A Differential Evolution Approach for Classification of Multiple Sclerosis Lesions

Ivanoe De Falco, Umberto Scafuri, Ernesto Tarantino
Institute of High Performance Computing and Networking - National Research Council of Italy
Via P. Castellino, 111 80131 Naples, ITALY
{ivanoe.defalco,umberto.scafuri,ernesto.tarantino}@na.icar.cnr.it

***Abstract*—The problem of automatically extracting novel and interesting knowledge from large amount of data is often performed heuristically when pattern extraction through classical statistical methods is found hard. In this paper an evolutionary approach, based on Differential Evolution, is proposed, which is able to perform the automatic discovery of comprehensible classification rules as a set of *IF...THEN* rules over a database of Multiple Sclerosis potential lesions. Moreover, this tool also determines which the most discriminant database attributes are in categorizing instances. Therefore, this evolutionary tool provides an efficient decision support system for clinical decisions, that could be a useful tool for medical experts to help them gain insight into the reasons for assessing the abnormality of a lesion.**

***Index Terms*—Multiple Sclerosis, classification, knowledge extraction, IF...THEN rules, Differential Evolution.**

I. Introduction

Multiple sclerosis (MS) is a long-lasting and disabling disease of the central nervous system which disrupts the information flow within the brain, and between the brain and body [1]. This results in several signs and symptoms, which include physical, mental, and in some cases psychiatric problems. Some such symptoms may be double vision, blindness in one eye, muscle weakness, sensorial or coordination troubles. MS is the most common autoimmune disorder affecting the central nervous system. In 2013, about 2.3 million people were affected worldwide, and about 20,000 people died from it. MS is complex and there is no single test that is proof-positive for its diagnosis. Although its not the sole test used to diagnose MS, magnetic resonance imaging (MRI) that makes detailed pictures of the body structures is very commonly used to visualize lesions [2] and represents a giant step in confirming a diagnosis and in evaluating its evolution. The use of MR images as a marker for MS needs the exploitation of all the knowledge of experts to correctly identify MS lesions.

In general, the process of finding out actual lesions for MS can be seen as a pipelining of three-task procedure: the *segmentation* of the MRI images into groups of homogeneous pixels/voxels representing tissues, the *labeling* of those tissues, and finally the actual *classification* step, meaning with this the assignment of each potential lesion detected.

Yet, this process is very laborious because of the high number of MR images that must be examined and of the variability in the number of MS lesions per image, as well as in their size and spatial distribution. Furthermore, the result of the analysis of an MR image is a set of potential lesions, some of which are actually lesions whereas others are not. This is a typical classification task, that has been up to now carried out prominently by human experts only.

Classification [3] is a well–recognized data mining task that uses supervised learning methods to predict the class which an instance belongs to. The operating way consists in a training and a testing phase. During the former a set of input instances (training dataset) is used to find a classification model that assigns them to predefined classes. Afterwards, the induced model is employed to classify the unknown instances (testing dataset) into their corresponding classes. The classification is of prime importance in the medical domain as an effective support in decision making with respect to diagnosis.

In this paper the goal is to design and implement an automatic tool, based on a Soft Computing methodology, that can be a useful decision support system (DSS) to doctors in their classification decisions about whether or not lesions in a patient's tissues evidenced by the segmentation and labeling steps are related to the presence of Multiple Sclerosis.

Not only should such a tool accurately discriminate between healthy and ill people, it should also support doctors by letting them have an insight into the reasons for this kind of prediction. This latter issue can be accomplished by automatically extracting explicit knowledge from the data and providing doctors with it under an easy–to-understand form. Our choice has been to organize such knowledge under the form of a set of explicit *IF–THEN* rules, which is typically easy to interpret. The set of rules will consist of some rules describing the conditions for illness and some other implying healthiness.

Hence, our approach is different from others in the literature [4], [5], [6], [7], in which Soft Computing–based tools are presented which often offer good performance in terms of classification rate, yet they cannot explain the reasons for their decisions, and, as such, are of little utility to doctors. Such tools are frequently termed as black-box classifiers.

