# A GA Driven Intelligent System for Medical Diagnosis

Grigorios Beligiannis[1], Ioannis Hatzilygeroudis[1], Constantinos Koutsojannis[1], and Jim Prentzas[1, 2]

[1] Department of Computer Engineering & Informatics, University of Patras
26500 Patras, Greece
[2] Dept of Informatics & Computer Technology, TEI of Lamia
35100 Lamia, Greece

**Abstract.** Manipulation of Male sexual dysfunction (or Impotence) that concerns 10% of the male population requires expertise and great experience. Different diagnostic approaches according to medical as well as to psychosocial and cultural characteristics of patients are usually followed. In this paper, a GA (genetic algorithm) driven intelligent system (GADIS) for diagnosis of male impotence is presented. The rule-base of GADIS has been constructed by using a genetic algorithm for rule extraction from a patients database. Experimental results show a very good diagnostic performance in terms of accuracy, sensitivity and specificity of the intelligent system. The rule-base can be refined each time the patient database is updated over a limit.

**Keywords:** GA driven intelligent system, intelligent medical diagnosis system, hybrid intelligent system.

## 1 Introduction

Human sexual dysfunction (or impotence) is characterized by disturbances in sexual desire and in psychophysiological changes associated with the sexual response cycle in men and women. An estimated 10% of the male population experience chronic Erectile Dysfunction (ED), a type of sexual dysfunction, however as few as 5% seek treatment. ED may affect 50% of men between the ages of 40 and 70 [1]. Most men experience this inability at some stage in their lives, usually by the age of 40, but are not psychologically affected by it. It has many causes, most of which are treatable. It is not an inevitable consequence of aging. Due to experts on this field, more men have been seeking help and returning to normal sexual activity because of improved, successful treatments for ED. Manipulation of the dysfunction requires expertise and great experience. Doctors, even urologists, cannot provide a typical evaluation and treatment strategy. A number of parameters and their possible impacts on the diagnosis and treatment are still under consideration and vogue. So, the creation of an intelligent system to assist non-expert doctors in making an initial diagnosis is quite desirable.

Rule-based intelligent systems are very popular in representing medical diagnosis processes. However, a problem with rule-based systems in general is the difficulty to acquire knowledge from experts, due to a variety of reasons [2]. Therefore, techniques for extracting rules from empirical data have been used. This can be done either

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part I, LNAI 4251, pp. 968 – 975, 2006.

indirectly (by constructing a neural network and then extracting rules from that) [3] or directly, by applying, for example, machine learning techniques (like the ID3, C4,5 etc algorithms) [4]. Recently, genetic algorithms (GAs) have been used for generating rules from empirical data. Most of existing efforts refer to generation of fuzzy rules from empirical data [5, 6]. A few of them have been used on medical data and have produced working systems [7]. However, generation of fuzzy rules is more complicated than that of simple symbolic rules, which are sometimes adequate for solving the classification problem at hand.

In this paper, we present a genetic algorithm driven intelligent system (GADIS) to make diagnoses and suggest appropriate treatments for male impotence. To acquire diagnosis rules from experts for such an application is not an easy task, given that not a consensus has been reached at yet [8]. GAs can help in such a task. GADIS uses a simple GA to evolve simple rules from a medical database with quite satisfactory results. The paper is organized as follows. In Section 2, we present the system architecture. In Section 3, the rule evolution process is described. Section 4 deals with the GA. Section 5 gives some implementation details, whereas Section 6 presents evaluation results. Finally, Section 7 concludes the paper.

## 2 System Architecture

The architecture of GADIS is depicted in Figure 1. It consists of the following units: user interface (UI), genetic algorithm unit (GA), rule extracting and updating unit (REU), medical database (MDB), rule base (RB) and inference engine (IE). Actually, RB and IE constitute the run-time system. GA, RE and MDB are used off-line.

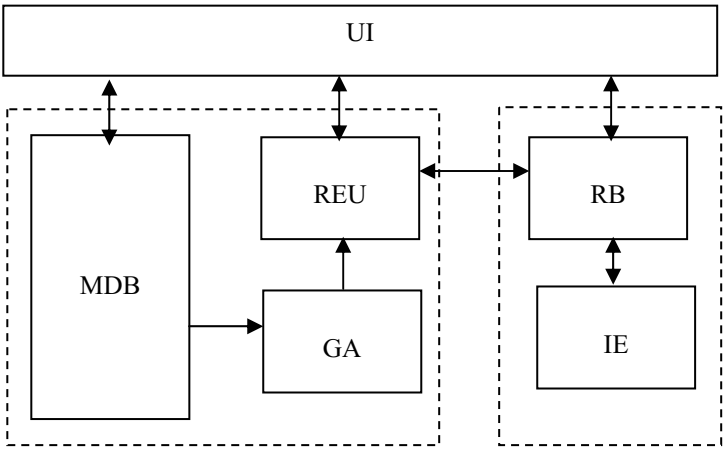


**Fig. 1.** GADIS Architecture

MDB contains patient data organized in groups. Each group refers to one of the possible diagnoses that the system can make. In the off-line mode, GA is successively applied to the data from each group in the MDB and produces a diagnosis classifier

