Medical Infobahn for Europe
A. Hasman et al. (Eds.)
IOS Press, 2000

Application of Genetic Programming for the Differential Diagnosis of Acid - Base and Anion Gap Disorders

Andrzej MALOLEPSZY[1], Edward KACKI[2], Tadeusz DOGDANIK[3]

[1]*Institute of Computer Science, Technical University of Lodz*
16/18 Sterlinga str., 90-217 Lodz, Poland,
e-mail: *malolep@ics.p.lodz.pl*
[2]*College of Computer Science in Lodz*
46 Pomorska str., 91-408 Lodz, Poland
[3]*Clinic of Acute Poisonings in Nofer Institute of Occupational Medicine in Lodz*
8 Teresy str., 90-950 Lodz, Poland

Abstract. Differential diagnostics of very complex disorders in the acid -base status and in the accompanying electrolyte balance belongs to the most difficult diagnostic procedures in clinical practice. The goal of our study was to create computer genetic programs, able to aid the diagnostics of acid-base disorders, taking into account a set of laboratory gasometric and electrolyte measurments. Seven (7) single acid-base disorders, eleven (11) double acid-base disorders and six (6) triple complicated disorders with accompanying anion gap alterations were approached by our study. A set of simulated laboratory measurements was prepared, providing 250 results for the evaluation of the fitness function of the designed computer genetic programs, plus some additional results for testing.

1. Introduction

Differential diagnosis of acid-base disorders requires an appropriate interpretation of relationships between electrolyte and arterial blood gas values.

In clinical practice, differential diagnoses are approached by consultants as priority procedures of choice over single diagnoses; computer aided diagnosing should follow the same pattern, especially that appropriate computer programs are able to juxtapose all gasometrical and electrolyte results. Fast-communicating computer programs, providing differential diagnoses of complicated disorders in the acid-base status and in the accompanying electrolyte balance, may be helpful diagnosis-support tools for clinicians in these rather difficult procedures.

The goal of our study was to create computer genetic programs, which could diagnose acid-base disorders, having a set of numerically processed laboratory, gasometrical, and electrolyte measurments as input.

2. Materials and Methods

In clinical practice, a physician can identify twenty four (24) acid-base disorders, taking into account values of serum pH, partial pressure of carbon dioxide (pCO_2), bicarbonate, base excess, sodium, potassium, chloride and anion gap. Apart of the normal acid-base status and according to the table, derived by Horn et al. [2] (Table 1), seven (7) single acid-base disorders, eleven (11) double acid-base disorders and six (6) triple complicated acid-base disorders can be identified, together with accompanying anion gap alterations. Since it would have been rather difficult to find patients, suffering from all posible types of acid-base disorders, a simulation of laboratory measurments was found appropriate. Accordingly, a set of simulated lab tests was designed, the set then divided into two subsets of equal length. One such subset included 250 cases of disorders and was used for evaluation of fitness function through the process of genetic programming. The second subset, also providing 250 inputs, was used for testing of the developed programs.

Table 1. Acid-Base disorders

1. Normal acid-base status
2. High anion gap metabolic acidosis
3. Normal anion gap metabolic acidosis
4. Metabolic alkalosis
5. Acute respiratory acidosis
6. Chronic respiratory acidosis
7. Acute respiratory alkalosis
8. Chronic respiratory alkalosis
9. Acute and chronic respiratory acidosis
10. Acute and chronic respiratory alkalosis
11. High anion gap metabolic acidosis and respiratory acidosis
12. High anion gap metabolic acidosis and respiratory alkalosis
13. High anion gap metabolic acidosis and metabolic alkalosis
14. Normal anion gap metabolic acidosis and respiratory acidosis
15. Normal anion gap metabolic acidosis and respiratory alkalosis
16. Normal anion gap metabolic acidosis and metabolic alkalosis
17. Metabolic alkalosis and respiratory acidosis
18. Metabolic alkalosis and respiratory alkalosis
19. High and normal anion gap metabolic acidosis
20. High and normal anion gap metabolic acidosis and respiratory acidosis
21. High and normal anion gap metabolic acidosis and respiratory alkalosis
22. High anion gap metabolic acidosis and metabolic alkalosis and respiratory acidosis
23. High anion gap metabolic acidosis and metabolic alkalosis and respiratory alkalosis
24. Normal anion gap metabolic acidosis and metabolic alkalosis and respiratory alkalosis
25. Normal anion gap metabolic acidosis and metabolic alkalosis and respiratory acidosis

Our sets of laboratory data – together with gasometrical values, applied by Horn et al. [2], -included the base excess (BE), calculated from the following equation:

$$BE = (1 - 0.014\ Hgb)\ (HCO_3 - 24) + 1.43\ (Hgb + 7.7)\ (pH - 7.4)$$

Table 2 present our example of ten simulated cases suffering from normal anion gap metabolic acidosis used to genetic programming.

