DETECTION OF DIABETES USING GENETIC PROGRAMMING

Muhammad Waqar Aslam and Asoke Kumar Nandi

The University of Liverpool,
Department of Electrical Engineering and Electronics,
Brownlow Hill, Liverpool, L69 3GJ, U.K.
{m.w.aslam, a.nandi}@liverpool.ac.uk

ABSTRACT

Diabetes is one of the common and rapidly increasing diseases in the world. It is a major health problem in most of the countries. Due to its importance, the need for automated detection of this disease is increasing. The method proposed here uses genetic programming (GP) and a variation of genetic programming called GP with comparative partner selection (CPS) for diabetes detection. The proposed system consists of two stages. In first stage we use genetic programming to produce an individual from training data, that converts the available features to a single feature such that it has different values for healthy and patient (diabetes) data. In the next stage we use test data for testing of that individual. The proposed system was able to achieve 78.5±2.2% accuracy. The results showed that GP based classifier can assist in the diagnosis of diabetes disease.

1. INTRODUCTION

Diabetes is a condition in which your body is unable to produce the required amount of insulin needed to regulate the amount of sugar in the body. This leads to various diseases including heart disease, kidney disease, blindness, nerve damage and blood vessels damage. There are two main types of diabetes. Type 1 results from the body's failure to produce insulin. Presently most persons with type 1 take insulin injections. Type 2 results from insulin resistance, a condition in which cells fail to use insulin properly, sometimes combined with absolute insulin deficiency. To survive, people with type 1 must have insulin delivered by injection or a pump. Many people with type 2 can control their blood glucose by following a healthy meal plan and exercise program, losing excess weight, and taking oral medication. Some people with type 2 may also need insulin to control their blood glucose. However, even if diabetes is under control it still contributes to heart disease.

A physician has to analyze a lot of factors before diagnosing the diabetes which makes physician’s job very difficult. Normally physicians make their decisions by comparing the test results of current patient with some previous patients who also had similar conditions. This depends not only on the physician’s knowledge but depends strongly on the experience of the physician as well. This is not an easy job as the physician has to consider a lot of factors while making a decision. Also there will be demand for a large number of physicians when everybody at risk will need to be tested. As physicians need to have a look at previous results while making their decision, they may need a tool for listing all the previous decisions made on the patients having similar conditions. So a classifier system is needed which can classify that list according to the decisions made by experts. There is no doubt that the most important factors in diagnosis are the data taken from the patient and expert’s opinion on that data but the use of different intelligence techniques and classifiers also helps a lot. That is why the use of classifier systems in medical diagnosis is increasing.

There has been numerous classification techniques used for the classification of diabetes data in the literature. Carpenter and Markuzon, presented an instance counting algorithm ARTMAP-IC and obtained 81% accuracy [1]. Deng and Kasabov obtained 78.4% classification accuracy with 10-fold cross-validation (FC) using ESOM [2]. Polat et al. used principal component analysis and neuro fuzzy inference for diabetes data classification [3]. They also proposed cascade learning system based on Generalized Discriminant Analysis (GDA) and Least Square Support Vector Machine (LS-SVM) for diagnosis of diabetes disease. They achieved 78.21% classification accuracy using LS-SVM with 10x FC and reported 79.16% classification accuracy using GDA–LS-SVM with 10x FC [4]. Kayaer and Yildirim used general regression neural networks to achieve an accuracy of 80.21% [5]. Hasan Temurtas et al. used neural networks for classification of this diabetes data and achieved 82.37% accuracy [6]. All of the above performance values do not report their standard deviations, so their robustness is unknown. There have been many other methods used for the classification of diabetes disease with accuracy between 59.5% and 77.7%. The performance values of these studies can be seen in Polat et al [4].