for each data group. The classifiers are then used by REU to produce the evolved rules, which are then suitably transformed and modify the RB. UI is the means for user communication and interference with the system, where and when it is necessary.

## 3 Rule Evolution Process

### 3.1 Medical Knowledge

GADIS concerns diagnosis and treatment of male impotence. There are four possible diagnoses, i.e. reasons, for male impotence: (a) psychogenic, (b) arteriopathy, (c) venoocclusive insufficiency and (d) specific neuropathy. To get to a diagnosis, information about a number of parameters is needed. To extract those parameters we used a statistical analysis method (Pearson analysis) to evaluate which of the parameters recorded in the patient records are significant for the intermediate and final diagnosis. We analyzed 75 patient records from the patient database of the 'Andrology Laboratory' of the Department of Urology of the University Hospital of Patras [8].

We finally came up with twenty two (22) parameters that may play a role in the diagnosis of male impotence, which are the following: onset, non-coital erection, onanism, diabetes mellitus, coronary artery, prostate, neuropathies, chronology, age, depression, smoking, alcohol, blood pressure, overweight, hormonal evaluation, cholesterol, NPT, PIP, Doppler, DICC, neurophysiological (the last five refer to medical tests). Some of them, like overweight, are yes-no parameters, whereas others can take a number of values (e.g. age can take one of nine values that represent age intervals).

### 3.2 The Evolution Process

The rule base evolution process is depicted in Figure 2. From the available medical data, 40% from each diagnosis group is used as a training set, while the remaining 60% is left to be used as a test set.

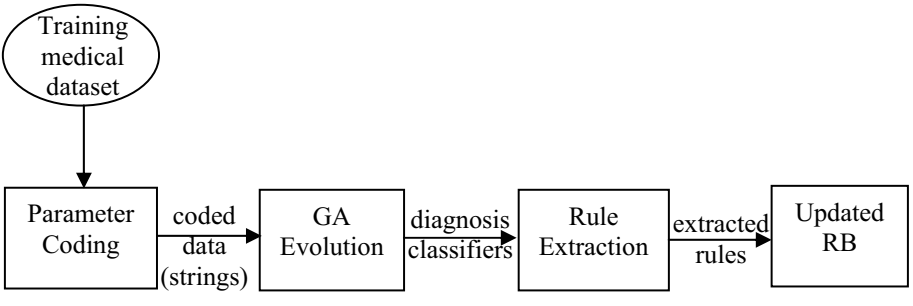


**Fig. 2.** GA Based Rule Evolution Process

The training dataset is given as input to the parameter coding unit (considered as part of the GA unit in the architecture of Figure 1). So, the system codes the training

data in order to create the initial population of the GA and triggers the GA evolution process. After a few generations, the GA provides four different binary strings each corresponding to one of the four different possible diagnoses (psychogenic, arteriopathy, venooclusive insufficiency and specific neuropathy). Each binary string constitutes a diagnosis classifier pattern for the corresponding diagnosis category.

Each such diagnosis classifier comprises a binary string that matches most of the provided training data of a specific diagnosis category. In this way, each diagnosis classifier contains the necessary information required by a rule that would correctly classify most of the provided training data. After that, the rule extraction procedure is taking over. Based on the characteristics of the four different diagnosis classifiers, this procedure results in a set of four different production rules each corresponding to a different diagnosis category. After the extraction of the rules is completed, the rule base is updated. We use the term 'updated' and not the term 'created', because this off-line process can be periodically repeated, given that adequate new patient data has been added to MDB.

### 3.3 Rule Extraction

The rule extraction process is as follows: After the GA evolution process is completed and the classifier for each different diagnosis category is produced, the system tries to identify, for each different classifier, the most important parameters of each diagnosis category. This is done as follows: For each classifier, it tries to find the parameters for which the specific classifier has different values compared to the other three classifiers. For example, the classifier for the psychogenetic case has different values compared to the other three classifiers for the parameters in the places 14 (NON-COITAL), 15 (ONSET), 17 (CHOLESTEROL), 18 (NPT) and 21 (DICC). These five parameters are used in order to construct the final form of the classifier. So, the rule extraction process is able to extract only the important parameters, which are significant for a specific diagnosis category, and omit all the others. After that, the process evolves the set of rules corresponding to different combinations of the significant parameters and finally results in the one that better classifies the cases of the training set.