Since our approach aims at supporting medical decisions, we do believe that its explicative feature is of very high significance to doctors. In fact, from our experience in cooperating with doctors, we have learned that they very seldom have confidence in automatic tools, and this is especially true for the ones that do not provide any interaction with them.

During the last decade, Differential Evolution (DE) [8], [9] has achieved great success on many real–world application problems [10]. DE is an evolutionary algorithm (EA) that has

978-1-5090-0679-3/16/$31.00 ©2016 IEEE

gained popularity because of its ease of implementation, of the request of few control parameters and of its low complexity [11]. Furthermore, such a technique often performs better, with regards to convergence speed, accuracy, and robustness, than several EAs, and many among the stochastic and direct search techniques for global optimization when used over benchmark problems and real-world ones [10], [12].

At our knowledge, apart from few papers [13], [14], [15], never has DE been employed for extracting comprehensible classification rules from databases. In fact, all other DE-based classification tools in the literature either merely look for centroids for the database classes instead of extracting explicit knowledge as rule sets [16], [17], [18], [19], or are hybrid systems in which DE just aims at optimizing some parameters while classification is done through other mechanisms [20], [21], [22], [23], [24].

Unlike the *Pittsburgh approach* [13], [14], [15] used to extract in one step the best set of explicit *IF–THEN* rules for all the classes, the approach presented within this paper finds separately the best set of rules for each class. The resulting algorithm is referred to as DE for Classification (DEC).

The paper is organized as follows: Section II illustrates the DE technique; Section III presents DE applied to the classification problem, and the DEC algorithm. In Section IV the experimental findings are shown together with the resulting explicit rules, and a discussion is presented. Finally, Section V is dedicated to conclusion remarks and future works.

II. Differential Evolution

Differential Evolution is a stochastic optimization algorithm [8], [9] based on a set, called *population*, of solutions, said *individuals*, which are represented as vectors of real numbers.

The basic idea underlying DE is in the use of difference vectors. A difference vector is computed as the difference between two population individuals. Essentially, DE creates new individuals, said trials, through the addition of one or two weighted difference vectors to another member in the population. A trial vector possessing a better objective function (*fitness function*) value than the original individual will replace this latter. A recombination mechanism, aimed at building trial vectors through the use of components of individuals in the current population, shuffles information about successful combinations, which makes the search for an optimum be focused on promising areas in the space of solutions.

In more depth, to face an optimization problem with n real–valued parameters, DE starts by randomly initializing a population of l individuals each consisting in n genes, each of which contains a real value. Then, from the current population, l individuals for the new population are obtained through a modification scheme, and this is repeated generation after generation. Many modification schemes exist. DE inventors [8], [9] advanced a naming–convention that names any DE strategy by means of a string of the form *DE/x/y/z*. In such a representation, x is a string specifying the individual that must be modified (*best* = the best one in the population, *rand* = an individual chosen at random, *rand–to–best* = a randomly chosen one, but the best one takes part in the modification as well), y is the number of difference vectors to be used in the modification of x (either 1 or 2), and z denotes the method for crossover (*exp* = exponential, *bin* = binomial).

As an example of a modification scheme, here the description of the *DE/rand–to–best/1/bin* strategy is given. According to this scheme, when creating the new i–th trial, the i–th individual in the current population is taken into account, and the random generation of two integer numbers r_1 and r_2 in $[1, \ldots, l]$ (differing each other and different from i) takes place. Moreover, the random generation of another integer number s in $[1, n]$ is performed. Now, the trial individual $\mathbf{x}'_i$ can be created, for which the generic j–th component is

$$x'_{i,j} = x_{i,j} + F \cdot (best_j - x_{i,j}) + F \cdot (x_{r_1,j} - x_{r_2,j}) \qquad (1)$$

on condition that either a real number ρ in $[0.0, 1.0]$, generated at random, is lower than the value of the *crossover rate* (CR) DE parameter, in $[0.0, 1.0]$ range as well, or the position j we are considering is equal to s. If neither is true, a copy is carried out: $x'_{i,j} = x_{i,j}$. F is the *factor*, another DE parameter, and is a real-valued constant used to control the largeness of the differential variation $(x_{r_1,j} - x_{r_2,j})$, whereas $best_j$ represents the j–th component of the best individual in the population.

