Feature Selection for a Cooperative Coevolutionary Classifier in Liver Fibrosis Diagnosis☆

Catalin Stoean[a], Ruxandra Stoean[a], Monica Lupsor[b], Horia Stefanescu[b], Radu Badea[b]

[a] *Department of Computer Science, University of Craiova, A. I. Cuza Str., No. 13, 200585, Craiova, Romania, Telephone number: +40251413728*
[b] *Department of Ultrasonography, 3rd Medical Clinic, University of Medicine and Pharmacy Cluj - Napoca, Croitorilor Str., No. 19-21, 400162, Cluj-Napoca, Romania, Telephone Number: +40264532525*

Abstract

This paper presents an automatic tool capable to learn from a patients data set with 24 medical indicators characterizing each sample and to subsequently use the acquired knowledge to differentiate between five degrees of liver fibrosis. The indicators represent clinical observations and the liver stiffness provided by the new, non-invasive procedure of Fibroscan. The proposed technique combines a hill climbing algorithm that selects subsets of important attributes for an accurate classification and a core represented by a cooperative coevolutionary classifier that builds rules for establishing the diagnosis for every new patient. The results of the novel method proved to be superior as compared to the ones obtained by other important classification techniques from the literature. Additionally, the proposed methodology extracts a set of the most meaningful attributes from the available ones.

Key words: classification, evolutionary algorithm, cooperative coevolution, feature selection, hill climbing, liver fibrosis staging

2000 MSC: 68T05, 68T20

1. Introduction

Hepatic fibrosis, the main pointer for the materialization of a liver disease within chronic hepatitis C, can be measured through several stages. The correct evaluation of its degree based on non-invasive procedures is essential since, in case of a significant level, an immediate antiviral therapy has to be administrated to the patient. Recent medical practice for treating the mentioned issue has moved from the standard liver biopsy, which is both invasive and also deceptive, to either imaging [1] or biochemical testing [2]. One of the most recent, non-invasive procedures for determining the stiffness of the liver tissue is the Fibroscan (Echosens, Paris, France) and its accuracy is impressive. Nevertheless, the complex interaction between its stiffness indicator and the other biochemical and clinical examinations with the purpose of even more rigorously determining the degree of liver fibrosis is hard to be manually discovered.

Such support for the decision-making based on the values that the patients have obtained both from the medical exams and the result from the non-invasive technique (Fibroscan, imaging or biochemical methodologies) could be addressed by an appropriate learning method from artificial intelligence with the aim of discovering the corresponding degree of fibrosis. In this respect, some papers have arisen recently, employing classical techniques like naïve Bayes and k-nearest neighbor [3] or more modern ones like neural networks [4] or support vector machines [1], [2].

Aiming to evaluate stiffness together with standard exams for predicting the liver fibrosis stage, present paper puts forward a novel approach that is comprised of two parts, each based on evolutionary algorithms (EAs): the first, a hill climbing algorithm, has the goal of finding a promising set of features for the second one, the cooperative coevolutionary algorithm (CCEA), which is the main engine of the proposed evolutionary method. The CCEA purpose is to *evolve* a set of rules with regard to the training samples that will be subsequently applied for determining the degree of hepatic fibrosis for new cases.

The aims of the current paper can be enumerated as follows:

- To offer a fast and reliable way for an automated prediction of liver fibrosis stages.
- To provide an accurate diagnosis.
- To highlight which are the most important features and also how they can be connected in order to give more accurate results.

The paper is organized as follows: next section presents the problem to be solved, that of diagnosing the degree of liver fibrosis based on several medical non-invasive indicators; the features are outlined and other previous research on the particular data set is discussed. After a brief introduction to evolutionary computation, section 3 presents the CCEA, its approach for classification, and the accompanying hill climbing algorithm used for feature selection. Experimentation is described in section 4 and the last part contains the final conclusions and ideas for future research.

2. The Classification Problem: Liver Fibrosis Staging

The chronic hepatitis C data set employed in this paper comes from the 3rd Medical Clinic, University of Medicine and Pharmacy, Cluj-Napoca, Romania, and consists of 722 samples, each described by 24 indicators, with a small number of missing values.

The medical attributes that are chosen to provide information that triggers a certain degree of liver fibrosis are outlined in Table 1. The first one is the stiffness indicator from the Fibroscan, while the others represent standard hematological and biochemical exams that are required in a patient with chronic hepatitis C. The five possible degrees of fibrosis have the following meaning and number of representatives:

- F0 (no fibrosis) – 29 examples;
- F1 (portal fibrosis without septa) – 227 examples;
- F2 (portal fibrosis and few septa) – 164 examples;
- F3 (numerous septa without cirrhosis) – 87 examples;
- F4 (cirrhosis) – 215 examples.

——————————— Table 1 about here ———————————

There has been a number of previous approaches to address the automated decision-making in the non-invasive evaluation of liver fibrosis using only the stiffness of the liver attribute. It is mostly ROC analysis that is employed for this purpose [5], [6], [7], [8]. There is also a study concerned with its application to a part of the medical data used within current work

[9], which will be next discussed more elaborately. Only 324 samples (of the 722 used within current paper) were considered, but the goal was only to differentiate between two classes like F0 vs. F1234 (87.74% obtained correct differentiation), F01 vs. F234 (78.62% accurate distinction), F012 vs. F34 (84.91% performance), F0123 vs. F4 (89.94% accuracy of prediction). [9] has made however no attempt to assign a patient to exactly one of the five possible degrees.

