Diabetes Diagnosis through the Means of a Multimodal Evolutionary Algorithm

Cătălin Stoean[1], Ruxandra Stoean[2], Mike Preuss[3] and D. Dumitrescu[4]

[1]University of Craiova
Faculty of Mathematics and Computer Science, Department of Computer Science
Str. Alexandru Ioan Cuza, No. 13, Craiova, 200585, Romania
(E-mail: catalin.stoean@inf.ucv.ro)
[2]University of Craiova
Faculty of Mathematics and Computer Science, Department of Computer Science
Str. Alexandru Ioan Cuza, No. 13, Craiova, 200585, Romania
(E-mail: ruxandra.gorunescu@inf.ucv.ro)
[3]University of Dortmund
Department of Computer Science
Chair of Algorithm Engineering and Systems Analysis
Joseph-von-Fraunhofer-Str. 20, 44227, Dortmund, Germany
(E-mail: mike.preuss@uni-dortmund.de)
[4]Babes - Bolyai University
Faculty of Mathematics and Computer Science, Department of Computer Science
Str. Mihail Kogalniceanu, No. 1, Cluj-Napoca, 400084, Romania
(E-mail: ddumitr@cs.ubbcluj.ro)

Abstract. A radii-based evolutionary algorithm is applied in solving a difficult classification problem concerning diabetes diagnosis. The algorithm was designed to treat multimodality and has recently successfully been applied in the optimization of several multimodal functions. The medical problem to be solved is to predict the diagnosis – either diabetic or not – for a set of patients, given some personal and medical conditions. Proposed algorithm gives a high accuracy of prediction and thus provides a good means of understanding the factors that doctors consider when diagnosing diabetes and a way of checking the consistency of decision making. A clear advantage over other computational techniques is that, besides the outcome for patients in the test set, the algorithm also provides simple rules that led to that decision.
Keywords: evolutionary computation, classification, rule discovery, genetic chromodynamics.

1. Introduction

Evolutionary algorithms are optimization techniques that have been applied, with very promising results, to optimization problems, classification, clustering etc.
In present paper, a recently developed multimodal evolutionary algorithm [13] is applied in deciding if, based on the values of eight factors, patients should be tested positive or negative for diabetes.
The algorithm uses some part of the data, the training set, in order to learn the most appropriate attributes values that made doctors decide whether a patient was ill or not; this way, IF-THEN rules are built, having as the condition part the values medically leading to the conclusion part, *i.e.* one of the two possible outcomes. These rules are built in present algorithm in an evolutionary manner, that is, they encode chromosomes that are *evolved* during this training step and then applied for the classification of the rest of the data, *i.e.* the test set. These obtained rules are themselves of high importance, as they can also provide the reasoning rules underlying the decision-making and not only the results.

The data set comes from the UCI repository of machine learning databases [8]. The diagnosis, which is binary-valued, represents whether the patient shows signs of diabetes according to World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care).
The algorithm uses the training set to generally establish two rules, one for each class. It sometimes detects more than two rules, meaning there is more than one for a class; the accuracy for classification in these cases is also good, indicating that in the two classes, there are clusters that are formed.
The paper is organized as follows: next section shortly presents the basic ideas underlying evolutionary algorithms; section 3 gives a brief description of the framework proposed method belongs to and depicts the new algorithm; section 4 shows the diabetes optimization problem and its solution, through the means of proposed evolutionary algorithm, is reached in section 5; finally, conclusions are reached in last section.

2. Evolutionary Context

In a regular evolutionary algorithm, the solution to the problem to be solved has to be *encoded* into a chromosome. More *possible* solutions (chromosomes) are initially considered; the values for the *genes* of these chromosomes are randomly chosen. Then, by the means of a fitness function that measures the quality of the chromosomes (and thus the quality of the solutions) and using variations operators, the possible solutions are evolved to better ones and, in the end, to reach the optimum. The optimum can be considered as the best chromosome from the last generation or the best chromosome from all generations.
The standard variants of evolutionary algorithms are the genetic algorithms [7], the evolution strategies [9] and evolutionary programming ([4], [5]).
In multimodal evolutionary algorithms, there are several solutions for the problem considered for solving; that means, in the end, not only the best chromosome is of interest, but more chromosomes that are connected to other solutions (local optima, for example, in the case of function optimization).
There are several techniques for solving multimodal problems, *e.g.* niching methods, island models, coevolution [3]. The algorithm [13] applied in present paper for classification belongs to the *Genetic Chromodynamics* (GC) framework [2] and has the advantage of an increased convergence speed over the original algorithm of GC, as shown in [13].
GC represents an evolutionary framework designed for solving problems with multiple solutions. It has been widely applied in the past for the optimization of multimodal functions, text categorization, classification and clustering ([6], [13], [14]).