The trial individual $\mathbf{x}'_i$ obtained in this way is compared against $\mathbf{x}_i$ in the current population in terms of fitness values, and the better enters the new population (at parity, the trial is chosen as well). This scheme is reiterated for a number of generations Gen, which is another DE parameter.

III. Differential Evolution applied to classification

The aim is to address a supervised classification problem by a DE tool that can extract in an automatic way from a database a set of easy–to–understand classification rules for each class. If we consider the number of database attributes and their ranges, it is evident that, whenever classification problems are complex, the amount of possible descriptions becomes huge. This makes an exhaustive search based on the enumeration of all the possible descriptions computationally unviable. Hence we appeal to an evolutionary system. It does not guarantee to find the global optimum, yet it often allows achieving a suboptimal solution in a reasonably low amount of time. In particular, the basic component of our approach is constituted by a DE model in which the population is constituted by l potential solutions, each representing a set of classification rules. During the evolution, each rule set competes with the other rule sets, and the best one is gradually improved by finding new, fitter ones. This search goes on until rule sets of good quality are found or other stopping criteria are satisfied. At the end of the evolution a best rule set emerges. Hence, in our approach more than one rule is discovered for each class.

With the definition of a fitness function, the classification problem becomes the problem of searching the best among all the possible descriptions, i.e. an optimization of the fitness function. For this latter problem several optimization techniques can be employed. The major steps of our DEC–based rule set search can be formalized as follows:

1) for each database class find the best rule set as follows:
   a) randomly create an initial population of rule sets
   b) evaluate each rule set by means of a fitness function;
   c) for each individual in the current population:
      - apply to it the chosen DE strategy to produce a new rule set;
      - insert this new rule set in the new population if its fitness is better than that of the starting one;
   d) repeat steps b) and c) until either a sufficient-quality rule set is obtained or the maximum number of generations is reached;
2) assign each example to one and only one class by resolving possible indeterminate cases.

Step (2) is very important because in all of the previous steps one class at a time is dealt with. Consequently, for a database containing Cl classes, it might occur that some instances cannot be allocated to one and only one class: such instances are termed *indeterminate*. This can happen either because an item is taken by more rules predicting for different classes ("*yes*" indeterminate item) or because no rule takes it ("*no*" indeterminate item). Hence a post–processing phase should be performed to assign each indeterminate case to one and only one class. The assignment phase for these indeterminate cases is described in detail in [14]. If, instead, a sample is taken by one or more rules predicting for one class, then the sample is assigned to that class.

A. Encoding

The supervised classification problem can be defined as follows. Suppose to have a database D constituted by N instances X_i with the corresponding class label C_i, i.e., $D = \{(X_1, C_1), (X_2, C_2), \ldots, (X_N, C_N)\}$. Each of these instances is a vector of n components that we call 'attributes'.

We apply DEC as the optimization algorithm to find for each class the optimal attribute ranges allowing us to predict the class. We face this as a supervised classification problem, so during a training phase the classifier builds a decision function as a set of rules starting from a part of the database. Then, in the testing set this decision function is used to predict the class of the previously unseen database instances.

Each individual is a real-valued vector representing a set of N_R rules connected in OR reported in sequence in the representation of the individual, called chromosome. Each rule is composed by a number of n conditional clauses, in which conditions over the attribute values are set, and by a predictive clause representing the class. A class together with its condition forms a classification rule R_i under the form '*IF*$<$ *conditions* $>$ *THEN* $<$ *class* $>$'.

The idea underlying our encoding is that the conditional part of the representation of any given rule rule should consist in a number of clauses equal to the number of the instance attributes, i.e. n. These clauses should be joined by means of logical AND connectors. Each clause should code for a zero–th order clause, i.e., one in which only one attribute is contained. To each attribute a range within the attribute domain should be associated.

