Coevolution for Classification

Catalin Stoean, Ruxandra Stoean, Mike Preuss
and D. Dumitrescu

No. CI-239/08

Technical Report ISSN 1433-3325 January 2008

Secretary of the SFB 531 · Technical University of Dortmund · Dept. of Computer Science/LS 2 · 44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the Technical University of Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

Coevolution for Classification

Catalin Stoean[1], Ruxandra Stoean[1], Mike Preuss[2], and D. Dumitrescu[3]

[1] University of Craiova, A. I. Cuza, 13, 200585, Craiova, Romania
{catalin.stoean, ruxandra.stoean}@inf.ucv.ro
[2] University of Dortmund, Otto-Hahn 14, 44221, Dortmund, Germany
mike.preuss@cs.uni-dortmund.de
[3] Babes-Bolyai University, M. Kogalniceanu, 1B, 400084, Cluj, Romania
ddumitr@cs.ubbcluj.ro

1 Introduction

A data mining field with daily, and sometimes even vital, practical applications, classification has been addressed by many powerful paradigms, among which evolutionary algorithms (EAs) play a successful role. Nevertheless, as evolutionary computation (EC) progresses, there appear new possibilities of developing simpler and yet robust classification techniques.

The aim of this paper is hence to put forward a novel evolutionary classification framework which embodies two contradictory prototypes coming from the state-of-the-art field of coevolution and which has proven to be a viable alternative.

Coevolution between individuals assumes two opposite interactions: cooperative and competitive. Analogously, coevolution for classification assumes two possible and opposed manners of solving the task. Within both approaches, the solution of a classification problem is regarded as a set of IF-THEN conjunctive rules in first order logic. As a consequence, learning will be driven either by the cooperation between rules towards a complete and accurate rule set or by the competition between rules and training samples in the direction of extensive and hard testing on each side.

The paper is organized as follows. The next section introduces a general point of view upon classification. Section three brings an overview on coevolution: The cooperative and competitive archetypes are outlined and explained. Section four describes the proposed manner of approaching classification from the cooperative side, while section five presents the application of the competitive counterpart. Experiments on three data sets, two benchmark and one real-world, are depicted in section six and the paper closes with the concluding remarks.

2 Classification. A Perspective

Classification can assume different characterizations, however this paper regards it from a general point of view. Given $\{(x_i, y_i)\}_{i=1,2,...,m}$, a training set where every $x_i \in R^n$ represents a data sample (values that correspond to a sequence of attributes or indicators) and each $y_i \in 1, 2, ..., p$ represents a class (outcome, decision attribute), a classification task consists in learning the optimal mapping that minimizes the discrepancy between the given classes of data sample and the actual classes produced by the learning machine. Subsequently, the learnt patterns are confronted with each of the test data samples, without an a priori knowledge of their real classes. The predicted outcome is then compared with the given class: If the two are identical for a certain sample, then the sample is considered to be correctly classified. The percentage of correctly labelled test data is reported as the classification accuracy of the constructed learning machine.

The data are split into the training set consisting of a higher number of samples and a test set that contains the rest of the data. The training and test sets are disjoint. In present discussion, the samples that form the training set are chosen in a random manner from the entire specific data set.

The aim of a classification technique is, consequently, to stepwise learn a set of rules that model the training set as good as possible. When the learning stage is finished, the obtained rules are applied on previously unseen samples within the test set.

3 Evolutionary Approaches to Classification

Apart from the hybridization with non-evolutionary specialized classification techniques, such as fuzzy sets, neural networks or decision trees, the evolutionary computation community has targeted classification through the development of special standalone EAs for the particular task.

On a broader sense, an evolutionary classification technique is concerned with the discovery of IF-THEN rules that reproduce the correspondence between the given samples and corresponding classes. Given an initial set of training samples, the system learns the patterns, i.e. evolves the classification rules, which are then expected to predict the class of new examples.

Remark: An IF-THEN rule is imagined as a first-order logic implication where the condition part is made of attributes and the conclusion part is represented by the class.

There are two state-of-the-art approaches to evolutionary classification techniques. The first direction ([11]) is represented by De Jong's classifier that is an evolutionary system which considers an individual to represent an entire set of rules. Rule sets are evolved using a canonical EA and the best individual from all generations represents the solution of the classification problem. The opposite related approach is Holland's classifier system ([10],

[11]). Here, each individual encodes only one conjunctive rule and the entire population represents the rule set. Thus, detection and maintenance of multiple solutions (rules) in a multiple sub-populations environment is required. As a canonical EA cannot evolve non-homogeneous individuals, Holland's approach suggested doubling the EA by a credit assignment system that would assign positive credit to rules that cooperate and negative credit to the opposite.

Another standard method is characterized by a genetic programming approach to rule discovery ([4], [5]). The internal nodes of the individual encode mathematical functions (e.g. AND, OR, +, -, *, <, =) while the leaf nodes refer the attributes. Given a certain individual, the output of the tree is computed and, if it is greater than a given threshold, a certain outcome of the classification task is predicted.

If discussion evolves around those classification evolutionary models that are specifically for coadapted components, then we must refer the abovementioned Holland's classifier system [10] and the REGAL system [6], where stimulus-response rules in conjunctive form were evolved by EAs. In Holland's system, cooperation is achieved through a bucket brigade algorithm that awards rules for collaboration and penalizes them otherwise. In the REGAL classifier, a problem decomposition is performed by a selection operator, complete solutions are found by choosing best rules from each component, a seeding operator maintains diversity and fitness of individuals within one component depends on their consistency with the negative samples and on their simplicity [19].

However, the existing evolutionary classification techniques have quite intricate engines and thus their application is not always straightforward: they use complex credit assignment systems that penalize or reward good rules, as well as very complicated schemas of the entire system.

To the best of our knowledge, there has been no attempt in applying either cooperative or competitive coevolution to classification based on individuals that encode simple conjunctive IF-THEN rules in first order logic.