On the other hand, with the goal of differentiating among the 5 distinct liver fibrosis stages, the application of neural networks, naïve Bayesian classification and the k-nearest neighbor algorithm has also been addressed for a primary database that included 125 patients and 26 attributes (stiffness from the Fibroscan and other clinical indicators), while investigating the importance of the stiffness indicator within the discrimination [3]. On the same data set, [4] employed a naïve Bayes classifier and a probabilistic neural network model; these were also used in conjunction with a feature selection algorithm, but there was gain only in runtime, without improving the accuracy. Overall, the best reported accuracy of prediction on this smaller data set was of 70%.

In comparison to earlier reported work, the proposed CCEA, in conjunction with the hill climbing algorithm for feature selection, aims to demarcate among the 5 different classes, supply information about the attributes that are most important and, moreover, even provide some automatically detected relations between the indicators that were considered for classification.

3. Cooperative Coevolution for Classification

A rough introduction to the evolutionary computation field precedes the presentation of the main aspects of the general cooperative coevolution framework. The bases are then set for the presentation of the CCEA approach for classification. In order to decrease the problem dimensionality and, at the same time, achieve some gain in classification accuracy, a hill climbing algorithm is used to choose the most significant attributes, prior to the application of the cooperative coevolution approach.

3.1. Evolutionary Computing

The power of evolution in nature is obvious since diverse species that make up our world manage to survive and adapt in their own niches, sometimes in very cruel environments. Thus, it is not surprising that some researchers chose natural evolution as a source of inspiration for problem solving.

A common artificial evolutionary process starts with a population of individuals that are randomly generated. Based on the fitness values, some of the individuals are *selected* to be the parents of the population in the next generation; descendants are obtained by applying *recombination* and/or *mutation* to the previously chosen individuals. Recombination takes place between two or more individuals and one or more *descendants* (or *offspring*) are obtained; descendants borrow particularities from each of the parents. When mutation is applied to a candidate solution, the result is one new candidate that is usually only slightly different from its parent. After applying these *variation operators* (mutation and recombination), a set of new individuals is obtained that will fight for survival with the old ones for a place in the next generation; the candidate solutions that are fitter are advantaged in this competition. The evolutionary process resumes and usually stops after a predefined computational limit is reached. A general scheme of an EA is presented in pseudocode in Algorithm 1 [10].

Algorithm 1 Pseudocode of an evolutionary algorithm

```
Require: A search/optimization problem
Ensure: The best obtained individual(s)
  Initialize population with random candidate solutions;
  Evaluate each candidate;
  while termination condition is not satisfied do
    Select parents;
    Recombine pairs of parents and obtain offspring;
    Apply mutation to offspring;
    Evaluate resulting offspring;
    Select individuals that will form the next generation;
  end while
```

3.2. Cooperative Coevolution

According to the Darwinian principles [11], an individual evolves through the interaction with the environment. An important part of its environment is, however, represented by other individuals so, as a consequence, evolution can be viewed as coevolution. This could be understood in two opposite ways:

on the one hand, the individuals could collaborate for the same purpose and thus construct the solution together or, on the contrary, they could compete against each other for the same resources. As a consequence, two types of artificial coevolutionary systems exist: cooperative and competitive, respectively.

Within cooperative coevolution, a solution for the problem to be solved is created through the unification of several individuals that evolve in different populations, while within the competitive approach, a candidate solution is evaluated based upon the results that it obtains after a set of competitions with several other individuals. We will further on discuss only the collaborative paradigm since it powers the proposed classifier.

The CCEA [12] requires that any candidate solution of the problem at hand is decomposed into several subsolutions and each of these separate components are evolved by a distinct EA. The only interaction between the different populations takes place when an individual is evaluated: its quality (or adaptation to the environment) cannot be measured separately because it represents only a part of a potential solution, but individuals from all the other populations have to be selected and brought together in order to construct a complete candidate solution that can be assigned a fitness score. The numerical value that stands for the evaluation of the latter is assigned as the fitness of the initial individual. There are many possibilities to choose the collaborators when computing the fitness evaluation [12]. Nevertheless, in order to keep the complexity of the problem low, in the current paper only one collaborator from each class is considered for the evaluation and each such collaborator is chosen following a random selection.

Algorithm 2 simulates the mechanisms of a standard CCEA. Each population is evolved by a different canonical EA.

3.3. The CCEA Approach for Classification

Recent research showed that cooperative coevolution can be successfully applied for classification purposes [13], [14]. A short account of a general classification problem is therefore outlined and followed by the description of the way the cooperative coevolution approach has been adapted for this type of problem.

3.3.1. Classification Problem Description

A set of n objects, each described by m attributes, $a_1, a_2, ..., a_m$, plus a decision value d_i, $i \in 1, 2, ..., k$, is divided into two subsets, one used for training an automated method and the other for testing its efficiency. The training and test sets are disjoint. Starting from the training set, the method has to automatically *learn* the relationship between the values of the attributes and the possible outcomes. Then, when applied to the test set, the technique finds one output for each object and the detected value is checked against the actual outcome. The prediction accuracy is then computed as the percent of objects from the test set that are correctly classified by the method.