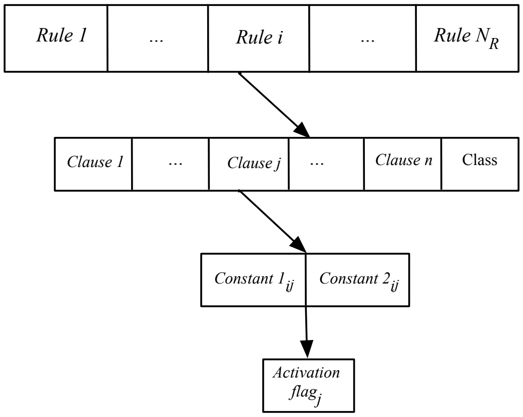

Fig. 1. The rule set structure.

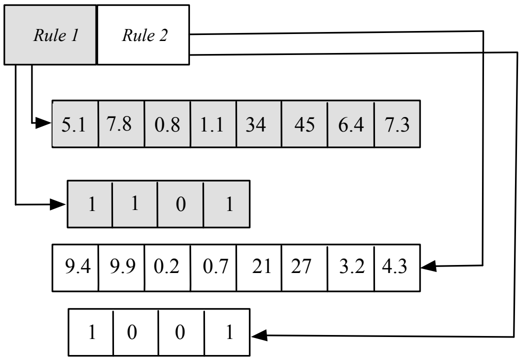

Fig. 2. An example of rule coding with the related range and active vectors.

To implement our view, we need two vectors.

Firstly, a *range vector* of size $2 \cdot n \cdot N_R$ is used that contains the minimal and maximal values for the range of each variable in each rule in the considered rule set.

Secondly, an *active vector* of size $n \cdot N_R$ specifies whether or not each clause of each rule in the data set is active, i.e. whether or not it should be explicited in the construction of the rule set. Each position of such a vector can take on two values $\{0{,}1\}$ indicating the absence and the presence of the related clause in the explicit form of the rule, respectively.

The structure of the rule set contained in a generic DEC individual is shown in Fig. 1. Going from top to bottom, it shows that a DEC individual contains N_R rules, each of which contains n clauses, each represented by two real values $Constant1_j$ and $Constant2_j$, and an *Activation* $flag_j$ stating if that clause contributes to the rule under construction.

Let's make an example. Supposing there are two rules and four attributes denoted with $A_1, \cdots, A_4$, if the range and active vectors are as in Fig. 2, then the classification rule set extracted for a class C is decoded as follows:

IF (($A_1 \in [5.1, 7.8]$) AND ($A_2 \in [0.8, 1.1]$) AND ($A_4 \in [6.4, 7.3]$)) OR (($A_1 \in [9.4, 9.9]$) AND ($A_4 \in [3.2, 4.3]$)) THEN C

The execution of DEC provides a rule set for each class that contains a combination of AND and OR connectors used to construct the classification rule for that class.

B. Fitness function

The fitness $\Phi(i)$ of the generic i-th individual in the population is computed as the percentage of correctly classified cases over the training set:

$$\Phi(i) = \frac{N_c}{N_{tr}} \cdot 100.0 \qquad (2)$$

where N_c is the number of cases in the training set that are correctly classified by the rule represented by the individual i, and N_{tr} is the total number of cases in the training set.

Given this definition, $\Phi(i)$ can vary in the range $[0.0-100.0]$ and the classification becomes a maximization problem.

To evaluate the effectiveness of the classifier over items never seen in this training phase, and to analyze the generalization ability of the rule set obtained, a testing set is taken into account, and several performance measures can be computed.

The first performance indicator is the *Accuracy* defined as the number of correctly classified samples with respect to the total number of samples in the testing set. It is evaluated by means of the equation (2) computed on the testing set. We will denote it as A in the following.

The accuracy in the classification greatly depends both on the method employed and on the characteristics of the specific database to be faced. It is well known that accuracy on its own may not be a sufficient measure for evaluating a model when an imbalanced distribution of the classes takes place, as it is the case for the problem we are facing in this paper.

Two more important indicators are defined in the medical domain, namely the sensitivity *Se* and the specificity *Sp*, that can be used as alternative/supplementary measures to accuracy [25]. These further measures represent an estimation of the posterior probability of obtaining a positive/negative decision in the presence of a positive/negative case, respectively, where the term ‘positive’ represents an unhealthy case. They quantify the ability to classify correctly examples as belonging to the positive and the negative class respectively.

