Expert Systems with Applications 40 (2013) 6019–6032

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Improving the performance of the BioHEL learning classifier system ☆

Xiao-Lei Xia [a,*], Huanlai Xing [b]

[a] School of Mechanical and Electrical Engineering, Jiaxing University, Jiaxing 314001, PR China
[b] School of Computer Science and IT, University of Nottingham, Nottingham NG8 1BB, UK

ARTICLE INFO

Keywords:
Learning Classifier Systems
Bioinformatics-oriented Hierarchical Evolutionary Learning
Estimation of Distribution Algorithms

ABSTRACT

The identification of significant attributes is of major importance to the performance of a variety of Learning Classifier Systems including the newly-emerged Bioinformatics-oriented Hierarchical Evolutionary Learning (BioHEL) algorithm. However, the BioHEL fails to deliver on a set of synthetic datasets which are the checkerboard data mixed with Gaussian noises due to the fact the significant attributes were not successfully recognised. To address this issue, a univariate Estimation of Distribution Algorithm (EDA) technique is introduced to BioHEL which primarily builds a probabilistic model upon the outcome of the generalization and specialization operations. The probabilistic model which estimates the significance of each attribute provides guidance for the exploration of the problem space. Experiment evaluations showed that the proposed BioHEL systems achieved comparable performance to the conventional one on a number of real-world small-scale datasets. Research efforts were also made on finding the optimal parameter for the traditional and proposed BioHEL systems.

1. Introduction

Recent decades have seen the successful application of Learning Classifier Systems (LCS) (Smith, 1980; Wilson, 1995) and Genetic-Based Machine Learning to a number of real-life problems. The BioHEL (Bacardit, Burke, & Krasnogor, 2009), an acronym for the Bioinformatics-oriented Hierarchical Evolutionary Learning, has been a very recent addition to the family of LCS algorithms. Its knowledge representation, which is preceded by an implicit feature selection procedure, is composed of only relevant attributes. It have been experimentally proven BioHEL's abilities to deliver maximally accurate, maximally general and compact rule sets for complex problems and its comparatively superior scalability to datasets of high dimensions.

In our recent studies, the BioHEL system was applied to 18 variants of checkerboard datasets in which a growing number of Gaussian noises were added. The BioHEL, with the default parameter setting had difficulties in maintaining satisfactory performance on datasets as the dimensions of noises grew to 7 and beyond. It was also discovered that multiple parameters for the knowledge representation, particularly the number of relevant attributes, have significant impact on the performance of the BioHEL.

An investigation was carried out to pinpoint the cause of the BioHEL's inability to cope with datasets contaminated by high dimensions of noises. It was observed that the significant attributes failed to prevail in terms of their occurrences in the population before the application of selection pressure. It was thus reduced the chances that the optimal attribute combination(s) and their respective optimal value range be identified.

It was conceived to introduce a mechanism to identify significant attributes for the BioHEL. Interestingly, other than the standard genetic algorithm operations which are selection, cross-over and mutation, the BioHEL also features two "special" operators: generalization and specialization. For each rule in the population, at each iteration by probability, generalization removes an attribute and specialization adds an extra attribute. These two types of operations, in fact, are explicit feature selection procedures. Nevertheless, the attribute is selected randomly, which, essentially, treats each attribute as equally significant. Consequently, this implicit feature selection procedure fails to benefit the BioHEL on the noisy variants of checkerboard datasets.

On the other hand, the fitness change resulting from the special operators is deemed informative about the significance of attributes. Intuitively, the loss of significant attributes, in general, worsens the fitness of associated rules across the population while the addition of significant attributes improves their fitness. Thus, this paper proposed to estimate the significance of attributes by constructing a probabilistic model based on the fitness change brought by the generalization or the specialization operator. A probability vector is built with each element gives the significance of each attribute. The vector is updated iteratively to reflect the

☆ This document is a collaborative effort.

* Corresponding author. Tel.: +86 573 83647354.
E-mail addresses: xxia01@qub.ac.uk (X.-L. Xia), psxhx@nottingham.ac.uk (H. Xing).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.05.025

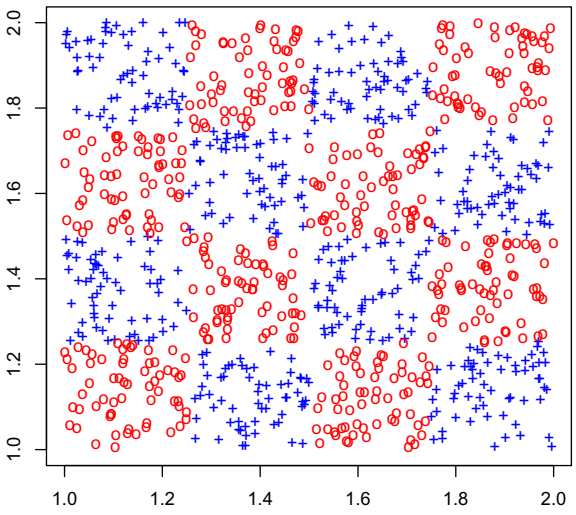

Fig. 1. The checkerboard datasets.

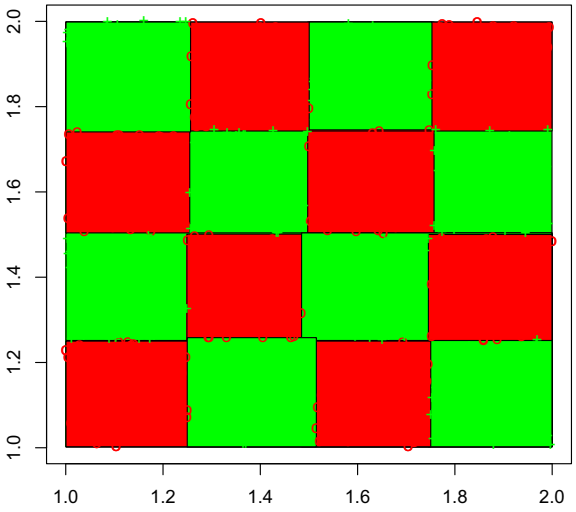

Fig. 2. The performance of the traditional BioHEL on the original checkerboard datasets.

most recent fitness change in associated rules. Meanwhile, the probability vector also provides guidelines for the special operations at the iteration that followed. Attributes whose addition or removal contributes to the improvement of fitness are given preference to more occurrence in candidate solutions, while the attributes that causes the fitness degradation are to occur less. This strategy can be categorized into the family of univariate Estimation of Distribution Algorithms (EDA) (Larranaga & Lozano, 2002) which estimate a model of the problem structure, assuming that attributes are independent of each other.

Experiments were performed on various datasets to evaluate the performance of the BioHEL integrated with the novel EDA technique. It was shown that with appropriate configurations of the parameters, the performance of BioHEL using the EDA algorithm remained stably satisfactory over the noisy variants of checkerboard datasets. Its performance were proven competitive on a variety of real-life problems datasets, both small and large.

Research efforts have also made on analyzing influence of multiple parameters on the generalization performance of different BioHEL systems. The optimal parameter settings were identified

Table 1
Parameter configuration of BioHEL for the 2-dimensional checkerboard dataset.

crossover operator	1px
default class	disabled
fitness function	MDL
initialization min classifiers	20
initialization max classifiers	20
iterations	100
MDL initial theory learning ratio	0.01
MDL iteration	10
MDL weight relax factor	0.90
population size	500
probability of crossover	0.6
probability of individual mutation	0.6
probability of one	0.75
selection algorithm	tournamentwor
tournament size	4
windowing ILAS	TRUE
dump evolution stats	
smart initialization	
class wise initialization	
coverage breakpoint	0.0625
repetitions of rule learning	TRUE
coverage ratio	0.90
knowledge representation hyperrectangle	
num expressed attributes init	15
hyperrectangle uses list of attributes	
probability generalize list	0.1
probability specialize list	0.1

Table 2
10-fold cross validation accuracies of LIBSVM, NaiveBayes (NB) and C4.5 on the checkerboard datasets.

	LIBSVM					
#Attr	C	λ	accuracy (%)	NB (%)	C4. 5(%)	BioHEL (%)
2	1024	4	97.98	47.98	49.90	97.73
3	256	4	94.66	52.22	49.90	97.34
4	512	2	88.61	50.30	49.90	97.87
5	64	8	80.75	49.40	49.90	96.39
6	2	8	73.69	48.39	49.90	96.06
7	8	8	69.35	48.59	49.90	94.84
8	2	8	64.92	49.09	49.90	90.12
9	1	8	61.29	51.51	49.90	85.25
10	1	8	59.27	50.60	49.90	79.14
11	8	8	58.47	51.21	49.90	70.72
12	8	8	56.55	50.60	49.90	63.79
13	1	0.125	54.94	49.90	49.90	61.82
14	0.25	0.125	55.65	51.41	49.90	61.83
15	0.5	0.125	55.04	51.41	49.90	58.76
16	1	0.0625	54.64	51.21	49.80	57.34
17	0.5	0.125	54.94	51.21	49.80	56.16
18	1	0.0625	54.74	50.91	49.80	54.95
19	4	0.03125	54.64	50.10	49.80	53.61
20	4	0.5	54.44	50.40	49.80	52.60

for the conventional and the proposed BioHEL systems in order to achieve the best performance.