## 4 The Genetic Algorithm

GAs, in general, operate on binary string structures, analogous to biological creatures (genomes). These structures are evolving in time according to the rule of survival of the fittest, by using a randomized, yet structured, information exchange scheme. Thus, in every generation, a new set of binary strings is created, using parts of the fittest members of the old set [9]. GAs process a binary coding of the parameter space and work on it. This coding (which is an essential part of the GA design procedure) results in formation of binary strings.

In our case, the coding scheme comprises the 22 parameters, specified in the previous section, that represent the data on which diagnosis of male impotence is based on. Each parameter is coded into a binary string. This binary string is in fact the genome of the GA that will be evolved. Each parameter is coded using a 31-bits binary string. For

example, parameter 'age', which can take nine different values (1 to 9), is coded into the binary strings $\underbrace{000...000}_{27}0001$ to $\underbrace{000...000}_{27}1001$ respectively (actually only the four last digits are required, the rest are used because of the implementation tool). On the other hand, parameter 'prostate', which is a yes-no parameter, is coded into binary strings $\underbrace{000...000}_{27}0001$ and $\underbrace{000...000}_{27}0010$ respectively. Each binary string constitutes a gene of the genome of the GA. As a result the whole genome length equals 682 (= 22 x 31) bits (Fig. 3). A set of such genomes constitutes the GA's population.

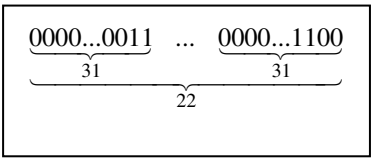


**Fig. 3.** The structure of the genome used in the evolution procedure

The steps of the GA are the classical ones. The initial population is generated using a set of random binary strings created by the parameter coding unit. Selection, crossover and mutation are used for the generation of new population, in each iteration. Evaluation of the objective function is made by applying it to the training set of the available medical data.

```
for i = 1 to m do
    fitness = 0;
    for j = 1 to 22 do
      for k =1 to n do
        if g[i, j] = d[k, j]
          then matches = matches+1;
        endif
      endfor
      if matches >= limit
        then fitness =fitness + 1;
      endif
    endfor
endfor
```

**Fig. 4.** Algorithm for fitness computation

The process is repeated for each different diagnosis category. The termination criterion is the number of generations, that is the evolution process is terminated when a specific number of generations (in our case 5) are completed. While the termination criterion is not met, the whole evolution process is taking place for the current population, i.e. selection, crossover and mutation are applied. When the termination criterion is met, the GA evolution process is terminated and results to one diagnosis classifier for each different diagnosis category.

The objective function estimates the goodness (fitness) of each individual (binary string) and is presented in Figure 4, where *g* is a two-dimension (*m*x22) array representing the *m* genomes of the current population (with 22 genes each representing 22 parameter values), *d* is also a two-dimension (*n*x22) array representing the *n* examples/patterns of the training medical dataset (with 22 parameter values each), *matches* is a counter that counts how many data patterns a gene of the current genome matches, *fitness* is a variable representing the fitness value of the current genome and *limit* is a constant representing the minimum number of matches that a gene must have with the corresponding genes of the patterns of the training dataset to add '1' to its genome's fitness value.

## 5 Implementation Details

As stated, the type of GA used is the classic simple GA. The representation used for the genomes of the genetic population is the classic binary string. As far as the reproduction operator is concerned, the classic biased roulette wheel selection is used. The crossover operator used is uniform crossover (with crossover probability equal to 0.9), while the mutation operator is the flip mutator (with mutation probability equal to 0.001. Except that, the size of the population is set to 50 while the GA uses linear scaling and elitism [10].

The GA was implemented using the C++ Library of Genetic Algorithms GAlib [11] and especially the GASimpleGA class for the implementation of the GA (non-overlapping populations) and the GABin2DecGenome class for the binary string genomes (an implementation of the traditional method for converting binary strings to decimal values). All the experiments were carried out on an Intel Pentium IV 2.7GHz PC with 256 MB RAM.