A further, often used, classification quality indicator is the “area under ROC curve”. It too ranges within 0.0 and 100.0, the higher the better. For details about it please see [25]. We will denote it as $AURC$.

IV. Results and Discussion

To carry out experiments a database, suitably anonymized, collected at the Department of Bio-Morphological and Functional Sciences of the University of Naples Federico II has been considered. To create this database, starting from a set of Magnetic Resonance brain images of 120 patients with clinically definite MS, they firstly applied a multiparametric segmentation procedure to the whole image set, so as to identify both normal brain tissues and clusters of potentially abnormal white matter voxels. Then they labeled these latter as White Matter Potential Lesions (WMPLs).

TABLE I
The Multiple Sclerosis Lesions database.

variable	unit range	description
surrounding white matter	0.32 - 1.00	Amount of White Matter enclosing a lesion
compactness	0.31 - 1.98	Degree of compactness of a lesion
tissue contrast	0.56 - 1.00	Minimum color contrast to detect a WML in the multiparametric space
volume	3 - 10,522	Lesion volume (number of voxels)
sphericity	0.01 - 1.23	Degree of sphericity of a lesion

Table I describes the five features contained in the resulting database, the sixth being the class (*no_MS_lesion / MS_lesion*). This database, prepared by medical experts, contains 2844 items, 1905 representing actual MS-related lesions (class 2) and 939 showing no actual MS-related lesions (class 1). All the database features contain real–valued data, apart from the volume that is expressed as an integer value, and the class, which is integer as well.

In our experiments we regard this database as the *gold standard*, we start from it and make use of DEC to extract explicit knowledge.

As concerns DEC, it has been written in C language, and has been run on a personal computer equipped with a 2.93 Ghz processor and 4GB 1067 MHz memory. A *DE/rand–to–best/1/bin* mutation strategy has been chosen in the DEC version used in these experiments, and the values for the parameters have been set as follows: population size $N_{Pop} = 30$, number of generations $Gen = 500$, crossover ratio $CR = 0.5$, scale factor $F = 0.5$. No preliminary tuning phase has been specifically effected over this sclerosis problem for the choice of these values, rather the same values as in [14] have been used. Moreover, we have decided to search for rule sets having three rules for MS lesions and three rules for non-MS lesions.

A DEC run performs a 10–fold cross-validation, so, for the generic i-th fold, testing uses the i-th 10% of the data, in their sequential order in the database, while training occurs on the remaining 90%. The classification result for this generic i-th fold is the accuracy over the related testing set $\%A^i_{Te}$. The result for the whole run, instead, is the average, over the 10 folds, of the 10 $\%A^i_{Te}$ values achieved, i.e. $Av_C = \langle \%A^i_{Te} \rangle$.

Actually, DE is a stochastic technique, whose results depend on an initial integer value assigned to a seed for a random number generator. To get rid of this, a total number of 25 runs, each of them being a 10–fold cross-validation, has been effected, and results have been averaged over the 25 runs.

For each of the four parameters accuracy, sensitivity, specificity, and area under the ROC curve, Table II reports the average values av achieved over the 25 10–fold runs, the related standard deviation st_dev, the highest (max) and the lowest (min) values. All this information is provided for the training set, the testing set, and the whole database.

A. The advantage of DEC: the IF-THEN rules

The clear advantage of DEC as a DSS consists in the fact that it provides users with explicit knowledge that is automat-

TABLE II
RESULTS IN TERMS OF ACCURACY, SENSITIVITY, SPECIFICITY, AND ROC CURVE AREA.