The paper is organized as follow. Section 2 reviews the related work, followed by a description of the BioHEL system in Section 3. Section 4 illustrates the performance of the standard BioHEL on the 2-dimensional checkerboard benchmark and the difficulties BioHEL encounters on variants of checkerboard datasets with a large amount of noisy disturbances. An investigation was carried out to pinpoint the root of the problem. Section 5 proposed the algorithms for constructing a probabilistic model which identifies significant attributes in order to make the generalization and the specialization operations better informed. It is followed by an analysis of parameter settings for different BioHEL systems. The conventional BioHEL and the proposed ones, each with their respective optimal parameter settings, are compared in terms of

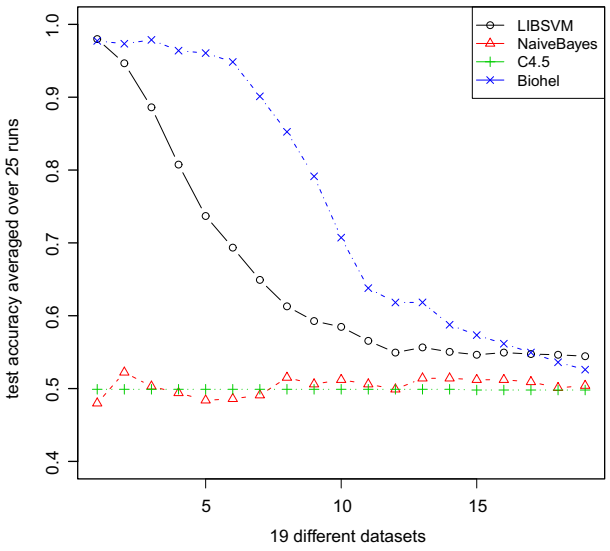

Fig. 3. The performance of BioHEL degrades to the growth in the number of noise attributes.

their generalization performances on the variants of the checkerboard. Section 6 reports the experimental results of the new BioHEL systems embedded the EDA algorithm on a number of benchmark datasets and its performance was compared to that of the conventional BioHEL. Conclusions are given in Section 7 with an outlook on future work.

2. Related work

In the community of evolutionary learning, research efforts has continued unabated to unveil the structural information of a problem efficiently, which has prompted the emergence of various techniques. Among these techniques are the Estimation of Distribution Algorithms (EDA) (Larranaga & Lozano, 2002) which are a class of evolutionary algorithms, identifying interactions between attributes. EDA techniques, in general, build a probabilistic model of candidate solutions from which offspring solutions are generated. For example, probabilistic models which contain the global structural information of the problem are built, respectively in the Compact Genetic Algorithm (cGA) (Harik, Lobo, & Goldberg, 2002), the Extended Compact Genetic Algorithm (ECGA) (Butz,

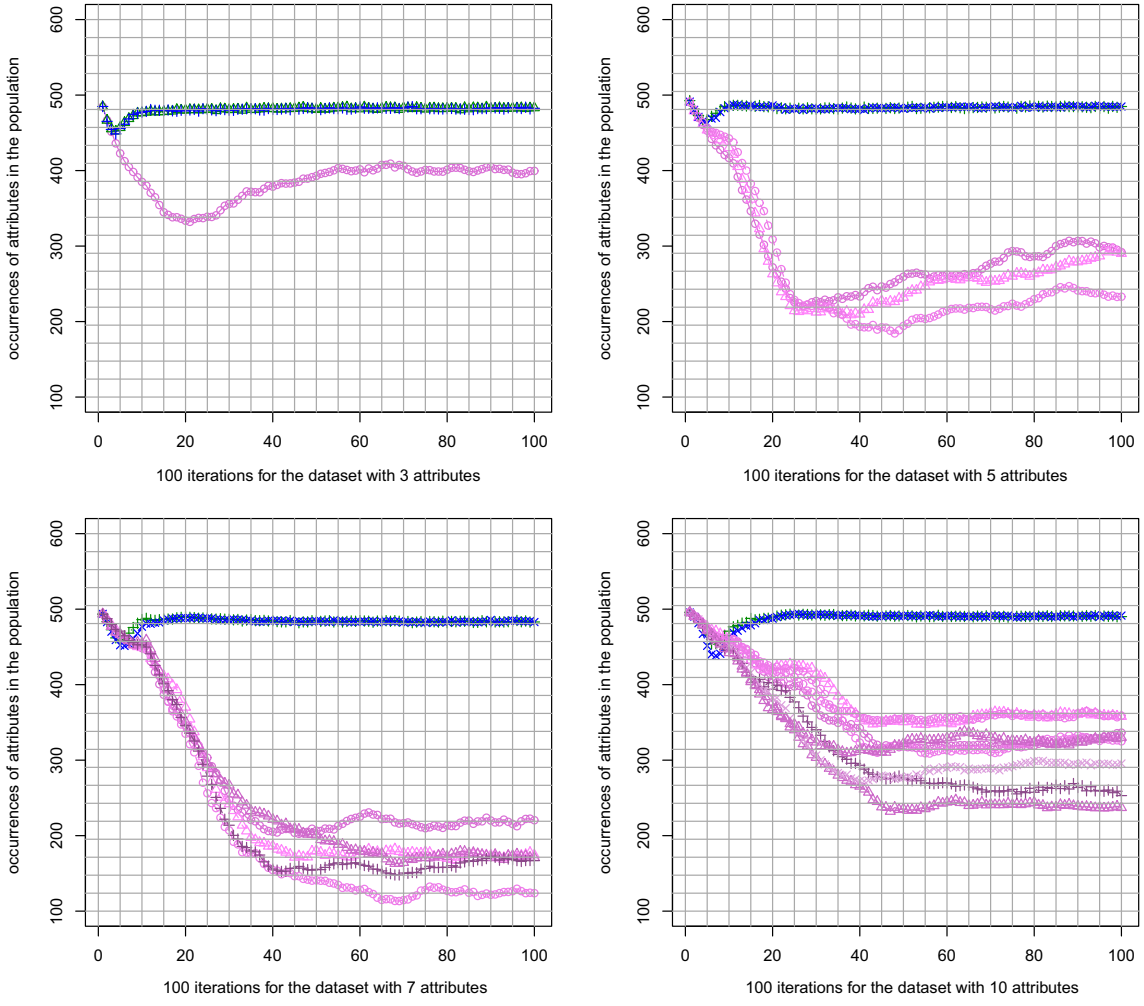

Fig. 4. The number of occurrences of attributes for datasets with respectively 3, 5, 7 and 10 attributes.

2006; Harik, Lobo, & Sastry, 2006) and the Bayesian Optimization Algorithm (BOA) (Pelikan, 2005).

The Learning Classifier Systems (LCS) has benefited greatly from these techniques which has made exploration of the problem space well-informed. The compact classifier system (CCS) (Llorà, Sastry, & Goldberg, 2005) is a Pittsburgh LCS incorporated with Compact Genetic Algorithm (cGA) (Harik et al., 2002) which is a typical EDA method. cGA is run repeatedly in order to produce and evolve accurate and maximally general rules. Different rules are sampled from different probability distribution vectors which are different perturbations of the initial probability vector of cGA. Probability vectors are generated and removed adaptively to ensure a compact rule set containing accurate and maximally general rules. However, the number of function evaluations required grows exponentially to the size of multiplexer problem. The poor scalability of the CCS prompted the development of χ-ary extended compact classifier system (χeCCS) (Llorà, Sastry, Goldberg, & de la Ossa, 2006) in which: (1) the probabilistic model is built using the aforementioned ECGA which search for least complex and most accurate model; (2) efficient rule niching is realized using restricted tournament replacement scheme which makes possible maintenance and evolution of multiple rules in a single run. Experimental evaluations showed that the χeCCs scales quadratically to the problem size for the multiplex applications.

The eXtended classifier system (XCS) which is a well-known LCS, has also used strategy of constructing probabilistic models (Wilson, 1995). The model, in effect, presents the XCS with well-informed recombination operators without require explicit global knowledge of the specific problem at hand. As a result,, the need for the conventional crossover operation is eliminated. More lately, a novel model was proposed which contains the structural information contained in the population of XCS system (Park & Oh, 2009). Using this model, different weights were assigned to the action that different inputs recommended for the prediction of a testing instance. The model is suggested to be capable of improving the performance of the traditional XCS by 10% for large-scale speaker identification problems.

Another important research avenue regarding the performance enhancement of LCSs is the application of Memetic Algorithms (Krasnogor & Smith, 2005), inspired by models of adaption in natural systems which hybridizes evolutionary adaptation of a population with individual learning procedures for local refinement. Recent research efforts include the LCS proposed in Wyatt and Bull (2005) where the memetic learning techniques were introduced to LCS to address continuous-valued problems. It is demonstrated that memetic technique can make the learning of LCS more accurate and more robust. Also in this category is the XCS which was incorporated with the gradient descent methods for multi-step problems (Butz, Goldberg, & Lanzi, 2005). Its superi-

