Soft Comput
DOI 10.1007/s00500-010-0678-8

ORIGINAL PAPER

REGAL-TC: a distributed genetic algorithm for concept learning based on REGAL and the treatment of counterexamples

L. Ignacio Lopez · Juan M. Bardallo · Miguel A. De Vega · Antonio Peregrin

Abstract This paper presents a proposal to improve REGAL, a concept learning system based on a distributed genetic algorithm that learns first-order logic multi-modal concept descriptions in the field of classification tasks. This algorithm has been a pioneer system and source of inspiration for others. Studying the philosophy and experimental behaviour of REGAL, we propose some improvements based principally on a new treatment of counterexamples that promote its underlying goodness in order to achieve better performances in accuracy, interpretability and scalability, so that the new system meets the main requirements for classification rules extraction in data mining. The experimental study carried out shows valuable improvements compared with both REGAL and G-Net distributed genetic algorithms and interesting results compared with some state-of-the-art representative algorithms in this field.

Keywords Concept learning · Distributed genetic algorithms · Cooperative evolution · Multi-modalities

L. I. Lopez · J. M. Bardallo · M. A. De Vega · A. Peregrin (✉)
Department of Information Technologies, University of Huelva, Palos de la Fra., Huelva, Spain
e-mail: peregrin@dti.uhu.es

L. I. Lopez
e-mail: nacho@uhu.es

J. M. Bardallo
e-mail: juanmanuel.bardallo@sic.uhu.es

M. A. De Vega
e-mail: miguelangelde.vega@sic.uhu.es

1 Introduction

In general terms in the field of Artificial Intelligence, classification problems can be faced by numerous algorithms (Lanzi 2008), which belong to different Machine Learning (ML) paradigms. Finding a correspondence between the problem and its optimum algorithm currently remains an open issue (Ho and Pepyne 2002).

The so-called Evolutionary Rule-Based Systems (Freitas 2003) are a type of Genetics-Based Machine Learning (GBML) that use rule sets as knowledge representation. One of the strengths of these approaches is the use of evolutionary algorithms as search mechanisms which allows for efficient searches over complex search spaces (Orriols-Puig and Bernadó-Mansilla 2009; Reynolds and de la Iglesia 2009; Rivero et al. 2009). Many approaches have been proposed in the field of symbolic GBML (Orriols-Puig et al. 2008; Fernández et al. 2010), which offer some advantages compared with non-symbolic techniques, such as the production of interpretable models, without assuming a priori information about the domain of the problem, not even prior relationships among attributes (Freitas 2001), with the possibility of obtaining compact and precise rule sets.

In the framework of data mining (Tan et al. 2006a; Witten and Frank 2005), obtaining classification models with high prediction in current databases could be a complex, inefficient and ineffective task due to the size of the data, so system scalability should necessarily be added to the desired virtues of the algorithms in terms of accuracy and interpretability, as a counterpart to the scaling problem (Yang and Wu 2006; Provost and Kolluri 1999; Orriols-Puig et al. 2008).

Regarding scalability, there are three main approaches to resolve this issue (Yang and Wu 2006):

Published online: 03 December 2010

- prior knowledge to guide the search,
- data reduction,
- algorithm scalability.

One way to achieve algorithm scalability is the use of distributed computation that involves computational decentralisation across a number of processors, which may be physically located in different components, subsystems, systems or facilities. REGAL (Giordana and Neri 1995) and G-Net (Giordana et al. 1997) are based on distributed genetic algorithms which increase computational resources via the use of data distribution along with an array of computers to achieve greater performance.

In this paper, we present REGAL-TC (REGAL with Treatment of Counterexamples), an enhanced version of REGAL which improves the convergence of the nodes, the quality of the rules and optimises the final classifier introducing a new treatment of the counterexamples, showing advantages in interpretability, accuracy and scalability after the experimental study and statistical analysis carried out, revealing an outperformance in these respects in comparison with their predecessor and interesting results with state-of-the-art representative algorithms.

The outline of our contribution is as follows. In Sect. 2, we review the context of genetic learning evolutionary systems based on rules, highlighting and emphasising the genetic cooperative-competitive type with distributed implementation. Section 3 is devoted to the analysis and synthesis of the proposed changes. The experimental study carried out is shown in Sect. 4, as is the statistic assessment, Sect. 5 being where we reach conclusions and comment some future works and proposals.

2 Preliminaries: genetic learning and REGAL

In Witten and Frank (2005), the authors propose a classification of the different data mining tasks. One of them consists of classification problems where the goal is to predict the value of a distinguished discrete variable (the class) using the values of the remaining ones. Due to the success obtained by evolutionary algorithms when applied to complex optimisation problems, they are shown to be one of the most robust methods to deal with real-world problems (Holden and Freitas 2009; Marín-Blázquez and Martínez Pérez 2009; Stout et al. 2009).

2.1 Genetic learning

Genetic algorithms were not specifically designed for learning but as optimisation algorithms based on a global search in the solution space. However, as Mitchell (1982) noticed, mutatis mutandi, the problem can be formulated as a search in a hypothesis space corresponding to candidate descriptions in a specified language. Genetic algorithms are appropriate searching engines to find the solution model which best fits with the learning task (Orriols-Puig et al. 2008).

When considering a rule system and focusing on learning rules, the different genetic methods follow two approaches in order to encode rules within a population of individuals (Fernández et al. 2010):

1. The Pittsburgh approach, in which each individual represents a rule set. In this case, each chromosome evolves a complete rule base and competes among them in an evolutionary way. GABIL (De Jong et al. 1993), GIL (Janikow 1993) and GAssist (Bacardit et al. 2007) are three paradigmatic examples that follow this approach.
2. In the second approach, each individual codifies a single rule, and the whole rule set is provided by juxtaposition of several individuals in the population (rule cooperation) or via different evolutionary runs.

The last approach embraces three generic proposals:

- The Michigan approach. These kinds of systems are usually called learning classifier systems (Holland and Reitman 1977): XCS (Wilson 1995) and UCS (Bernadó-Mansilla and Garrell-Guiu 2003) belong to this category.
- IRL (Iterative Rule Learning) approach, in which each chromosome represents a rule. Chromosomes compete in every GA run, choosing the best rule per run. The global solution is formed by the best rules obtained when the algorithm is run multiple times. SIA (Venturini 1993), ESIA (Liu and Kwok 2000) and HIDER (Aguilar-Ruiz et al. 2003) are proposals that follow this approach.
- GCCL (Genetic Cooperative-Competitive Learning) approach, in which the complete population, or a subset of it, encodes the rule base. In this model, the chromosomes compete and cooperate simultaneously. COGIN (Greene and Smith 1993), REGAL (Giordana and Neri 1995), G-Net (Giordana et al. 1997), OCEC (Jiao et al. 2006), EDGAR (Rodríguez et al. 2010) and DOGMA (Hekanaho 1997) are examples that can be located in this framework.

We may consider that REGAL has been a pioneer system in the so-called niching genetic algorithms and a source of inspiration for others. Because of its inherent qualities, this algorithm served as a starting point for our research.

Our proposal is focused on the study of treatment of the counterexamples, which will drive the different improvements described in the next section. The main aim is to obtain a better performance in accuracy, interpretability and scalability.

2.2 An overview of REGAL

REGAL is a learning system based on a distributed genetic algorithm that is capable of learning multi-modal concepts described in first-order logic.

In REGAL, each individual, also called a disjunct, encodes a partial solution, i.e., it consists of a conjunctive formula, and the whole population is a redundant set of these partial solutions.

The task of learning concepts by the disjunction of conjunctive formulas, where each formula covers a region or group of examples (modalities), structurally leads to a methodology of niches and species. The "ad hoc" hybrid architecture follows the underlying philosophy of REGAL of processing the aforementioned niches and species in a distributed way.

The REGAL architecture, as described in Cantú-Paz (1998), Alba and Troya (1999), and Alba et al. (2002), can be seen in Fig. 1, consisting of a network of nodal genetic algorithms, also known as Genetic Algorithm Learners (GALs), coordinated by a Supervisor (Bianchini et al. 1995; Weilie et al. 2000; Nojima et al. 2008).