4 Coevolution. Prerequisites

According to the Darwinian principles, an individual evolves through the interaction with the environment. However, a significant segment of its surroundings is, in fact, represented by other individuals. As a consequence, evolution actually implies coevolution. This interactive process may assume collaboration towards the achievement of a specific mutual purpose, or, on the contrary, competition for the common resources in the spirit of the survival of the fittest. Accordingly, two kinds of artificial coevolutionary systems exist: cooperative and competitive, respectively.

In cooperative coevolution, collaborations between two or more individuals are necessary in order to evaluate one complete potential solution, while in competitive coevolution, the evaluation of an individual is determined by a set of competitions between the current individual and several others.

Coevolutionary algorithms bring an interesting angle of perception upon evolution, as they promote a different manner of fitness evaluation of a candidate solution, which takes into account its relation to the other surrounding individuals. In addition, the coevolutionary evaluation is continuously altered throughout the existence of an individual as a result of various tests.

4.1 Cooperative Coevolution

Cooperative coevolution implies a decomposition of a candidate solution of the problem to be solved into a number of components [18], [25]. Each of these parts is subsequently attributed to a population (species) of an EA. The species evolve independently (although concurrently), while interactions between populations appear only at the moment when fitness is computed. Each individual of a species stands for a part of the solution, therefore, a candidate for each component in turn cannot be evaluated separately from the complementary ones. Hence, when the fitness of an individual is assessed, collaborators from each of the remaining populations are selected in order to form a complete solution. The performance of the established solution is measured and returned as the fitness evaluation of the considered individual.

Evolution is thus directed by the collaboration between species towards the joint goal of assembling a near optimal solution to the problem.

Algorithm 1 simulates the mechanisms of a canonical cooperative coevolutionary method.

Algorithm 1 A canonical cooperative coevolutionary algorithm

```
$t \leftarrow 0$;
for each species $s$ do
  randomly initialize population $\mathrm{Pop}_s(t)$;
end for
for each species $s$ do
  evaluate $\mathrm{Pop}_s(t)$;
end for
while termination condition = false do
  $t \leftarrow t + 1$;
  for each species $s$ do
    select population $\mathrm{Pop}_s(t)$ from $\mathrm{Pop}_s(t - 1)$;
    apply variation operators to $\mathrm{Pop}_s(t)$;
    evaluate $\mathrm{Pop}_s(t)$;
  end for
end while
```

The evolutionary process starts with the initialization of each population. In order to evaluate the initial fitness of each individual, a random selection of collaborators from each of the other populations is performed and obtained solutions are measured and attributed accordingly. After this starting phase, each population is evolved through a canonical EA. Subsequently, the evaluation of a member of one species is performed through its fusion to individuals of the complementary population, which are this point selected through a certain strategy.

The main issue within cooperative coevolution concerns the choice of collaborators. As a result, there are three attributes (parameters) that control this option, whose values have to be properly decided.

1. *Collaborator selection pressure* refers to the manner in which individuals are chosen from each of the complementary populations with the purpose of forming complete solutions to the problem; it must be decided whether we pick the best individual according to its previous fitness score, pick a random individual or use a classic selection scheme.
2. *Collaboration pool size* represents the number of collaborators that are selected from each population.
3. *Collaboration credit assignment* decides the way to compute the fitness of the current individual. This attribute appears solely in the case when the *collaboration pool size* is higher than one. In this situation, the evaluation of an individual consists of several collaborations. Since every such collaboration has its personal score for the objective function, these multiple values must be somehow encapsulated into a single quality value. There are three methods for deciding the final assignment:
   a) *Optimistic* - the fitness of the current individual is the value of its best collaboration.
   b) *Hedge* - the average value of its collaborations is returned as the fitness score.
   c) *Pessimistic* - the value of its worst collaboration is assigned to the considered individual.

Algorithm 2 demonstrates the modality of evaluation of an individual c with respect to the three mentioned attributes. We presume that we have a maximization problem.

In order to evaluate an individual c from a certain population, a number of complete potential solutions are formed according to the chosen collaboration pool size. In order to aggregate a solution, collaborators from each population different from that of c are selected through a certain strategy (collaboration selection pressure). Each solution is evaluated according to the objective function of the current problem. Once all candidate solutions are gathered and assessed, the preferred type for the collaboration credit assignment decides the value that will be returned as the performance of the individual c.

Cooperative coevolution was introduced as an alternative evolutionary approach to function optimization [18]. For this task, one considers as many populations as the number of variables of the function, i.e. each variable represents a component of the solution vector and is separately treated using any

Algorithm 2 Fitness evaluation within cooperative coevolution

```
for each $i = 1, 2, ...,$ collaboration pool size (cps) do
    select one collaborator $d_j$, $j = 1, 2, ...,$ number of species from each population
    different from that of $c$;
    form a complete potential solution;
    compute the fitness $f_i$ of the solution in the terms of the objective criterion;
end for
if Collaboration credit assignment = Optimistic then
    $evaluation \leftarrow max_{i=1}^{cps}(f_i)$;
else
    if Collaboration credit assignment = Pessimistic then
        $evaluation \leftarrow min_{i=1}^{cps}(f_i)$;
    else
        $evaluation \leftarrow avg_{i=1}^{cps}(f_i)$;
    end if
end if
```

type of EA. Several functions with multiple local optima and one global optimum were considered and the cooperative coevolutionary algorithm proved to be effective [18], [25].

The cooperative coevolutionary technique has been recently successfully applied to develop a rule-based control system for agents; two species were considered, each consisting of a population of rule sets for a class of behaviours [20].

4.2 Competitive Coevolution

Within the competitive model [16], the complementary evolution between species is achieved through an inverse fitness interaction process. This implies that success attained on one side is regarded as failure among the individuals of the other side; the latter species will have to react in order to maintain its chances of survival.

Competitive coevolution represents a predator-prey complex: The strong evolutionary pressure determines the prey to defend itself better while, as a response, the predator develops better attacking strategies. This results in a stepwise adaptation and complexity of involved species. Therefore, the competitive interaction between species represents the force that drives evolution forward.