In this study genetic programming (GP) has been used for the detection of diabetes. GP has been used in the past as a classifier but it has not been used for this problem. Guo and Nandi proposed a method for breast cancer diagnosis using the feature generated by GP [7]. They used a modified Fisher criterion for this purpose. Guo, Jack and Nandi used GP for feature generation. They created new features from original data set using GP. Then they used those features for fault classification in rotating machines [8]. Kishore et al. explored the feasibility of applying GP to multi classification problems [9]. Day and Nandi introduced the idea of comparative partner selection (CPS) in order to emphasize the importance of phenotype in GP [10]. The same technique has been used here for diabetes problem.

2. THE PROPOSED SYSTEM

In this study two variations of genetic programming have been used and their results have been compared with those obtained in literature. One is referred as standard GP and the other is called GP with comparative partner selection (CPS). They differ in the way parents are crossed with each other to produce children. GP Lab tool box is used in this study for the experiments (http://gplab.sourceforge.net/). The proposed system consists of two stages. In the first stage diabetes disease features are given as input to the GP system which gives us a single individual as output using training data. This individual in the output gives us a single feature such that it has different values for healthy and patient. In the second stage we test that individual using test data. The block diagram of the proposed method is given in Fig. 1. The details of these stages are given in the next section.

2.1 Standard Genetic Programming

GP belongs to the class of evolutionary algorithms which emulate Darwinian model of natural evolution. In GP candidates or individuals compete with each other to get transferred to the next generation. In this strategy individuals are tested and they are given a rank or fitness value. This fitness value is then used to compare the individuals and the individuals with better fitness values go to the next generation. GP produces new individuals in every generation and fittest of all the individuals lasts in the end. So this is the game of ‘Selection of the fittest individual’. The new individuals (offsprings) are created by using genetic operators on the current individuals (parents). Genetic operators create new offsprings typically by crossing copies of two parents’ genes (crossover) or by mutating a copy of single parents’ gene (mutation). In this way we get a new generation different from the current generation and with a higher chance of improvement as the parents are the better or fitter individuals in the current generation. The idea of GP can be described by the following equation.

$$g_{t+1} = g_o\ (f\,(g_t))$$

where g_{t+1} is new generation being produced, g_t is the current generation, function ‘f’ chooses the fittest individuals in the current generation and the function g_o applies genetic operators on the current generation to produce new individuals.

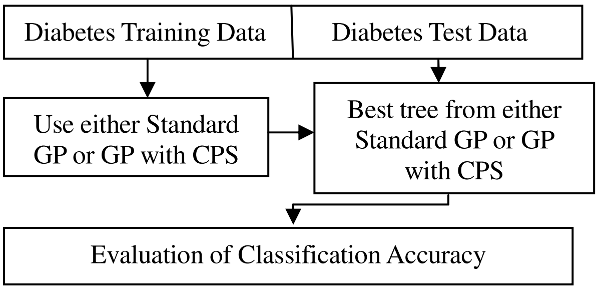
Diabetes Training Data
Diabetes Test Data
Use either Standard GP or GP with CPS
Best tree from either Standard GP or GP with CPS
Evaluation of Classification Accuracy

Fig.1. Block diagram of proposed system

The individuals or the solutions can be represented in different ways but the most common representation is using a tree. So the same tree structure is used here as well. Three main parts of a tree are leaf nodes, intermediate nodes and root. The leaf nodes are the terminals of the tree which are the inputs given to the tree. Intermediate nodes represent different functions which will operate on these terminal values and the root of the tree represents the final output of the tree.

2.2 Basic Terms of Standard GP

2.2.1 Function Pool

The function pool contains all the functions that will be used by the intermediate nodes in the structure of a tree. A function pool can contain different number of functions depending upon the nature of the problem. These functions have different number of inputs while their output is always single. For example for logical problems, logical functions like AND, OR etc. are used and for a non-linear problem nonlinear functions are preferred. The function pool used in this study is given in Table 1.

2.2.2 Fitness Function

The most important parameter that drives the GP algorithm is the fitness value of an individual. This is the value that is used to decide which individuals are going to be transferred to the next generation. Whenever a generation is created each individual is given a fitness value and then according to this fitness value all the individuals are sorted. Then from this sorted list the best individuals are picked. This fitness value is calculated using a fitness function which is user defined function and it depends upon the type of problem.