	A				*Se*				*Sp*				*AURC*			
Database	*av*	*std_dev*	*max*	*min*	*av*	*std_dev*	*max*	*min*	*av*	*std_dev*	*max*	*min*	*av*	*std_dev*	*max*	*min*
Training set	75.53	2.27	84.14	72.81	74.37	2.93	86.20	71.10	87.37	7.70	98.92	70.53	85.60	6.42	95.25	71.61
Testing set	74.51	4.84	85.92	66.90	73.97	5.45	93.64	66.67	82.87	11.56	100.00	56.52	81.51	9.84	96.69	58.45
All database	75.43	2.21	83.51	72.96	74.32	2.89	86.87	71.99	86.90	7.87	98.54	70.28	85.18	6.57	94.84	71.39

ically obtained from the database and presented as *IF-THEN* rules. In fact, these rules can be profitably used for diagnosis purposes. Moreover, DEC also executes feature extraction, because the obtained rules may contain just some among the attributes in the database, which could be a very useful support when performing diagnoses. This is the case reported in [14]. Thanks to both these characteristics, physicians may be helped with useful information. Obviously, their opinion on the correctness of the extracted rules, and on their helpfulness as well, is of fundamental importance for medical practice. In the following the best set of rules found for the Multiple Sclerosis problem is reported. Namely, they are those with the highest accuracy value on the testing set obtained on a fold among all the 25 runs. This set of rules has been obtained in the 12-th run for fold 3.

$$\textit{IF } (0.40 \leq \mathit{compactness} \leq 0.91) \textit{ AND } (0.07 \leq \mathit{sphericity} \leq 0.82) \\ \textit{THEN MS\_lesion}$$

$$\textit{IF } (0.63 \leq \mathit{compactness} \leq 1.93) \textit{ THEN MS\_lesion}$$

$$\textit{IF } (0.36 \leq \mathit{surrounding\_white\_matter} \leq 0.73) \textit{ AND} \\ (4241 \leq \mathit{volume} \leq 6945) \textit{ THEN MS\_lesion}$$

$$\textit{IF } (0.38 \leq \mathit{surrounding\_white\_matter} \leq 0.73) \textit{ AND} \\ (0.31 \leq \mathit{compactness} \leq 0.69) \textit{ THEN no\_MS\_lesion}$$

$$\textit{IF } (0.31 \leq \mathit{compactness} \leq 0.41) \textit{ THEN no\_MS\_lesion}$$

$$\textit{IF } (0.33 \leq \mathit{compactness} \leq 0.55) \textit{ AND} \\ (0.64 \leq \mathit{tissue\_contrast} \leq 0.91) \textit{ THEN no\_MS\_lesion}$$

Thus, even if the OR connector is not explicitly present in the single DEC rules, the whole rule set implicitly executes an OR over each class through the use of a user-defined number of rules to achieve satisfactory classification performance.

The values of accuracy, sensitivity, specificity, and area under ROC curve for this set of rules are reported in Tab.III over the training set, the testing set, and the total database.

TABLE III
THE CLASSIFICATION ACCURACY OF THE BEST SET OF RULES.

	A	*Se*	*Sp*	*AURC*
Training set	81.21%	82.06%	78.69%	79.00%
Testing set	85.92%	88.18%	80.25%	80.72%
All database	81.68%	82.64%	78.87%	79.19%

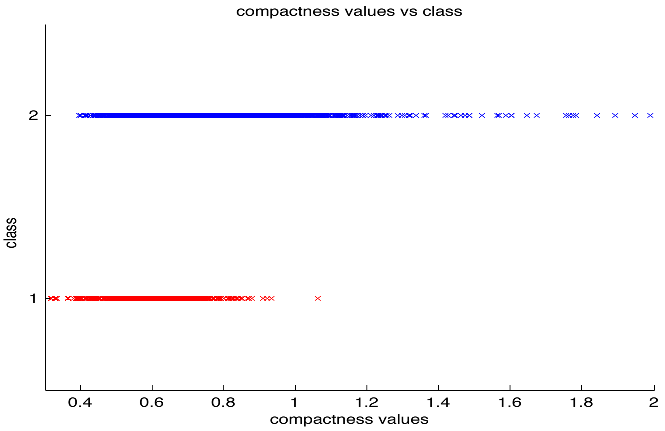

Fig. 3. Compactness values in items belonging to the two classes.