Accordingly [17], one species corresponds to certain tests a solution must satisfy and the other to the potential solutions for the given task. Competition is achieved through encounters between one individual from the tests population and one from the solution species. The two selected individuals are checked against each other and, if the solution passes the test, then the former is rewarded while the latter is penalized; if it fails, credits are assigned in a reverse manner. Moreover, each individual has a history of its encounters which embodies the penalizations/rewards it has received. The fitness of the individual is computed on this basis, as the sum of its most recent behaviours (successes/failures).

An important remark is that, since tests are a priori defined, it is only the population of potential solutions that evolves; the opposite species contains the same individuals (tests) until the end of the evolutionary process. The only fluctuation that appears within the latter population solely regards the ranking of the individuals according to fitness (their satisfiability hardness). It must be however noted that, in certain cases when tests cannot be exhaustively given, the tests population may also evolve.

Canonical competitive coevolution can be described as in Algorithm 3.

Algorithm 3 A canonical competitive coevolutionary algorithm

```
$t \leftarrow 0$;
randomly initialize solutions population $Pop_{Sol}(t)$;
create *history* and evaluate individuals in $Pop_{Sol}(t)$;
create *history* and evaluate individuals in $Pop_{Test}(t)$;
**while** termination condition = false **do**
  $t \leftarrow t + 1$;
  **for** $i$ = 1, 2, ..., *number of encounters* **do**
    select solution from $Pop_{Sol}(t-1)$;
    select test from $Pop_{Test}(t-1)$;
    obtain result from encounter between solution and test;
    update *history* and evaluation of solution according to result;
    update *history* and evaluation of test according to result;
  **end for**
  select two solutions from $Pop_{Sol}(t-1)$;
  apply variation operators to obtain one offspring;
  evaluate offspring;
  $Pop_{Sol}(t) \leftarrow Pop_{Sol}(t-1)$
  insert offspring into $Pop_{Sol}(t)$;
**end while**
```

The initial evaluation of the individuals in both populations is based on the results of random encounters between solutions and tests. When such an encounter takes place, only the current individual is rewarded / penalized without the inverse score attribution for its competitor happening as well.

An evolution cycle is then entered. A predefined number of encounters between solutions and tests takes place. Those opposite individuals that meet are decided following a ranking selection. As a result, the fittest solutions and tests are more frequently involved in such "tournaments": The best performing solutions must prove their superiority more often, while, concomitantly, the algorithm focuses upon the most difficult tests. As soon as the reward/penalization is established for the two selected competitors, their corresponding history is updated: The score of the most recent encounter replaces that of the oldest one and evaluation is revised.

After the considered encounters are finished, a single offspring is created. Two parents are selected according to the same selection scheme and recombination and mutation on the resulting solution are subsequently applied. A personal history of the offspring is created through a number of encounters equal to the defined history length. The tests are again selected according to a ranking scheme. Following such an encounter, only the history of the offspring is modified; unlike a standard encounter between a solution and a test, no simultaneous penalization/reward of the involved test is conducted. This stems from the simple reason that a mediocre offspring might lead to an unreliable change in the behaviour of the considered test. After the offspring is evaluated, it will replace the weakest individual in the solutions population.

During the entire evolutionary process, the tests population suffers no variation.

The two species thus evolve together, through the inverse fitness interaction mechanism: As soon as the potential solutions satisfy certain tests, the latter receive a weaker evaluation score which leads to omission from further selection. As a result, other more difficult tests are subsequently more often selected for tournaments, while the solutions must evolve to adapt to the new requirements that must be fulfilled.

The parameters that are associated with competitive coevolution are the history length of an individual (the number of meetings that provide a measure of its performance) and the number of encounters between solutions and tests within an evolutionary cycle.

The importance of the personal history is manifold [16]. For one, it offers a continuous evaluation of an individual. Then, its partial nature leads to a major decrease in the computational expense of testing a potential solution against all the given tests, while it offers dynamics and keep of pace between the two species.

The competitive paradigm has been applied to a wide range of problems, i.e. path planning [14], constraint satisfaction [13] and classification. As classification is concerned, known techniques involve the evolution of neural networks [12], decision trees [21], cellular automata rules [8], [15] and the use of genetic programming for the problem of intertwined spirals [9]. Again, it has to be stated that to the best of our knowledge, the competitive coevolution between simple IF-THEN rules and the training set has not been achieved yet.

5 Cooperative Coevolution Approach to Classification

The solution to the classification problem is imagined as to be represented by a set of rules that contains at least one rule for each class. Therefore, the decomposition of each potential problem solution into components is performed by assigning to each species (population) the task of building the rule(s) for one certain class [22], [23], [24]. Thus, the number of species equals the number of outcomes of the classification problem. A rule is considered to be a first logic entity in conjunctive form, i.e.:

$$\text{if } (a_1 = v_1) \wedge (a_2 = v_2) \wedge ... \wedge (a_n = v_n) \text{ then class } k$$

where $a_1, a_2, , ..., a_n$ are the attributes, $v_1, v_2, , ..., v_n$ are the values in their domain of definition and $k = 1, 2, ..., p$.

5.1 Training Stage. The Evolutionary Algorithm Behind

Recall the training data set $\{(x_i,\ y_i)\}_{i=1,2,...,m}$, where $x_i \in R^n$ and $y_i \in \{1,$ $2,\ ...,\ p\}$. As the task of the cooperative coevolution technique is to build p rules, one for each class, p populations are considered, each with the purpose of evolving one of the p individuals.

Representation

Each individual (or rule) c in every population follows the same encoding as a sample from the data set, i.e. it contains values for the corresponding attributes, $c = (c_1,\ c_2,\ ...,\ c_n)$. As already stated, individuals represent simple IF-THEN rules having the condition part in the attributes space and the conclusion in the classes space. Within the cooperative approach to classification, an individual will not however encode the class, as all individuals within a population have the same outcome.

Initialization

The values for the genes of all individuals are randomly initialized following a uniform distribution in the definition intervals of the corresponding attributes in the data set.

In case the considered data set is normalized, the values for the genes of the individuals are initialized in the interval [0, 1], again following a uniform distribution.