2.2.3 Genetic Operators

Genetic operators are the programs which operate on the current generation and then produce the next generation which we expect to be better than the current generation. These genetic operators operate similar to any other reproduction process for example sexual and asexual reproduction. Three genetic operators were used for this diabetes problem.

Crossover

This is the genetic operator used mostly in the reproduction process. As the name suggests that two parents are crossed with each other in this process. In this process a node is randomly chosen on both the parents and then both the sub trees from this node downwards are swapped with each other to produce off-springs. An example of a crossover is shown in Fig. 2.

Mutation

Mutation is a different genetic operator which takes a single parent as an input and also returns a single child as output. It alters the parent in some random way to create the child. It randomly chooses a node on the parent and replaces the tree downwards from that node with a randomly generated sub tree.

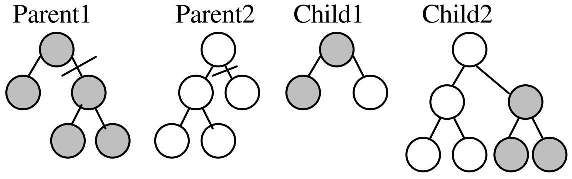

Fig. 2. Example of a Crossover

Reproduction

It is similar to cloning or asexual reproduction. It simply copies the individual into next generation. The amount of reproduction used is usually very low in state of the art systems.

2.3 GP with Comparative Partner Selection (CPS)

In standard GP, genetic operator crossover chooses the individuals based solely on their fitness values. But between these individuals there are no criteria which individuals will be crossed with whom. They are just picked randomly. The idea of choosing the parents only on a single fitness value may be limiting in the sense that individual is judged considering only one aspect. A typical GP problem has a number of training cases. GP individual may be very good in some of the training cases and may not be good in the remaining cases. So assigning an overall fitness ignores the task-wise performance of GP individual. An individual may be strong in one direction and weak in other direction. We need to explore the strengths and weaknesses of an individual.

A simple way to explore strengths and weaknesses of an individual could be to check for which cases in the training set it performs best. The strengths and weaknesses can be considered as which examples are classified correctly and which examples are not. For binary problems we can create a binary string which places a 1 in the binary string, for an example which is correctly solved and 0 in the string for an example which GP has not been able to solve. This binary string is called Binary String Fitness Characterisation (BSFC) where 1 represents strength and 0 represents weakness.

Then in order to remove the weaknesses of an individual crossover is encouraged between the individuals, if one individual shows strength in an area in which other is weak. And crossover is discouraged if both individual have weakness in the same area. This process is shown in Fig. 3 where grey and black colours represent strength and weakness of an individual respectively.

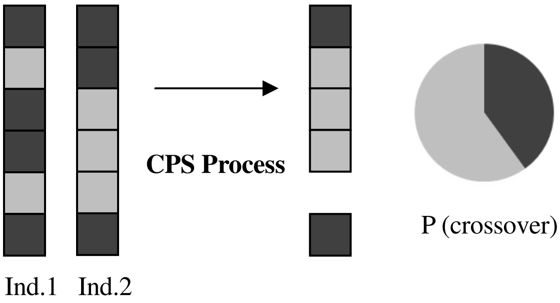

Fig. 3. CPS Process

As the nature of BSFC is binary so logic operations are performed to check whether two individuals should crossover or not. The probability of crossover can be calculated as given in equation 1.

$$P_c(b_1,b_2) = \frac{\sum XOR\ (b_1,b_2)}{\sum XOR\ (b_1,b_2) + \sum NOR(b_1,b_2)} \qquad (1)$$