## 6 System Evaluation

We run GADIS using the test dataset (each case in the test dataset is accompanied by the correct diagnosis, which was really verified). The patient cases included in the test dataset were also presented to three urology residents to make their own diagnoses (based on the given data for each case). The results of GADIS were compared to the diagnoses of the three residents via three metrics, commonly used for this purpose: accuracy, sensitivity and specificity (abbreviated as Acc, Sen and Spec respectively), defined as follows:

$$Acc=(a + d)/(a + b + c + d),$$
$$Sen = a/(a + b),$$
$$Spec = d/(c + d)$$

where, *a* is the number of positive cases correctly classified, *b* is the number of positive cases that are misclassified, *d* is the number of negative cases correctly classified and *c* is the number of negative cases that are misclassified. By 'positive' we mean that a case belongs to the group of the corresponding diagnosis and by negative that it doesn't.

**Table 1a.** Evaluation results for Psychogenic diagnoses

| Metrics | 3 Residents (mean score) | GADIS |
|---|---|---|
| **ACCURACY** | 61 | 88 |
| **SENSITIVITY** | 57 | 94 |
| **SPECIFICITY** | 65 | 78 |

**Table 1b.** Evaluation results for Arteriopathy diagnoses

| Metrics | 3 Residents (mean score) | GADIS |
|---|---|---|
| **ACCURACY** | 63 | 89 |
| **SENSITIVITY** | 57 | 73 |
| **SPECIFICITY** | 72 | 96 |

**Table 1c.** Evaluation results for Venooclusive diagnoses

| Metrics | 3 Residents (mean score) | GADIS |
|---|---|---|
| **ACCURACY** | 62 | 89 |
| **SENSITIVITY** | 57 | 73 |
| **SPECIFICITY** | 67 | 96 |

The comparison results are presented in Tables 1a, b and c and show a clear overall superiority of the performance of GADIS. GADIS is much better as far as all three metrics are concerned. However, the three residents have a better balance between sensitivity and specificity than GADIS for each type of diagnosis individually. Also, this balance is uniform in the three residents, but not in GADIS (Sen is larger than Spec in Psychogenic diagnoses -Table 1a, but the reverse is true in the other two types of diagnoses -Tables 1b and c). So, the three residents tend to better classify negative cases in all types of diagnosis, whereas GADIS does it for two of them, but the reverse for the other type of diagnosis (Psychogenic).

## 7 Conclusions

In this paper, a genetic algorithm driven intelligent system (GADIS) for diagnosis of male impotence is presented. The rule-base of GADIS has been constructed by using a genetic algorithm for rule extraction from a medical database. Experimental results show a very good diagnostic performance in terms of accuracy, sensitivity and specificity of the intelligent system. The rule-base can be refined each time the patient database is adequately updated. A further research step could be, to find a way to generate a a fuzzy rule-base and examine whether it would give better results.

## Acknowledgements

We thank the European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program PYTHAGORAS I, for partially funding the above work.

## References

1. A. Jardin, G. Wagner, S. Khoury, F. Giuliano, H. Padman Nathan and R. Rosen, Erectile Dysfunction, ISSIR,Pr S. Khoury, (Eds)(1999), pp 115-138.
2. Gonzalez, A. and Dankel, D. (1993), The Engineering of Knowledge-Based Systems: Theory and Practice, Prentice-Hall, Inc.
3. Andrews, R., Diederich, J. and Tickle, A. (1995), "A Survey and Critique of Techniques for Extracting Rules From Trained Artificial Neural Networks", Knowledge-Based Systems, vol. 8(6), pp. 373-389.
4. R. S. Michalski, J. D. Carbonell and T. M. Mitchell, "Machine Learning, an AI approach", Tioga Publishing Company, 1983.
5. O. Gordon, F. Herrera and P. Villar (2001), "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base", IEEE Transactions on Fuzzy Systems, 9(4), 667-674.
6. R. A. K. Mehdi, H. Khali and A. Araar (2006), "Generating fuzzy rules for classification problems using genetic algorithms", Proceedings of the IASTED Conference on Artificial Intelligence and Applications (AIA-06), Innsbruck, Austria, Feb. 13-16.
7. A. Tsakonas, G. Dounias, J. Jantzen, H. Axer, B. Bjerregaard and D. G. von Keyserlingk (2004) "Evolving rule-based systems in two medical domains using genetic programming", AI in Medicine, 32, 195-216.
8. Perimenis P, Gyftopoulos K, Giannitsas K, Markou SA, Tsota I, Chrysanthopoulou A, Athanasopoulos A, Barbalias G. A comparative, crossover study of the efficacy and safety of sildenafil and apomorphine in men with evidence of arteriogenic erectile dysfunction. Int J Impot Res. 2004 Jan;16(1):2-7.
9. Z. Michalewicz (1999), Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin.
10. D. E. Goldberg (1989), Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, Mass.
11. GAlib - A C++ Library of Genetic Algorithm Components, Matthew Wall, Massachusetts Institute of Technology (MIT) (http://lancet.mit.edu/ga/).