In order to cross-validate the given data and verify the generalization ability of probabilistic methods for the classification task, it is customary to try several random ways of splitting the samples into training and test sets, apply the technique for these configurations and compute an average over all obtained accuracies.

Algorithm 2 A canonical cooperative coevolutionary algorithm

```
Require: A search/optimization problem
Ensure: The best obtained individual(s)
  $t \leftarrow 0$;
  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 is not satisfied 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
```

3.3.2. Cooperative Coevolution for Classification

Within a possible evolutionary treatment of a classification problem [13], [14], the aim is to perform a generation of rules for each class. Rules are first randomly created, subsequently tested against the training set and continually adjusted in order to increase the training accuracy they provide.

As stated in the previous subsection, in order to solve a problem by means of a cooperative coevolution engine, a potential solution has to be decomposed into several items and every component is treated by a separate EA. For the classification problem, each population may evolve rules for a certain class and thus the number of species equals the number of outcomes of the classification problem. A complete candidate solution may therefore represent an entire set of rules that optimally associates the indicators with the fibrosis levels.

Each rule has the same representation as the samples in the data set to be classified, i.e. it has the same number of features and one outcome. The value for each attribute is initially randomly generated following a uniform distribution between the definition bounds for that specific feature, that is between the minimum value that exists for that attribute in the data set and the maximum one. Individuals can be interpreted as simple IF-THEN rules having the condition part in the attributes space and the conclusion in the classes space (1). However, an individual does not specifically encode the decision class, as this is implicit from the population to which the rule belongs.

$$\text{IF } a_1 = v_1 \text{ AND } a_2 = v_2 ... \text{ AND } a_m = v_m \text{ THEN } d_i, i \in \{1, 2, ..., k\} \tag{1}$$

In order to evaluate an individual (rule) from a certain population, a complete set of rules has to be formed, in the sense that one rule from each of the other classes has to be selected. The entire rule set is then applied to the training data: for every sample, similarities between each rule in the set and the current object are computed and the found class is concluded to be the one of the rule that is closest. A prediction accuracy over all samples is obtained and assigned as the fitness of the rule to be evaluated.

In order to calculate how close the current rule is to a sample from the training/test set, a distance measure has to be employed. In the experiments conducted within the current paper, it is the Manhattan distance (2) that is considered in this respect; x and s represent an individual and a sample, respectively, and x_i is the i-th component of the potential solution. However, there is obviously no obstacle in using any other desired distance measure.

$$d(x,s) = \sum_{i=1}^{n} \mid x_i - s_i \mid \tag{2}$$

At the end of the run, CCEA provides several population of rules, each one with those prototypes that define a certain class. In order to apply these rules for samples in the test set, individuals are selected once more from each population, objects are labeled accordingly and the prediction accuracy is achieved.

3.4. Feature Selection by Means of a Hill Climbing Algorithm

It is customary that techniques for automated diagnosis in biology and medicine [15], [16] make an a priori use of some mechanism of selecting the most relevant indicators in the data set. The reason lies in the assumption that some attributes might only hinder the search for the accurate solutions or even block the entire method under the curse of dimensionality.

One of the commonly used feature extraction mechanisms, Principal Component Analysis (PCA), is thus also employed in our study. This leads to a sizeable reduction of the data dimensionality from 24 to only 6 assembled attributes. When applying CCEA to the new data, the results are improved by only 1.4%, as opposed to those achieved through the direct application on the original data. It has to be mentioned that there is, however, a major improvement in runtime.

Therefore, a different way to considerably improve the results accuracy has to be discovered. In this respect, the classification method can be aided by an incorporated dynamic feature selection mechanism. In order to make the additional procedure efficient, a limited number of applications of the CCEA technique is desired. For that reason, a hill climbing algorithm is used with the purpose of choosing the attributes that make the CCEA perform more beneficially.

An individual is represented as binary, has a number of genes equal to the number of features that exist in the data set and a value of 1 means that the corresponding attribute is taken into consideration, while 0 that the attribute is skipped. An individual is randomly constituted and the selected attributes are considered for the CCEA. The algorithm generates rules based on the newly defined training set and then applies them to the test set; an accuracy is obtained and that value represents the fitness evaluation of the hill climber. Perturbation is then applied for the individual, a new climber is obtained, it is evaluated and, if fitter, it replaces the previous one.

4. Experimental Results

In order to closely follow the conducted tests and make the entire experimentation clear, the section is divided into several parts: early observations are mentioned, the task is accurately defined, the experimental preparation is described and finally the results are disseminated, following the order in which they were pursued by the authors.

Pre-experimental planning: As regards the fibrosis problem, first of all, the issue of missing data had to be resolved: in this respect, the common method of mean substitution per class is chosen for their imputation. Second of all, within the initial experiments, we tried CCEA on the entire data set directly, without any prior feature selection. However, after some shallow tuning of the CCEA parameters, the average results over 30 repeated runs only reached 51.8% test accuracy. This fact determined us to support CCEA by an additional mechanism that would preprocess the data set. PCA came as an alternative, it compressed the data from 24 to only 6 dimensions, but the accuracy only reached 53.2%.

In order to eliminate the noisy indicators, our next choice was to use a genetic algorithm for selecting the attributes; however, because a CCEA run may take up to 10 seconds, a hill climbing algorithm appeared as the more proper choice for this assignment.