Fitness Evaluation

In order to measure the quality of a rule, this has to be integrated into a complete set of rules which is to be subsequently applied to the training set. The obtained accuracy reflects the quality of the initial rule. Of course, the value of the accuracy very much depends on the other rules that are selected in order to form a complete set of rules: For a more objective assessment of its quality by means of the accuracy value, the rule is tested within several different sets of rules, i.e. different values for the collaboration pool size are considered.

We will further on denote by *cps* the value for the collaboration pool size parameter. For evaluating an individual from a certain population–that is a rule of a certain outcome–a collaborator from each of the other populations is selected n times according to the collaborator selection pressure choice. Every time, the set of rules is applied to the entire training collection. Obtained accuracy represents the evaluation of the current individual. The fitness of an individual c may be given by the best of the *cps* acquired accuracies (optimistic assignment), by the worst one of them (pessimistic assignment) or by the average of all *cps* accuracies (hedge assignment). Algorithm 4 describes the way evaluation takes place in these cases.

Algorithm 4 Fitness evaluation of an individual c by means of either optimistic, pessimistic or hedge collaboration credit assignment

```
for $i = 1$ to $cps$ do
  $correct_i = 0$;
  select a random collaborator from each population different from that of $c$
  according to the collaborator selection pressure parameter;
  for each sample $s$ in the training set do
    find the rule $r$ from the set of all collaborators that is closest to $s$; found
    class for $s = r$'s class;
    if found class for $s$ = real class of $s$ then
      $correct_i = correct_i + 1$;
    end if
  end for
end for
if $optimistic$ then
  $success = max_{i=1}^{n}(correct_i)$
else
  if $pessimistic$ then
    $success = min_{i=1}^{n}(correct_i)$
  else
    $success = avg_{i=1}^{n}(correct_i)$
  end if
end if
$accuracy = 100 * success$ / number of training samples;
```

In addition to the classical cooperative coevolutionary ones, we propose a novel type of assignment (Algorithm 5). For each sample s in the training set, multiple sets of rules are formed and applied in order to predict its class. All rules within a set have different outcomes. Scores are computed for the sample s, for each of the possible outcomes in the following manner: when a rules set is applied to a sample, a certain outcome is established for it. The score of that outcome is increased by unity. Each of the *cps* sets of rules are applied to s. Finally, the class of s is concluded to be the class that obtains the highest score.

Algorithm 5 Score-based fitness evaluation for an individual c

```
for each sample $s$ in the training set do
    set the score for each possible outcome of $s$ to 0;
end for
for $i = 1$ to $cps$ do
    select a random collaborator from each population different from that of $c$
    according to the collaborator selection pressure parameter;
    for each sample $s$ in the training set do
        find the rule $r$ from the set of all collaborators that is closest to $s$; increase
        the score of $r$'s class for $s$ by one unit
    end for
end for
$success = 0$;
for each sample $s$ in the training set do
    if the real class of $s$ equals the class that had the higher score for $s$ then
        $s$ is correctly classified;
        $success = success + 1$;
    end if
end for
$accuracy = 100 * success$ / number of training samples;
```

Independently of the chosen algorithm for calculating fitness evaluations, the distance between individuals and samples from the data set has to be computed when one decides which rule is *closer* to each sample in the training set. In our conducted experiments, we adopted normalized Manhattan as the distance measure (1). However, other distance measures may be as well employed, depending of the considered problem. Note that the distance does not depend on the class of the sample/individual.

$$d(c, x_i) = \sum_{j=1}^{n} \frac{\mid c_j - x_{ij} \mid}{b_j - a_j} \tag{1}$$

We denoted by $x_i = (x_{i1}, x_{i2}, ..., x_{in})$ a sample from the training set, while by $c = (c_1, c_2, ..., c_n)$ an individual (or rule). a_j and b_j represent the lower and upper bounds of the j-th attribute. As usually the values for the attributes belong to different intervals, the distance measure has to refer to their bounds. Obviously, if data is normalized, the denominator disappears as all attributes have their values between 0 and 1.

In both algorithms 4 and 5, the fitness of an individual is computed as the percent of correctly classified samples from the training set (variable *success* in the algorithms specifies the number of samples that were successfully labelled).

In Algorithm 5, situations may appear when, for a certain sample, there exist more classes that have the same maximum score. In this case, one class has to be decided and it was considered to choose the first one in the order of outcomes. As herein all combinations of rules count in the determination of accuracies, we might state that the new choice of assignment is closer to the classical hedge type.

Selection and Variation Operators

The selection operator presently discussed refers to the selection for reproduction within each population, not to the collaborators selection. We employed fitness proportional selection, but any other selection scheme [3] may be successfully applied.

Intermediate recombination was used – having two randomly selected parents P and Q, the value of a gene i of the offspring O is obtained according to (2).

$$O_i = P_i + R \cdot (Q_i - P_i), \tag{2}$$

where R is a uniformly distributed random number over [0, 1]. The obtained offspring individual replaces the worst of its two parents.

Mutation with normal perturbation was used for the experiments performed in current paper – a value of the gene i of an individual P is changed according to (3).

$$P_i = P_i + R \cdot (b_i - a_i)/ms, \tag{3}$$

where R is a random number with normal distribution, b_i and a_i are the upper and lower bounds of the i-th attribute in the data set and ms is the mutation strength parameter. As the domains for the values of the attributes in the data set have different sizes, we again have to refer to the size of the interval for each attribute when we perturb the values of the genes through mutation. In case the data set is normalized, the way the value of the gene i is modified changes to (4).

$$P_i = P_i + R \cdot ms \tag{4}$$

We cannot imagine any obstacle for using any other recombination or mutation operators [3].

Stop Condition

In our experiments, we set a fixed number of generations for the evolutionary process.

5.2 Cooperative Coevolution Parameters

In order to achieve the optimal configurations for the parameters of the cooperative approach, experiments were carried out as follows.

Concerning the *collaborator selection pressure* attribute, we used random selection, on the one hand, and, on the other hand, we employed a fitness proportional scheme.