The Supervisor dynamically assigns a subset of the complete dataset to the nodes according to a long-term strategy, the so-called cooperative evolution (Neri 2002), which is aimed at distributing different species on different nodes in order to reduce the genetic pressure of large disjuncts on small ones (Carvalho and Freitas 2002; Orriols-Puig and Bernadó-Mansilla 2005). In this respect, we have used the cooperative strategy DTSU (Describe Those Still Uncovered) as described by Neri (2002).

The Supervisor, constantly monitoring the state of all GALs, can easily focus the space exploration by modifying the set of examples assigned to any GAL. In this way, nodes with associated different subsets of examples shall become niches where different species can grow up. Therefore, the mating between individuals of different niches can be controlled by tuning the migration parameter μ. Finally, the Supervisor periodically extracts a rule base from the global population, i.e., the best individual from each node, and, when it finds a satisfactory one, according to a chosen stopping criterion, it halts the whole system.

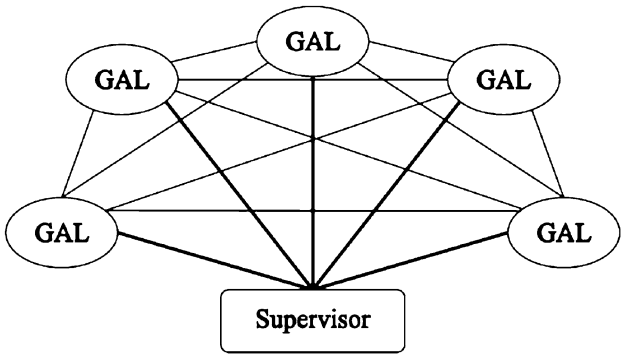

Fig. 1 Abstract view of REGAL

In order to let the system learn at least a reasonable rule base, when it is not able to improve the best description found at the moment, a freezing mechanism has been introduced. The freezing mechanism is a restarting of the learning process. It is assumed that this description represents a target concept modality, and that it is not further improvable. Consequently, it can be saved and its covered examples can be removed from the learning set, in order to move the focus of the hypothesis space exploration.

A GAL process is basically a classic genetic algorithm. It uses binary fixed string chromosomes based on VL21 language (Michalski 1983) totally suitable for the task of this algorithm, which is an important point taking into account that "the representation scheme can severely limit the windows by which the system observes its world" (Michalewicz 1996).

Figure 2 shows an example of bit strings encoding formulas for a given concept Λ formed by the conjunction of the predicates: colour and shape, each one having a set of attributes. Taking this concept into account, a set of individuals can be generated setting to 1 or 0 those values that are selected or not, respectively.

Regarding the genetic operators, it uses selection, crossover, mutation and seeding operators. The most relevant differences compared with a classical genetic algorithm concern seeding and selection.

The seeding operator is used to dynamically generate new individuals to cover an example. It can be seen as a more sophisticated version of the *new event* operator used in GIL and of the *creation* operator used in SIA. More precisely, for a given example, the seeding operator returns an individual covering it. The process is as follows:

Seeding (ξ)
  Let ξ be an example
  Generate a random bit string s
  defining an individual φ
  Turn to 1 the smallest set of bits in s
  necessary to satisfy φ on ξ
  Return (φ)

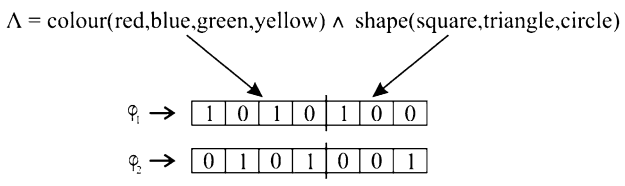

Fig. 2 Bit strings corresponding to the individuals φ_1 and φ_2 obtained for the concept Λ

The selection operator implemented is Universal Suffrage (US). The basic idea is that the individuals to be mated are not chosen directly from the current population; instead, it is the examples that choose which individuals are selected. The selection procedure works as follows.

A number $gM \leq M$ of examples is randomly selected, where M is the cardinality of the population and g is the *generation gap* and represents the proportion of formulas selected for mating. To each selected example ξ_k, a set $R(\xi_k)$, containing the formulas covering it, is associated. The set $R(\xi_k)$ corresponds to a "roulette wheel" r_k, divided into sectors, each one associated to a $\varphi_j \in R(\xi_k)$. The extension of the sector associated with φ_j is proportional to the ratio between φ_j's total fitness, calculated as the fitness of the individual multiplied by its multiplicity (number of formulas that are equal in the population), and the sum of the total fitness values of all the formulas in $R(\xi_k)$. For each spin of the wheel r_k, the winning individual is chosen.

```
Universal Suffrage Selection
    Let B(t) = ∅
    Randomly select gM examples
    for each selected example ξ_k do
        if R(ξ_k) ≠ ∅ then
                Spin r_k and add the winning individual to B(t)
        else
                Create a new individual φ covering ξ_k by
                applying the seeding operator and add to B(t)
    end
```

Its ability to let subpopulations not disappear, irrespective of the cardinality, reaching an equilibrium state has been theoretically and experimentally proven (Neri and Saitta 1996).

The US selection operator favours those individuals with higher coverage. If an example is not yet covered, it would be sensible to dynamically generate new individuals covering it using the *seeding* operator.

The fitness function is very simple in comparison with similar algorithms, such as G-Net and DOGMA, which use MDL (Rissanen 1989) to implement their fitness function, although in REGAL's fitness only the consistency and simplicity of the solution are considered, while the completeness is considered by the US operator. The fitness of an individual φ is given by the formula:

$$f(\varphi) = f(z, w) = (1 + Az)\, \mathrm{e}^{-w}$$

where w is equal to the number of counterexamples covered by φ, and z is a measure of φ's simplicity ($z \in [0, 1]$), calculated as the average number of bits equal to 1 in the bit string. The parameter A is user-tunable and its value was fixed by the authors at $A = 0.1$.

REGAL uses four crossover operators; there are the well-known two-point and uniform crossovers, and the generalising and specialising crossovers, specifically designed for the task at hand. The synergy of multiple crossover operators has been studied previously (Yoon and Moon 2002).

The generalising and specialising crossovers need additional explanation. As shown in Fig. 2, the bit string of an individual can be divided into substrings, each corresponding to a specific predicate. In both crossover operators, a set of predicates is randomly selected. The bit strings in the parent individuals corresponding to the predicates not selected are copied unchanged into the corresponding offsprings. Then, for each selected predicate, a new substring is generated by AND-ing (OR-ing) the bits of the corresponding substring of the parents. The generated substring is then copied in both offsprings.

Given a pair of bit strings (s_1, s_2), representing two individuals φ_1 and φ_2 selected for mating, crossover will be applied with an assigned probability p_c. Then, the specific crossover type is selected stochastically by taking into account the features of s_1 and s_2. The conditional probabilities p_u of uniform crossover, p_{2pt} of two-point crossover, p_s of specialising crossover and p_g of generalising crossover are computed as follows:

$$\begin{aligned} p_u &= (1 - a \cdot f_n) \cdot b \\ p_{2pt} &= (1 - a \cdot f_n) \cdot (1 - b) \\ p_s &= a \cdot f_n \cdot r \\ p_g &= a \cdot f_n \cdot (1 - r) \end{aligned} \tag{1}$$

In expressions (1), a and $b (a, b \in [0, 1])$ are tunable parameters, f_n is the normalised mean value of the fitness of the two individuals φ_1 and φ_2 :

$$f_n = \frac{f(\varphi_1) + f(\varphi_2)}{2 f_{\max}}$$

where $f_{\max}$ is the highest value of fitness of the two individuals and r is the ratio:

$$r = \frac{n^+(\varphi_1) + n^-(\varphi_1) + n^+(\varphi_2) + n^-(\varphi_2)}{2(E + C)} \tag{2}$$

where $n^+(\varphi)$ and $n^-(\varphi)$ are the number of examples and counterexamples covered by φ, respectively, whereas E and C are the number of the training examples and counterexamples, respectively.

As far as the mutation operator is concerned, it is identical to the classical one. It is applied to generate offspring with probability $p_m \ll p_c$ and can affect any bit of the string.

