	7.18	0	3.92	2.0	2	1.78	6.51	0	2.18	1.70	1
Rule 4	3	7.42	6.7	1	3.44	3.12	0	1.78	6.5	0	0.9	2.34	2
	c_i, x	c_i, a	c_i, b	c_i, x	c_i, a	c_i, b	c_i, x	c_i, a	c_i, b	c_i, x	c_i, a	c_i, b	k

Figure 1: An example rule-based classifier applied to the Iris Dataset [23, 24].

Figure 1 sketches the application of such a rule-based classifier system to the well-known Iris Dataset [23, 24], where each data sample t stands for a flower and is characterized by four real attributes ($t_1 \dots t_4$). In our approach, each rule of the classifier system therefore consists of four conditions $c_1 \dots c_4$. Each condition c_i , in turn, consists of one operation selector c_i, x and two real values c_i, a and c_i, b . If $c_i, x = 0$, the condition matches if $t_i < \min\{c_i, a, c_i, b\}$ and for $c_i, x = 1$, it is true if and only if $t_i > \max\{c_i, a, c_i, b\}$. t_i must fall into the real interval $[\min\{c_i, a, c_i, b\}, \max\{c_i, a, c_i, b\}]$ if $c_i, x = 2$ and $c_i, x = 3$ stand for *don't care* (like the * symbol in LCSs). All rules of a classifier system are matched against a data sample t and the corresponding class of the rule with the least unmet conditions is returned in the end.

Figure 1 illustrates the application of such a classifier system to the third instance of the Iris Dataset. By chance, the third classifier has the fewest mismatching conditions (one) and therefore, the data sample is classified with class 1.

Since we use a GA to evolve the rule-based classifiers, a binary encoding is needed for the real values c_i, a and c_i, b in the conditions. We use at most four bits for each condition. In cases where an attribute can take on more than sixteen values, the linear scaling mechanism defined in Equation 1 is applied to translate values z (four bits) to the corresponding condition c_i, a and c_i, b respectively.

$$\min\{t_i : \forall t \in T\} + z \frac{\max\{t_i : \forall t \in T\} - \min\{t_i : \forall t \in T\}}{15} \quad (1)$$

1) *Related Approaches*: LCSs are among the most common applications of GAs to classification. In many practical scenarios, approaches derived from their more straightforward variant, the Pittsburgh LCSs, are used. The evolution of binary-coded rule sets in different forms, very often also involves notations for *don't care* symbols, has been researched in several application areas [25, 26].

Our rule set evolution is similar to the approach used by Corcoran and Sen [27]. In their approach, classification rules are encoded in a real-valued genome where each condition is defined as a range $[c_i.a, c_i.b]$ given by two real values $c_i.a$ and $c_i.b$. A *don't care* situation occurs when $c_i.a > c_i.b$. Different from ours, they use a fixed-length genome where the number of rules is predetermined. Notably, they consider only exact matching rules during the classification process. In our approach, the rule that has the least errors wins, if no perfect match could be found. Furthermore, Corcoran and Sen [27] use an internal voting mechanism in case of draws between two rules matching a sample. For us, the rule with the lower index wins. We can use this much simpler approach since our EAs are able to permute rules via crossover and thus, can find their best order during the course of the evolution.

B. Tree-based Classifiers

The synthesis of decision trees is a very common data mining method. The leaf nodes of decision trees contain classes $k \in K$ and each inner node usually has two children. The starting point for the classification of a data sample $t \in A$ is the root of the tree. From there on, a path is followed to one of the leaves which then represents the estimated class of t . At each node on this path, a decision is usually made based upon the comparison of one attribute value with a constant. According to this decision, either the left or the right child of the node is visited. Eventually, the control token will reach a leaf node and the class attached to this node is returned.

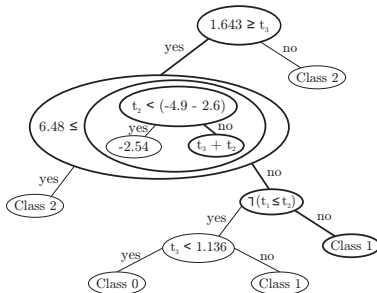


Figure 2: An evolved decision tree applied to the same data sample as used in Figure 1.

We extend this approach by releasing all constraints on the expressions leading to the decisions. In other words, instead of a simple comparison like $t_3 > 1.4$, complex formulas like

$(t_3 * t_1 - 4 \leq 2 * t_2) \vee (t_3 - t_1 > t_2 * 0.2)$ can be attached to a node. Furthermore, the shape of the trees is subject to optimization as well. GP [28] is a type of EAs where the search space consists of trees and therefore, is predestined for the creation of decision trees. In order to evolve arbitrarily structured decision trees, we apply GP with a function set $F = \{+, -, *, \wedge, \vee, \neg, =, >, <, \leq, \geq, \neq, \text{if-then-else}\}$ and a terminal set consisting of constants, references to the attributes, and class identifiers.

In Figure 2, an evolved decision tree with complex node expressions is applied to the same data sample used in Figure 1. The control flow is highlighted by marking the nodes and transitions with bold lines.