An interesting issue about these numerical results is that this rule set is effective in classification over the whole data set in terms of correct classification rate, and shows high values for sensitivity. This is of extreme importance in medicine, where the error of assigning sick people as healthy should be highly avoided, since this error would lead to not providing the patient with the necessary care. The other error, instead, i.e. assigning healthy people to the sick class, is related to specificity values, and, although may have dangerous psychologic consequences, does not have lethal consequences. Our rule set effectively deals with this scenario as well.

As it is evident from this rule set, DEC is able to suggest to doctors that compactness is a very discriminating parameter for detecting actual MS lesions, because it is present in five out of the six rules, i.e. two rules for lesions and all the three rules for non lesions. Basically, these rules say that if compactness is low, the lesion is not related to MS: the three intervals for compactness in the three non-MS rules are: [0.31 - 0.69], [0.31 - 0.41], and [0.33 - 0.55]. On the contrary, if compactness value is high, then the lesion is related to MS: in this case the two intervals for compactness in the two MS rules containing this parameter are, in fact, [0.40 - 0.91] and [0.63 - 1.93].

Aiming at further investigating this issue, in Fig. 3 we report the distribution of the values for the compactness parameter in items belonging to the two classes.

This figure shows that, actually, lesions not related to MR, i.e. those in class 1, have low compactness values (average: 0.54, min: 0.31, max: 1.06), typically from 0.33 up to about 0.70, apart from a few outliers with higher values. MR-related lesions (class 2), instead, exhibit higher values (average: 0.71, min: 0.40, max: 1.98), very often up to 1.25, many outliers showing values up to the maximum value, i.e. 1.98, and just few down to 0.40. On the other hand, the two classes strongly overlap in an intermediate range, say from 0.45 up to 0.75.

To effectively discriminate within this intermediate range, the other parameters come to our help. Each of them is contained in just one or two rules, but they are of high importance to support compactness parameter in correctly assigning items to the two classes.

It is interesting to contrast the knowledge that has been extracted by DEC against that provided by medical experts.

In fact, the medical knowledge needed to classify WMPLs has been defined in cooperation with a team of physicians, starting from the sclerosis features contained in the database, and can be stated, in natural language, as follows: *The tissue composing a WMPL is abnormal if the lesion is somewhat surrounded by WM, characterized by a strong compactness and greatly contrasted in the multiparametric space. The sphericity is moderate or high in small lesions, whereas, as their volume increases, the sphericity starts decreasing progressively. Finally, as volume increases and sphericity starts lessening, a lesion can be surrounded by gradually decreasing WM and its compactness still remains high.*

As it can be seen, high values for compactness are considered by doctors as a serious hint for the presence of an MS-related lesion. Yet, in their description, it seems as if this knowledge is hidden within many other considerations about all the other parameters, as if they all were equally important for discrimination. Even worse, these considerations are expressed in a kind of a fuzzy language, through the use of terms and expressions as: somewhat, moderate, starts decreasing progressively, starts lessening, can be, gradually decreasing, and so on. DEC, instead, proposes a clear way for assessing whether or not a lesion is related to MS.

V. Conclusions and future work

In this paper, an approach relying on Differential Evolution for the automatic classification of potential lesions in a real-world Multiple Sclerosis database has been proposed to implement a valuable decision support system that could be useful to doctors. Namely, the DEC tool has been utilized to extract in an automatic way explicit knowledge from the database as a set of *IF-THEN* rules, each composed by AND–connected literals on the database attributes, and to show them to the users.

The experiments have confirmed the viability of the approach presented, and the numerical results have evidenced its effectiveness in terms of satisfactory percentage of correct classification rate over previously unseen patients.

The most effective set of rules in terms of highest classification accuracy has been shown and discussed.

The advantage of DEC has resulted in its ability to supply users with both explicit knowledge and a selection among the database attributes, unlike many other black-box classification algorithms. In fact, a simple look at the extracted rules evidences that one of the database parameters is of main importance.

Since DEC is based on the basic version of DE, to improve its performance future work will involve the investigation of several enhanced DE versions appeared in the last years with the aim at softening the main problem DE suffers from, i.e. that of a limited amount of search moves.