Figure 3 shows the general schema of a GAL. Basically, the process in a GAL is the same as in a classical genetic algorithm. The difference resides in the communication among nodes, which allows a node to share its individuals with others. In each cycle of the algorithm, the GAL selects a subpopulation $Bn(t)$ and combines with some individuals from other nodes according to the US operator, then it

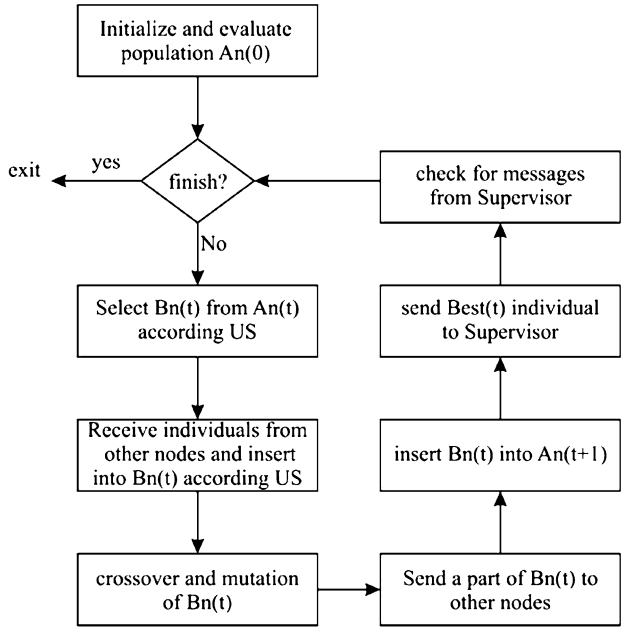

Fig. 3 GAL scheme

performs crossover and mutation and sends a part of $Bn(t)$ to other nodes. Finally, the GAL obtains the population for the next cycle $An(t+1)$ combining $An(t)$ and $Bn(t)$ and sends the current best individual to the Supervisor.

3 Description of our proposal: REGAL-TC

In this section, we describe the proposed improvements. In order to achieve an optimum and predictable system, we have introduced enhancements in aspects which are mostly related with a new treatment of counterexamples. In short, the counterexamples will be the ones that will modulate the strategies in that point. These improvements let the algorithm reach a better consistency, accuracy and scalability.

3.1 Balance between generalisation and specialisation crossovers

Regarding the generalisation and specialisation crossovers, we propose a new theoretical reformulation providing a new expression to work out the probability of the crossing specialisation/generalisation operators. In this way, we try to achieve a better balance between the uses of each of them.

As we seen before, in REGAL, the probabilities for the specialisation and generalisation crossover operators are given by p_s and p_g, where it is r that drives the use of one of the two crossover operators by means of (1). Notice that the formula (2) of r has a tendency to use the specialising crossover p_s in most cases. This is due to the fact that the numerator does not distinguish between examples or counterexamples but only the absolute values; this makes the specialisation operator be used, even though the number of negative cases covered is zero, in which case it would be more suitable to use the generalisation.

Instead of the expression (2), we propose the equation:

$$r = \frac{n^-(\varphi_1) + n^-(\varphi_2)}{n^+(\varphi_1) + n^-(\varphi_1) + n^+(\varphi_2) + n^-(\varphi_2)}$$

We have examined the new expression, which yields a convenient balance between the numbers of runs of p_s and p_g. In this case, r represents the ratio of negative coverage, i.e., the number of counterexamples covered by the two individuals, thus, the higher the value of $n^-(\varphi_1)$ and $n^-(\varphi_2)$, the higher the value of r and therefore the specialisation crossover will be used in order to try to reduce the coverage of the counterexamples. On the contrary, for lower values of $n^-(\varphi_1)$ and $n^-(\varphi_2)$, the tendency is to use the generalising crossover.

3.2 Seeding operator

In the same way as in the new r equation, the counterexamples will be the ones which will direct the process to reformulate the philosophy and implementation of the seeding operator, which is used when the GAL needs mainly to create a new subpopulation or find a new individual for these examples which are not covered by any individual, so that it is necessary to make up a individual for this target. The question is to obtain good rules with a low negative coverage by means of this complementary strategy from the beginning, to speed up the convergence and the consistency of the final classifier.

The modification is based on the study of the individual yielded by the seeding operator, comprehensively analysing the negative covering of each attribute and calculating the entropy taking into account the formula proposed in Zhang et al. (2005), we can calculate the negative information of each attribute of a individual by means of the formula:

$$I(A|R) = \log_2 \frac{P\left(C = c_m | \Pi_{i=1}^{n}(A = v_i)\right)}{P(C = c_m)}$$

where $P\left(C = c_m | \Pi_{i=1}^{n}(A = v_i)\right)$ is the proportion of attribute A with value v_i under $C = c_m$, $P(C = c_m)$ is the proportion of class C with value c_m in the training set. Notice that, in our case, c_m corresponds to the value of the counterexamples class.

Once the negative information for each attribute is calculated, we try to reduce the number of counterexamples covered by the individual; the process works as follows:

- For each attribute of the individual, the negative information is calculated by means of the above formula.
- With all the attributes, a tournament is performed, taking into account the value of the negative information, in order to choose one of them.
- In the last stage, we drop all the values of the attribute without compromising the positive coverage (coverage of the examples) of the rule, i.e., leaving the value that makes the coverage of the seeded example possible.

With this new seeding approach, we obtain more consistent rules reducing the number of counterexamples covered by it. Due to the fact that we use the negative information of the rule, we decide to name this new seeding as NIR-Seeding (Negative Information of the Rule Seeding).

3.3 Convergence of GALs

Taking into account the GALs, we improve the speed up to the local minimum or maximum, preventing excessive interchange between nodes and Supervisor.

As shown in Fig. 3, in REGAL, each node delivers an individual per cycle. We propose more iteration per node with the stopping criterion to send the Supervisor the best individual that does not improve any further in this stage. In this way, the genetic canonical algorithm is allowed more interrelation and hence, more exploration in the searching space according to the underlying philosophy of these algorithms.

Figure 4 shows the new schema of a GAL; notice that the difference resides in the stopping criterion before sending the best individual to the Supervisor. With this new strategy, a GAL will only send the best individual when this fails to improve.

3.4 Dropping strategy

In order to achieve a better interpretability and dramatically lower the overfitting, we propose a new policy that attempts to discard the irrelevant rules that may appear during the evolution process.

When the classifier calculated in the Supervisor is unable to improve, the freezing strategy is launched. This consists of saving the actual classifier, at the same time removing all the examples covered, whereupon the system restarts the process trying to cover the remaining examples.

Once the freezing is applied, it could be possible for the remaining examples to belong to specific regions of the search space, inducing new extremely specific individuals which draw proportionately too many counterexamples.

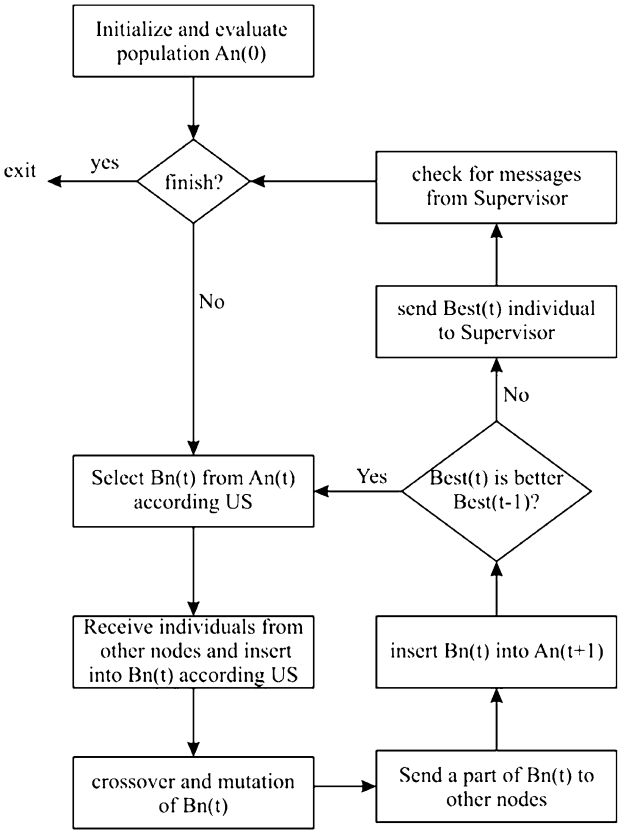

Fig. 4 New schema of a GAL

We have observed experimentally that in complex datasets those low qualities individuals worsening simplicity and making overfitting have appeared.