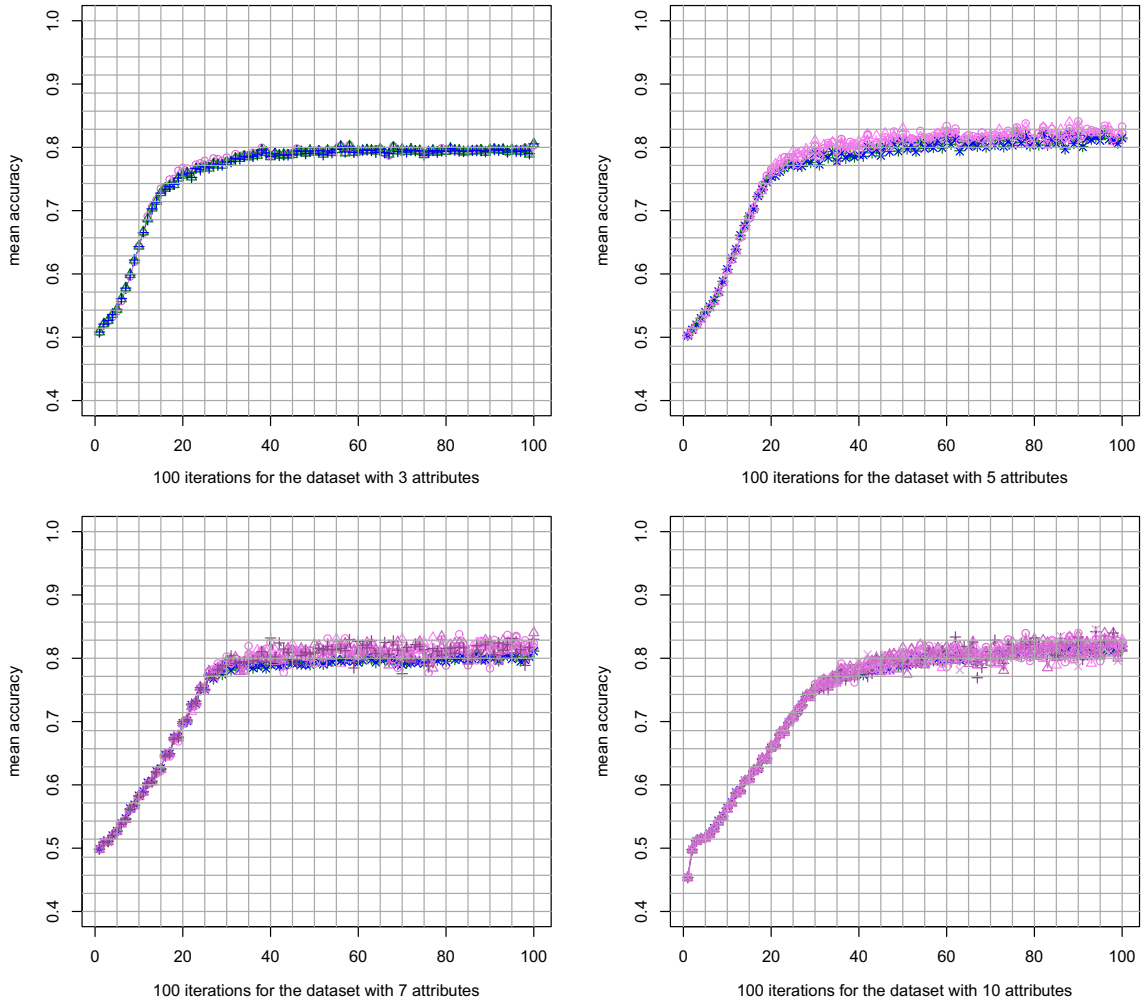

Fig. 5. The mean accuracy for datasets with respectively 3, 5, 7 and 10 attributes.

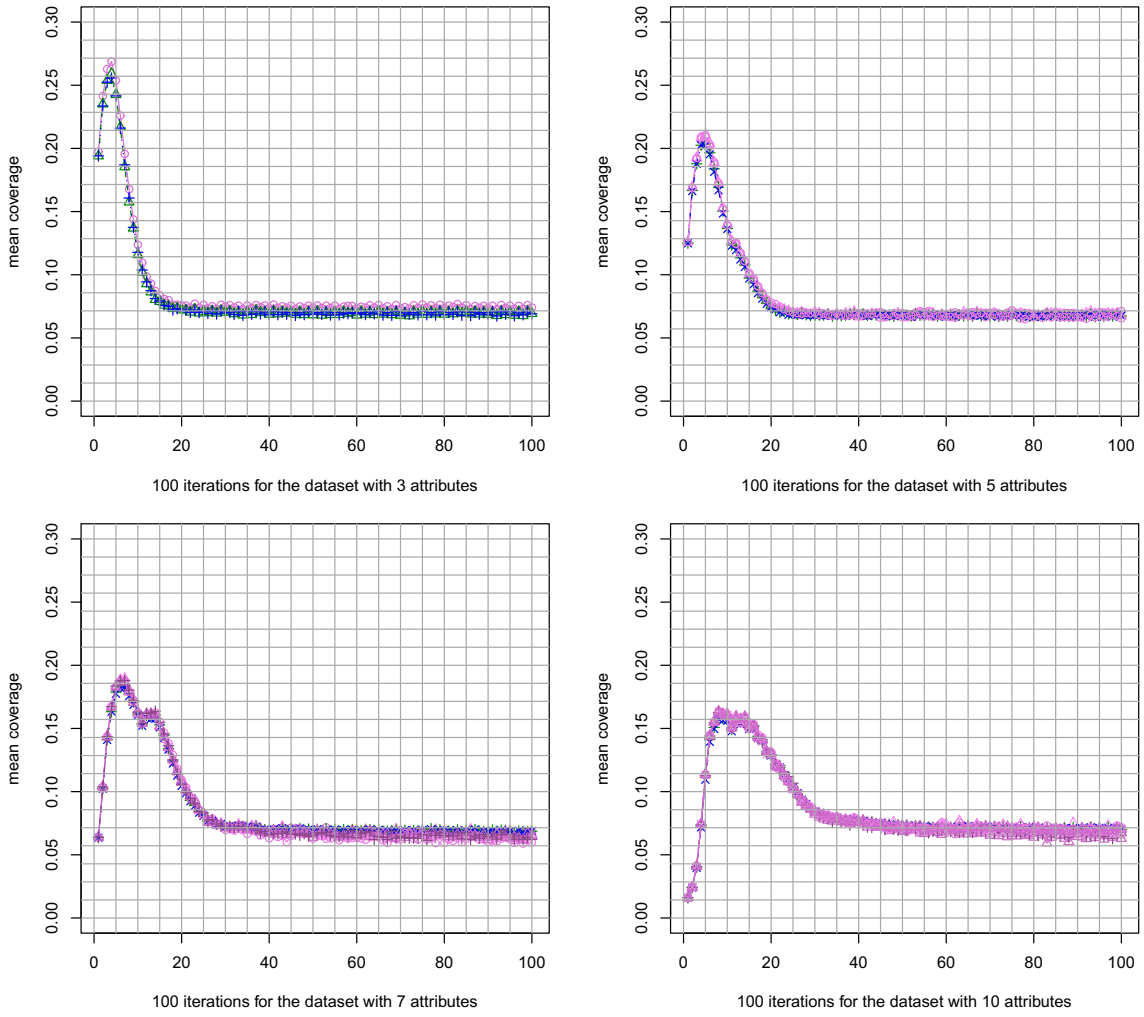

Fig. 6. The mean coverage for datasets with respectively 3, 5, 7 and 10 attributes.

ority to the standard XCS is experimentally confirmed. Within the domain of multistep applications, the SAMUEL system (Grefenstette, 1991) is another LCS in which various learning operators, inspired by memetic learning mechanisms, are defined.

More recently, a smart crossover operator was proposed within the framework of Pittsburgh LCS (Bacardit & Krasnogor, 2006). The operator recombines rules from multiple parents and heuristically identify rule set which is maximally accurate and compact. In the extension of this LCS (Bacardit & Krasnogor, 2009), multiple new operators are defined to reinforce the specificity/generalization pressure for individual rules. Experiments showed that the Pittsburgh LCS reinforced a proper combination of these operators can achieve comparable performance to that XCS/BOA and χeCCS.

3. Bioinformatics-oriented Hierarchical Evolutionary Learning System

The Bioinformatics-oriented Hierarchical Evolutionary Learning System (BioHEL) is a Pittsburgh LCS which employes Iterative Rule Learning (IRL) paradigm (Venturini, 1993). The system employs genetic algorithm to evolve a population of candidate rules iteratively. At each iteration, a rule is learnt and appended to the rule set which is initially empty. Meanwhile, the instances that are covered by the rule are removed from the training set. The resultant rule set, with the rule elements in chronological order, forms the solution to the problem.

Each rule in BioHEL takes the general form of IF < condition > THEN < class > . For continuous attributes, the rule specifies the eligible value range by specifying a lower upper and an upper bound. For discrete attributes, the rule uses binary representation where each optional value is assigned a bit. The disjunctive of the bits which is assigned a value of "1" is the set of all the optional values for a single attribute. Different attributes are combined by logical conjunction into the condition part. Each rule is identified by the following parameters: (1) an integer specifying the number of attributes selected into the rule; (2) a vector containing the indexes of the attributes; (3) a vector specifying the feasible value range for each attribute; (4) the class label. For large-scale datasets, only a subset of attributes are selected randomly into each rule during initialization, which is, in effect, an explicit feature selection process for each rule.

Exploration operators defined for this rule representation include the traditional crossover operator and the mutation operator. The crossover operator selects a random cut point within each parent rule and recombines into two offsprings. The mutation operator selects an attribute of a rule randomly and alters its value interval or its value options. It is worth attention that, in a single rule, the mutation operator "virtually" removes an attribute when the interval for an attribute is expanded to its feasible or all the op-