Here P_c is the probability of crossover, b_1 and b_2 are binary strings of two individuals, and summation represents the summation of each bit in the binary string. The denominator in equation 1 can be replaced by a single NAND operation, so the equation becomes

$$P_c\ (b_1,b_2) = \frac{\sum XOR\ (b_1,b_2)}{\sum NAND(b_1,b_2)} \qquad (2)$$

The procedure for selecting the parents is as follows: a single individual is selected on the basis of fitness value and then second partner is also selected using the same criteria. Then the probability of crossover between the two individuals is calculated using equation 2. This probability will be between 0 and 1. Then a random number between 0 and 1 is generated (just to include some probability in the method) and if that random number is less than the probability of cross over calculated through equation 2, crossover takes place. If it is greater than crossover probability then crossover does not take place between the two. If crossover does not take place then the first parent is kept and second parent is selected again according to fitness value and the same process is repeated to see if they crossover or not. If GP is unable to find a suitable second parent for crossover in N/2 trials (where N is total population), the second parent is selected randomly, ignoring the CPS criteria and crossover takes place between the two. The crossover and mutation probabilities are fixed initially to 0.6 and 0.4 respectively at the start of the experiment. They remain the same throughout in standard GP while they change during the run in CPS. If a parent is not able to find a suitable partner after N/2 iterations and a crossover takes place outside the CPS criteria then the probability of crossover in the current generation is decreased by 1/N and the probability of mutation is increased by 1/N. For the next generation these probabilities go back to their initial values of 0.6 and 0.4. The probability of reproduction is taken as 0.05. Before selection of genetic operators a random variable between 0 and 1 is generated and if that random variable is less than reproduction probability, reproduction is chosen otherwise crossover or mutation is selected according to their probability values.

3. THE EXPERIMENTAL RESULTS

This section first explains the diabetes dataset used in the experiments. It then explains the experiments conducted to solve this problem. Finally, the experimental results and comparison of our results with other results presented in the literature is given.

3.1 Diabetes Disease Dataset

The dataset used in this problem was obtained from UCI Repository of Machine Learning Databases (http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes). National Institute of Diabetes and Digestive and Kidney Diseases is original owner of this data. All patients were Pima-Indian females who were at least 21 years old. There are eight input variables which are shown in Table 1. There is one output variable which has either a value of ‘1’ or ‘0’, where ‘1’ means positive test for diabetes and ‘0’ means negative test for diabetes. There are 268 (34.9%) cases for class ‘1’ and 500 (65.1%) cases for class ‘0’.

There were 8 attribute in total for the diabetes disease dataset– (1) Number of times pregnant, (2) Plasma glucose concentration a 2 hours in an oral glucose tolerance test, (3) Diastolic blood pressure (mm Hg), (4) Triceps skin fold thickness, (5) Hour serum insulin (mu U/ml), (6) Body mass index, (7) Diabetes pedigree function and (8) Age (years). The details of this data set are given in Table 2. Since the mean of different attributes is quite different so the attributes were preprocessed to get a mean value of 0 and standard deviation of 1 before presenting them as inputs to GP algorithm.

Table 1
Parameters used for the experimental work

Parameter	Standard value
No. of Generations	125(Exp I), 500(Exp II)
Population Size	100(Exp I), 100(Exp II)
Function Pool	{+, -, *, /, square, √, sin, cos, asin, acos, tan, tanh, reciprocal, log, abs, negate}
Terminal Pool	8 attributes
Genetic Operators	{Crossover, mutation, reproduction}
Operator Probabilities	{0.6,0.4,0.05}
Tree Generation	Ramped half-n-half
Initial Maximum Depth	6
Maximum Depth	28
Selection Operator	Roulette
Elitism	Half-elitism

3.2 Fitness Value and CPS Criteria

These eight attributes of diabetes data will be given as input to GP algorithm. GP should try to derive such a function so that it can clearly differentiate between these two classes. It will produce a random population of individuals and then will assign each individual a fitness value. It will then choose the individuals which have better fitness. Then this fitness value will be the driving function to arrive at an individual through generations and generations which is able to separate the two classes. The final individual that will be obtained will have some of the attributes of diabetes dataset as an input and then it will perform some actions on those attributes and will give a single output feature. This feature will contain two distributions of data. One for class '1' and other for class '0'. These two distributions of classes should be as apart as possible. Our aim is to increase the distance between these classes (i.e. increase the intra class variance) and decrease the distance between the points within each class (i.e. decrease the inter class variance). So the fitness function should be such which serves this purpose and the fitness function is a key function here.