We propose a policy to filter those individuals which will be part of the final classifier, worsening the simplicity and the accuracy of the final solution.

Based on the ideas exposed in Domingos (1995) and Freitas (2001), we have implemented a strategy in the Supervisor as an abortion method, dropping those individuals that do not reach a certain quality threshold, which depend on the dataset in question.

As in Michalski (1980), we used the weighted sum of the consistency and coverage as a measure of individual quality, which is one of the simplest of those studied in An and Cercone (2000) and is given by the formula:

$$Q_{\mathrm{WS}} = w_1 \cdot \mathrm{cons}(R) + w_2 \cdot \mathrm{cov}(R)$$

where w_1 and w_2 are user-defined weights belonging to $[0, 1]$ and summed 1. And $\mathrm{cons}(R)$ and $\mathrm{cov}(R)$ are the consistency and the coverage of the individual, respectively.

Each of the improvements has a different impact and of course in different respects, but what is observed empirically is the synergistic advantage when they work together, increasing accuracy, simplicity and above all

scalability, a necessary condition when mining large databases.

4 Experimental study

This section describes the experimental study developed to test the proposed method and analyses the results obtained. It is divided into four subsections: in the first one, we describe the algorithms set-up and the datasets employed; in the second, we develop a study of the One-vs-One (Knerr et al. 1990) and One-vs-All (Anand et al. 1995; Clark and Boswell 1991) schemes in order to check the behaviour of REGAL-TC in multi-class problems; then, we compare our proposal with other reference distributed methods, REGAL and G-Net. Finally, we present a comparison of REGAL-TC with some state-of-the-art algorithms.

4.1 Experimental framework

4.1.1 Algorithms considered for comparison

Apart from REGAL and G-Net as representative distributed algorithms, we consider some algorithms for comparison based on the study developed in Fernández et al. (2010), selecting some representative methods. These are OCEC (Jiao et al. 2006) and COGIN (Greene and Smith 1993) as GCCL methods. Due to CORE (Tan et al. 2006b) has shown a low performance, we decided to replace it by GIL (Janikow 1993), which is a GBML algorithm that only works with nominal values, as in our proposal. We have also chosen C4.5 (Quinlan 1993) and RIPPER (Cohen 1995) as non-evolutionary reference. All the methods selected are included in the KEEL software (Alcalá-Fdez et al. 2009). Table 1 summarises the main features of each algorithm indicating the name, family and default parameters used.

4.1.2 Datasets

For the study developed, we selected 25 datasets from the UCI repository (Asuncion and Newman 2007). As in Fernández et al. (2010), we removed the missing values (cleveland, breast cancer, dermatology, hepatitis and wisconsin) and stratified the sample at 10% for the largest datasets (abalone, nursery and penbased). Table 2 shows the characteristics of each one, indicating the id and the name of the dataset, the number of examples (#Ex.), the number of attributes (#Atts.), the number of numerical (#Num.) and nominal (#Nom.) attributes, and the number of classes (#Cl.).

Note that REGAL, REGAL-TC, G-Net, OCEC, COGIN and GIL do not cope with numerical values. A preprocessing discretisation step thus became necessary. We used the Class-Attribute Dependent Discretiser (Ching et al. 1995). In those cases where the discretiser removes all the values of an attribute, we have fixed four as the minimum number of intervals.

4.1.3 Performance measure

To evaluate the performance of the algorithms, the Cohen's kappa measure (Cohen 1960; Ben-David 2007) was used. Cohen's kappa scores the successes independently for each class and aggregates them. It can be calculated using the confusion matrix by means of the following expression:

$$\text{kappa} = \frac{n\sum_{i=1}^{m} h_{ii} - \sum_{i=1}^{m} T_{ri}T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri}T_{ci}}$$

where h_{ii} is the cell count in the main diagonal (the number of true positives for each class), n is the number of examples, m is the number of class labels, and T_{ri}, T_{ci} are the rows' and columns' total counts, respectively ($T_{ri} = \sum_{j=1}^{m} h_{ij}, T_{ci} = \sum_{j=1}^{m} h_{ji}$). Cohen's kappa ranges from -1 (total disagreement) through 0 (random classification) to 1 (perfect agreement).

In the comparison of REGAL-TC with REGAL and G-Net, we also used the number of rules obtained as a measure of the interpretability of each system.

We validate the results obtained with each algorithm by means of a fivefold cross-validation method. The original sample is randomly partitioned into five subsamples. Of the five subsamples, a single one is retained as the validation data for testing the model (test set), and the remaining four subsamples are used as training data (training set). The cross-validation process is then repeated five times (the folds), with each of the five subsamples used exactly once as the validation data. This process is repeated six times using different random seeds. Thus, we have 30 results which are averaged to produce a single estimation.

4.1.4 Statistical tests for performance comparison

To provide statistical support for the analysis of the results, we use the hypothesis testing techniques. Specifically, we use non-parametric test (Demsar 2006; Garcí a et al. 2010).

In this study, we used Wilcoxon Signed-Rank Test (WSRT) (Wilcoxon 1945; Garcí a et al. 2009) to perform a pairwise comparison between two methods. It is a non-parametric alternative to the paired t test which ranks the differences in performance of two classifiers for each dataset, ignoring the signs, and compares the ranks for the positive and negative differences. WSRT can reject the null

Table 1 Methods considered for comparison

Method	Family	Parameters
COGIN	GCCL	Misclassification error level = 2, gen. limit = 1,000, crossover rate = 0.9, negation bit = yes
OCEC	GCCL	Number of total generations = 500, number of migrating/exchanging members= 1
GIL	Pittsburgh	Pop. size = 40, number of gen. = 1,000, $w_1 = 0.5$, $w_2 = 0.5$, $w_3 = 0.01$, rules exchange = 0.2, rule exchange selection = 0.2, rules copy = 0.1, new event = 0.4, rules generalisation = 0.5, rules drop = 0.5, rules specialisation = 0.5, rule split = 0.005, nominal rule split = 0.1, linear rule split = 0.7, condition drop = 0.1, conjunction to disjunction = 0.02, introduce condition = 0.1, rule directed split = 0.03, reference change = 0.02, reference extension = 0.03, reference restriction = 0.03, condition level prob. = 0.5, lower threshold = 0.2, upper threshold = 0.8
REGAL	GCCL	Number of nodes = 6, population size per node = 133, maximum number of generations = 500, maximum number of iterations for freezing = 30, generation gap = 0.9, cross probability = 0.6, $A = 0.5$, $B = 0.5$, mutation probability = 0.001, migration rate = 0.2
G-Net	GCCL	Macro-cycles = 20, micro-cycles = 200, G-nodes = 6, population = 100
REGAL-TC	GCCL	Number of nodes = 6, population size per node = 133, maximum number of generations = 500, maximum number of iterations for freezing = 30, generation gap = 0.9, cross probability = 0.6, cross $A = 0.5$, cross $B = 0.5$, mutation probability = 0.001, fitness $A = 0.1$, migration rate = 0.2, $Q_{ws} = 0.05$, $w_1 = 0.5$, $w_2 = 0.5$
C4.5	non-Evolutionary	Prune = true, confidence level = 0.25, minimum number of item-sets per leaf = 2
RIPPER	non-Evolutionary	Size of growing subset = 66, repetitions of the optimisation stage = 2

hypothesis (Zar 2007) (equal accuracy and interpretability for compared algorithms in our study) when α is smaller than 0.05.