3. Genetic Chromodynamics Framework. Elitist Generational Genetic Chromodynamics algorithm

GC belongs to the family of radii-based multimodal evolutionary frameworks, as it builds and maintains subpopulations connected each to local or global optima of the problem to be solved. This is achieved by introducing a set of restrictions such as the way recombination takes place or the way selection is applied. For reproduction, a local interaction principle is considered, meaning that only chromosomes similar under a given threshold recombine. For selection, each chromosome represents a stepping-stone for the forming of the new generation – each chromosome is taken into account for reproduction. After crossover takes place, the offspring fights for survival with its first parent only, that is the *stepping-stone*.
GC introduces a new operator that *merges* very similar chromosomes into a single one that is often chosen to be the best one of them with respect to the fitness evaluation. It is a very useful

operator as it leads to a better computational time, obtained by reducing the size of the population, thus meaning less fitness evaluations.
It is important to note that, firstly, the merging operator saves evaluations only where they most likely will not make sense because chromosomes are very near to each other. Secondly, it can be used to decrease the population size in a meaningful way, so that in present example only two rules remain (which is what it is desired, since there are two classes of diagnosis). Without merging, the algorithm would end up with a lot of rules.
The elitist generational genetic chromodynamics framework (EGGC) algorithm applied in present paper achieves an even higher speed up in convergence by changing the way the offspring enters the population and the way selection for reproduction is carried out.
The EGGC algorithm is outlined below.

Algorithm 1. EGGC algorithm used for Diabetes Diagnosis

- $t = 0$;
- The initial population $P(t)$ is randomly chosen;
- Repeat
    - Evaluate each chromosome;
    - For $i = 1$ to n do
        - Randomly choose a chromosome c from the current population;
        - If there are chromosomes in the mating region of c:
            - Use proportional selection to choose a chromosome for crossover;
            - Crossover takes place and a descendant is obtained;
            - The descendant replaces the worst chromosome in its own mating region with respect to the fitness function provided the descendant is better.
        - Else, if there are no chromosomes in the mating region of c, mutation is applied to c. The descendant replaces the parent chromosome only if it is better.
    - Merging is applied to all chromosomes;
    - $t = t + 1$;
- Until (stop condition)

In the initialization of the population, a number n of chromosomes are considered – the values for genes are randomly taken from their intervals. Evaluation of a chromosome is done by measuring its quality; the quality (or fitness) evaluation is differently defined for each problem.
The distance between two chromosomes is computed. Let a chromosome c be considered: the distance between c and all the other chromosomes in the population is computed. The mating region of c contains all chromosomes that are at a distance from c of less than a given threshold that represents the mating radius.
Usually, crossover is considered to take place only between pairs of two chromosomes and one offspring is obtained from two parents. The offspring *fights* to enter the current generation with the chromosomes in its personal mating region. Mutation causes only minor perturbation to a chromosome.
The stop condition can refer to a given number of generations (size of t) or to the case when there is not any major improvement achieved after many generations, as in present paper.

4. Diabetes Diagnosis Problem

The Diabetes data set was given to the UCI repository by the Johns Hopkins University. Prior to that, the university selected cases from a larger database owned by the National Institute of Diabetes and Digestive and Kidney Diseases to create it.
All patients in the dataset are females of at least 21 years old, of Pima Indian heritage, living near Phoenix, Arizona, USA. There are eight attributes (either discrete or continuous) containing personal data, e.g., age, number of pregnancies, and medical data, e.g., blood pressure, body mass index, result of glucose tolerance test etc (see Table 1).