All the three types of fitness *assignment* presented in Algorithm 4 together with the one based on scores are tested.

As for the *collaboration pool size*, we varied the number of collaborators in order to find the optimum balance between accuracy and runtime.

5.3 Test Stage. Rules Application

After the stop condition is reached, we dispose of p populations of rules that were evolved against the training set. In order to form a complete set of rules, we have to choose an item from each population. There rules may be selected randomly, the best ones can be considered or a selection scheme can be used. In the last two cases, we take into account the final fitness evaluations of the individuals. It is not always the case that, by selecting the fittest rule from each population, we obtain the best accuracy on the test set. Even if these best rules would give very good results on the training set, they may be in fact not general enough to be applied to previously unseen data.

In the experiments we conducted, for a number of *cps* times, we randomly selected one rule from each population in order to form *cps* complete sets of rules. Each time, the rule set is applied to the test data in a similar manner to the fitness calculation in Algorithm 5 and the classification accuracy is acquired.

6 Competitive Coevolution Approach to Classification

Similarly to the cooperative approach for classification, the aim of the competitive classifier is to construct, based on a training set, a set of rules that model the data and which will be subsequently applied to the test set. Within proposed competitive approach, the population of tests is represented by the samples in the training data, while the other population, that of solutions, will contain only the rules that are to be evolved.

6.1 Training Stage. The Evolutionary Algorithm Behind

Keeping the same notations as in the cooperative approach, the task in this case will be once more to build p rules, one for each class. Consequently, in order to form a solution to the classification problem, a complete set of rules has to be selected from the solutions population, which must therefore contain rules for every outcome.

Representation

The same representation for the individuals (rules) as in the cooperative approach is adopted. The only difference is that herein a better tracking of the class for each rule has to be kept. Since, in the cooperative case, the outcome of each rule was identified with the population class, in present methodology all rules, indifferent of the class they have, lie in the same population. As a result, this time the rules will also encode the class, i.e. $c = (c_1, c_2, ..., c_n \mid k)$, $k = 1, 2, ..., p$.

Initialization

As previously stated, at least p rules have to be obtained. Recombination will take place only between individuals with the same outcome, therefore, the size of the solutions population has to be of at least $2p$ individuals, i.e. differentiating two individuals per class. However, we only state here the minimum size of the rules population; a higher number of individuals would obviously bring a better covering of the search space.

Fitness Evaluation

For each individual and for each sample from the tests population, we have to construct a *history* of the scores they obtained during encounters. The actual fitness evaluation of each individual/sample will be equal to the sum of all scores in their history.

The main question is: How are the scores given? When an encounter between a rule and a sample takes place, the distance (we used the same normalized Manhattan distance as before) between them is computed. The task for the rules is to be as similar as possible to the samples in the training set, therefore the aim is to minimize the distance between them and the samples from the training set with the same outcome. The score that is attached to a rule is given by the negative value of the distance; the maximum score a rule aims to attain is thus 0, meaning that the rule is identical to the sample it encountered.

Conversely, for a sample in the tests population, we attach the actual value of the distance between it and the rule that it met. As rules get closer to a certain pattern of samples, they will subsequently encounter other samples that have larger fitness values (and as a consequence higher chances of being selected for encounters) because they are very different from those rules. Thus, new fitter samples are continuously selected in order to adapt the rules so that they will resemble them too, i.e. evolve the solutions according to a high variety of tests.

Immediately after the initialization of the rules population, the fitness evaluations for both the rules and the samples have to be computed. In this respect, for each rule, a sample with the same outcome as its label is randomly selected and the encounter takes place: This has to be performed for a number of times equal to the history length. Each individual will thus posses a history and, as a result, an evaluation. In a similar manner, for a number of times equal to the history length, each sample from the training set will be considered and random individuals with the same outcome are selected in order to form the encounters that will complete their histories.

After the initial fitness calculation, each time an encounter takes place, solutions and samples are chosen from each population by means of ranking selection. As encounters take place only between solutions and samples with the same outcome, they will occur separately and in turn for each class (Algorithm 6). After the encounter, the new score is added to the history queue and the oldest score is removed, such that the same history length is maintained.

An individual that is obtained after the variation operators is evaluated as follows: for a number of times equal to the history length, a sample with the same outcome as its own is selected using ranking selection and encounters take place. Its fitness may now be computed by summing all the scores in the history. It is then included in the population by replacing the individual with the worst fitness evaluation.

Algorithm 6 The competitive coevolution approach to classification

```
$t \leftarrow 0$;
randomly initialize solutions population $Pop_{Sol}(t)$;
create *history* and evaluate individuals in $Pop_{Sol}(t)$;
create *history* and evaluate individuals in $Pop_{Test}(t)$;
**while** termination condition = false **do**
  $t \leftarrow t + 1$;
  **for** $j$ = 1, 2, ..., *number of classes* **do**
    **for** $i$ = 1, 2, ..., *number of encounters* **do**
      select solution labelled by $j$ from $Pop_{Sol}(t-1)$;
      select test labelled by $j$ from $Pop_{Test}(t-1)$;
      obtain result from encounter between solution and test;
      update *history* and evaluation of solution with -result;
      update *history* and evaluation of test with +result;
    **end for**
    select two solutions with class $j$ from $Pop_{Sol}(t-1)$;
    apply variation operators to obtain one offspring;
    evaluate offspring;
    insert offspring into $Pop_{Sol}(t)$;
  **end for**
**end while**
```

Selection and Variation Operators

In our experiments we only tried the ranking scheme, as it is usually advised in the general framework of competitive coevolution.

As regards the variation operators, the same types of recombination and mutation as in the cooperative approach were employed. Recombination takes place only between individuals within the same class and, therefore, the offspring inherits the outcome of the parents. The mutation operator does not apply to the class gene.

Again, any other variation operators may be successfully tried.

Remark: Variation and replacement take place for every class in turn (Algorithm 6).

Stop Condition

Competitive techniques usually require more iterations than a canonical EA, as in each generation there is only one new descendant that enters the population. However, in our approach for classification, we apply the variation operators for individuals of every class in the same generation, a change that makes several individuals (descendants), i.e. p instances (one for each class), enter the population within that iteration.