In the case of the number of rules, to make the differences comparable, we propose to adopt a normalised difference DIFF, defined by the expression:

$$\text{DIFF} = \frac{\text{MEAN(other)} - \text{MEAN(reference)}}{\text{MEAN(other)}}$$

where MEAN(x) represents the number of rules means obtained by the x algorithm. This difference expresses the improvement percentage of the reference algorithm on the other one.

In the case of multiple comparisons, we use the Friedman test (Friedman 1937; Garcí a et al. 2010) to detect statistical differences among a group of results and the Finner post hoc test (Finner 1993) to observe the difference in performance among the methods and the retention or rejection of the hypothesis with the level of significance fixed.

4.2 Experimental study of binarization strategies

As Table 2 shows, we have chosen some multi-class datasets for the statistical comparison. However, REGAL-TC is a binary classifier, so it is necessary to use binarization techniques to deal with multi-class datasets. These techniques consist of dividing the original data set into two-class subsets, learning a different binary model for each new subset.

The most common strategies are called "One-vs-One" (OVO) (Knerr et al. 1990), consisting of dividing the dataset into as many binary subsets as possible combinations between pair of classes, and "One-vs-All" (OVA) (Anand et al. 1995; Clark and Boswell 1991), where one class is distinguished from all other classes obtaining as many subsets as number of classes.

In both strategies, once a classifier is obtained for each subset, an aggregation method is applied to calculate the predicted class for a given problem. The simplest way is the application of a voting strategy (Friedman 1996), where each classifier votes for the predicted class and the one with the largest amount of votes is predicted.

The aim of this study is to check which of these techniques (OVO or OVA) works better with REGAL-TC. We run our algorithm with all multi-class datasets shown in Table 2 using both strategies. The results obtained can be seen in Table 3, indicating the dataset ID, the kappa measure in test of REGAL-TC with OVO strategy and OVA strategy, and the number of rules obtained by REGAL-TC with the OVO strategy and with the OVA strategy.

To check for statistical differences between the two strategies, we performed a Wilcoxon Signed-Rank Test whose results are shown in Table 4 for the kappa measure and in Table 5 for the number of rules. We can reject the null hypotheses with a 95% of confidence for the kappa since the p value obtained is 0.02000 and is lower than 0.05. In the case of the number of rules, we cannot reject the null hypotheses. In subsequent sections, we therefore use OVO strategy for REGAL and REGAL-TC to perform the comparisons.

Table 2 Datasets summary description

id	Dataset	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
aba	abalone	418	8	7	1	28
aus	australian credit approval	690	14	8	6	2
bal	balance scale	625	4	4	0	3
bre	breast cancer	286	9	0	9	2
car	car evaluation	1,728	6	0	6	4
cle	cleveland	297	13	13	0	5
con	contraceptive method choice	1,473	9	6	3	3
crx	japanese credit screening	125	15	6	9	2
der	dermatology	366	33	1	32	6
eco	ecoli	336	7	7	0	8
fla	solar flare	1,389	10	0	10	6
ger	german credit data	1,000	20	6	14	2
gla	glass identification	214	9	9	0	7
hab	haberman	306	3	3	0	2
hea	heart	270	13	6	7	2
hep	hepatitis	155	19	6	13	2
iri	iris	150	4	4	0	3
lym	lymphography	148	18	3	15	4
new	new-thyroid	215	5	5	0	3
nur	nursery	1,296	8	0	8	5
pen	pen-based recognition	1,099	16	16	0	10
tic	tic-tac-toe endgame	958	9	0	9	2
veh	vehicle	846	18	18	0	4
wis	wisconsin	683	9	9	0	2
zoo	zoo	101	17	0	17	7

Table 3 Results obtained by REGAL-TC using OVO and OVA strategies

Dataset	Kappa		#R	
	REGAL-TC_{OVO}	REGAL-TC_{OVA}	REGAL-TC_{OVO}	REGAL-TC_{OVA}
aba	0.0768	0.1235	181.100	696.393
bal	0.4719	0.4237	34.933	32.200
car	0.9295	0.9784	58.700	39.167
cle	0.1977	0.2617	73.333	86.133
con	0.0188	0.0863	2.900	12.433
der	0.8589	0.9110	20.400	22.700
eco	0.5413	0.5348	68.533	78.900
fla	0.4634	0.6341	71.133	88.133
gla	0.4207	0.4834	53.300	66.133
iri	0.8067	0.8000	12.067	6.700
lym	0.5190	0.5177	20.933	13.367
new	0.6508	0.6762	18.793	11.300
nur	0.9201	0.9172	83.667	39.167
pen	0.6430	0.7597	151.600	195.500
veh	0.3134	0.3784	143.600	134.600
zoo	0.9337	0.9303	8.667	21.433
Avg. values	0.5478	0.5885	62.729	96.516

Table 4 Wilcoxon Signed-Rank Test for kappa

Algorithm	R^+	R^-	p value
REGAL-TC$_{OVO}$–REGAL-TC$_{OVA}$	113.0	23.0	0.02000

Table 5 Wilcoxon Signed-Rank Test for rules

Algorithm	R^+	R^-	p value
REGAL-TC$_{OVO}$–REGAL-TC$_{OVA}$	58.0	78.0	0.60510

4.3 Comparison of distributed methods

In this section, we perform a comparison between REGAL-TC, REGAL and G-Net. On the one hand, we study the results obtained in terms of Cohen's kappa (as a measure of accuracy) and the number of rules (as a measure of interpretability); on the other, we perform a scalability study to check the behaviour of the algorithms when the number of nodes grows.

4.3.1 Performance study

Table 6 shows the kappa means for REGAL, G-Net and REGAL-TC and the mean number of rules obtained in test by each one for all the datasets shown in Table 2. The average rank and the rank position are also included.

The result obtained for the p value by the Friedman test for kappa measure is 0.02065, which is lower than 0.05. This implies that we can reject the null hypotheses that all methods are equivalent and, therefore, we can perform the Finner post hoc procedure. The results obtained by this procedure are shown in Table 7 where REGAL-TC is the control algorithm. We can conclude that REGAL-TC is the best method outperforming REGAL and G-Net.

For the number of rules, the Friedman test obtains a p value of 0.00035. Table 8 shows the results obtained for the Finner post hoc procedure. In this case, we can conclude that G-Net obtains the least number of rules outperforming REGAL, although there is no statistical

Table 6 Average kappa value and number of rules obtained in test

	Test			#R		
	REGAL	G-Net	REGAL-TC	REGAL	G-Net	REGAL-TC
aba	0.0728	0.0208	0.1235	862.27	23.80	696.39
aus	0.4605	0.6458	0.6322	53.13	17.77	32.20
bal	0.3776	0.5293	0.4237	47.83	6.40	39.17
bre	0.2082	0.1805	0.1864	49.80	34.90	86.13
car	0.9645	0.2653	0.9784	41.63	9.53	22.70
cle	0.2586	0.1620	0.2617	150.37	72.63	78.90
con	0.0153	0.0243	0.0863	241.10	9.13	88.13
crx	0.6143	0.6613	0.6285	66.53	31.70	66.13
der	0.7791	0.5877	0.9110	166.37	65.23	6.70
eco	0.5463	0.4407	0.5348	106.67	22.17	13.37
fla	0.5688	0.2903	0.6341	214.47	17.03	11.30
ger	0.2399	0.1554	0.2558	181.23	90.93	39.17
gla	0.4416	0.4051	0.4834	94.57	44.07	195.50
hab	−0.0007	0.0825	0.0921	9.80	2.47	134.60
hea	0.5264	0.4896	0.5032	30.83	38.00	14.93
hep	0.2644	0.3849	0.3583	9.87	7.73	21.43
iri	0.7800	0.7950	0.8000	7.93	6.50	25.70
lym	0.5458	0.5044	0.5177	25.67	24.53	28.43
new	0.6292	0.6838	0.6762	12.27	14.53	38.67
nur	0.9822	0.5035	0.9172	157.07	31.33	35.13
pen	0.7068	0.3272	0.7597	600.50	108.37	6.70
tic	0.8856	0.8840	0.9927	34.87	50.60	1.00
veh	0.3160	0.1304	0.3784	312.70	120.10	1.13
wis	0.6901	0.9066	0.6864	23.13	12.20	9.57
zoo	0.9308	0.9335	0.9303	22.03	8.70	16.27
Avg. values	0.5122	0.4398	0.5501	140.91	34.81	68.37
Avg. rank	2.12(2)	2.32(3)	1.56(1)	2.60(3)	1.48(1)	1.92(2)

Table 7 p value of Finner post hoc procedure for kappa

i	Algorithm	p_{Finn}
1	G-Net	0.01437
2	REGAL	0.04771

REGAL-TC is the control method

Table 8 p value of Finner post hoc procedure for number of rules

i	Algorithm	p_{Finn}
1	REGAL	0.00015
2	REGAL-TC	0.11979

G-Net is the control method

difference from REGAL-TC, since the p value is 0.11979, which is higher than 0.05.