No.	Attribute	Interval
1	Number of times pregnant	0…5
2	Plasma glucose concentration in an oral glucose tolerance test	0…199
3	Diastolic blood pressure	0…122
4	Triceps skin fold thickness	0…99
5	2-Hour serum insulin	0…846
6	Body mass index	0…67
7	Diabetes pedigree function	0.078…2.42
8	Age	21…81

Table 1. Attributes and their corresponding ranges in Pima Diabetes problem

The last attribute is a discrete one and it offers the diagnosis, which is either 0 (negative) or 1 (positive). 34.9% of the patients in the dataset are assigned diabetes positive. The total number of cases is 768. The data is complete, according to its documentation; however, there are some 0 values of attributes that were not reported as missing data, but look a bit strange. No replacement or deletion of these values was undertaken in present paper. Some brief statistical analysis is presented in Table 2.

Attribute No.	Mean	Standard deviation
1	3.8	3.4
2	120.9	32.0
3	69.1	19.4
4	20.5	16.0
5	79.8	115.2
6	32.0	7.9
7	0.5	0.3
8	33.2	11.8

Table 2. Statistical analysis of attribute values in Pima Diabetes problem

5. Diabetes Diagnosis Problem approached by Evolutionary Computation

Each chromosome will encode an IF-THEN rule. A chromosome contains therefore nine genes, one for each attribute; first eight genes are real valued while the last is a binary one and it gives the output of the chromosome (diagnosis of the patient encoded).

First, n chromosomes are randomly generated with genes taking values from the intervals presented in Table 1. These chromosomes form the initial population. The quality of these chromosomes is next measured.

5.1 Fitness evaluation

Patients from the database are considered as being vectors of values for the nine attributes, in conclusion, they are further on represented in the same way chromosomes are. The distance between two chromosomes refers to the first eight attributes only; as the values for the eight attributes belong to different intervals, the distance measure has to refer to the bounds of the intervals. Having a chromosome $c = (c_1, c_2, \ldots, c_8, c_9)$ and a patient from the training set $p = (p_1, p_2, \ldots, p_8, p_9)$, the distance between c and p is computed by

$$d(c, p) = \sum_{i=1}^{8} \frac{| c_i - p_i |}{b_i - a_i} \qquad (1)$$

where a_i and b_i represent the lower and upper bounds of the i-th attribute.
When measuring the quality of a chromosome from the current population, one has to refer to all patients in the training set that have the same outcome as it does. The distance between the chromosome to be evaluated and all patients from the training set that have the same outcome as the chromosome has to be minimized in order to achieve good chromosomes (rules) for an outcome. In conclusion, a rule is good if its values for the nine attributes very much suit the values for the attributes patients with same outcome from the training set have.

5.2 Application of rules on the test set

The EGGC technique, as presented in Algorithm 1, is applied then and rules are evolved. Mating and merging take into consideration only chromosomes with same outcomes. Convex crossover and mutation with normal perturbation are used.
Usually, two rules are obtained, one for each outcome. The rules are then applied to the test set. For each patient from the test set, the distance between it and each of the resulted rules is computed as in (1). The outcome for the patient is concluded to be the same with the outcome of the rule that has the lowest value for the distance between it and the patient. The outcome is considered to be correct only if it is the same the patient already had assigned for real in the data set.

5.3. Obtained rules

The rules that are obtained within a run of proposed algorithm are approximately the following:

- IF *number of times pregnant* = 2 AND *plasma glucose concentration* = 106.85 AND *diastolic blood pressure* = 69.99 AND *triceps skin fold thickness* = 21.32 AND *2-Hour serum insulin* = 42.97 AND *body mass index* = 29.99 AND *diabetes pedigree function* = 0.34 AND age = 27 THEN ***healthy***

- IF *number of times pregnant* = 4 AND *plasma glucose concentration* = 137.92 AND *diastolic blood pressure* = 72.02 AND *triceps skin fold thickness* = 27.60 AND *2-Hour serum insulin* = 27.88 AND *body mass index* = 34.19 AND *diabetes pedigree function* = 0.47 AND age = 35 THEN ***ill***

5.4 Experimental results

The values used for the parameters of the evolutionary algorithm are given below in Table 3.

Number of chromosomes	***Mutation probability***	***Mating region***	***Merging radius***	***Mutation strength***	***No improvement times***
100	0.4	0.3	0.03	dif_i / 100	10

Table 3. Parameters of the EGGC algorithm

In previous table, dif_i denotes the difference between the bounds of the interval corresponding to attribute i.