The individuals that encode the rules within CCEA are randomly generated using a uniform distribution; it has been also tried to include some samples from the training set among the rules of each class (up to 60% of the population) in order to help each of them evolve faster to the best prototype. Unexpectedly, this conducted the search to even worse results than the populations with random rules, so this idea was abandoned.

Task: A hill climbing algorithm is used to dynamically pick the proper attributes from the data set and then the CCEA is employed to classify the selected data into five different classes. Our task is to obtain an accuracy that is better than the ones acquired by other state of the art classification methods like support vector machines (SVMs) or neural networks (NNs) applied on both the raw data set or on a PCA dimensionally reduced one.

Experimental Setup: The hill climber starts from a randomly generated binary configuration of 24 genes (the number of attributes from the data set), the indicators that have a corresponding value of 1 are considered as selected and the CCEA is applied 30 times to the data set referring only the chosen features. The average accuracy obtained over the 30 repeats represents the fitness evaluation of the hill climber. In each of the 30 runs, the training and test sets are randomly chosen in order to have a more objective evaluation.

Mutation is then applied and the genes values can be flipped depending on the probability set for this purpose. The generated offspring is evaluated and, if fitter, it replaces the parent hill climber. The process continues and, if there is no improvement in fitness for a number of iterations, a new individual is generated and the process restarts. A fixed budget of fitness evaluations is set for the hill climber as a stop condition for the algorithm. The values for all parameters involved in maneuvering the hill climber can be found in the first line of Table 2. All values are empirically set. The mutation probability is decided to be small in order to have only slight steps from one configuration to another (about 2-3 bits are changed when mutation occurs). The number under generations refers to the number of iterations that pass without improvement (necessary to restart the hill-climber), while the number under fitness evaluation calls regards the stop condition for the hill climbing algorithm.

All samples in the data set are normalized, therefore the genes for the individuals that encode the rules within the CCEA are generated in the interval $[0, 1]$. The established evolutionary parameters of the CCEA are given in the second line of Table 2. Each involved population is set to a size of 50 and, since there exist five classes, there are 250 individuals evolved overall.

——————————— Table 2 about here ———————————

The utilized genetic and variation operators within CCEA are the ones commonly employed for the EA application to real-valued domains [10]:

- Binary tournament selection, where every two individuals fight to enter the parents pool based on their fitness values.

- Intermediate recombination with probability P_r, so that the genes of an offspring O are obtained from two parents P and Q from the same species according to (3), where R is a uniformly distributed random number over [0, 1].
$$O = P + R \cdot (Q - P). \tag{3}$$

- Mutation with normal perturbation with probability P_m. A gene of an individual X selected to be modified through mutation is changed according to (4); MS and $N(0, 1)$ represent the mutation strength and a normally distributed random variable with expected mean 0 and variance 1, respectively.

$$X' = X + MS * N(0, 1). \tag{4}$$

A high probability is set for recombination with the purpose of bringing homogeneity within each species, while for the mutation probability a small value is chosen in order to gradually explore the search space and avoid the rapid change of the entire genotype. As the values of the genes are set between 0 and 1, the mutation strength MS is set to a small number that allows the search to perform fine tuning. A number of 80 generations is set as a stop condition, as it has been observed during pre-experimentation that it provides sufficient time to reach an (almost) optimum. All parameter values are established as a result of manual tuning; there were however other configurations that yielded good results, however the one proposed herein also showed consistency as regards the standard deviation.

The hill climber uses 1000 fitness evaluations and begins by generating a random configuration which is evaluated, suffers mutation and, if changed, is re-evaluated. If fitter, it replaces the previous hill-climber. If there is no improvement for 20 generations, a new individual is generated and the process is restarted, but resuming the count of the fitness evaluations.

In order to validate the results against the ones obtained by other artificial intelligence methods, the state-of-the-art SVMs and NNs are chosen accordingly. Since there are no previous applications of the techniques to the same data set, it is decided to use the implementations within the R software and environment. Three R packages are necessary for the purpose, $e1071$ for both implementations, *kernlab* for SVMs and *nnet* for NNs. For the former, the pre-experiments demonstrated that a linear kernel is preferable to a polynomial of higher degree or radial, as it achieved the best performance. The chosen implementation presumes an implicit one-against-one separation. For the NN method, the default parameters are changed as the necessity is indicated by some prolonged parameter tuning: the number of units in the hidden layer is raised to 30, the rang is set to 0.1 and the maximum number of iterations is increased to 500. 30 random trials are executed by both the SVM and NN in the same manner as for the CCEA, i.e. the training and test sets are randomly chosen in each run.

In addition, the two methods are applied to the reduced data attained by the PCA mechanism. In the following, when the two techniques refer the transformed data, they are denoted by SVM+PCA and NN+PCA, respectively.

Results and Visualization: Depending on the selected attributes, the best accuracy result obtained as the average over 30 repeated runs of random cross-validation of the CCEA is of 62.11% correctly classified patients. Over the 1000 fitness evaluations of the hill climber, the average accuracy is of 55.93%, while the worst test accuracy is of 47.92%. The individual that yields the best obtained accuracy only selected 9 attributes out of the 24 available and these are the following: stiffness, triglycerides, HDL cholesterol, aspartate aminotransferase, gama glutamyl transpeptidase, alkaline phosphatase, prothrombin index, prolonged activated partial thromboplastin time and hematocrit.