1) *Related Approaches*: Besides Forsyth's experiments [29], another set of early results in the area of evolving decision trees has been contributed by Koza [30], as he applied a GP approach to an example problem also used by Quinlan for illustrating the ID3 algorithm – the predecessor of the popular C4.5 algorithm [3].

Freitas [31] used a GP framework to create classification rules composed of comparison and logical operators. In problems with more than two classes ($|K| > 2$), he applied an approach similar to the Michigan-style LCSs: a niching method which allows the selection of $|K| - 1$ classification rules, each matching a different class.

The evolution of *oblique* decision trees has been pursued by Cantú-Paz and Kamath [32]. The decision expressions attached to the nodes of such trees have the form $\sum_{i=1}^n a_i t_i > 0$ where the factors $a_1 \dots a_n$ are subject to optimization. The trees evolved in our work are much less restricted in their form and can, in fact, use arbitrary mathematical expressions. The size of the expressions in the nodes of our approach may vary and is subject to optimization too, which increases the information density in the classifiers.

In [33], a multi-tree approach for solving classification problems with $|K| \geq 2$ is presented. This approach differs from ours in that it uses one distinct tree for deciding whether a data sample belongs to one of the $|K|$ classes or not, whereas we use one single tree only whose leaves are labels which may belong to any of the classes.

C. Objective Functions and Voting

1) *Objective Functions*: In our approaches, the optimization processes are driven by three objective functions, $f_1 \dots f_3$, all subject to minimization. Their values are determined by applying the classifiers to each sample in the training data A . f_1 corresponds to the number of samples which have been classified correctly. Since we generally use minimization, we negate this value. The value of f_2 is determined by a cost matrix which can be specified *a priori*. It also allows the introduction of different penalties for certain misclassifications. For instance, classifying an *iris setosa* as *iris virginica* might be less critical than classifying it as an *iris versicolor*. If no cost matrix is provided, the

identity matrix is used instead and f_1 and f_2 become interchangeable. The third optimization criterion (f_3) is the size of the classifier, i.e., the number of rules in a rule set or the number of nodes in a decision tree. The smaller a classifier is, the less likely it is overfitted.

2) *Voting*: If the three objective functions are applied in Multi-objective Evolutionary Algorithms [34, 35] together with a Pareto-ranking based fitness assignment process [19], the result will be a set of classifiers instead of a single solution. If evolving decision trees, for instance, the Pareto set will contain trees of several different sizes. A classifier C_1 with fewer nodes (better f_3 value) which performs slightly worse (in respect of f_2 or f_1) than one (C_2) with more nodes, is neither better nor worse. In terms of the Pareto relation, this means that C_1 is not dominated by C_2 and vice versa.

Instead of using only one of the evolved classifiers, we combine the complete Pareto set resulting from the optimization process to a single *super classifier*. We therefore introduce two alternative voting mechanisms **V1** and **V2**. If a data sample is to be classified, all the members of the Pareto set are applied to it and their results (the suggested classes) are noted. If **V1** is applied, each classifier can cast a number of votes inversely proportional to the costs it has caused (f_2); and with variant **V2**, the number of votes of a classifier is inversely proportional to a combination of f_1 and f_3 . The class which receives the most votes is then the result of the classification.

Compared to other related evolutionary data mining methods, our approaches here have one obvious advantage: they utilize the full potential of Multi-objective Evolutionary Algorithms by incorporating the whole Pareto set into one “super classifier”. With this, none of the evolved solutions is lost and higher robustness against overfitting can be achieved. Another advantage of this is that results from multiple runs of the optimizer may also be combined.

Boosting and ensemble learning methods [36, 37] are other approaches that also combine different learners or classifiers in order to increase precision. They follow a concept different from our simple voting mechanism by involving learning on different subsets of the training data set. In our case, all classifiers are trained with the same data and we do not explicitly aim to create different classifiers; they simply result from the nature of multi-objective optimization. Voting is only used to utilize information which the EAs extract from their input data.

IV. EXPERIMENTS AND RESULTS

A. Data sets

For the purposes of verifying the performances of our approaches, we applied them to five standard data sets from supervised data mining. Except for \mathbf{A}_3 which was already divided into two subsets by default (*Erratum*: and \mathbf{A}_4 , where we used 100 samples for testing), 70% of the samples of

the data sets \mathbf{A}_i were used as training samples (A) and the remaining 30% as test samples \bar{A} .

The first data set used \mathbf{A}_1 is the *Iris Dataset* [23, 24] already introduced in Section III-A. Besides this data set, others include the *Wine Dataset*, the *Heart Disease Dataset*, the *Wisconsin Breast Cancer Dataset* and the *Hepatitis Dataset*.

The *Wine Dataset* (\mathbf{A}_2) by [38] represents the results of a chemical analysis of different wines. The three examined wines ($|K| = 3$) all grow in the same region in Italy but stem from different cultures. For each of the 178 samples t , thirteen characteristic components $t_1 \dots t_{13}$ have been measured.