Three ways of choosing the training and test sets were considered. Each time, the two sets were disjoint and there was a correspondence of 75% training - 25% test cases, as established in [8] with respect to the diabetes task.

The first way of splitting the data set was cross-validation. The first 75% of the cases made the training set and the last 25% composed the test set.
The second way was performed according to the best rules that should be used for the diabetes problem, as established in [8] and called the sequential manner of splitting. The data set is sequentially split into 75% training - 25% test cases in such a way that there result four different combinations of these two sets:

- first 75% of the cases for training and last 25% for test
- reversely, first 25% data for test and last 75% for training
- first 50% data for training, next 25% for test and last 25\% for training, as well
- first 25% data for training , next 25% for test and last 50% for training, as well

The third way was random cross-validation. The cases that went into training and test, respectively, were chosen randomly.

The algorithm was applied for 100 runs in all three cases. The results range from 69.5% to 75% accuracy and are given in Table 4. However, in many tests, it was noticed that when the chromosome pool still has four chromosomes left and has not converged yet, a higher accuracy of 80% is obtained. There are actually two rules for each of the two classes and that means there are two subclusters of patient patterns inside the two clusters defined by the presence or absence of diabetes.

Obtained results have been compared to others reported by literature, with respect to the Diabetes Diagnosis problem. Accuracies obtained by other techniques, namely artificial neural networks or one method closer to evolutionary algorithms, ***i.e.*** hybridization of evolutionary computation again with neural networks, range between 62% and 80.7%.

Below, a comparison with such methods that specify exactly the values of variables of the testing environment is reached. Such variables include size of training/test, method of choosing these sets, replacement or deleting of missing data, number of runs of the algorithm. Although neither affirmed nor denied, it is supposed authors did not operate on the 0 values of the data set. It is mentioned however that they did not delete any of the tuples containing missing data. Whenever not specified, the number of runs is presumed to be ten.

In [10] a neural network algorithm to forecast the onset of diabetes mellitus was used. From the 768 samples, an equal number of 170 samples were selected randomly to represent each of the two possible results of diabetes test: positive and negative. The remaining 428 were used as validating samples. The mean of five runs of the best neural configuration was 75.12%.

In [1], a total of 30% of the records were randomly selected as test set. Rules were mined from the remaining 70% of the data. The algorithm was applied ten times. If the authors were to define a baseline accuracy to mean the accuracy obtained by simply assigning the most frequently occurring values to the attributes being predicted, it is 65.1%.

One approach our results can be directly compared to is ([11], [12]). Using a neural networks heuristic, 75% training - 25% test cases, the rules established by Prechelt for choosing training/test sets, 100 trials and no replacement or deletion of missing data, just as in the present algorithm, the mean accuracy was obtained as 65.55%.

Another approach to which an objective comparison of results can be achieved is [15]. A new evolutionary system to evolve artificial neural networks was proposed, cross-validation was used and 30 runs of the algorithm were conducted and again no replacement or deletion of missing data was done. The obtained mean accuracy on the test set was of 77.6%. As the issue of best results produced is concerned, the evolved artificial neural network obtained an accuracy of 80.7%.

A comparison with the results the original algorithm of GC obtains was not performed, as it had been shown that the modified algorithm converges faster to the same optima a problem has [13].

Table 4 shortly depicts the comparison with the last two other results discussed above.

Algorithm	***Number of runs***	***Accuracy (%)***
EGGC with cross-validation	100	75
EGGC with sequential splitting	100	69.67
EGGC with random cross-validation	100	69.5
EGGC best accuracy instead of last	100	80
Neural Network with sequential splitting	100	65.5
Evolved Neural Network with cross-validation	30	77.6
Evolved Neural Network with cross-validation - best result within specified number of runs	30	80.7

Table 3. Results of different techniques for the Diabetes Diagnosis Problem in comparison to EGGC

Although the accuracy of present algorithm is comparable to that of other techniques, proposed algorithm has a clear advantage over the others as it also provides the means to understand, not only to achieve, the decision-making. This is not the case with artificial neural networks. Present algorithm is thus as good as the best-known approaches and additionally provides simple rules that underlie the medical decision process.