The stop condition we used refers to a fixed number of generations, just like in the cooperative approach.

6.2 Competitive Coevolution Parameters

There are two important parameters related to the competitive coevolution technique: The history length and the number of encounters that take place within one generation. The larger the values for both of them, the more accurate the fitness evaluation is for an individual/sample. Unfortunately, together with the raise in the values for either of the two, the runtime of the algorithm also increases.

The value for the number of encounters parameter directly depends on the population size of the two species: If there are many individuals in any of the populations, then a high value for the number of encounters have to be set in order to update the fitness evaluations of a great amount of the individuals.

A value that is too small for the history length parameter could make the fitness evaluation of an individual/sample change too drastically after each encounter and thus the fitness evaluation would not objectively reflect the quality of the individual/sample in contrast to the other population.

6.3 Test Stage. Rules Application

After the evolutionary process stops, one rule for every class is selected and these are applied to the test set. For each sample in the test set, the dissimilarity to each of the rules is computed. The found outcome of the sample is taken from the rule it resembles the most.

7 Experiments

For each of the two coevolutionary approaches for classification, we consider the same data sets for experiments: Two data sets concerning benchmark classification problems coming from the University of California at Irvine (UCI) Repository of Machine Learning Databases[4], i.e. Wisconsin breast cancer diagnosis and iris recognition, are selected for reasons of validation and comparison. Besides, the former is a two-class instance, while the latter represents a multi-class task, which should reveal whether the classification algorithms remain flexible and feasible with some increase in the number of outcomes.

Finally, a real-world data set, courtesy of the University Hospital in Craiova, Romania, is also considered with the purpose of testing and application on an unpredictable environment that is usually associated with raw data. For all grounds mentioned above, the selection of test problems certainly contains a variety of situations that is necessary for the objective validation of the new framework of coevolution in application to classification.

In all conducted experiments, for each parameter setting, the training set is formed of randomly picked samples and the test set contains the rest of the samples. In order to prove the stability of the approaches, each reported average result is obtained after 30 runs of the algorithm.

The experiments section is organized as follows: A small description of each data set is first outlined; it is then continued with the results obtained for both the cooperative coevolution technique and the competitive counterpart. The section closes by undertaking a comparison to accuracies obtained by standard data mining techniques.

7.1 Data sets description

The significant information on the each of the considered classification tasks is given in the following lines.

Breast Cancer Diagnosis

The data set contains 699 observations on nine discrete cytological factors and reflects a chronological grouping of the data. The objective is to identify whether a sample corresponds either to a benign or a malignant tumour. Class distribution is 65.5% for benign and 34.5% for malignant. There are 16 missing values for attribute 6; we replaced them by the average value for that attribute.

Iris Plants Classification

There are 150 samples with three possible classes pertaining to this data set. Each sample consists of four attributes which denote the length and width for

[4] Available at http://www.ics.uci.edu/~mlearn/MLRepository.html

petals and sepals of the iris flowers. Samples are equally distributed among classes.

Hepatic Cancer Early Diagnosis

Hepatocellular carcinoma (HCC/hepatoma) is the primary malignancy (cancer) of the liver that ranks fifth in frequency among all malignancies in the world. In patients with a higher suspicion of HCC, the best method of diagnosis involves a scan of the abdomen, but only at a high cost. A cheap and effective alternative consists in detecting small or subtle increases for serum enzymes levels. Consequently, based on a set of fourteen significant serum enzymes, a group of 299 individuals and two possible outcomes (HCC and non-HCC), we aim to provide an efficient computational means of checking the consistency of decision making in the early detection of HCC at significantly low expense.

7.2 Experiment 1: Cooperative Classification Validation

Pre-experimental planning: In preliminary experiments, we tested different settings for the coevolutionary parameters in order to verify their suitability for the classification problems. However, in these initial experiments, we tested the technique only on the breast cancer and iris data sets.

We observed that there are not major differences between results obtained when different types of collaboration credit assignments are used: Slightly better results appeared to be achieved for the score-based fitness.

As it was expected, when the collaboration pool size value is increased, the runtime of the algorithm also raises. As concerning the results, they also seem to be improved to some extent by the increase of the value for this parameter. The technique had been tested from one up to seven collaborators.

As regards the collaborator selection pressure parameter, we initially employed random selection which drove the coevolutionary process to very competitive results. We then chose the best individual from each population for collaborations, but, surprisingly, the obtained results were worse than in the case of a random selection pressure. The next step we took was that of using a selection scheme for choosing the collaborators. Proportional selection was employed and it proved to be efficient as results were slightly better than those obtained through random selection.

Task: We want to evaluate whether the cooperative approach produces viable results if compared to those obtained by other approaches (information on these is given in subsection 7.4) and how appropriate parameters will be chosen.

Setup: The values for all the parameters were manually tuned. Table 1 contains the values for both the parameters of the EA and the coevolutionary ones. The population size refers to the number of individuals from each of the considered species. We only outlined the values for which we obtained the best average results in 30 runs. In each of the runs, the training and test sets are formed from randomly selected samples.

Table 1. Parameter values for the cooperative coevolution approach

	Breast Cancer	Iris	Hepatic Cancer
Evolutionary Parameters			
Population size	100	150	100
Recombination probability	0.5	0.4	0.5
Mutation probability	0.6	0.6	0.6
Mutation strength	0.01	150	100
Generations	120	200	100
Cooperative Coevolution Parameters			
Collaboration pool size	3	3	5
Collaborator selection pressure	proportional	proportional	proportional
Collaboration credit assignment	hedge	hedge	worst

The only normalized data were the ones regarding breast cancer.

For all considered implementations, we used fitness proportional selection, intermediate recombination - two-parent and one offspring - and mutation with normal perturbation. The offspring replaces the parent only if fitter.

Results: Table 2 outlines the average results obtained by the cooperative approach with the chosen parameter values. For all three data sets, the technique proved to be competitive and stable.