The *Heart Disease Dataset* [39] (\mathbf{A}_3) contains cardiological diagnostic samples from Single Proton Emission Computed Tomography (SPECT3) images. Each of the 267 samples (representing a patient) has 22 binary attributes and is classified into either of the two classes $k_1 = \text{normal}$ and $k_2 = \text{abnormal}$.

The *Wisconsin Breast Cancer Dataset* [39] (\mathbf{A}_4) consists of documented clinical samples which have 10 attributes, each stands for one measurement from a breast cancer examination with a scale from one to ten. The 699 samples are again to be classified into two classes.

\mathbf{A}_5 , the *Hepatitis Dataset* [40] consists of 19 attributes which indicate whether a person suffering from hepatitis will survive this infection or not ($|K| = 2$). There are categorical, integer-valued, and also real attributes. The data set has 155 samples.

B. Classifiers

For comparison, we also applied some well-established decision tree-based classification approaches to these data sets with the *Weka* framework [41, 42].

The Random-Forest algorithm by [11] trains multiple decision trees on sub-samples drawn with the *bagging* [43] approach from the input data. Like our approach, the trees are combined with a voting mechanism. In *Weka*, we used the default settings for both the tree size as well as the features in our experiments.

The second approach used for comparison is the popular C4.5 algorithm by [3] which creates decision trees where an arbitrary number of branches can follow each node. Its implementation in *Weka*, called *J48*, has separate tree branches for each value of the attribute used for decision at a given node. We used this approach with a confidence factor of 0.25 and a minimum number of samples per leaf of 2.

Finally, we applied the *Weka*-specific *RepTree* algorithm which uses information gain/variance reduction and prunes the trees with reduced-error pruning and backfitting. In the experiments, we used the default settings, i.e., a minimum variance proportion of 0.001 and a minimal sample weight of the leaf nodes of 2.0.

C. Settings

In the experiments with our evolutionary data mining approaches, we used EAs with a population size of 2000 individuals, Pareto ranking, tournament selection with three contestants, and the simple convergence prevention (SCP) algorithm from [19, 44]. We ran multiple series for 1000 generations each, with different mutation rates $mr \in \{0.6, 0.7\}$, crossover rates $cr \in \{0.3, 0.4\}$, both voting methods $v \in \{V_1, V_2\}$, and either with steady-state ($ss = 1$) or generational ($ss = 0$) population handling strategies. All randomized algorithms, i.e., the EAs and the Random-Forest method, were applied at least ten times to each data set.

D. Results

With our experiments, we aim to answer two questions:

- 1) Does the different parameter settings of EAs lead to significant differences in the classification accuracy of the evolved classifiers?
- 2) How is the performance of our approaches compared to the well-known approaches like C4.5 and Random-Forest?

We measured the median cA of the percentage of correctly classified samples from the training data A , the median $c\bar{A}$ of correctly classified samples on the test data \bar{A} , and the best result from all runs on the test data $b\bar{A}$.

In Table I, we present the results achieved with different configurations of the EAs in the tree-based ($ap = t$) and rule-based ($ap = r$) approaches ordered by $c\bar{A}$. From the table, it is clear that the best median precision on the test data is always reached by the rule-based approach. In all but one case, it is also the method with the best $b\bar{A}$ value. The combination of a mutation rate of 30% and a crossover rate of 70% seems to be more effective than a 40%/60% setting. Interestingly, the classifiers with the best precision are always found by the generational EAs ($ss = 0$) while the steady-state EAs ($ss = 1$) always achieve the best median result. Between the two voting mechanisms, no supremacy can be detected.

We further analyzed the influence of each single parameter with two-tailed Sign Tests [45] and the Wilcoxon's Signed Rank Test [46]. We found that no single parameter alone has a significant positive or negative influence when applying the tests with a significance level of 5%. In other words, our approaches are very robust and deliver results with similar qualities for a variety of settings.

When comparing the columns cA and $c\bar{A}$, we found that the classifiers are only moderately overfitted except in the *Heart Disease Dataset* where the evolved classifiers are almost 20% better on the training data than on the test samples.

In Table II, we show the performances of our approaches (i.e. Rule-EA and Tree-EA) in comparison with the other