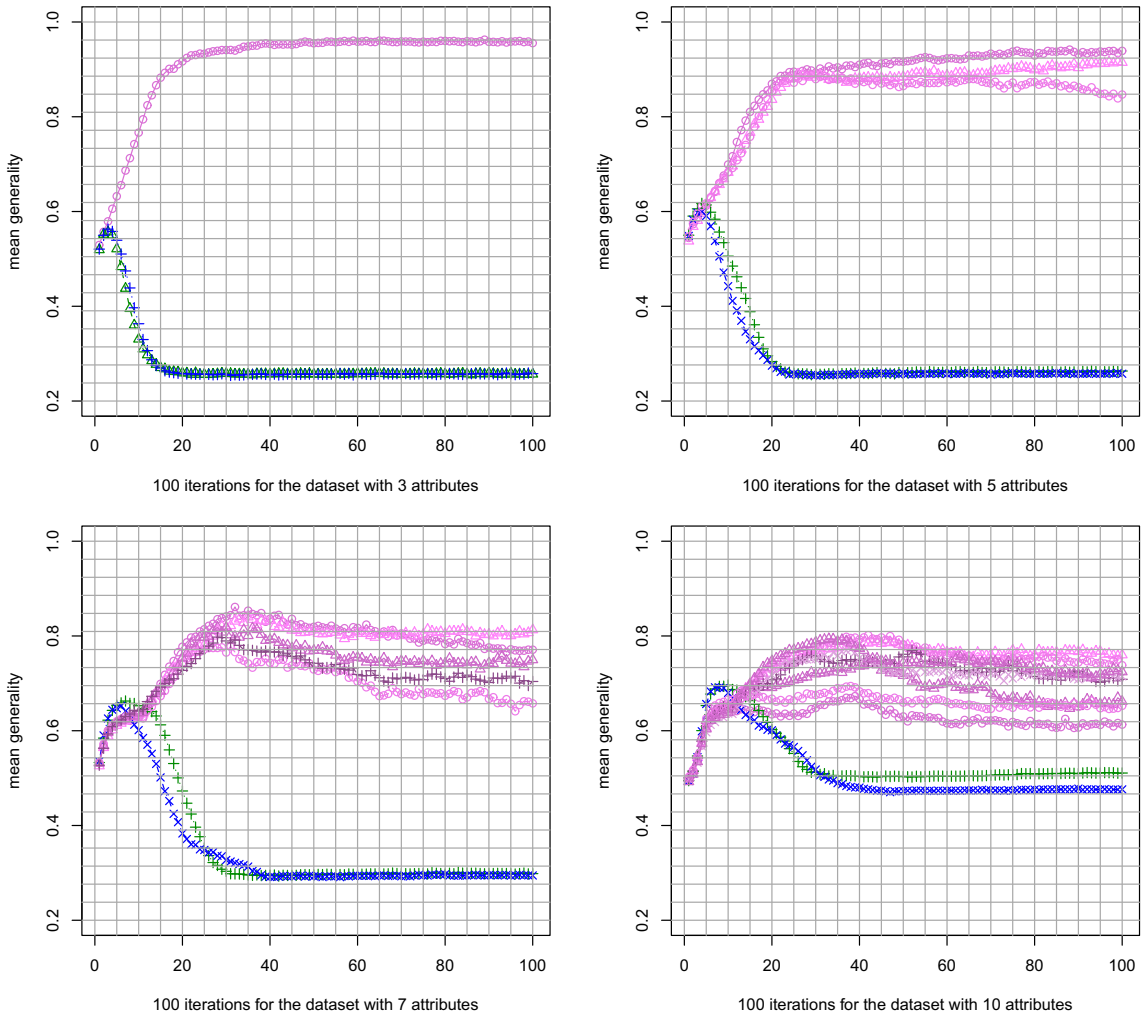

Fig. 7. The mean generality for datasets with respectively 3, 5, 7 and 10 attributes.

tional values for the attribute are allowed. Two special operators are also defined which are respectively generalization operator and specialization operator. The generalization operator removes an attribute from a rule and specialization operator adds an extra attribute to a rule. These two operators, in actual fact, perform feature selection on a rule and ensure that attribute combinations as many as possible are evaluated.

The fitness function of the BioHEL follows the principle of Minimum Description Length (MDL) (Rissanen, 1978). The MDL principle seeks the optimal trade-off between the complexity and the accuracy of a rule. BioHEL searches for the rule with the minimal fitness value. The fitness function is formulated as:

$$\text{Fitness} = \text{TL} \cdot W + \text{EL} \tag{1}$$

where TL, the acronym for theory length, is indicative of the complexity of a rule while EL, the acronym for exception length, is indictive of the accuracy of the rule. W is the parameter which adjusts the tradeoff between the two terms.

For a rule denoted as R, its TL is calculated as:

$$\text{TL}(R) = \frac{\sum_{i=1}^{\#Attr} (1 - \text{Size}(R_i))/\text{Size}(D_i)}{\#Attr} \tag{2}$$

where #Attr is the number of the relevant attributes, Size (R_i) is the width of the interval for ith attribute in the rule R and Size (D_i) is the width of the feasible range for ith attribute.

EL, on the other hand, is formulated as:

$$\text{EL}(R) = 2 - \text{ACC}(R) - \text{COV}(R) \tag{3}$$

where ACC (R) represents the accuracy of the rule R, defined as the number of examples correctly classified among all the examples that are covered by the rule. And COV (R) stands for the coverage of the rule R defined as the number of examples matched by the rule divided by the current number of total training examples. The tradeoff between ACC (R) and COV (R) is controlled by a parameter called the Coverage Ratio (CR). Meanwhile, the coverage term is a function of the number of examples that are matched by the rule R, matched (R). COV (R) grows linearly to matched (R) until it reaches a threshold which is parameterized by a threshold termed as the Coverage Breakpoint (CB). After reaching the CB, COV (R) grows at a slower rate.

Techniques which are introduced to improve the performance of the BioHEL system also include: (1) incremental learning with alternating strata (ILAS) windowing scheme in which a portion of, rather than all, training examples are used to evaluate the fitness of each rule. (2) ensemble learning scheme in which the BioHEL run multiple times with different random seeds. Each run cast a vote on the membership of an example which is eventually

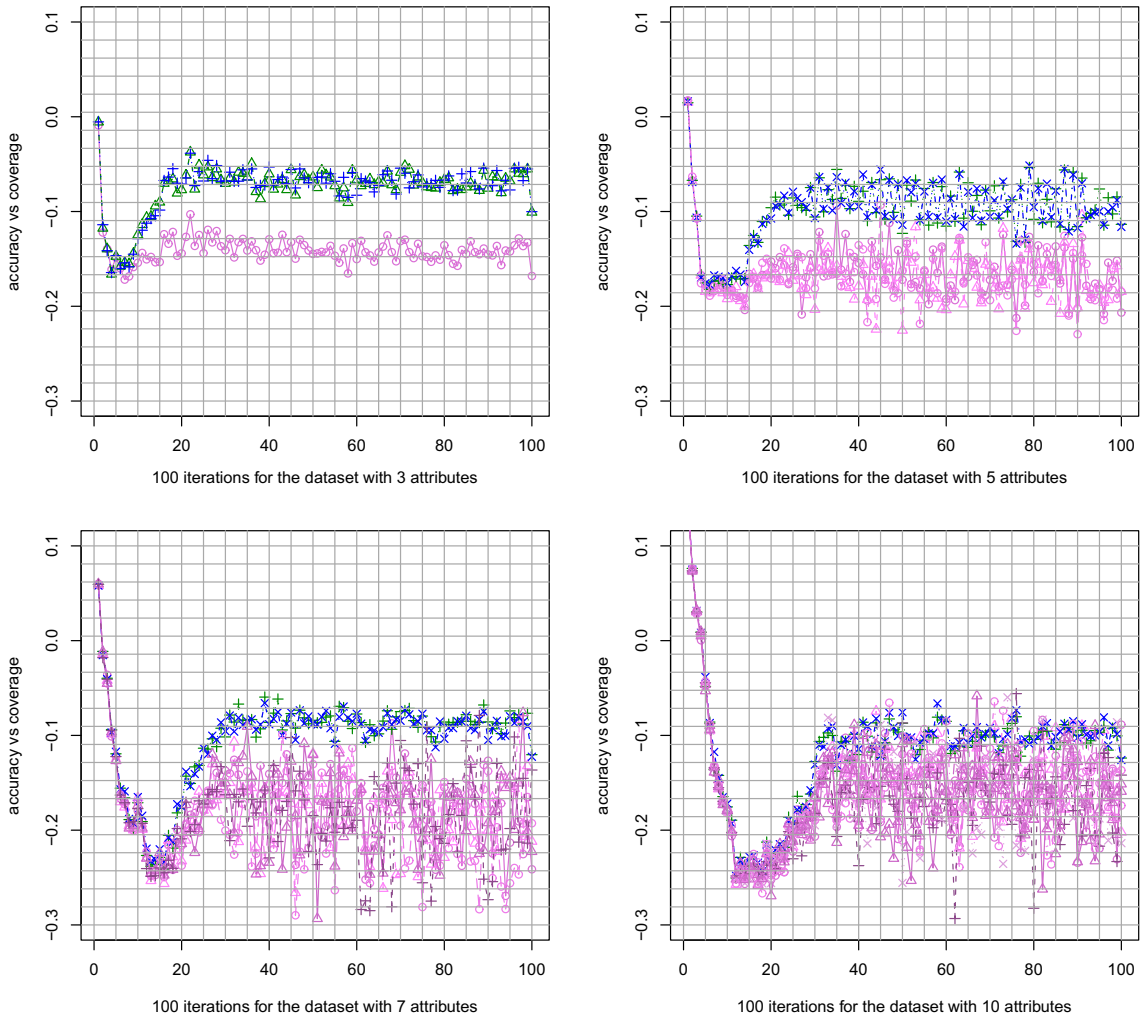

Fig. 8. The pearson correlation coeffiecent between accuracy and coverage for each attribute of datasets with respectively 3, 5, 7 and 10 attributes.

assigned the label of the class winning the majority of votes. (3) default class policy which allows examples to be grouped into a certain class in order to make the ultimate rule set more compact.

4. Experiments on checkerboard datasets

4.1. Performance of various algorithms on the checkerboard datasets

The checkerboard dataset contains 992 data points from two classes on X–Y plane, with 496 for each class, as shown by Fig. 1. Fig. 2 displays the checkerboard pattern successfully recognized by the BioHEL using the parameter setting in Table 1. The BioHEL achieved a 10-fold cross-validation accuracy of 97.17%.

In order to examine BioHEL's ability of identifying signification attributes, the checkerboard datasets were added, incrementally, 1 to 18 dimensions of Gaussian noises, resulting in 18 synthetic datasets. Each noisy attribute, as well as the original two, has roughly the same mean of 1.50. The standard deviations of the noises are also at the same level of around 0.14, while they are around 0.28 for the original two attributes.

Ten-fold cross validation accuracies of LIBSVM (Chang & Lin, 2001), NaiveBayes (NB) and C4.5 on the checkerboard datasets were reported in Table 2 and depicted by Fig. 3. For LIBSVM, the following Gaussian Radial Basis Function (RBF) was used:

$$K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\lambda \|\mathbf{X}_i - \mathbf{X}_j\|^2} \tag{4}$$

From Fig. 3, it can be seen that both NaiveBayes and C4.5 were unable to perform well while the performance of both the BioHEL and LIBSVM degraded as the number of noises grows.