4.3.2 Scalability study

The goal we pursue is to check the behaviour of the algorithms when the number of nodes is increased for a given problem.

We executed REGAL, G-Net and REGAL-TC for all the datasets in Table 2, fixing the number of generations and the values of the common parameters between the algorithms. For each dataset, we performed the execution with a different number of nodes, from 1 up to 32.

In the following figures, we can see the results obtained for the *tic-tac-toe* and *nursery* datasets as a representative sample. It can be seen that REGAL-TC and G-Net keep approximately the same kappa and the number of rules while in REGAL the kappa decreases and the number of rules grows as the number of nodes increases.

Figures 5 and 6 show the kappa results and number of rules obtained for the *tic-tac-toe* problem, respectively. It can be seen that REGAL-TC and G-Net keep the kappa while REGAL loses performance when the number of nodes increases. In this case, REGAL-TC obtains the best performance in all cases. Regarding the number of rules, we can observe that REGAL-TC and G-Net maintain a similar number of rules regardless of the number of nodes, with REGAL-TC obtaining the lower number of rules. However, REGAL greatly enlarges the number of rules when the number of nodes increases.

Figures 7 and 8 show the kappa results and number of rules obtained for the *nursery* dataset, respectively. As in the previous case, REGAL-TC and G-Net obtain approximately the same performance indistinctly the number of nodes, but the kappa value obtained by G-Net is lower than

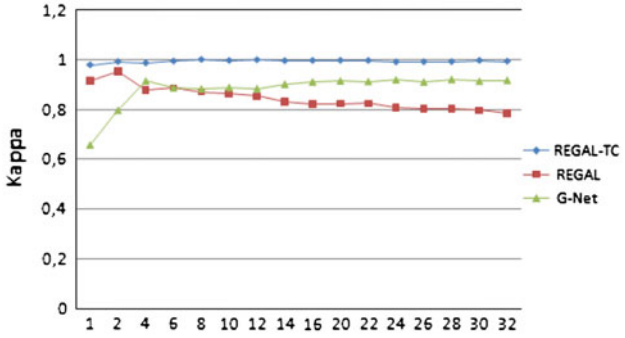

Fig. 5 Kappa results obtained in test for the tic-tac-toe dataset

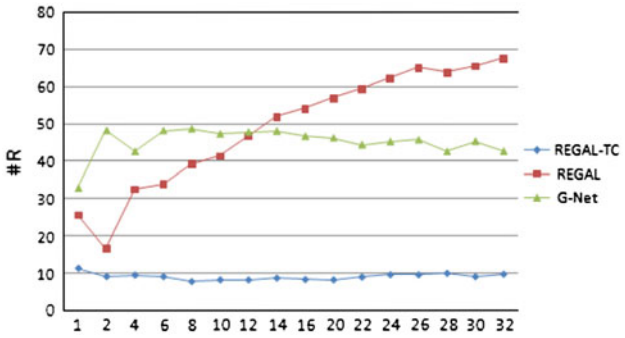

Fig. 6 Number of rules obtained in test for the tic-tac-toe dataset

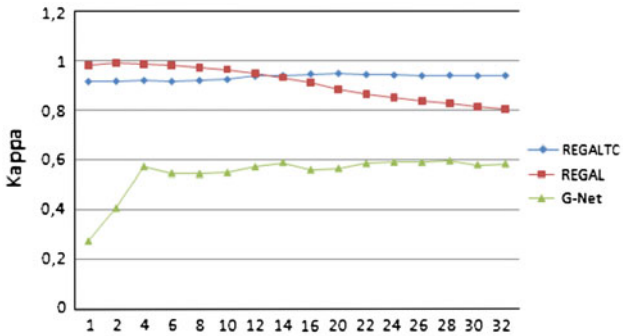

Fig. 7 Kappa results obtained in test for the nursery dataset

REGAL-TC. Regarding REGAL, its performance slightly decreases when the number of nodes rises.

For the number of rules, as in the *tic-tac-toe* dataset, REGAL increases the number of rules when we use a higher number of nodes. In this case, both REGAL-TC and G-Net maintain a similar number of rules regardless the number of nodes.

With the results obtained, it is proven that REGAL-TC and G-Net behave stably for both the accuracy (kappa measure) and interpretability (number of rules) regardless of the increasing number of nodes. In contrast, the performance in REGAL slightly decreases when the number

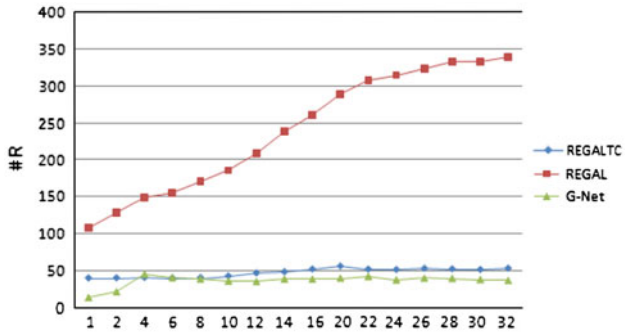

Fig. 8 Number of rules obtained in test for the nursery dataset

of nodes rises. This loss of performance comes about because the greater the increase in the number of nodes, the higher the number of irrelevant rules introduced in the classifier. REGAL does not have any mechanism to avoid this problem, resulting in overfitting and decreasing of both the interpretability and accuracy. On the contrary, REGAL-TC achieves a better interpretability and accuracy by means of rules with lower negative coverage and the dropping strategy to optimise the classifier obtained, dramatically lowering the overfit. REGAL-TC always presents better performance. So, we may say in the light of the results obtained that both algorithms (G-Net and REGAL-TC) are scalable.

4.4 Experimental study of the state-of-the-art

In this section, we establish a comparison with some state-of-the-art algorithms. From the study developed in (Fernández et al. 2010), we try to choose those algorithms, such as REGAL, G-Net and REGAL-TC, which need a pre-processing discretisation step. These are OCEC, COGIN and GIL. Furthermore, C4.5 and RIPPER were chosen as non-evolutionary representative methods. In this study, we