Iris Dataset (A_1)								
Rank	ap	mr	cr	ss	v	cA	$c\bar{A}$	$b\bar{A}$
1	r	0.7	0.3	1	V2	97.98	<u>98.04</u>	98.04
2	t	0.7	0.3	1	V2	97.98	<u>98.04</u>	98.04
3	t	0.6	0.4	1	V2	97.98	<u>98.04</u>	98.04
4	t	0.7	0.3	0	V1	98.48	<u>98.04</u>	98.04
...
15	r	0.7	0.3	0	V1	100.00	94.12	<u>100.00</u>
...
Wine Dataset (A_2)								
Rank	ap	mr	cr	ss	v	cA	$c\bar{A}$	$b\bar{A}$
1	r	0.7	0.3	1	V1	100.00	<u>91.67</u>	96.67
2	r	0.6	0.4	0	V1	100.00	<u>91.67</u>	96.67
3	t	0.6	0.4	1	V2	96.61	90.83	93.33
4	t	0.7	0.3	1	V2	97.03	90.83	93.33
...
16	t	0.7	0.3	0	V1	97.46	86.67	<u>98.33</u>
Heart Disease Dataset (A_3)								
Rank	ap	mr	cr	ss	v	cA	$c\bar{A}$	$b\bar{A}$
1	r	0.7	0.3	1	V1	92.41	<u>73.93</u>	76.34
2	r	0.6	0.4	0	V2	91.14	73.12	<u>81.18</u>
3	t	0.7	0.3	1	V1	89.87	72.90	75.81
4	t	0.7	0.3	1	V2	86.71	72.04	77.41
...
Breast Cancer Dataset (A_4)								
Rank	ap	mr	cr	ss	v	cA	$c\bar{A}$	$b\bar{A}$
1	r	0.6	0.4	1	V1	99.42	<u>99.00</u>	99.00
2	t	0.7	0.3	0	V1	98.24	98.50	99.00
3	t	0.7	0.3	1	V2	97.99	98.00	99.00
4	t	0.6	0.4	0	V1	98.16	98.00	99.00
...
7	r	0.7	0.3	0	V2	99.25	98.00	<u>100.00</u>
...
Hepatitis Dataset (A_5)								
Rank	ap	mr	cr	ss	v	cA	$c\bar{A}$	$b\bar{A}$
1	r	0.7	0.3	1	V2	98.10	<u>86.28</u>	90.19
2	r	0.7	0.3	1	V1	99.10	84.62	88.26
3	t	0.7	0.3	0	V2	91.43	84.31	86.27
4	t	0.7	0.3	0	V1	96.67	84.31	88.23
...
7	r	0.6	0.4	0	V2	98.10	83.33	<u>92.16</u>
...

Table I: The best configurations of the classifier-evolving EAs.

		cA	$c\bar{A}$	$b\bar{A}$			cA	$c\bar{A}$	$b\bar{A}$
Rule-EA	A_1	97.89	98.04	<u>100.00</u>	Tree-EA	A_1	97.98	98.04	98.04
	A_2	100.00	91.67	96.67		A_2	96.61	90.83	<u>98.33</u>
	A_3	92.41	73.93	<u>81.18</u>		A_3	89.87	72.90	77.41
	A_4	99.42	99.00	<u>100.00</u>		A_4	98.24	98.50	99.00
	A_5	98.10	<u>86.28</u>	<u>92.16</u>		A_5	91.43	84.31	88.23
Rand.For.	A_1	100.00	<u>98.99</u>	<u>100.00</u>	J48	A_1	97.97	98.04	98.04
	A_2	100.00	93.33	93.33		A_2	100.00	93.33	93.33
	A_3	93.67	74.73	74.73		A_3	84.81	73.66	73.66
	A_4	99.67	99.00	99.00		A_4	95.15	<u>100.00</u>	<u>100.00</u>
	A_5	100.00	80.39	82.25		A_5	92.38	<u>86.28</u>	86.28
RepTree	A_1	95.96	96.08	96.08					
	A_2	94.92	90.00	90.00					
	A_3	81.01	75.80	75.80					
	A_4	94.82	<u>100.00</u>	<u>100.00</u>					
	A_5	80.95	76.47	76.47					

Table II: The approaches in comparison.

three decision tree algorithms. From the table, two things are obvious:

- 1) The evolutionary approaches have the highest best precision $b\bar{A}$ (in ten runs) in all five data sets.
- 2) In terms of median results, each of the evolutionary approaches outperforms or is at least as good as other approaches in at least one data set.

The rule-based classifiers, for instance, were better than Random-Forest on A_5 , J48 on A_3 , and RepTree on A_1 in terms of $c\bar{A}$. In contrast, they were not as good as Random-Forest on A_1 , J48 on A_2 , and RepTree on A_4 . The tree-based classifiers fared better against Random-Forest on A_5 but did not do well on A_1 . They scored even with J48 on A_1 but lost out on A_2 . On A_1 , they performed better than RepTree but not on A_3 .

These results clearly indicate that our evolutionary data mining approaches are indeed useful and may lead to very good results even with small numbers of runs. Although they are not superior than the other approaches, their performance is very competitive.

1) *Runtime*: The runtime requirements of our data mining approaches are higher than those used for comparison. 1000 generations of the rule-based classifier took between half an hour and 10 hours. The evolution of decision trees needed between 1.5 hours and 6 hours on an off-the-shelf computer. There are two reasons for these higher time requirements:

- 1) EAs are population-based, iterative optimization algorithms which evaluate all candidate solutions in the population in each generation. With our population size of 2000 individuals, this makes 2 000 000 evaluations in total.
- 2) Evaluating a classifier means applying it to each of the data samples in the training set. In the Random-Forest method, for instance, the individual classifiers are constructed using subsets of the training data only.

Nevertheless, EAs are very suitable for parallelization, which would reduce the runtime requirements significantly. Moreover, in most of the problems, exhausting the full 1000 generations was not necessary. Instead, the optimization process converged much faster to one optimum most of the time and thus could have been terminated earlier, as we will show in the following section.