Table 2. Average results after 30 runs for the cooperative coevolution approach

Data set	Average accuracy (%)	Standard deviation (%)
Breast cancer	94.5	1.8
Iris	95.4	3.0
Hepatic cancer	90.5	2.4

Observations: In order to verify how many generations are necessary for the algorithm to reach a constant and good result, we applied the evolved rules to the test set from an early stage of the evolution process. This test was performed solely on the breast cancer data set. In the initial (approximatively 10) generations, the results were very unstable, jumping from 20% to 80% and in-between, reaching then a certain stability of about 80% accuracy and growing slowly, but constantly. In the final generations, there are only minor modifications of the test accuracy of up to one percent. However, the results greatly depend of the way the training/test sets were generated as there exist cases when the accuracy starts with 80% even from the early stages of the evolutionary process.

It has to be noted that there is not a very strong dependence between the EA parameters and obtained results as very competitive accuracies are obtained for a large scale of their values.

Concerning the coevolution parameters, changes within the collaboration credit assignments do not bring vital modifications to the final average results: The differences between various settings as regards the final accuracies do not overcome one percent.

As previously stated, the collaboration pool size parameter directly influences the runtime of the program that implements the approach; the final test accuracy is also affected by the increase in this value, but a balance has to be established between runtime and accuracy. For the two test problems we considered in the pre-experimental stage, better results were obtained for an odd number of collaborators. Another important observation is that when a certain threshold for the collaboration pool size parameter is surpassed no further gain in accuracy is reached. We generally achieved the best results when three collaborators were considered.

The cooperative parameter that brought the most important change in the final result of the algorithm was the collaborator selection pressure. A selection scheme is preferred to a random selection; selecting only the best collaborator seems to be the worst of the three choices.

Discussion: The proposed classification approach based on cooperative coevolution provides very accurate results in a relatively small amount of time. For instance, on the breast cancer data set, the runtime lasts from 7 seconds per run when the collaboration pool size is one, up to 24 for three collaborators and to around 37 seconds when five collaborators are used. Note that for experiments, we used a computer with the following characteristics: Pentium IV CPU 3.0 GHz and 1 GB of RAM.

7.3 Experiment 2: Competitive Classification Validation

Pre-experimental planning: The same two benchmark data sets from the UCI repository were used for preliminary experiments. The first observation in these tests refers to the high amount of time necessary for the algorithm to run: The explanation lies in the fact that the tests population is very large and, at the beginning of the evolutionary process, all samples are evaluated. This means that for each sample, for a number of times equal to the history length, an individual is selected and an encounter takes place between the two, assigning a score to the sample.

We also noticed at this stage the importance of the two competitive coevolution parameters: history length and the number of encounters. They also significantly influence the runtime of the program that implements the algorithm. However, a more exact evaluation of an individual or sample is obtained if the history length value is large and, on the other hand, a high value for the number of encounters updates the evaluations of individuals/samples.

Task: It will be investigated if the competitive classification technique can perform as well as the cooperative approach.

Setup: The values for the parameters were manually tuned, like in the cooperative case. Found values are indicated in Table 3.

In the current experiment, in order to enhance the speed of the algorithm, we tried to reduce the population size as much as possible: Less individuals means they will have more encounters with samples from the other species and their fitness will be updated very often.

None of the considered data sets was normalized.

Table 3. Parameter values for the competitive coevolution approach

	Breast Cancer	Iris	Hepatic Cancer
Evolutionary Parameters			
Population size	100	50	100
Mutation probability	0.5	0.5	0.4
Mutation strength	8	1	10
Generations	100	300	150
Competitive Coevolution Parameters			
History length	30	30	30
Number of encounters	20	30	20

Results: The average results that were obtained after 30 runs by applying the competitive classification technique are illustrated in Table 4.

Table 4. Average results after 30 runs for the competitive coevolution approach

Data set	Average accuracy (%)	Standard deviation (%)
Breast cancer	92.9	2.9
Iris	91.1	3.5
Hepatic cancer	84.7	3.4

Observations: The average results show that the competitive approach is significantly weaker than the cooperative one. Not only the final results prove that this approach is much poorer than the cooperative one, but there is also a great difference as concerns runtime: To make an objective comparison, we measured the average runtime for the same breast cancer data set, by using the competitive approach with the parameters indicated in Table 3; the obtained value was of around 480 seconds, that is almost 13 times slower than the cooperative approach with 5 collaborators.

The standard deviation of the results is also significantly higher, which indicates the fact that this technique is not as stable as the cooperative approach. Nevertheless, it has to be stated that in an objective judgement, there are not too many perturbations that appear in 100, or even 300 generations, since only two individuals per class recombine during one generation and mutation is applied solely to the obtained offspring. To conclude, at least 1000 generations would probably be necessary for the variation operators to considerably change the population. That would, on the other hand, slow down the program even more.

Discussion: The obtained results and the large runtime of the competitive technique indicate the cooperative approach as much more viable as compared to the state-of-the-art techniques for classification. Note however that this is only the first time the competitive approach for classification is proposed and we believe that it represents a good starting point, as there is definitely potential within this technique as well. To outline some ideas for future research concerning the competitive technique for classification, perhaps a preprocessing technique step could be first applied to the training data in order to substantially reduce them. In conjunction with that (or by itself), we presume that employing a chunking technique in order to pick only small parts from the training set and use them as the static species could significantly improve runtime (maybe even the accuracy). Then, after the rules are specialized on the selected samples, the tests species could bring new ones, while the dynamic population of rules could resume the evolution.

An important enhancement could be brought if the very good rules that are evolved at a certain point could be blocked for further modifications: Make one such individual a *tabu rule* and maybe move it in a rules archive that will be applied when the termination condition is reached. In the way the technique is now built, these good rules have the highest chances to be selected over and over again, therefore modified many times, maybe for the worse.

In the end of the evolutionary run, the best rule of each class in the final population was taken and the entire formed set was applied to the test data. Obviously, a different way of choosing the rules could be imagined, e.g. take several rules for one class or apply an archive variant as suggested above.