Table 2
Brief analysis of diabetes dataset

Attribute No.	Mean	Standard Deviation	Min/Max
1	3.8	3.4	0/17
2	120.9	32.0	0/199
3	69.1	19.4	0/122
4	20.5	16.0	0/99
5	79.8	115.2	0/846
6	32.0	7.9	0/67.1
7	0.5	0.3	0.078/2.42
8	33.2	11.8	21/81

The fitness function used here is given in equation 3.

$$\text{Fitness} = \left[\frac{|\, m_1 - m_2|}{\sqrt{(\sigma_1{}^2 + \sigma_2{}^2}}\right]^{-1} \tag{3}$$

where m_1, m_2 are the means of two classes and σ_1, σ_2 represent standard deviations of two classes. This fitness function tries to increase the distance between the means of two classes while minimizing the variance of two classes. As far as the CPS is concerned, the criteria for making a binary string is that the points closer to the mean of the class are given preference over those away from the mean of the class.

3.3 Results and Discussion

3.3.1 Experiment I

In the first experiment 125 generations were used with a population size of 100. The fitness function used is given in equation 3. Fitness function tries to separate the two classes. The lower the fitness, the better the individual and greater is the distance between the two distributions. The data used for training purpose was 90% of dataset and the remaining 10% was used for testing. After going through 125 generations a tree is obtained which is then tested for the test data. The experiment is done 40 times and then the average performance is taken.

Both the standard GP and CPS were tested. The fitness graph is shown in Fig. 4. One can clearly see the difference in performance of standard GP and CPS. As one can see the fitness is sill decreasing and it will continue to decrease if run for more generations. The only problem that restricted in going above 125 generations was tree size. As numbers of generations go above 125, trees created become too big and are difficult to handle. It is evident from Fig. 5. that number of nodes are quite high at 125 generations and increasing still. In order to counter this problem Experiment II was done.

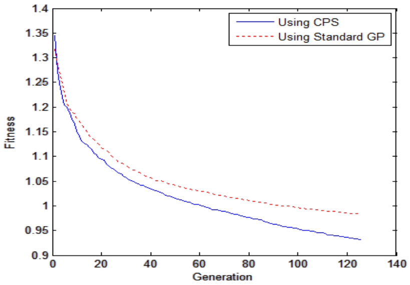

Fig. 4. Average fitness of best of generations over 125 generations for 40 runs of diabetes problem

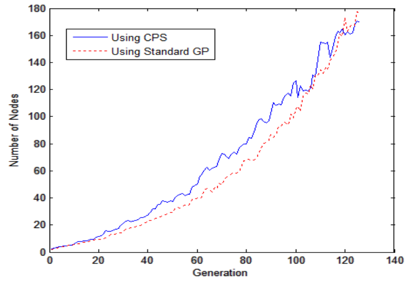

Fig. 5. Average number of nodes of best of generations over 125 generation for 40 runs of diabetes problem

3.3.2 Experiment II

As evident from Experiment I a method was needed which can penalise the runs having big trees. So a condition was set on the number of nodes that if the numbers of nodes of an individual in any run go above a certain number then that run should stop at that generation and next run should start. A cut at 300 nodes was used, so if a generation produces an individual having more than 300 nodes, it will stop there and go to the next run. The numbers of generations used for this experiment were 500 with 100 individuals and the percentage of training and test data was the same as in Experiment I.

3.3.3 Results

First we compare our results from Experiment I with the results obtained so far in the literature. Table 3 gives the classification accuracies of our method and previous methods where classification accuracy represents the percentage of instances correctly classified using test data. GP has not been tried in the past for this problem. As it can be seen from results that our method achieved 78.2±2.5% for standard GP and 78.4%±2.2 for CPS which are quite good.

Table 3
Classification accuracies obtained using our system and other proposed methods in the literature.