2) *Convergence*: Figure 3¹ illustrates the convergence behavior of the evolutionary approaches for synthesizing classifiers in terms of the first objective function f_1 . These figures illustrate that most of the progresses in terms of classification accuracy on the training samples are always made in the first 200 generations. In the *Wine Dataset* A_2 , the evolution could have been stopped after this phase without any loss in precision. Generally, significant progress rarely occurs after 400 generations. Hence, the runtime

¹Erratum: In the original version of the paper, the figures for data set A_4 and data set A_5 were mixed up.

measured in the previous section is more than twice as high as what would actually be required. Another observation is that the evolution of rule-based classifiers converges faster than the synthesis of decision trees.

V. CONCLUSION

In this paper, we have presented two evolutionary approaches for building classifiers in supervised data mining tasks. We showed that both approaches perform very competitively compared to the standard decision tree algorithms on five general benchmark data sets. The performances are robust across a variety of configurations.

We have also shown that even simple classifier structures can be synthesized by EAs in a way which is no worse than the highly sophisticated algorithms such as Random-Forest – they obtained the highest precision in all experiments and outperformed each traditional approach in at least one area.

The problem of runtime has not been considered in this work. In our experiments, EAs took significantly longer to derive classifiers. The basic principles of evolutionary optimization require creating and evaluating many candidate solutions. Here, the evaluation, i.e., the application of the evolved classifiers to the training data sets, most significantly contributes to the runtime. Determining the fitness of the solutions using only a subset of training data may be one way to speed up the process. Another straightforward method for reducing the runtime may be to use efficient parallelization or distribution techniques [19, 47, 48, 49]. Future work will consider these.

REFERENCES

- [1] T. Weise and R. Chiong, “Evolutionary Data Mining Approaches for Rule-based and Tree-based Classifiers,” in *Proceedings of the 9th IEEE International Conference on Cognitive Informatics (ICCI'10)*, F. Sun, Y. Wang, J. Lu, B. Zhang, W. Kinsner, and L. A. Zadeh, Eds. IEEE Computer Society Press: Los Alamitos, CA, USA, 2010, pp. 696–703. [Online]. Available: <http://www.it-weise.de/documents/files/WC2010EDMAFRBATBC.pdf>
- [2] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, “Knowledge Discovery in Databases: An Overview,” *AI Magazine*, vol. 13, no. 3, pp. 213–228, 1992. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.1674>
- [3] J. R. Quinlan, *C4.5: Programs for Machine Learning*, ser. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993.
- [4] M. Mehta, R. Agrawal, and J. Rissanen, “SLIQ: A Fast Scalable Classifier for Data Mining,” in *Advances in Database Technology – 5th International Conference on Extending Database Technology (EDBT'96)*, ser. Lecture Notes in Computer Science

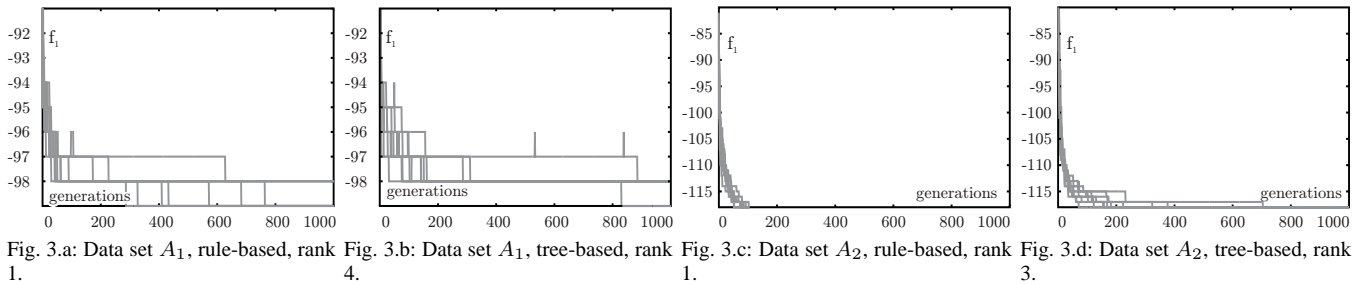


Fig. 3.a: Data set A_1 , rule-based, rank 1. Fig. 3.b: Data set A_1 , tree-based, rank 4. Fig. 3.c: Data set A_2 , rule-based, rank 1. Fig. 3.d: Data set A_2 , tree-based, rank 3.