7.4 Comparison to Standard Data Mining Approaches

Comparison of obtained results can be made only for the breast cancer and iris data sets, since they are public benchmark problems. The resulting rules of the hepatic data set were however confronted with the specialized opinion of the physician and were found to be consistent with the medical diagnosis.

A summary of best and worst accuracies in literature concerning the two considered UCI data sets is presented in Table 5. These results come from surveys on several canonical data mining techniques in [1], [2] and [7]. Comparison cannot be objective, however, as outlined methods either use different sizes for the training/test sets or other types of cross-validation and number of runs or employ various preprocessing procedures for feature or sample selection.

Table 5. Comparison to accuracies of data mining techniques reviewed in [1], [2] and [7]

Task	Worst reported accuracy	Best reported accuracy
Breast cancer	94.2%	97.2%
Iris	93.47%	96.31%

The highest and lowest obtained accuracy for the breast cancer diagnosis problem are reported in [1], [2]. However, the authors used 10-fold cross-validation and removed the samples that had missing values.

On the other hand, the two results for iris are given in [7]; a similar way of selecting the training and test sets was used. The difference to our approach is that 80% of the samples from the Iris data were used for training and the rest (less samples than in our approach) for testing and that average accuracies are obtained after 500 runs.

8 Concluding Remarks

The two types of coevolution within which species either cooperate or compete are herein proposed as tools for solving classification tasks. The cooperative approach, probably even due to the fact that it has been more extensively tested, proved to be more much efficient than the competitive one, as concerns both the accuracy and runtime. An important drawback of the cooperative technique is however the fact that that the number of populations must increase with the number of classes of the problem.

The presented coevolution framework brings a natural manner of targeting classification, with a simple and concise representation and a straightforward fitness assignment as an advantage over existing evolutionary possibilities.

We have consequently assembled this work with the purpose of showing that coevolution can provide new insights and successes into the demanding and crucial field of classification.

References

1. Bennett K. P., Blue J. (1997) A Support Vector Machine Approach to Decision Trees, R.P.I Math Report No. 97–100 Rensselaer Polytechnic Institute Troy, New York
2. Duch W., Adamczak R., Grabczewski K., Zal G. (1999) Hybrid neural-global minimization method of logical rule extraction, Journal of Advanced Computational Intelligence, Volume 3, Number 5: 348–356
3. Eiben A. E., Smith J. E. (2003) Introduction to Evolutionary Computing. Springer–Verlag, Berlin Heidelberg New York
4. Freitas A. A. (2002) A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery, Advances in Evolutionary Computation, 819–845
5. Freitas A. A. (2002) Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer–Verlag, Berlin Heidelberg
6. Giordana A., Saitta L., Zini F. (2000) Learning Disjunctive Concepts by Means of Genetic Algorithms, Proceedings of the 11th International Conference on Machine Learning: 96–104
7. Iacus S., Porro G. (2006) Missing Data Imputation, Classification, Prediction and Average Treatment Effect Estimation via Random Recursive Partitioning, UNIMI - Research Papers in Economics, Business, and Statistics
8. Juille H., Pollack J. B. (1996) Dynamics of Co-evolutionary Learning, Proceedings of the 4th International Conference on Simulation of Adaptive Behavior: 526–534
9. Juille H., Pollack J. B. (1998) Coevolutionary Learning: a Case Study, Proceedings of the 15th International Conference on Machine Learning: 251–259
10. Holland J.H., (1986) Escaping Brittleness: The possibilities of General Purpose Learning Algorithms Applied to Parallel Rule-Based Systems, Machine Learning 2:593–623
11. Michalewicz Z. (1996) Genetic Algorithms + Data Structures = Evolution Programs. Springer–Verlag, New York
12. Paredis J. (1994) Steps towards Co-evolutionary Classification Neural Networks, Artificial Life IV: 102-108
13. Paredis J. (1994) Coevolutionary Constraint Satisfaction, Proceedings of the 3rd Conference on Parallel Problem Solving from Nature, LNCS, 866: 46-55
14. Paredis J., Westra R. (1997) Coevolutionary Computation for Path Planning, Proceedings EUFIT 1997, 1: 394–398
15. Paredis J. (1997) Coevolving Cellular Automata: Be Aware of the Red Queen! Proceedings of the 7th International Conference on Genetic Algorithms: 393–399
16. Paredis J. (1999) Constraint Satisfaction with Coevolution, In: New Ideas in Optimization, Corne D., Dorigo M., Glover F. (Eds.), McGraw-Hill, London: 359–366
17. Paredis J. (2002) Coevolutionary Algorithms, www.evalife.dk/ToEC2002/publications/paredis_coevo_survey_HEC.ps
18. Potter M. A., De Jong K. A. (1994) A Cooperative Coevolutionary Approach to Function Optimization, Proceedings of the 3rd Conference on Parallel Problem Solving from Nature: 249–257
19. Potter M. A., De Jong K. A., (2000) Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents, Evolutionary Computation, 8(1): 1-29

20. Potter M. A., Meeden L. A., Schultz A. C. (2001) Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergence of Specialists, Proceedings of the 17th International Conference on Artificial Intelligence: 1337-1343
21. Siegel E. V. (1994) Competitively Evolving Decision Trees Against Fixed Training Cases for Natural Language Processing, Advances in Genetic Programming, 409–423
22. Stoean C., Dumitrescu D., Preuss M., Stoean R. (2006) Cooperative Coevolution for Classification, Bio-Inspired Computing: Theory and Applications 2006: 289–298
23. Stoean C., Preuss M., Dumitrescu D., Stoean R. (2006) Cooperative Evolution of Rules for Classification, IEEE Post-proceedings Symbolic and Numeric Algorithms for Scientific Computing 2006: 317-322
24. Stoean C., Stoean R., El-Darzi E. (2007) Breast Cancer Diagnosis by Means of Cooperative Coevolution, 3rd ACM International Conference on Intelligent Computing and Information Systems: 493–497
25. Wiegand R. P. (2003) Analysis of Cooperative Coevolutionary Algorithms, PhD Thesis, George Mason University, Virginia