For some configurations of the training-test sets and with the attributes mentioned above, the test accuracy even reaches above 65% correctly classified samples. This is the case with the run of CCEA that is represented in Figure 1, where both the training and test accuracies are illustrated from generation 1 and up to generation 80. Furthermore, the dissimilarity between the individuals from the same species was measured during the evolutionary process; in Figure 2 it can be observed that the distance between rules of each class decreases almost in the same manner. When randomly generated, all dissimilarities start from a large value and converge to a comparable small value after 80 generations. The dissimilarity within a species is obtained by adding the distances from each individual to the mean of that population.

——————————— Figure 1 about here ———————————

——————————— Figure 2 about here ———————————

As concerns the quality of the results, a confusion matrix is outlined in Table 3, thus illustrating how far the misclassified samples are from the actual classes and which degrees are better interpreted. It is taken from a run that achieved an accuracy of 65% on the test set. The correctly classified samples are the ones on the principal diagonal. The test set comprises of 180 samples, all from the F1 class are correctly classified, but, unfortunately many from F2 are also labeled as F1. Most of the data are split between levels F1 (59) and F4 (62) and the CCEA basically concentrates on recognizing these two classes as good as possible. However, it has to be underlined that, in each run, the training and test data are different and the confusion matrices may change considerably.

——————————— Table 3 about here ———————————

The test accuracies obtained by CCEA and by the techniques used for comparison are highlighted in Table 4. Besides the mean value over the 30 repeated runs, the minimum, maximum and the standard deviation are reported. CCEA obtains the best average value and the smallest standard deviation, demonstrating through the latter better stability than the rest of the techniques. NN delivers the worst mean result, but also the best maximum accuracy obtained in one run: this and the fact that it has the highest standard deviation proves that the technique is the least stable among the ones tested.

——————————— Table 4 about here ———————————

In order to verify the significance of the results and to validate the hypothesis formulated in the task subsection, two statistical tests are conducted for the results obtained in the 30 repeats of the algorithms. A t-test for independent samples is used to assess the difference in means between CCEA and every other technique that is used for comparison. An alternative given by the Wilcoxon rank-sum test is also employed, since in some instances it gives greater power to reject the null hypothesis than the t-test. The statistical results are presented in Table 5. A p-value that is smaller than 0.05 demonstrates significant difference in results. The only method that, from a statistically point of view, is not at a sizeable difference from the dominating CCEA is the SVM applied on the data modified by PCA, although the mean difference is of almost one percent. Nevertheless, the proposed method has the essential advantage that it reveals sets of attributes that, even if at first glance may not appear as the most suitable choice for a medical expert, it is through association that they provide a better accuracy than when using the entire set of features.

——————————— Table 5 about here ———————————

Discussion: Naturally, besides the discussed configuration, there are several others found by the hill climbing algorithm which also produce good results. It is very interesting to observe that there are not always the same attributes that are selected; the hill climbing algorithm rather discovers sets of features that in connection perform better. However, there are some attributes that are included more often into many successful configurations and, in order to discover them, best 3% of the 1000 different configurations that were evolved are examined and the importance of the features is illustrated in Figure 3. Overall, among the attributes that seem more decisive are the following: stiffness, sex, cholesterol, glycemia, prothrombin index and alkaline phosphatase. However, the attributes were not necessarily considered in this combination and there are others that immediately follow in importance, like prolonged activated partial thromboplastin time or haematids; the enumeration could continue in the order of weight, according to the automated artificial intelligence composed methodology used in the current paper.

——————————— Figure 3 about here ———————————

It has to be underlined that the most significant feature, the one that has been chosen in most of the successful combinations, was the liver stiffness indicator, fact that is also acknowledged by the medical experts.

As concerns the prediction power on the training and test sets (see Figure 1), the test accuracy is most of the time above the training one, meaning that overfitting is avoided. Approximately from the second half of the evolution time (generation 45), there is no major fluctuation with respect to either of the two accuracies as the methods stabilizes around the final result.

5. Conclusions and Future Work

A hill climbing algorithm for choosing a set of features to be used by a classifier is employed together with a cooperative coevolutionary method that provides rules for differentiating between fibrosis degrees, following several medical indicators of patients with chronic hepatitis C.

The CCEA is kept to its simplest formulation with the coevolutionary parameters set to the easiest default values in order to simplify the job of the user that has to establish only the parameters of the EA; these can also be straightforwardly handled, as experimentation provided good results for a large range of configurations. Moreover, the proposed technique performed better than recognized classification techniques like SVMs of NNs, with or without a preprocessing method like PCA. Additionally to that, when having evolved the rules, it takes less than a 10th of a second for the testing part to take place on an average performing computer. Most importantly, the method can provide prototypes of rules for each fibrosis stage and information about the importance and interaction of the attributes.

As a future step to be undertaken, it is planed to focus more on the most important feature, the liver stiffness, as a part of the patients can be directly classified by using that attribute only while, for the other part, it gives important information about the classes they are close to; the classification method proposed herein could then be adapted and employed only for these delicate cases.

Additionally, images obtained by clinical ultrasound examination are to be used in the near future as another indicator that could help in identifying a more precise diagnosis like in [17].