4.2. Investigation into BioHEL's performance degradation

In order to investigate the cause of BioHEL's performance degradation, different variables in the system and their relations were analyzed in detail. First analyzed was the average change of different variables through the 100 iterations to generate the first rule. Figs. 4–7 depicts respectively the number of occurrences, the mean accuracy, the mean coverage and the mean generality for each attribute across rules, on datasets with 3, 5, 7 and 10 attributes. For each graph, variables depicted are, in a direction of clockwise, the number of occurrences, accuracies, generality and coverage of different attributes across the entire population. The value of these variables averaged over 25 runs on different random seeds. The blue and the green curves represented the two significant attributes while the rest in different shades of cyan are all noises.

It can be seen that the number of occurrence and the coverage of different attributes both can help distinguish the significant attributes from the noises on datasets of 3 attributes. However,

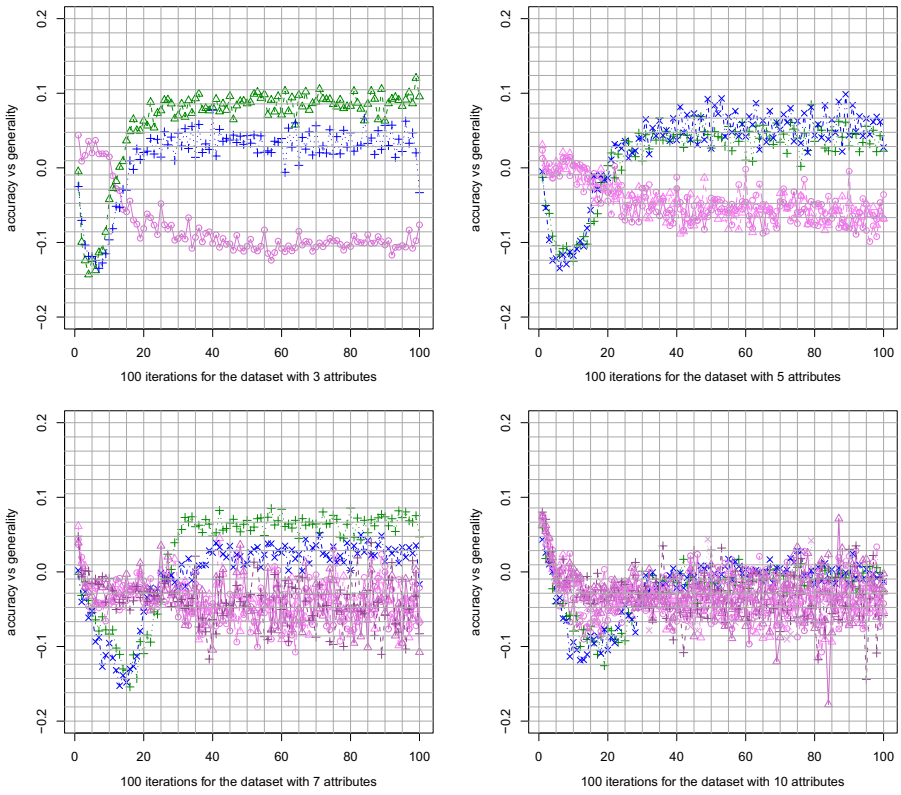

Fig. 9. The pearson correlation coeffiecent between accuracy and generality for each attribute of datasets with respectively 3, 5, 7 and 10 attributes.

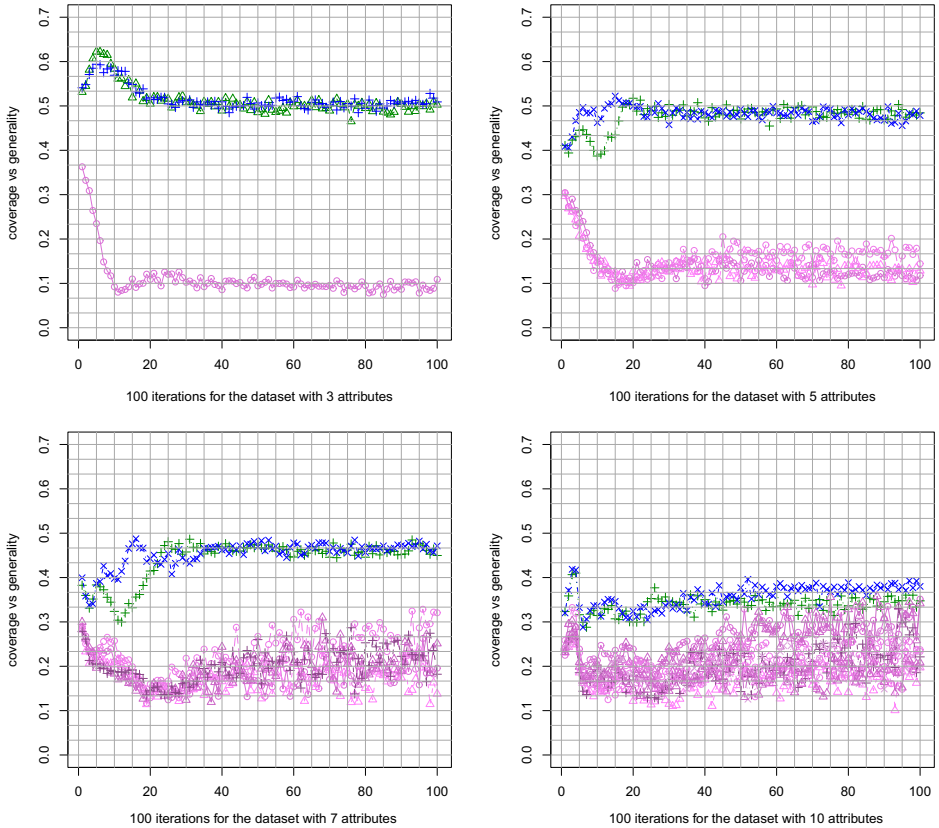

Fig. 10. The pearson correlation coeffiecent between coverage and generality for each attribute of datasets with respectively 3, 5, 7 and 10 attributes.

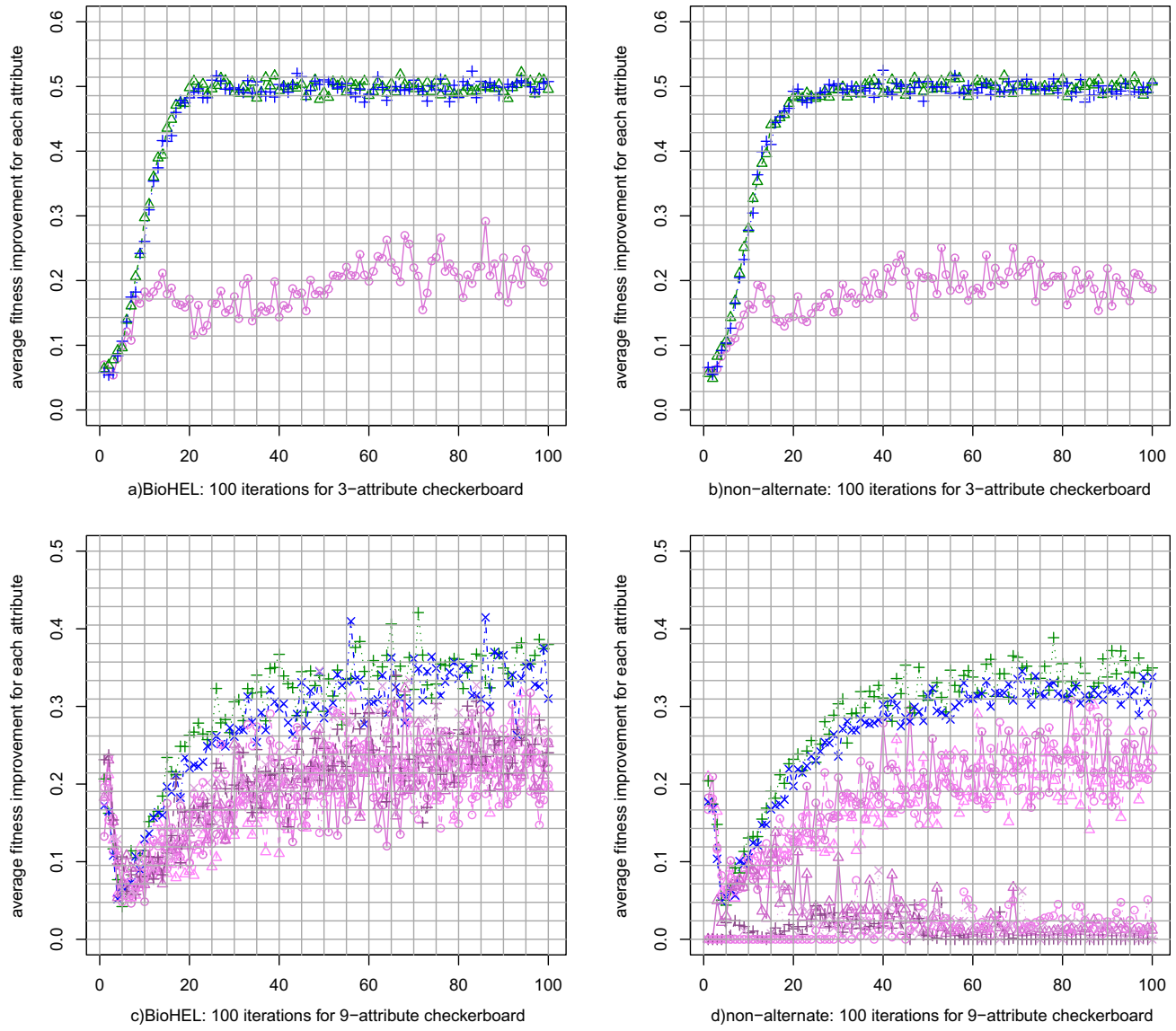

Fig. 11. the average fitness improvement of the original BioHEL and the non-alternate system on the checkerboard datasets with respectively 3 and 9 attributes.

as the number of attributes grows, for both variables, curves of significant attributes tend to merge with those of the noise attributes, particularly on datasets of 15 and 20 attributes.