Table 9 Average kappa value obtained in test for multi-class datasets

	Test							
	OCEC	COGIN	GIL	C4.5	RIPPER	REGAL	G-Net	REGAL-TC
aba	0.0701	0.1044	0.0511	0.0873	0.0935	0.0728	0.0208	0.1235
aus	0.7162	0.6912	0.6897	0.6799	0.6300	0.4605	0.6458	0.6322
bal	0.5113	0.5363	0.3414	0.5922	0.3178	0.3776	0.5293	0.4237
bre	0.2082	0.1612	0.1966	0.2330	0.1716	0.2082	0.1805	0.1864
car	0.5535	0.1759	0.6413	0.7986	0.7591	0.9645	0.2653	0.9784
cle	0.3067	0.2132	0.1894	0.2257	0.2068	0.2586	0.1620	0.2617
con	0.1943	0.1410	0.1891	0.2568	0.2723	0.0153	0.0243	0.0863
crx	0.7262	0.6761	0.7003	0.7043	0.6387	0.6143	0.6613	0.6285
der	0.7543	0.8227	0.8033	0.9048	0.8513	0.7791	0.5877	0.9110
eco	0.4768	0.4185	0.4652	0.6998	0.6559	0.5463	0.4407	0.5348
fla	0.6290	0.5851	0.5309	0.6716	0.5891	0.5688	0.2903	0.6341
ger	0.2400	0.1508	0.3138	0.2826	0.2510	0.2399	0.1554	0.2558
gla	0.3376	0.3701	0.4098	0.5742	0.5288	0.4416	0.4051	0.4834
hab	0.0862	0.0318	0.0879	0.1521	0.1432	−0.0007	0.0825	0.0921
hea	0.5583	0.5562	0.5380	0.5866	0.5017	0.5264	0.4896	0.5032
hep	0.3646	0.1109	0.2985	0.1240	0.3191	0.2644	0.3849	0.3583
iri	0.8220	0.7427	0.8500	0.9000	0.8960	0.7800	0.7950	0.8000
lym	0.5508	0.5394	0.5602	0.5367	0.5627	0.5458	0.5044	0.5177
new	0.7132	0.6330	0.6925	0.8140	0.8769	0.6292	0.6838	0.6762
nur	0.7353	0.7412	0.7391	0.8382	0.8386	0.9822	0.5035	0.9172
pen	0.6415	0.7007	0.4319	0.8818	0.8412	0.7068	0.3272	0.7597
tic	0.5909	0.8213	0.4109	0.6766	0.9375	0.8856	0.8840	0.9927
veh	0.3647	0.3755	0.2845	0.6248	0.6104	0.3160	0.1304	0.3784
wis	0.9072	0.8990	0.9033	0.8904	0.9122	0.6901	0.9066	0.6864
zoo	0.9166	0.8838	0.9192	0.9217	0.8828	0.9308	0.9335	0.9303
Avg. values	0.5190	0.4833	0.4895	0.5863	0.5715	0.5122	0.4397	0.5501
Avg. rank	4.18(4)	5.48(7)	4.92(5)	2.72(1)	3.68(2)	5.14(6)	6.04(8)	3.84(3)

Table 10 Adjusted p values

i	Algorithm	p_{Finner}
1	G-Net	0.00001
2	COGIN	0.00024
3	REGAL	0.00111
4	GIL	0.00262
5	OCEC	0.04878
6	REGAL-TC	0.12250
7	RIPPER	0.16586

C4.5 is the control method

used all the datasets shown in Table 2. Table 9 shows the results obtained for the algorithms in test using Cohen's kappa. We also include the average ranking and the rank position of each algorithm using the Friedman test.

The result obtained for the p value by the Friedman test is 0.00001, which is lower than 0.05, so we can perform the Finner post hoc procedure. The results obtained by this procedure are shown in Table 10, where C4.5 is the control algorithm. We may conclude that C4.5 is the best method, although there is no statistical difference with RIPPER and REGAL-TC. OCEC and GIL are better than REGAL while G-Net is outperformed for all methods.

Finally, in view of the results obtained in the experimental framework, we can assert that the main aim of this work has been achieved, i.e., to improve REGAL. Although the C4.5 algorithm obtained the best results with the selected datasets, our approach seemed to perform better than other methods that handle nominal values, in our case OCEC and COGIN, which are of the GCCL family, GIL, and distributed GCCL REGAL and G-Net.

5 Conclusions

In this paper, we presented REGAL-TC, an improved version of REGAL adding some new features based mainly on a new treatment of the counterexamples to achieve a more accurate, interpretable and scalable system.

We reported three sets of experiments on REGAL-TC. In the first, we study the binarization techniques OVO and OVA to verify which of these techniques works best with REGAL-TC when dealing with multi-class datasets; in the second, we compared REGAL-TC with two distributed algorithms (REGAL and G-Net) in terms of performance and scalability; finally, we perform a comparison of REGAL-TC with some state-of-the-art algorithms. Taking into account the results obtained, it appears that our refined algorithm favourably competed with its predecessor and achieved interesting results compared with some state-of-the-art representative algorithms in this field.

Based on the experimentation described so far, we may affirm that REGAL-TC provides solutions with a good accuracy, finding a lower number of rules in all cases, which is a desirable condition in most data mining systems.

In terms of scalability, it can be seen that regardless of the number of GALs selected, REGAL-TC reaches approximately the same accuracy while managing to keep the number of rules. This property is very important, so that the new system meets the main requirements for classification rules extraction in data mining: accuracy, interpretability and scalability.

As we have already commented, there is no theory that matches a problem with its suitable model. Being aware of this, we intend to go one step further, trying to fit our model to the problem at hand without prior knowledge.

The ongoing works are related to several aspects, which we think could be improved, mainly by achieving adaptability in all genetic operators and in those aspects with a static setting and implementing a new cooperative coevolution method (De Jong et al. 1995; Mendes et al. 2001; Kim and Ryu 2007). In this respect, our future works will focus on achieving automatic adaptation to the problem (Herrera and Lozano 2003; Gallagher and Bo 2005) in order to reach what might be termed, metaphorically speaking, the natural system resonance frequency.

Acknowledgments This paper was supported in part by the Spanish Ministry of Education and Science under grant no. TIN2008-06681-C06-06 and the Andalusian government under grant no. P07-TIC-03179.

References

Aguilar-Ruiz JS, Riquelme JC, Toro M (2003) Evolutionary learning of hierarchical decision rules. IEEE Trans Syst Man Cybern Part B Cybern 33(2):324–331

Alba E, Troya JM (1999) A survey of parallel distributed genetic algorithms. Complexity 4(4):31–52

Alba E, Nebro AJ, Troya JM (2002) Heterogeneous computing and parallel genetic algorithms. J Parallel Distrib Comput 62(9):1362–1385

Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell-Guiu JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318

An A, Cercone N (2000) Rule quality measures improve the accuracy of rule induction: an experimental approach. In: Foundations of intelligent systems. Lecture Notes in Computer Science, vol 1932. Springer, Berlin, pp 119–129

Anand R, Mehrotra K, Mohan CK, Ranka S (1995) Efficient classification for multiclass problems using modular neural networks. IEEE Trans Neural Netw 6(1):117–124

Asuncion A, Newman DJ (2007) UCI machine learning repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html

Bacardit J, Goldberg D, Butz M (2007) Improving the performance of a Pittsburgh learning classifier system using a default rule. In: Kovacs T, Llorà X, Takadama K, Lanzi P, Stolzmann W, Wilson S (eds) Learning classifier systems. Lecture Notes in Computer Science, vol 4399. Springer, Berlin, pp 291–307

Ben-David A (2007) A lot of randomness is hiding in accuracy. Eng Appl Artif Intell 20(7):875–885

Bernadó-Mansilla E, Garrell-Guiu JM (2003) Accuracy-based learning classifier systems: models, analysis and applications to classification tasks. Evolut Comput 11(3):209–238

Bianchini R, Brown CM, Cierniak M, Meira W (1995) Combining distributed populations and periodic centralized selections in coarse-grain parallel genetic algorithms. In: Proceedings of the international conference on artificial neural networks and genetic algorithms 1995, pp 483–486

Cantú-Paz E (1998) A survey of parallel genetic algorithms. Calculateurs Paralleles 10:141–171

Carvalho DR, Freitas AA (2002) A genetic algorithm with sequential niching for discovering small-disjunct rules. In: Proceedings of the genetic and evolutionary computation conference. Morgan Kaufmann Publishers Inc., San Francisco, pp 1035–1042

Ching JY, Wong AKC, Chan KCC (1995) Class-dependent discretization for inductive learning from continuous and mixed-mode data. IEEE Trans Pattern Anal Mach Intell 17(7):641–651

Clark P, Boswell R (1991) Rule induction with CN2: some recent improvements. In: Kodratoff Y (ed) Machine learning EWSL-91. Lecture Notes in Computer Science, vol 482. Springer, Berlin, pp 151–163

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46

Cohen WW (1995) Fast effective rule induction. In: Proceedings of the 12th international conference on machine learning. Morgan Kaufmann, pp 115–123

De Jong KA, Spears WM, Gordon D (1993) Using genetic algorithms for concept learning. Special Issue Genet algorithms 13(2–3): 161–188

De Jong KA, Potter M, Grefenstette JJ (1995) A coevolutionary approach to learning sequential decision rules. In: Proceedings of the sixth international conference on genetic algorithms. Morgan Kaufmann, pp 366–372

Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7(7):1–30

Domingos P (1995) Rule induction and instance-based learning a unified approach. In: Proceedings of the fourteenth international joint conference on artificial intelligence, vol 2, pp 1226–1232

Fernández A, García S, Luengo J, Bernadó-Mansilla E, Herrera F (2010) Genetics-based machine learning for rule induction: state of the art, taxonomy and comparative study. IEEE Trans Evolut Comput (in press)

Finner H (1993) On a monotonicity problem in step-down multiple test procedures. J Am Stat Assoc 88(423):920–923

Freitas AA (2001) Understanding the crucial role of attribute interaction in data mining. Artif Intell Rev 16(3):177–199

Freitas AA (2003) A survey of evolutionary algorithms for data mining and knowledge discovery. In: Ghosh A, Tsutsui S (eds) Advances in evolutionary computing: theory and applications. Springer-Verlag New York, Inc., New York, pp 819–845

Friedman JH (1996) Another approach to polychotomous classification. Tech. rep. Department of Statistics, Stanford University, Stanford, CA. http://www-stat.stanford.edu/jhf/ftp/poly.ps.Z

Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32(200):675–701

Gallagher M, Bo Y (2005) A hybrid approach to parameter tuning in genetic algorithms. In: Proceedings of 2005 IEEE congress on evolutionary computation, IEEE, vol 2, pp 1096–1103

García S, Fernández A, Luengo J, Herrera F (2009) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput Fusion Found Methodol Appl 13(10):959–977

García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inform Sci 180(10): 2044–2064

Giordana A, Neri F (1995) Search-intensive concept induction. Evolut Comput 3(4):375–416

Giordana A, Saitta L, Bello GL (1997) A coevolutionary approach to concept learning. In: ISMIS '97: Proceedings of the 10th international symposium on foundations of intelligent systems, vol 1325. Springer, London, UK, pp 257–266

Greene DP, Smith SF (1993) Competition-based induction of decision models from examples. Mach Learn 13(2):229–257

Hekanaho J (1997) GA-based rule enhancement in concept learning. In: Proceedings of the third international conference on knowledge discovery and data mining. AAAI Press, pp 183–186

Herrera F, Lozano M (2003) Fuzzy adaptive genetic algorithms: design, taxonomy and future directions. Soft Comput 7(8): 545–562

Ho Y, Pepyne D (2002) Simple explanation of the no-free-lunch theorem and its implications. J Optim Theory Appl 115(3): 549–570

Holden N, Freitas A (2009) Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation. Soft Comput 13(3):259–272

Holland JH, Reitman JS (1977) Cognitive systems based on adaptive algorithms. In: Waterman DA, Hayes-Roth F (eds) Pattern directed inference systems. Academic Press, New York, pp 313–329

Janikow CZ (1993) A knowledge-intensive genetic algorithm for supervised learning. Mach Learn 13(2):189–228

Jiao L, Liu J, Zhong W (2006) An organizational coevolutionary algorithm for classification. IEEE Trans Evolut Comput 10(1): 67–80

Kim MW, Ryu JW (2007) An efficient coevolutionary algorithm using dynamic species control. In: Proceedings of the third international conference on natural computation (ICNC 2007), vol 3. IEEE, Haikou, pp 431–435

Knerr S, Personnaz L, Dreyfus G (1990) Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Fogelman J (ed) Neurocomputing: algorithms, architectures and applications, vol F68. Springer, NATO ASI, New York, pp 41–50

Lanzi PL (2008) Learning classifier systems: then and now. Evolut Intell 1(1):63–82

Liu JJ, Kwok JTY (2000) An extended genetic rule induction algorithm. In: Proceedings of the 2000 congress on evolutionary computation, vol 1, CEC00 (Cat. No. 00TH8512), IEEE, La Jolla, CA, pp 458–463

Marín-Blázquez J, Martínez Pérez G (2009) Intrusion detection using a linguistic hedged fuzzy-xcs classifier system. Soft Comput 13(3):273–290

Mendes RRF, Voznika FDB, Freitas AA, Nievola JC (2001) Discovering fuzzy classification rules with genetic programming and co-evolution. In: Proceedings of the fifth European conference on principles of data mining and knowledge discovery. Lecture Notes In Computer Science, vol 2168. Springer, London. pp 314–325

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, London, UK

Michalski RS (1980) Pattern recognition as rule-guided inductive inference. IEEE Trans Pattern Anal Mach Intell 2(4):349–361

Michalski RS (1983) A theory and methodology of inductive learning. Artif Intell 20(2):111–161
Mitchell TM (1982) Generalization as search. Artif Intell 18(2):203–226
Neri F (2002) Relational concept learning by cooperative evolution. J Exp Algorithm 7:12–37
Neri F, Saitta L (1996) An analysis of the universal suffrage selection operator. Evolut Comput 4(1):87–107
Nojima Y, Ishibuchi H, Kuwajima I (2008) Parallel distributed genetic fuzzy rule selection. Soft Comput 13(5):511–519
Orriols-Puig A, Bernadó-Mansilla E (2005) The class imbalance problem in learning classifier systems. In: Proceedings of the 2005 workshops on genetic and evolutionary computation, GECCO '05. ACM Press, New York, pp 74–78
Orriols-Puig A, Bernadó-Mansilla E (2009) Evolutionary rule-based systems for imbalanced data sets. Soft Comput 13(3):213–225
Orriols-Puig A, Casillas J, Bernadó-Mansilla E (2008) Genetic-based machine learning systems are competitive for pattern recognition. Evolut Intell 1(3):209–232
Provost F, Kolluri V (1999) A survey of methods for scaling up inductive algorithms. Data Min Knowl Discov 3(2):131–169
Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA
Reynolds A, de la Iglesia B (2009) A multi-objective grasp for partial classification. Soft Comput 13(3):227–243
Rissanen J (1989) Stochastic complexity in statistical inquiry theory. World Scientific Publishing Co., Inc., River Edge, NJ
Rivero D, Dorado J, Rabual J, Pazos A (2009) Modifying genetic programming for artificial neural network development for data mining. Soft Comput 13(3):291–305
Rodríguez M, Escalante DM, Peregrín A (2010) Efficient distributed genetic algorithm for rule extraction. Appl Soft Comput (in press)
Stout M, Bacardit J, Hirst J, Smith R, Krasnogor N (2009) Prediction of topological contacts in proteins using learning classifier systems. In: Special issue on evolutionary and metaheuristics based data mining (EMBDM), vol 13. Springer, Berlin, pp 245–258
Tan KC, Yu Q, Ang JH (2006a) A dual-objective evolutionary algorithm for rules extraction in data mining. Comput Optim Appl 34(2):273–294
Tan KC, Yu Q, Ang JH (2006b) A dual-objective evolutionary algorithm for rules extraction in data mining. Int J Syst Sci 37(12):835–864
Venturini G (1993) SIA: a supervised inductive algorithm with genetic search for learning attributes based concepts. In: Machine learning: ECML-93. Lecture Notes in Computer Science, vol 667. Springer, Berlin, pp 280–296
Weilie Y, Qizhen L, Yongbao H (2000) Dynamic distributed genetic algorithms. In: Proceedings of the 2000 congress on evolutionary computation, vol 2. IEEE, La Jolla, CA, pp 1132–1136
Wilcoxon F (1945) Individual comparisons by ranking methods. Biometr Bull 1(6):80–83
Wilson SW (1995) Classifier fitness based on accuracy. Evolut Comput 3(2):149–175
Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco
Yang Q, Wu X (2006) 10 challenging problems in data mining research. Int J Inform Technol Decis Mak 5(4):597–604
Yoon HS, Moon BR (2002) An empirical study on the synergy of multiple crossover operators. IEEE Trans Evolut Comput 6(2):212–223
Zar JH (2007) Biostatistical analysis, 5th edn. Prentice-Hall, Inc., Upper Saddle River, NJ
Zhang X, Luo M, Pi D (2005) Effective classifier pruning with rule information. In: Hoffmann A, Motoda H, Scheffer T (eds) Discovery science. Lecture Notes in Computer Science, vol 3735. Springer, Berlin, pp 392–395