Acknowledgements

Present work was supported by the research grant No. 41071/2007, entitled "SONOFIBROCAST" (Diagnostic algorithm for the stage and prediction of evolution of hepatic fibrosis using non-invasive ultrasound techniques, optimized by stochastic analysis and image analysis), financed by the Romanian Ministry of Education and Research and the National Authority for Scientific Research (MEdC & ANCS).

The authors would like to thank the reviewers for their helpful comments and suggestions that brought insight and improvement to the paper during the revision.

References

[1] W. Yeh, S.W. Huang, P.C. Li, Liver fibrosis grade classification with B-mode ultrasound, Ultrasound Med. Biol. 29(9), 1229–1235 (2003).

[2] Z. Jiang, K. Yamauchi, K. Yoshioka et al, Support vector machine-based feature selection for classification of liver fibrosis grade in chronic hepatitis C, J. Med. Syst. 20, 389–394 (2006).

[3] M. Gorunescu, S. Belciug, A.-B. Salem et al, A machine learning-based diagnosis for liver diseases using the Fibroscan medical ultrasound technique, ACM International Conference on Intelligent Computing and Information Systems, 243–248 (2009).

[4] S. Belciug, M. Lupsor, R. Badea, Features selection approach for non-

invasive evaluation of liver fibrosis, Annals of the University of Craiova, Mathematics and Computer Science Series 35, 15–20 (2008).

[5] J. Foucher, E. Chanteloup, J. Vergniol et al, Diagnosis of cirrhosis by transient elastography (FibroScan): a prospective study, Gut. 55 403–408 (2006).

[6] L. Sandrin, B. Fourquet, J.M. Hasquenoph et al, Transient elastography: a new noninvasive method for assessment of hepatic fibrosis, Ultrasound Med. Biol. 29, 1705–1713 (2003).

[7] M. Ziol, A. Handra-Luca, A. Kettaneh, Noninvasive assessment of liver fibrosis by measurement of stiffness in patients with chronic hepatitis C, Hepatology 41, 48–54 (2005).

[8] L. Castera, J. Vergniol, J. Foucher, Prospective comparison of transient elastography, Fibrotest, APRI, and liver biopsy for the assessment of fibrosis in chronic hepatitis C, Gastroenterology 128, 343–350 (2005).

[9] M. Lupsor, R. Badea, H. Stefnescu et al, Analysis of histopathological changes that influence liver stiffness in chronic hepatitis C. Results from a cohort of 324 patients, J. Gastrointestin. Liver Dis. 17(2), 155–163 (2008).

[10] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing. Springer-Verlag, Berlin Heildelberg New York (2003).

[11] Charles Darwin, On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. John Murray, London (1859).

[12] M.A. Potter and K. de Jong, Cooperative coevolution: An architecture for evolving coadapted subcomponents, Evol. Comput. 8(1), 1–29 (2000).

[13] C. Stoean, Mike Preuss, D. Dumitrescu, R. Stoean, Cooperative evolution of rules for classification, IEEE Intl. Conf. on Symbolic and Numeric Algorithms for Scientific Computing, Lisa O'Conner (Ed.), 317–322, IEEE Press, Los Alamitos, CA, USA (2006).

[14] C. Stoean, R. Stoean, Evolution of cooperating classification rules with an archiving strategy to underpin collaboration, H.-N. Teodorescu, J. Watada, and L.C. Jain (Eds.), Intelligent Systems and Technologies, Studies in Computational Intelligence 217, 47–65, Springer-Verlag, Berlin Heidelberg New York (2009).

[15] Q. Shen, Z. Meia, B.-X. Ye, Simultaneous genes and training samples selection by modified particle swarm optimization for gene expression data classification, Comput. Biol. Med. 39(7), 646–649 (2009).

[16] Y. Marinakis, G. Dounias, J. Jantzen, Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification, Comput. Biol. Med. 39(1), 69–78 (2009).

[17] G. Cao, P. Shi, B. Hu, Liver fibrosis identification based on ultrasound images captured under varied imaging protocols, J. Zhejiang Univ. Sci. B. 6(11), 1107-1114 (2005).