References

[1] J. A. Bobholz and S. Gremley, "Multiple sclerosis and other demyelinating disorders," in *The Little Black Book of Neuropsychology: A Syndrome-Based Approach*, M. R. Schoenberg and J. G. Scott, Eds. Springer, 2011, ch. 20, pp. 647–662.

[2] D. Miller, R. Grossman, S. Reingold, and M. H.F., "The role of magnetic resonance techniques in understanding and managing multiple sclerosis," *Brain*, vol. 121, pp. 3–24, 1998.

[3] J. Han and M. Kamber, *Data mining: concept and techniques*. Morgan Kaufmann, 2001.

[4] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Trans. on Neural Networks*, vol. 8, no. 3, pp. 694–713, 1997.

[5] D. B. Fogel, E. C. Wasson, E. M. Boughton, V. W. Porto, and P. J. Angeline, "Linear and neural models for classifying breast masses," *IEEE Trans. on Medical Imaging*, vol. 17, no. 3, pp. 485–488, 1998.

[6] G. Fung and O. L. Mangasarian, "Semi–supervised support vector machines for unlabeled data classification," Department of Computer Science, University of Wisconsin, Tech. Rep., October 1999.

[7] C. H. Lee and D. G. Shin, "A multistrategy approach to classification learning in databases," *Data & Knowledge Engineering*, vol. 31, pp. 67–93, 1999.

[8] K. Price and R. Storn, "Differential evolution," *Dr. Dobb's Journal*, vol. 22, no. 4, pp. 18–24, 1997.

[9] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[10] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: a Practical Approach to Global Optimization*. Springer Verlag, 2005.

[11] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state–of–the–art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[12] S. Das, A. Konar, U. K. Chakraborty, and A. Abraham, "Differential evolution with a neighborhood–based mutation operator: a comparative study," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[13] I. De Falco, "A differential evolution-based systems supporting medical diagnosis through automatic knowledge extraction from databases," in *Proceedings of the International Conference on Bioinformatics and Biomedicine*. IEEE Press, 2011, pp. 321–326.

[14] ——, "Differential evolution for automatic rule extraction from medical databases," *Applied Soft Computing*, vol. 13, no. 2, pp. 1265–1283, 2013.

[15] G. Sannino, I. De Falco, and G. De Pietro, "An automatic rules extraction approach to support osa events detection in an mhealth system," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1518–1524, 2014.

[16] I. De Falco, A. Della Cioppa, and E. Tarantino, "Automatic classification of handsegmented image parts with differential evolution," in *Lecture Notes in Computer Science*. Springer–Verlag, 2006, pp. 403–414.

[17] P. Luukka and J. Lampinen, "Differential evolution classifier in noisy settings and with interacting variables," *Applied Soft Computing*, vol. 11, no. 1, pp. 891–899, 2011.

[18] I. S. U. Maulik, "Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery," *Pattern Recognition*, vol. 42, no. 9, pp. 2135—2149, 2009.

[19] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Transaction on Systems Man and Cybernetics A*, vol. 1, pp. 218–237, 2008.

[20] J. Wu and Z. Cai, "Attribute weighting via differential evolution algorithm for attribute weighted naive bayes (WNB)," *Journal of Computational Information Systems*, vol. 7, no. 5, pp. 1672–1679, 2011.

[21] I. Triguero, S. García, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44, pp. 901–916, 2011.

[22] R. A. Aliev, W. Pedrycz, B. G. Guirimov, R. R. Aliev, U. Ilhan, M. Babagil, and S. Mammadli, "Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization," *Information Sciences*, vol. 181, no. 9, pp. 1591—1608, 2011.

[23] F. Al-Obeidat, N. Belacel, J. A. Carretero, and P. Mahanti, "Differential evolution for learning the classification method PROAFTN," *Knowledge-Based Systems*, vol. 23, pp. 418–426, 2010.

[24] M. Esposito, I. De Falco, and G. De Pietro, "An evolutionary-fuzzy DSS for assessing health status in multiple sclerosis disease," *International Journal of Medical Informatics*, vol. 80, no. 12, pp. e245–e254, 2011.

[25] J. Han and M. Kamber, *Data mining: Concepts and techniques*. Morgan Kaufmann, 2000.