6. Conclusions and future work

A multimodal evolutionary algorithm has been applied to the problem of diabetes diagnosis. The data came from the UCI repository of machine learning databases and originally belonged to Johns Hopkins University.
The application correctly classified up to 80% of the cases. Perhaps a better understanding and analysis of the Diabetes data set might improve accuracy. Also, another distance measure or other issues lying at the border of evolutionary computation and classification of diabetes cases might give better results. And maybe a training - validation - testing procedure, as common in genetic programming, might lead to significant improvement. Another thing that we may investigate in the future is the question whether the clustering is over-determined by the eight attributes given. Maybe one or two of the attributes only add noise because they do not really have an influence on the decision. Therefore, a sort of 'incomplete' rule (leaving out some attributes) may achieve even better than 75 or 80%.
Nevertherless, even as it is now, proposed application achieves the goals that were set, those of creating a means of understanding and supporting medical decision making in the question of diabetes diagnosis.

7. References

[1] AU W.-H. and CHAN K. C. C. (2001): *Classification with degree of membership: A fuzzy approach*. In Proc. of the 1st IEEE Int'l Conference on Data Mining, San Jose, CA.

[2] DUMITRESCU D. (2000): *Genetic Chromodynamics*. In Studia Universitatis Babes-Bolyai Cluj-Napoca, Ser. Informatica, vol. 45, no. 1, pp. 39-50.

[3] EIBEN A.E., SMITH J.E. (2003): *Introduction to Evolutionary Computing*. Springer – Verlag.

[4] FOGEL, D. B. (1995): *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ

[5] FOGEL, L. (1999): *Artificial Intelligence through Simulated Evolution* Wiley – Interscience, 2nd edition

[6] GORUNESCU, R. and MILLARD, P., H. (2004): *An evolutionary model of a multidisciplinary review panel for admission to long-term care*. In Proc. of ICCC 2004, I Dzitac, T. Maghiar and C. Popescu, pp. 181-185.

[7] HOLLAND, J. (1992): *Adaptation in Natural and Artificial Systems* MIT Press, 2nd edition

[8] PRECHELT, L. (1994): *Proben 1 – a set of benchmark and benchmarking rules for neural network training algorithms*. University of Karlsruhe, Institute for Program Structures and Data Organization (IPD), Tech. Rep. 21/94.

[9] SCHWEFEL, H. P. (1995): *Evolution and Optimum Seeking*, John Wiley & Sons

[10] SMITH J. W., EVERHART J. E., DICKSON W. C., KNOWLER W. C., and JOHANNES R. (1988): *Using the ADAP learning algorithm to forecast the onset of diabetes mellitus*. In Proceedings of 12th Symposium on Computer Applications in Medical Care, R. A. Greenes, Ed. IEEE Computer Society Press, pp. 261–265.

[11] SMITHIES R., SALHI S., and QUEEN N. (2004): *Adaptive hybrid learning for neural networks*. *Neural Computation*, vol. 16, no. 1, pp. 139–157.

[12] SMITHIES R. G. (2004): *Neural network heuristics for realworld classification – an application to predict cancer recurrence*. Ph.D. dissertation, University of Birmingham, submitted.

[13] STOEAN, C., GORUNESCU, R. and DUMITRESCU D. (2005): *A New Evolutionary Model for the Optimization of Multimodal Functions*. The Anniversary Symposium Celebrating 25 Years of the Seminar "Grigore Moisil" and 15 Years of the Romanian Society for Fuzzy Systems and A.I., 2005, International Participation, Iasi, Romania, Intelligent Systems, Selected Papers, H. N. Teodorescu, J. Watada, J. Gil Aluja, M. Mihaila (Eds.), Performantica Press, pp. 65 – 72.

[14] STOEAN, C., GORUNESCU, R., PREUSS, M. and DUMITRESCU D. (2004): *An evolutionary learning spam filter system*. In Proc. 6th International Symposium, SYNASC04 - Symbolic and Numeric Algorithms for Scientific Computing, D. Petcu, D. Zaharie, V. Negru and T. Jebelean, Eds. Timisoara, Romania: Mirton Publishing House, pp. 512–522, iSBN 973-661-441-7.

[15] YAO X., LIU Y. (1997): *A new evolutionary system for evolving artificial neural networks.* IEEE Transactions on Neural Networks 8(3), pp. 694-713.