Indicator	F0	F1	F2	F3	F4	Total
Stiffness ($A1$)	4.3 ± 0.9	6.1 ± 2	7.8 ± 3.6	11.9 ± 6.9	32.3 ± 18	14.9 ± 15.4
Sex ($A2$)	Male: 8	Male: 78	Male: 55	Male: 32	Male: 118	Male: 291
BMI (body mass index) ($A3$)	25.2 ± 5	23.8 ± 8.1	24.9 ± 6.3	24.3 ± 8.2	24 ± 9.6	24.3 ± 8.1
Glycemia ($A4$)	110 ± 79.8	83.4 ± 41.4	80.5 ± 47.4	99 ± 44.6	101.9 ± 49.3	91.2 ± 48.5
Triglycerides ($A5$)	125.7 ± 90	92.6 ± 80	87.5 ± 64.7	98.9 ± 64	106.6 ± 74.9	97.7 ± 74.3
Cholesterol ($A6$)	207.5 ± 73.9	154.4 ± 94.4	150 ± 86.2	164 ± 73.8	147.3 ± 74.5	154.6 ± 84.4
HDL Cholesterol ($A7$)	39.2 ± 28.5	33.4 ± 32.5	37.4 ± 32.9	34 ± 32.6	27.9 ± 29.5	33 ± 31.7
Aspartate aminotransferase ($A8$)	32.4 ± 20.4	39 ± 29	47.5 ± 36.5	68.6 ± 54.4	77.2 ± 54.9	55.6 ± 46
Alanin aminotransferase ($A9$)	53.7 ± 37.9	63.8 ± 51.2	75.6 ± 60.6	102.8 ± 94.9	77.2 ± 59.8	74.8 ± 63.2
Gama glutamyl transpeptidase ($A10$)	63.6 ± 63.5	49.2 ± 46.7	62.5 ± 71	78.3 ± 84.3	137.4 ± 208.4	82.6 ± 130.7
Total bilirubin ($A11$)	0.6 ± 0.3	0.6 ± 0.4	0.6 ± 0.5	0.7 ± 0.4	2.2 ± 7.7	1.1 ± 4.3
Alkaline phosphatase ($A12$)	188.8 ± 146.3	148.2 ± 97	150.3 ± 110.5	179 ± 106.8	249 ± 163.3	184 ± 133
Prothrombin index ($A13$)	87 ± 42.8	76.6 ± 47.5	63.8 ± 50.5	70.2 ± 44.3	69.4 ± 36.8	71.2 ± 45
TQS (Quiq Time) ($A14$)	12.4 ± 5.9	10.8 ± 10.4	9.1 ± 7.8	10.6 ± 7.9	16 ± 7	12 ± 8.9
INR (prothrombin time ratio) ($A15$)	0.8 ± 0.4	0.8 ± 0.4	0.7 ± 0.5	0.8 ± 0.5	1.1 ± 0.5	0.9 ± 0.5
Prolonged activated partial thromboplastin time ($A16$)	24.8 ± 12.3	20.4 ± 14.2	17.9 ± 15.1	21 ± 14.4	28.3 ± 11.7	22.4 ± 14.2
Haematids (erythrocytes) ($A17$)	4.1 ± 1.7	3.9 ± 2.1	30 ± 336.7	4.2 ± 1.9	6.3 ± 29.1	10.6 ± 161.2
Hemoglobin ($A18$)	11.9 ± 5.1	11.5 ± 6	11.1 ± 6.4	12.6 ± 5.4	12.9 ± 4.3	12 ± 5.5
Hematocrit ($A19$)	35.2 ± 14.9	31 ± 19.7	25.7 ± 21.4	30.9 ± 19.9	36.1 ± 14.4	31.5 ± 18.9
Medium erytrocity volume ($A20$)	69 ± 36.2	57.7 ± 41.6	48.9 ± 44.1	58.8 ± 42	72.8 ± 35.7	60.8 ± 41.3
Avg. erytrocitary hemoglobin ($A21$)	22.2 ± 12.8	18.7 ± 14.4	16.4 ± 15	19.2 ± 14.6	26.3 ± 15.8	20.7 ± 15.4
Avg. concentration of hemoglobin in a red blood cell ($A22$)	16.3 ± 17.2	17.3 ± 17	15.2 ± 16.8	18.6 ± 17.1	21.7 ± 17.2	18.2 ± 17.2
Thrombocytes ($A23$)	209.7 ± 73.4	184.6 ± 105.5	170.9 ± 105.3	175.8 ± 85.2	116.3 ± 76.3	161 ± 98.6
Sideraemia ($A24$)	56.6 ± 56	69.4 ± 58.5	73.3 ± 62.7	86 ± 74.2	85.3 ± 77.7	76.5 ± 67.8

Table 1: Description of fibrosis data features: their values in mean and standard deviation.

Method	*Pop. Size*	P_r	P_m	MS	*Eval. calls*	*Generations*
Hill Climbing	1	-	0.1	-	1000	20
CCEA	50	0.9	0.2	0.1	-	80

Table 2: Parameter values for the hill climbing algorithm and the CCEA. Pop. size refers to the size of the population, P_r and P_m are the probabilities for recombination and mutation, MS is the mutation strength and the last two parameters represent the stop conditions for the two combined methods.

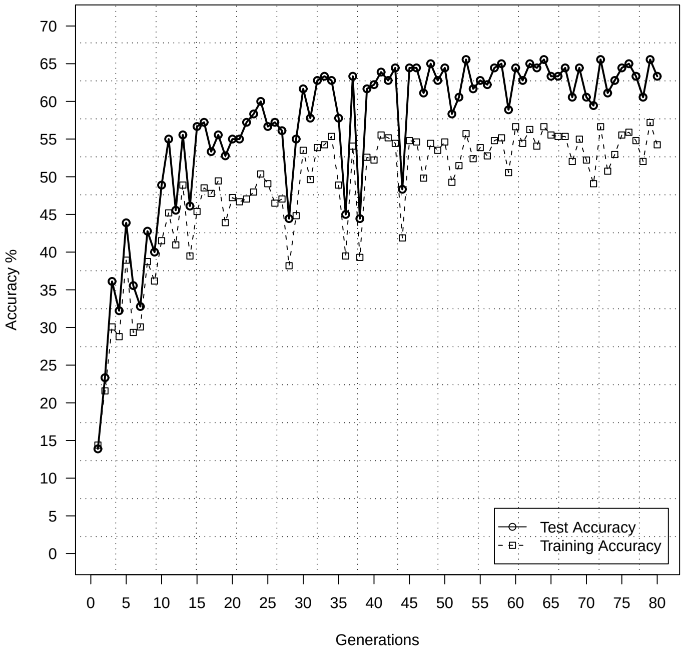

Figure 1: Accuracies for a well-performing run of the best configuration found by the hill climbing algorithm.