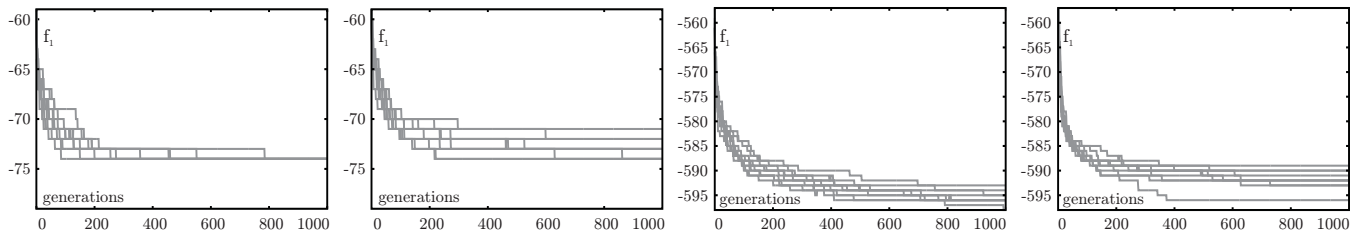


Fig. 3.e: Data set A_3 , rule-based, rank 1. Fig. 3.f: Data set A_3 , tree-based, rank 3. Fig. 3.g: Data set A_4 , rule-based, rank 1. Fig. 3.h: Data set A_4 , tree-based, rank 3.

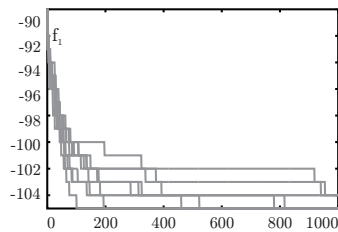


Fig. 3.i: Data set A_5 , rule-based, rank 1.

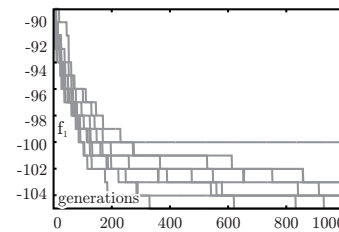


Fig. 3.j: Data set A_5 , tree-based, rank 3.

Figure 3: Convergence of the evolutionary data mining approaches.

- (LNCS), P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, Eds., vol. 1057. Springer-Verlag GmbH: Berlin, Germany, 1996, pp. 18–32. [Online]. Available: <http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0405/KDD/paper/MAR96.pdf>
- [5] J. C. Shafer, R. Agrawal, and M. Mehta, “SPRINT: A Scalable Parallel Classifier for Data Mining,” in *Proceedings of 22th International Conference on Very Large Data Bases (VLDB’96)*, T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, Eds. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1996, pp. 544–555.
- [6] J. W. Grzymala-Busse, “A New Version of the Rule Induction System LERS,” *Fundamenta Informaticae – Annales Societatis Mathematicae Polonae, Series IV*, vol. 31, no. 1, pp. 27–39, 1997.
- [7] J. Stefanowski and K. Slowinski, “Rough Sets as a Tool for Studying Attribute Dependencies in the Urinary Stones Treatment Data Set,” in *Rough Sets and Data Mining: Analysis of Imprecise Data*, T. Y. Lin and N. Cercone, Eds. Kluwer Academic Publishers: Norwell, MA, USA, 1996, pp. 177–196.
- [8] J. Gehrke, R. Ramakrishnan, and V. Ganti, “RainForest – A Framework for Fast Decision Tree Construction of Large Datasets,” in *Proceedings of 24rd International Conference on Very Large Data Bases (VLDB’98)*, A. Gupta, O. Shmueli, and J. Widom, Eds. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1998, pp. 416–427. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.474>
- [9] R. Rastogi and K. Shim, “PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning,” in *Proceedings of 24rd International Conference on Very Large Data Bases (VLDB’98)*, A. Gupta, O. Shmueli, and J. Widom, Eds. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1998, pp. 404–415. [Online]. Available: <http://eprints.kfupm.edu.sa/60034/>
- [10] W.-H. Au, K. C. C. Chan, and X. Yao, “A Novel Evolutionary Data Mining Algorithm with Applications to Churn Prediction,” *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, vol. 7, no. 6, pp. 532–545,