Also analyzed was the pearson correlation coefficient between each pair of different variables, as illustrated by Figs. 8–10. It is demonstrated that the expected correlation between coverage and generality can be detected for all attributes. For datasets of 3 attributes, the correlation is relatively constant while for datasets with large amounts of noises, these correlation coefficients show significant fluctuation. It is noted that, as the number of noises grow, the correlation coefficients between different variables for significant attributes become more and more indistinguishable from those for noise attributes.

5. BioHEL integrated with EDA techniques

The above various analysis exposed the BioHEL's inability of identifying significant attributes for noisy datasets. This fact necessitates the development of a mechanism which can help identify significant attributes. Although there have been a number of Estimation of Distribution Algorithms (EDA) existing in the literature which exploits the structure of the problem space to find the useful attributes or attribute combinations, they were unable to be directly applied to the BioHEL. Nonetheless, it is observed that the fitness change brought by the special operation, which include the generalization and the specialization operators, can be informative about the relevance of the attributes. Fig. 11 (a) and (c), for the checkerboard datasets with 3 and 9 attributes respectively, plot the average fitness improvement versus the number of total occurrences for each attribute at each iteration. In the figures, the blue and the green lines indicates the original two attributes and the cyan the noise one(s). For the specialization operator, the fitness improvement refers to only the positive change in the value of a rule's fitness after the addition of a attribute. For the generalization operator, the fitness improvement refers to that the negative change in fitness upon the removal of the attribute. The fitness improvement for a specific attribute is summed and then averaged over the total occurrence of special operations, which results in the average fitness improvement for the attribute.

From Fig. 11(a) and (c), it can be observed that the average fitness improvement of both significant attributes is much higher

Table 3
Special operators incorporated with the EDA technique: non-alternate style.

(1) Initialize the probability vector
for i: = 1 to ℓdo
$p[i] = 1/\ell$
(2) Initialize the sum of fitness improvement and the occurrence count of special operations for each attribute
for i: = 1 to ℓdo
$s[i] = 0; c[i] = 0$
(3) Apply the specialization or the generalization operator to each individual by probability.
for i: = 1 to n do
generate a random number represented by $p_s \in [0,1]$
if p_s is greater than probability of specialization, then
{
generate a random number between [0,1]
for j: = 1 to ℓdo
if $\sum_1^j p[i]$ is greater than the random number
then break;
add the jth attribute to the i-th rule;
$c[j] = c[j] + 1$;
if the fitness of the rule after specialization is increased by $\Delta F > 0$ then
$s[j] = s[j] + \Delta F$
}
elseif p_s is greater than greater than probability of generalization, then
{
generate a random number between [0,1]
for j: = 1 to ℓdo
if $\sum_1^j (2 - p[i])$is greater than the random number
then break;
remove the jth attribute from the ith rule;
$c[j] = c[j] + 1$;
if the fitness of the rule after generalization is reduced by $\Delta F > 0$ then
$s[j] = s[j] + \Delta F$
}
(4) Update the probability vector
for i: = 1 to ℓdo
$p[i] = s[i]/\sum_1^\ell s[i]$
(5) Go to step 2 and repeat
ℓ: the number of attributes
n: the number of individuals in the population

Table 4
Special operators incorporated with the EDA technique: alternate style.

(1) xmllabelp0245 Initialize the probability vector
for i: = 1 to ℓdo
$p[i] = 1/\ell$
(2) Initialize the sum of fitness improvement and the occurrence count of special operations for each attribute
for i: = 1 to ℓdo
$s[i] = 0; c[i] = 0$
(3) Depending on the count of iterations, apply the specialization or the generalization operator
to each individual by probability.
if the current iteration count is an odd number then
for i: = 1 to n do
generate a random number represented by $p_s \in [0,1]$
if p_s is greater than probability of specialization, then
{
generate a random number between [0,1]
for j: = 1 to ℓdo
if $\sum_1^j p[i]$ is greater than the random number
then break;
add the jth attribute to the ith rule;
$c[j] = c[j] + 1$;
if the fitness of the rule after specialization is increased by $\Delta F > 0$ then
$s[j] = s[j] + \Delta F$
}
if the current iteration count is an even number then
{
generate a random number between [0,1]
for j: = 1 to ℓdo
if $\sum_1^j (2 - p[i])$ is greater than the random number
then break;
remove the jth attribute from the ith rule;
$c[j] = c[j] + 1$;
if the fitness of the rule after generalization is reduced by $\Delta F > 0$ then
$s[j] = s[j] + \Delta F$
}
(4) Update the probability vector
for i: = 1 to ℓdo
$p[i] = s[i]/\sum_1^\ell s[i]$
(5) Go to step 2 and repeat
ℓ: the number of attributes
n: the number of individuals in the population

than that of noises, which successfully separate the two types of attributes. This observation is in consistent with the intuition that, the loss of significant attributes, in general, worsens the fitness of associated rules across the population while the addition of significant attributes improves their fitness.

Fig. 11(c) shows that, for the checkerboard with 9 dimensions of noise, the average fitness improvement still manages to set the significant attributes apart from the noise attributes to a certain extent. However, by comparing Fig. 11(c) to (a), it can be inferred that as the amount of noises grows larger, the curves of average fitness improvement between significant attributes and noises one become harder to separate.

Thus, it is proposed in this paper: (1) to estimate the significance of attributes by constructing a probabilistic model based on the fitness improvement resulting from the application of special operators; (2) to use the probabilistic model, in turn, as guidance for the exploration of the search space by the special operators. Table 3 lists the pseudocode for the updated special operators at a single iteration.

It is worth attention the algorithm in Table 3 disallows the generalization and the specialization both to be performed at one iteration. Consequently, if the probability of specialization is set to be $a \in [0,1]$, the probability of generalization can not exceed $1 - a$. To address this issue, a second algorithm is proposed which performs generalization and specialization alternately between iterations. Table 4 gives the pseudocode for the algorithm of alternate style. The two algorithms are referred to, respectively, as the non-alternate and the alternate algorithms in this paper.

Fig. 11(b) and (d) depict, for checkerboard with respectively 3 and 9 attributes, the fitness improvement in the non-alternate system brought by different attributes. From Fig. 11(b), it can be observed that the non-alternate systems identified the significant attributes for the dataset with 3 attributes. Also can be seen is that less overlapping were displayed in Fig. 11(d) than in Fig. 11(c), which suggests that the non-alternate system set the attributes and the noise further apart than the traditional BioHEL.

5.1. Parameter tuning of different BioHEL systems

The parameter setting, on the other hand, have a major impact on the performance of the BioHEL algorithm. In an effort to find the optimal parameter setting for the BioHEL, the non-alternate and the alternate algorithms, the following three parameters are tested.

(1) number expressed attributes at initialization: 5, 10, 15, 20,
(2) the interval of the initial population: [0,0.1], [0,0.25],[0.25,0.50], [0.25,0.75],
(3) probability of generalization/specialization: 0.1, 0.3, 0.6.

Table 5
The rank of different parameter settings for the BioHEL on checkerboard datasets.

#Attr	interval	prob.	Rank
10	[0.25,0.75]	0.6	3.33
5	[0.25,0.75]	0.6	4.89
20	[0.25,0.75]	0.6	5.67
5	[0.25,0.75]	0.3	6.33
15	[0.25,0.75]	0.6	6.83
10	[0.25,0.75]	0.3	9.17
5	[0.25,0.75]	0.1	9.33
20	[0.25,0.5]	0.3	13.11
20	[0,0.25]	0.1	14.06
15	[0.25,0.5]	0.3	14.22
20	[0,0.25]	0.3	14.33
15	[0,0.25]	0.1	14.72
15	[0.25,0.75]	0.3	14.83
20	[0.25,0.75]	0.3	15.00
15	[0,0.25]	0.3	16.83
10	[0.25,0.75]	0.1	18.19
10	[0.25,0.5]	0.3	18.89
15	[0.25,0.5]	0.1	19.14
5	[0.25,0.5]	0.3	19.17
20	[0.25,0.5]	0.1	20.00
5	[0.25,0.5]	0.1	20.50
10	[0.25,0.5]	0.1	21.39
20	[0.25,0.5]	0.6	22.50
20	[0,0.1]	0.1	23.89
15	[0,0.1]	0.1	26.56
10	[0,0.25]	0.1	26.89
15	[0.25,0.5]	0.6	27.28
20	[0,0.25]	0.6	27.94
5	[0,0.25]	0.1	28.22
5	[0,0.25]	0.3	28.33
10	[0,0.25]	0.3	28.39
10	[0.25,0.5]	0.6	28.83
15	[0,0.25]	0.6	28.83
5	[0.25,0.5]	0.6	29.17
20	[0,0.1]	0.3	31.17
15	[0.25,0.75]	0.1	31.33
20	[0.25,0.75]	0.1	31.39
15	[0,0.1]	0.3	33.39
5	[0,0.25]	0.6	36.61
10	[0,0.1]	0.1	36.83
10	[0,0.25]	0.6	37.94
5	[0,0.1]	0.1	41.83
10	[0,0.1]	0.3	42.28
20	[0,0.1]	0.6	43.56
15	[0,0.1]	0.6	43.78
5	[0,0.1]	0.3	44.89
10	[0,0.1]	0.6	46.56
5	[0,0.1]	0.6	47.67

Table 6
The rank of different parameter settings for the non-alternate system on checkerboard datasets.