Author	Method	Accuracy (%)
Carpenter & Markuzon [1]	ARTMP-IC	81.0
Polat & Gunes [4]	LS-SVM	78.2
	GDA-LS-SVM	79.2
Kayaer & Yildirim [5]	GRNN	80.2
	MLNN with LM	77.1
Hasan Temurtas et al. [6]	MLNN with LM	82.4
	PNN	78.1
Gadaras & Mikhailov[11]*	Fuzzy Classification	92.3
This Study	Standard GP (Exp I)	78.2±2.5
	GP with CPS (Exp I)	78.4±2.2
	Standard GP (Exp II)	77.4±2.2
	GP with CPS (Exp II)	78.5±2.2
Detailed list can be found in Polat et al.[4]		

*They used 50% training and test data partition.

Experiment II was also run for 40 runs. The classification accuracies obtained for Experiment II are 77.4±2.2% and 78.5±2.2% for standard GP and CPS respectively which are not much different from previous results. One can see in Table 3 that most of the methods have mentioned their best performance without any upper and lower limits while the values mentioned in our study show their mean values along with standard deviation. If the best performance is taken then the best performance achieved in this study is 81.8% for standard GP (Exp II) and 84.4% for CPS (Exp II). The values mentioned in the Table are averaged over 40 runs.

4. CONCLUSION

In this study GP and a modified version of GP (CPS) has been used for the classification of diabetes disease. For the first time GP has been used for this problem. The results strongly suggest that GP based classifier can assist in the diagnosis of diabetes disease. GP showed quite good classification accuracies for both variations. Much more can be explored in GP for classification of diabetes disease. We hope more interesting results will follow on further exploration of GP regarding this problem.

ACKNOWLEDGMENTS The Authors would like to thank National Institute of Diabetes and Digestive and Kidney Diseases for diabetes data. Muhammad Waqar Aslam would like to acknowledge the financial support of the University of Azad Jammu & Kashmir, Pakistan.

REFERENCES

[1] Gail A. Carpenter and Natalya Markuzon, “ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases”, Neural Networks, vol. 11, Issue 2, 31 March 1998, pp. 323-336.

[2] D. Deng and N. Kasabov, “On-line pattern analysis by evolving self-organizing maps”, In Proceedings of the fifth biannual conference on artificial neural networks and expert systems (ANNES), 2001, pp. 46–51.

[3] Kemal Polat and S. Gunes, “An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease”, Digital Signal Processing, vol. 17, July 2007, pp. 702-710.

[4] K. Polat, S. Gunes and A. Aslan, “A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine”, Expert Systems with Applications, vol. 34(1), 2008, pp. 214–221.

[5] K. Kayaer and T. Yıldırım, “Medical diagnosis on Pima Indian diabetes using general regression neural networks “, In Proceedings of the international conference on artificial neural networks and neural information processing (ICANN/ICONIP), 2003, pp. 181–184.

[6] T. Hasan, Y. Nejat, T. Feyzullah, “A comparative study on diabetes disease diagnosis using neural networks”, Expert Systems with Applications, vol. 36, May 2009, pp. 8610-8615.

[7] Hong Guo and A. K. Nandi, “Breast cancer diagnosis using genetic programming generated feature”, Pattern Recognition, vol. 39, 2006, pp. 980-987.

[8] Hong Guo, L. B. Jack, A. K. Nandi, “Feature generation using genetic programming with application to fault classification”, IEEE Transactions on Systems, Man and Cybernetics, vol. 35, Feb. 2005, pp. 89-99.

[9] J. K. Kishore et al., “Application of genetic programming for multicategory pattern classification”, IEEE Transactions on Evolutionary Computation, vol. 4, Sep 2000, pp. 242-258.

[10] P. Day, A. K. Nandi, “Binary String Fitness Characterization and Comparative Partner Selection in Genetic Programming”, IEEE Transaction on Evolutionary Competition, vol. 12, 2008, pp. 724-735.

[11] I. Gadaras, L. Mikhailov, “An interpretable fuzzy rule-based classification methodology for medical diagnosis”, Artificial Intelligence in Medicine, vol. 47, 2009, pp. 25-41.