2003. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.8230>
- [11] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: <http://www.springerlink.com/content/u0p06167n6173512/fulltext.pdf>
- [12] S. F. Smith, "A Learning System based on Genetic Adaptive Algorithms," Ph.D. dissertation, University of Pittsburgh: Pittsburgh, PA, USA, 1980.
- [13] J. H. Holland, "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems," in *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. William Kaufmann: Los Altos, CA, USA, 1986, vol. II, pp. 593–623.
- [14] M. Fidelis, H. S. Lopes, and A. A. Freitas, "Discovering Comprehensible Classification Rules with a Genetic Algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'00)*, vol. 1. IEEE Computer Society: Piscataway, NJ, USA, 2000, pp. 805–810. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1828>
- [15] P. L. Hsu, R. Lai, and C. C. Chiu, "The Hybrid of Association Rule Algorithms and Genetic Algorithms for Tree Induction: An Example of Predicting the Student Course Performance," *Expert Systems with Applications – An International Journal*, vol. 25, no. 1, pp. 51–62, 2003.
- [16] J. J. Tapia, E. Morett, and E. E. Vallejo, "A Clustering Genetic Algorithm for Genomic Data Mining," in *Foundations of Computational Intelligence – Volume 4: Bio-Inspired Data Mining*, ser. Studies in Computational Intelligence, A. Abraham, A.-E. Hassanien, and A. Ponce de Leon F. de Carvalho, Eds. Springer-Verlag: Berlin/Heidelberg, 2009, vol. 204/2009, pp. 249–275.
- [17] R. Chiong, Ed., *Nature-Inspired Algorithms for Optimisation*, ser. Studies in Computational Intelligence. Springer-Verlag: Berlin/Heidelberg, 2009, vol. 193/2009.
- [18] R. Chiong, F. Neri, and R. I. McKay, "Nature that Breeds Solutions," in *Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery: Implications in Business, Science and Engineering*, R. Chiong, Ed. Information Science Reference: Hershey, PA, USA, 2009, ch. 1, pp. 1–24.
- [19] T. Weise, *Global Optimization Algorithms – Theory and Application*. it-weise.de (self-published): Germany, 2009. [Online]. Available: <http://www.it-weise.de/>
- [20] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Inc.: New York, NY, USA, 1996.
- [21] W. M. Spears and K. A. De Jong, "Using Genetic Algorithms for Supervised Concept Learning," in *Proceedings of the 2nd International IEEE Conference on Tools for Artificial Intelligence (TAI'90)*. IEEE Computer Society Press: Los Alamitos, CA, USA, 1990, pp. 335–341. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.147>
- [22] K. A. De Jong and W. M. Spears, "Learning Concept Classification Rules using Genetic Algorithms," in *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91-II)*, J. Mylopoulos and R. Reiter, Eds., vol. 2. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1991, pp. 651–656. [Online]. Available: <http://citeseer.ist.psu.edu/dejong91learning.html>
- [23] E. Anderson, "The Irises of the Gaspé Peninsula," *Bulletin of the American Iris Society*, vol. 59, pp. 2–5, 1935.
- [24] R. A. Sir Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936. [Online]. Available: <http://digital.library.adelaide.edu.au/coll/special/fisher/138.pdf>
- [25] H. Min, T. G. Smolinski, and G. M. Boratyn, "A Genetic Algorithm-based Data Mining Approach to Profiling the Adopters And Non-Adopters of E-Purchasing," in *Third International Conference on Information Reuse and Integration (IRI'01)*, W. W. Smari, Ed. International Society for Computers and Their Applications, Inc. (ISCA): Cary, NC, USA, 2001, pp. 1–6. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.4711>
- [26] A. Kamrani, W. Rong, and R. Gonzalez, "A Genetic Algorithm Methodology for Data Mining and Intelligent Knowledge Acquisition," *Computers & Industrial Engineering*, vol. 40, no. 4, pp. 361–377, 2001.
- [27] A. L. Corcoran and S. Sen, "Using Real-Valued Genetic Algorithms to Evolve Rule Sets for Classification," in *Proceedings of the First IEEE Conference on Evolutionary Computation (CEC'94), 1994 IEEE World Congress on Computation Intelligence (WCCI'94)*, Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel, and H. Kitano, Eds., vol. 1, IEEE Computer Society: Piscataway, NJ, USA. IEEE Computer Society: Piscataway, NJ, USA, 1994, pp. 120–124. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.1864>
- [28] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, ser. Bradford Books. MIT Press (Stanford University): Cambridge, MA, USA, 1992, 1992 first edition, 1993

second edition.