#Attr	interval	prob.	Rank
10	[0.25,0.75]	0.3	3.00
5	[0.25,0.75]	0.3	3.72
15	[0.25,0.75]	0.3	4.83
5	[0.25,0.75]	0.1	5.78
20	[0.25,0.75]	0.3	8.17
10	[0.25,0.75]	0.1	12.11
10	[0.25,0.75]	0.6	12.50
15	[0.25,0.75]	0.6	12.50
20	[0.25,0.75]	0.6	13.06
5	[0.25,0.75]	0.6	13.67
5	[0,0.25]	0.6	14.83
5	[0.25,0.5]	0.1	16.50
15	[0,0.25]	0.3	17.33
20	[0,0.25]	0.6	17.44
5	[0.25,0.5]	0.6	17.56
10	[0,0.25]	0.6	18.00
20	[0,0.25]	0.3	18.50
15	[0,0.25]	0.6	19.00
15	[0.25,0.5]	0.6	19.50
10	[0.25,0.5]	0.6	19.83
20	[0.25,0.5]	0.6	20.61
10	[0.25,0.5]	0.1	21.56
5	[0,0.25]	0.3	22.22
5	[0.25,0.5]	0.3	22.28
20	[0.25,0.5]	0.3	23.67
10	[0.25,0.5]	0.3	24.33
5	[0,0.25]	0.1	25.17
15	[0.25,0.5]	0.1	25.17
15	[0.25,0.5]	0.3	25.17
15	[0.25,0.75]	0.1	26.11
20	[0.25,0.5]	0.1	26.28
20	[0.25,0.75]	0.1	26.44
15	[0,0.25]	0.1	27.28
10	[0,0.25]	0.1	28.39
10	[0,0.25]	0.3	28.67
20	[0,0.25]	0.1	30.17
15	[0,0.1]	0.1	37.22
10	[0,0.1]	0.1	38.00
5	[0,0.1]	0.1	38.17
20	[0,0.1]	0.1	39.44
15	[0,0.1]	0.3	40.28
20	[0,0.1]	0.3	40.50
5	[0,0.1]	0.3	43.33
10	[0,0.1]	0.3	44.22
15	[0,0.1]	0.6	44.83
20	[0,0.1]	0.6	45.17
5	[0,0.1]	0.6	46.44
10	[0,0.1]	0.6	47.06

The resultant 48 combinations are listed in order in Table 5, where "#Attr" refers to number of relevant attributes, "prob" probability of generalization and specialization, and "interval" the value interval for the initial population. For each setting, the BioHEL system which performed the best is identified in the last column.

Empirically, for both the BioHEL and the alternate algorithms, the optimal parameter setting is proved to be 10,[0.25,0.75],0.6 for the three parameters. For the non-alternate method, it is 10,[0.25, 0.75],0.3. From Tables 5–7, it can be seen that, with respect to the interval parameter, the 3 BioHEL learning systems favored a more general setting as [0.25,0.75] while the setting of [0,0.1] resulted in poorer performance. The different systems showed a certain level of flexibility on the parameter of #Attr since the four optional settings achieved comparable performance to each other. The top ranking settings for different systems also suggest a bias towards higher probability for the generalization and the specialization operations in order to combat the large number of noises.

After performing friedman test on the three algorithms each with their respective optimal parameter setting, the average rank for BioHEL, non-alternate and alternate are respectively 2.28, 2.17 and 1.56 with p-value = 0.06573. It indicated that the alternate method performed better than the original BioHEL and the non-alternate algorithms.

6. Experiments on small-scale real-world problems

Experiments were further performed on 30 datasets from UCI repository (Blake & Merz, 1998), whose detailed information is given in Table 8, with the 48 value combinations for the 3 parameters discussed previously. The setting on the rest of the parameters was given in Table 9.

The rank of the 48 value combinations for the different BioHELs were given in Tables 10–12. The optimal parameter setting for both the BioHEL, the non-alternate and the alternate can be read respectively as 20,[0.25,0.5],0.1,15, [0,0.25],0.1 and 15,[0.25,0.75],0.1. The different BioHEL systems exhibited insensitivity, to some extent, to the value setting of the "interval" parameter. And in general, they favored low probability value for generalization and specialization

Table 7
The rank of different parameter settings for the alternate system on checkerboard datasets.

#Attr	interval	prob.	Rank
10	[0.25,0.75]	0.6	4.22
5	[0.25,0.75]	0.6	4.81
5	[0.25,0.75]	0.3	5.78
15	[0.25,0.75]	0.6	6.25
20	[0.25,0.75]	0.6	7.78
5	[0.25,0.75]	0.1	9.22
10	[0.25,0.75]	0.3	9.89
5	[0.25,0.5]	0.3	11.61
10	[0.25,0.5]	0.3	13.22
15	[0.25,0.5]	0.3	14.44
5	[0.25,0.5]	0.1	16.06
15	[0,0.25]	0.3	17.17
20	[0.25,0.5]	0.6	17.28
15	[0,0.25]	0.6	17.61
15	[0.25,0.75]	0.3	17.67
10	[0.25,0.75]	0.1	17.78
20	[0,0.25]	0.6	18.28
5	[0.25,0.5]	0.6	18.33
15	[0.25,0.5]	0.6	18.39
10	[0.25,0.5]	0.6	19.50
20	[0.25,0.5]	0.3	19.78
10	[0.25,0.5]	0.1	20.83
5	[0,0.25]	0.3	21.89
20	[0.25,0.75]	0.3	22.06
5	[0,0.25]	0.1	23.72
20	[0,0.25]	0.3	24.61
5	[0,0.25]	0.6	25.61
15	[0.25,0.5]	0.1	26.00
10	[0,0.25]	0.3	26.89
10	[0,0.25]	0.1	27.00
15	[0,0.25]	0.1	27.06
10	[0,0.25]	0.6	29.00
15	[0.25,0.75]	0.1	29.33
20	[0.25,0.5]	0.1	31.06
20	[0.25,0.75]	0.1	31.44
20	[0,0.25]	0.1	34.56
15	[0,0.1]	0.1	37.94
10	[0,0.1]	0.3	38.28
15	[0,0.1]	0.3	38.28
10	[0,0.1]	0.1	38.83
5	[0,0.1]	0.1	39.11
15	[0,0.1]	0.6	40.33
5	[0,0.1]	0.3	41.17
20	[0,0.1]	0.6	41.33
20	[0,0.1]	0.1	42.00
20	[0,0.1]	0.3	42.17
10	[0,0.1]	0.6	43.83
5	[0,0.1]	0.6	46.61

Table 8
summary of datasets.

code	# Instances	# Attributes	# Classes
bal	625	4	3
bpa	345	6	2
bre	286	9	2
cmc	1473	9	3
col	368	22	2
cr-a	690	15	2
gls	214	9	6
h-c1	303	13	2
hep	155	19	2
h-h	294	13	2
h-s	270	13	2
ion	351	34	2
irs	150	4	3
lab	57	16	2
let	20000	16	26
lym	148	18	4
pen	10992	16	10
pim	768	8	2
prt	339	17	21

Table 8 (*continued*)

code	# Instances	# Attributes	# Classes
sat	6435	36	6
seg	2310	19	7
son	208	60	2
thy	215	5	3
vot	435	16	2
wav	5000	40	3
wbcd	699	9	2
wdbc	569	30	2
wine	178	13	3
wpbc	198	33	2
zoo	101	16	7

Table 9
Base configuration of different BioHEL systems.

crossover operator	1px
default class	disabled
fitness function	MDL
initialization min classifiers	20
initialization max classifiers	20
iterations	100
MDL initial theory learning ratio	0.25
MDL iteration	10
MDL weight relax factor	0.90
population size	500
probability crossover	0.6
probability individual mutation	0.6
probability one	0.75
selection algorithm	tournamentwor
tournament size	4
windowing ILAS	2
dump evolution stats	TRUE
smart initialization	TRUE
class wise initialization	TRUE
coverage breakpoint	0.25
repetitions of rule learning	2
coverage ratio	0.90
knowledge representation hyperrect	TRUE
hyperrectangle uses list of attributes	TRUE

Table 10
The rank of different parameter settings for the conventional BioHEL on real-life problems.

#Attr	interval	prob.	Rank
20	[0.25,0.5]	0.1	18.22
15	[0,0.25]	0.3	18.33
20	[0,0.25]	0.1	19.20
15	[0,0.1]	0.1	20.87
20	[0,0.1]	0.1	20.93
20	[0,0.25]	0.3	20.95
10	[0.25,0.5]	0.1	21.05
10	[0,0.25]	0.1	21.07
15	[0.25,0.75]	0.1	21.47
15	[0,0.25]	0.1	21.70
20	[0.25,0.5]	0.6	22.00
15	[0.25,0.5]	0.6	22.08
5	[0,0.1]	0.1	22.60
20	[0.25,0.5]	0.3	22.78
20	[0.25,0.75]	0.6	23.63
10	[0,0.25]	0.3	23.75
20	[0.25,0.75]	0.3	23.78
20	[0,0.1]	0.6	23.80
5	[0.25,0.5]	0.1	23.82
15	[0.25,0.5]	0.3	23.90
15	[0.25,0.75]	0.6	24.08
20	[0.25,0.75]	0.1	24.08
15	[0.25,0.75]	0.3	24.10
20	[0,0.1]	0.3	24.20
15	[0,0.1]	0.3	24.42
15	[0.25,0.5]	0.1	24.88
10	[0.25,0.75]	0.3	25.10

Table 10 (*continued*)

#Attr	interval	prob.	Rank
5	[0,0.1]	0.3	25.23
15	[0,0.25]	0.6	25.28
10	[0.25,0.75]	0.6	25.30
10	[0.25,0.75]	0.1	25.60
5	[0,0.25]	0.1	25.65
5	[0,0.25]	0.3	26.00
10	[0.25,0.5]	0.6	26.75
20	[0,0.25]	0.6	26.78
10	[0.25,0.5]	0.3	26.80
10	[0,0.1]	0.3	26.82
5	[0.25,0.5]	0.6	26.88
5	[0.25,0.75]	0.3	27.20
15	[0,0.1]	0.6	27.70
5	[0,0.25]	0.6	27.73
5	[0.25,0.75]	0.1	27.78
5	[0.25,0.75]	0.6	28.08
10	[0,0.1]	0.1	28.12
5	[0,0.1]	0.6	28.18
10	[0,0.25]	0.6	28.38
5	[0.25,0.5]	0.3	28.52
10	[0,0.1]	0.6	30.40

Table 11
The rank of different parameter settings for the non-alternate system on real-life problems.