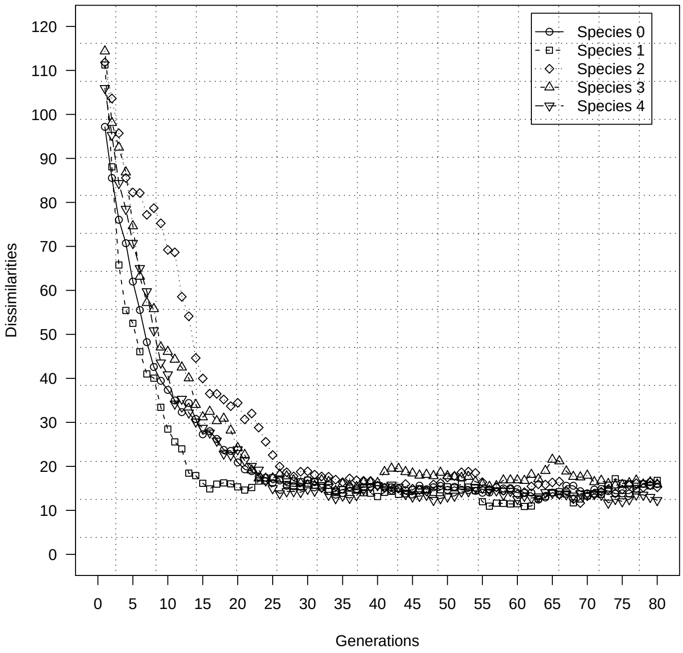

Figure 2: Dissimilarities in each of the five species for the same run as in Figure 1.

		Actual				
		F0	F1	F2	F3	F4
Predicted	F0	**0**	0	2	0	0
	F1	7	**59**	27	7	1
	F2	0	0	**4**	0	8
	F3	0	0	0	**8**	8
	F4	0	0	3	2	**46**

Table 3: Example of a confusion matrix of CCEA for differentiating the correct labeling from the misclassifications: predicted outcomes on the rows, actual classes on the columns, number of correct predictions on the diagonal.

Method	Minimum	Maximum	Mean (%)	St. dev. (%)
CCEA	**58.89**	65.55	**62.11**	2.14
SVM	53.33	63.89	60.11	3.74
NN	53.33	**66.67**	58.89	4.29
SVM+PCA	57.78	64.44	61.33	2.56
NN+PCA	52.22	66.67	59.61	3.54

Table 4: The comparison between the accuracy results obtained by the considered techniques averaged over 30 repeated runs. The minimum and maximum values over the 30 trials, as well as the standard deviations are included.

CCEA vs.	**p-values**	
	t-test	Wilcoxon Rank-Sum Test
SVM	**0.01**	0.07
NN	**4.3e-4**	**6.8e-4**
SVM+PCA	0.2	0.3
NN+PCA	0.06	**0.02**

Table 5: The p-values calculated through a t-test for independent samples and a Wilcoxon rank-sum test for CCEA vs. the other techniques as applied for the classification of the liver fibrosis staging data.

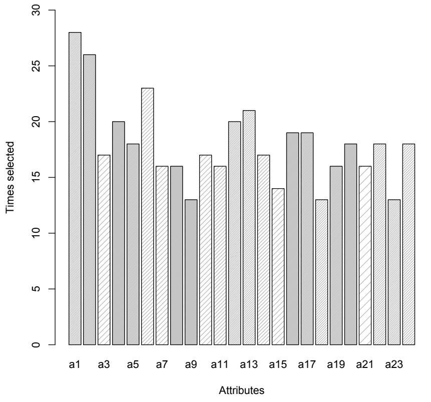

Figure 3: Most selected attributes (from 30 configurations) that yield the best results. The attributes are represented in the order in which they were previously presented in Table 1.

Summary

A rule generator based on cooperative coevolution and powered by a hill climber to choose the most appropriate and crucial medical attributes is employed in present paper for the determination of the liver fibrosis degree within chronic hepatitis C. Clinical/biochemical examinations and the liver stiffness result from the state-of-the-art non-invasive Fibroscan device play together a complex role that must be discovered by the proposed methodology. The combination between the hill climber and the cooperative coevolutionary classifier aims not only towards a rigorous feature selection and accurate prediction capacity, but additionally for an illustration of the existing interaction between those attributes that actually trigger the outcome. Hepatic fibrosis is a crucial indicator of a liver disease within chronic hepatitis C. The more precise the identification of its stage, the more increased the chances of an immediate proper treatment and survival of the patient. The particular data set employed in this study comes from the 3rd Medical Clinic of the University of Medicine and Pharmacy of Cluj-Napoca, Romania, and consists of 722 samples with 24 attributes and 5 possible degrees of fibrosis. A hill climber searches for a possible boolean selection of attributes through the search space. Every candidate configuration is measured against the training set by means of a cooperative coevolutionary approach to classification. For the given selection of features, corresponding genotypic rules evolve within different species for each class. The performance of a resulting set of rules is the fitness credited to the potential combination of attributes. The medical expert opinion on the significance and relationships of the consequential features confirms the computational findings.