- [29] R. Forsyth, "BEAGLE – A Darwinian Approach to Pattern Recognition," *Kybernetes*, vol. 10, no. 3, pp. 159–166, 1981, received December 17, 1980. (copy from British Library May 1994). [Online]. Available: http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/kybernetes_forsyth.pdf
- [30] J. R. Koza, "Concept Formation and Decision Tree Induction using the Genetic Programming Paradigm," in *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature (PPSN I)*, ser. Lecture Notes in Computer Science (LNCS), H.-P. Schwefel and R. Männer, Eds., vol. 496/1991. Springer-Verlag GmbH: Berlin, Germany, 1990, pp. 124–128. [Online]. Available: <http://citeseer.ist.psu.edu/61578.html>
- [31] A. A. Freitas, "A Genetic Programming Framework for Two Data Mining Tasks: Classification and Generalized Rule Induction," in *Proceedings of the Second Annual Conference on Genetic Programming (GP'97)*, J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, H. Iba, and R. L. Riolo, Eds. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1997, pp. 96–101. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.2449>
- [32] E. Cantú-Paz and C. Kamath, "Using Evolutionary Algorithms to Induce Oblique Decision Trees," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'00)*, L. D. Whitley, D. E. Goldberg, E. Cantú-Paz, L. Spector, I. C. Parmee, and H.-G. Beyer, Eds. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2000, pp. 1053–1060. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.7300>
- [33] D. P. Muni, N. R. Pal, and J. Das, "A Novel Approach to Design Classifiers using Genetic Programming," *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, vol. 8, no. 2, pp. 183–196, 2004.
- [34] D. A. van Veldhuizen and L. D. Merkle, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," Ph.D. dissertation, Air University, Air Force Institute of Technology: Wright-Patterson Air Force Base, OH, USA, 1999. [Online]. Available: <http://citeseer.ist.psu.edu/old/vanveldhuizen99multiobjective.html>
- [35] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.5848>
- [36] R. Avnimelech and N. Intrator, "Boosting Regression Estimators," *Neural Computation*, vol. 11, no. 2, pp. 499–520, 1999. [Online]. Available: <http://citeseer.ist.psu.edu/avnimelech99boosting.html>
- [37] R. E. Schapire, "The Strength of Weak Learnability," *Machine Learning*, vol. 5, pp. 197–227, 1990. [Online]. Available: <http://citeseer.ist.psu.edu/schapire90strength.html>
- [38] M. Forina, S. Lanteri, C. Armanino, and et al., "PARVUS – An Extendible Package for Data Exploration, Classification and Correlation," Institute of Pharmaceutical and Food Analysis and Technologies: Genoa, Italy, Tech. Rep., 1988.
- [39] W. H. Wolberg and O. Mangasarian, *Breast Cancer Wisconsin (Original) Data Set*. UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems, Donald Bren School of Information and Computer Science, University of California: Irvine, CA, USA, 1989. [Online]. Available: [http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))
- [40] G. Gong and B. Cestnik, *Hepatitis Data Set*. UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems, Donald Bren School of Information and Computer Science, University of California: Irvine, CA, USA, 1988. [Online]. Available: [http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))
- [41] G. Holmes, A. Donkin, and I. H. Witten, "WEKA: A Machine Learning Workbench," in *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems (ANZIIS'98)*. IEEE Computer Society Press: Los Alamitos, CA, USA, 1994, pp. 357–361. [Online]. Available: <http://www.cs.waikato.ac.nz/~ml/publications/1994/Holmes-ANZIIS-WEKA.pdf>
- [42] E. Frank, M. A. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg, "WEKA – A Machine Learning Workbench for Data Mining," in *The Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer Science+Business Media, Inc.: New York, NY, USA, 2005, ch. 62, pp. 1305–1314. [Online]. Available: http://www.cs.waikato.ac.nz/~ml/publications/2005/weka_dmh.pdf
- [43] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.9399>
- [44] T. Weise, "Evolving Distributed Algorithms with Genetic Programming," Ph.D. dissertation, University of Kassel, Fachbereich 16: Elektrotechnik/Informatik, Distributed Systems Group: Kassel, Germany, 2009. [Online]. Available: <http://www.it-weise.de/documents/files/W2009DISS.pdf>
- [45] S. Siegel and N. J. Castellan Jr., *Nonparametric Statistics for The Behavioral Sciences*, ser. Humanities/Social Sciences/Languages. McGraw-Hill: New York, NY, USA, 1956.

- [46] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945. [Online]. Available: <http://sci2s.ugr.es/keel/pdf/algorithm/articulo/wilcoxon1945.pdf>
- [47] W. N. Martin, J. Lienig, and J. P. Cohoon, "Island (Migration) Models: Evolutionary Algorithms Based on Punctuated Equilibria," in *Handbook of Evolutionary Computation*, ser. Computational Intelligence Library, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. Oxford University Press, Inc.: New York, NY, USA, Institute of Physics Publishing Ltd. (IOP): Dirac House, Temple Back, Bristol, UK, and CRC Press, Inc.: Boca Raton, FL, USA, 1997, ch. C6.3, pp. 448–463. [Online]. Available: http://www.cs.virginia.edu/papers/Island_Migration.pdf
- [48] E. Alba Torres and M. Tomassini, "Parallelism and Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, vol. 6, no. 5, pp. 443–462, 2002.
- [49] T. Weise and K. Geihs, "DGPF – An Adaptable Framework for Distributed Multi-Objective Search Algorithms Applied to the Genetic Programming of Sensor Networks," in *Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Applications (BIOMA'06)*, ser. Informacijska Družba (Information Society), B. Filipič and J. Šilc, Eds. Jožef Stefan Institute: Ljubljana, Slovenia, 2006, pp. 157–166. [Online]. Available: <http://www.it-weise.de/documents/files/W2006DGPFc.pdf>

Preview

This document is a preview version and not necessarily identical with the original.

<http://www.it-weise.de/>

```
@inproceedings{WC2010EDMAFRBATBC,
  title      = {Evolutionary Data Mining Approaches for Rule-based and
                Tree-based Classifiers},
  author     = {Thomas Weise and Raymond Chiong},
  booktitle  = {Special Session on Evolutionary Computing of the 9th IEEE
                International Conference on Cognitive Informatics (ICCI 2010)},
  month     = {jul # {~7--9}},
  year      = {2010},
  location  = {Tsinghua University, Beijing, China},
  publisher  = {IEEE Computer Society Press: Los Alamitos, CA, USA},
  pages     = {696--703},
  isbn      = {978-1-4244-8040-1},
  keywords  = {Data Mining, Evolutionary Algorithms, Genetic Programming,
                EC, GP, Learning Classifier Systems, LCS, Rule-based Classifiers,
                Tree-based Classifiers, Random Forest, Iris, Wine, Heart,
                Hepatitis, Breast Cancer, Decision Trees},
}
```