#Attr	interval	prob.	Rank
15	[0,0.25]	0.1	18.13
15	[0,0.1]	0.1	18.47
20	[0,0.1]	0.1	19.78
10	[0,0.25]	0.1	19.92
15	[0.25,0.5]	0.6	20.35
20	[0.25,0.75]	0.3	20.93
15	[0.25,0.5]	0.1	21.18
20	[0.25,0.5]	0.1	21.28
20	[0.25,0.5]	0.6	21.62
20	[0.25,0.75]	0.6	22.00
15	[0.25,0.75]	0.3	22.37
10	[0.25,0.75]	0.1	22.50
20	[0,0.1]	0.3	22.72
10	[0,0.1]	0.1	22.80
20	[0,0.25]	0.1	22.83
20	[0.25,0.75]	0.1	22.93
15	[0.25,0.5]	0.3	22.95
10	[0.25,0.75]	0.3	23.18
15	[0,0.1]	0.3	23.70
5	[0.25,0.75]	0.1	23.75
5	[0,0.1]	0.1	24.15
5	[0.25,0.5]	0.1	24.17
10	[0.25,0.5]	0.1	24.20
20	[0.25,0.5]	0.3	24.28
15	[0.25,0.75]	0.1	24.52
10	[0.25,0.5]	0.3	25.10
10	[0.25,0.5]	0.6	25.17
10	[0,0.25]	0.3	25.23
20	[0,0.25]	0.3	25.23
15	[0.25,0.75]	0.6	25.27
5	[0,0.25]	0.1	25.47
5	[0.25,0.5]	0.6	25.52
10	[0.25,0.75]	0.6	25.70
5	[0,0.25]	0.6	25.93
15	[0,0.25]	0.6	26.25
20	[0,0.1]	0.6	26.57
5	[0.25,0.75]	0.6	26.90
5	[0.25,0.75]	0.3	26.92
20	[0,0.25]	0.6	26.92
5	[0.25,0.5]	0.3	26.95
10	[0,0.1]	0.3	27.30
10	[0,0.25]	0.6	27.33
5	[0,0.25]	0.3	27.75
5	[0,0.1]	0.3	28.13
15	[0,0.25]	0.3	29.00
10	[0,0.1]	0.6	29.17
5	[0,0.1]	0.6	31.18
15	[0,0.1]	0.6	32.30

Table 12
The rank of different parameter settings for the alternate system on real-life problems.

#Attr	interval	prob.	Rank
15	[0.25,0.75]	0.1	18.77
20	[0.25,0.5]	0.1	19.42
15	[0,0.25]	0.1	19.75
20	[0,0.1]	0.3	19.77
20	[0.25,0.75]	0.6	19.80
15	[0,0.1]	0.1	20.28
20	[0,0.1]	0.1	20.28
20	[0.25,0.5]	0.3	20.47
10	[0,0.1]	0.1	21.10
10	[0.25,0.5]	0.1	21.38
15	[0.25,0.75]	0.3	21.52
15	[0.25,0.5]	0.6	21.75
20	[0.25,0.5]	0.6	22.10
15	[0.25,0.5]	0.1	22.35
20	[0,0.1]	0.6	22.35
20	[0.25,0.75]	0.3	22.42
20	[0,0.25]	0.3	23.08
20	[0,0.25]	0.1	23.40
20	[0.25,0.75]	0.1	23.52
15	[0,0.25]	0.6	23.62
15	[0.25,0.5]	0.3	23.87
15	[0,0.1]	0.3	24.32
20	[0,0.25]	0.6	24.38
5	[0,0.25]	0.1	24.43
10	[0.25,0.75]	0.1	24.82
10	[0.25,0.75]	0.6	25.25
15	[0,0.1]	0.6	25.25
10	[0,0.1]	0.3	25.33
10	[0,0.25]	0.1	25.43
5	[0.25,0.75]	0.6	25.50
10	[0.25,0.5]	0.6	25.57
5	[0,0.1]	0.6	25.97
15	[0.25,0.75]	0.6	26.32
15	[0,0.25]	0.3	26.38
10	[0.25,0.75]	0.3	26.63
5	[0.25,0.5]	0.1	26.77
5	[0,0.25]	0.6	27.62
5	[0,0.1]	0.1	27.65
10	[0,0.25]	0.6	27.78
5	[0,0.25]	0.3	27.88
10	[0,0.1]	0.6	27.88
5	[0.25,0.75]	0.3	27.92
10	[0,0.25]	0.3	28.13
5	[0,0.1]	0.3	28.40
10	[0.25,0.5]	0.3	29.37
5	[0.25,0.75]	0.1	29.85
5	[0.25,0.5]	0.6	29.87
5	[0.25,0.5]	0.3	30.32

and a higher setting on "#Attr". It can be justified by the fact that, comparatively, these datasets contain much smaller amount of irrelevant attributes than the noisy checkerboard datasets. As a result, it did not depend on the generalization and specialization operations as heavily to identify significant attributes.

The BioHEL, the non-alternate and the alternate learning machines, each using their respective optimal parameter setting, were then applied to the 30 classification problems. Friedman tests were performed on the three algorithms, and the average rank for the BioHEL, the non-alternate and the alternate are respectively 2.0, 2.08 and 1.92 with p-value = 0.8. The alternate method is proven to be the best method among the three.

7. Conclusions

In order to improve the performance of the newly-emerged BioHEL learning systems, the paper proposed to build a probabilistic model which identifies the significant attributes according to the fitness improvement upon the application of generalization and

specialization operators. Experiments on synthetic checkerboard datasets showed that, the introduction of the probabilistic model lead to the successful recognition of the checkerboard pattern even the original benchmark was added with 18 dimensions of Gaussian noises. In contrast, the conventional BioHEL system can only maintain satisfactory performances up to 7 extra attributes of noises. Experiments on small-scale datasets further confirmed the superiority, in terms of the classification accuracies, of the proposed BioHEL systems over the traditional one. It was also analyzed the impact of different parameter settings on the generalization of BioHEL systems. Future work will focus on the application of proposed BioHEL systems to large-scale problems.

Acknowledgement

This work was supported by grants from the project of Zhejiang Provincial Natural Science Foundation of China (Project LQ13F030011).

References

Bacardit, J., Burke, E., & Krasnogor, N. (2009). Improving the scalability of rule-based evolutionary learning. *Memetic Computing, 1*(1), 55–67.

Bacardit, J., & Krasnogor, N. (2006). Smart crossover operator with multiple parents for a pittsburgh learning classifier system. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 1441–1448). ACM.

Bacardit, J., & Krasnogor, N. (2009). Performance and efficiency of memetic Pittsburgh learning classifier systems. *Evolutionary Computation, 17*(3), 307–342.

Blake, C.L., Merz, C.J., (1998). UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/MLRepository. html>.

Butz, M. (2006). *Rule-based evolutionary online learning systems: A principled approach to LCS analysis and design*. Springer Verlag.

Butz, M., Goldberg, D., & Lanzi, P. (2005). Gradient descent methods in learning classifier systems: Improving XCS performance in multistep problems. *IEEE Transactions on Evolutionary Computation, 9*(5), 452–473.

Chang, C., Lin, C., (2001). LIBSVM: A Library for Support Vector Machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm> 80 (pp. 604–611).

Grefenstette, J. J. (1991). Lamarckian learning in multi-agent environments. In *Proceedings of the fourth international conference on genetic algorithms* (pp. 303–310). Morgan Kaufmann.

Harik, G., Lobo, F., & Goldberg, D. (2002). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation, 3*(4), 287–297.

Harik, G., Lobo, F., & Sastry, K. (2006). Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ecga). *Scalable Optimization via Probabilistic Modeling*, 39–61.

Krasnogor, N., & Smith, J. (2005). A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation, 9*(5), 474–488.

Larranaga, P., & Lozano, J. (2002). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Netherlands: Springer.

Llorà, X., Sastry, K., & Goldberg, D. (2005). The compact classifier system: Motivation, analysis, and first results. In *Proceedings of the 2005 conference on Genetic and evolutionary computation* (pp. 1993–1994). ACM.

Llorà, X., Sastry, K., Goldberg, D., & de la Ossa, L., (2006). The χ-ary extended compact classifier system: Linkage learning in pittsburgh lcs. In *Proceedings of the ninth international workshop on learning classifier systems*.

Park, W., & Oh, J. (2009). New entropy model for extraction of structural information from XCS population. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 1283–1290). ACM.

Pelikan, M. (2005). Bayesian optimization algorithm. *Hierarchical Bayesian Optimization Algorithm*, 31–48.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*(5), 465–471.

Smith, S., (1980). A learning system based on genetic adaptive algorithms. PhD Thesis, University of Pittsburgh Pittsburgh, PA, USA.

Venturini, G. (1993). SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts. In *Machine learning: ECML-93* (pp. 280–296). Springer.

Wilson, S. (1995). Classifier fitness based on accuracy. *Evolutionary Computation, 3*(2), 149–175.

Wyatt, D., & Bull, L. (2005). A memetic learning classifier system for describing continuous-valued problem spaces. *Recent Advances in Memetic Algorithms*, 355–395.