Table 2. Normal anion gap metabolic acidosis

Nr of patient 's	PH	pCO_2 mm Hg	HCO_3 act. mmol/l	BE mEq/l	Sodium mmol/l	Potassium mmol/l	Chloride mmol/l	Anion gap mEq/l	Nr of Diagnosis
21	7,33	36	19	-5,5	131	4	100	16	3
22	7,31	36	18	-6,5	130	4	100	16	3
23	7,29	35	17	-7,0	120	5	97	12	3
24	7,25	36	16	-9,0	124	4	98	14	3
25	7,20	37	16	-12,0	118	5	96	9	3
26	7,15	36	13	-15,0	120	5	98	14	3
27	7,10	37	12	-17,0	116	4	98	10	3
28	7,05	37	11	-19,5	121	4	100	14	3
29	7,00	35	9	-22,0	120	4	100	15	3
30	6,90	35	7	-23,0	119	4	100	16	3

All the presented values of gasometrical parameters are arterial blood values. In case of venous blood, 0.03 is to be added to venous pH in order to obtain the arterial value of pH, according to Goldberg et al. [1].

3. Genetic Programming Paradigm

The goal of automatic programming is an automatic creation of computer program that enables a computer to solve a problem. Genetic programming [3] is a domain-independent approach to automatic programming, in which computer programs are evolved to solve, or approximately solve, problems.

Genetic programming is a branch of genetic algorithms, the representation of solution being the main difference between genetic programming and genetic algorithms. Genetic programming creates computer programs in LISP or in scheme computer languages as solution. Within the genetic programming system, structures, which undergo adaptation, are hierarchical computer programs, based on LISP-like symbolic expression. Thus the solution of a problem becomes a search through all the possible combinations of symbolic expressions, as defined by programmer.

The entire process of genetic programming can be divided into a number of sequential steps [4];

1. Creation of a random population of programs, using provided symbolic expressions.
2. Evaluation of each program, together with fitness value assignment, following the criteria of pre-specified fitness function which indicates the problem solution potential of created programs.
3. Using some predefined reproduction technique copying of the existing programs into the new generation.
4. Genetic recombination of the new population of programs with the crossover function from a randomly chosen set of parents.

5. Repetition of Step 2 for the new population so long until the pre-specified termination criterion has been satisfied or a fixed number of generations has been completed.
6. The genetic program with the best fitness within the whole generations stands for the solution of a given problem.

The creation of a genetic programs is the combination of the domain dependent symbolic expressions predefined by the designer. These symbolic expression are divided into two sets, if the expression requires arguments it is placed in a function set otherwise it is placed within terminal set. Some combination of expressions must be sufficient to solve the problem.

4. Genetic Programming in the Problem of Differential Diagnosis

A system designed to assist in diagnosing is shown on Fig. 1.

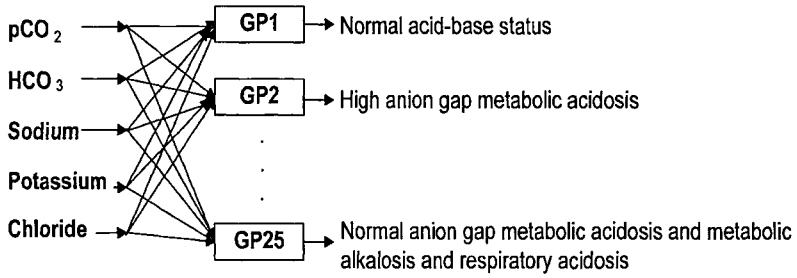

Fig. 1 The structure of diagnosing system

The system consists of twenty-five (25) independent genetic programs, from **GP1** to **GP25,** automatically created as described below. Raw laboratory data, namely: pCO_2 mm Hg, HCO_3 act. mmol/l, Sodium mmol/l, Potassium mmol/l, Chloride mmol/l are presented as the input to the system. As solution, the system should generate a high level signal in one output only, the others outputs to be close to zero. Each of the 25 genetic programs was obtained via the following procedure.

Five major steps in genetic programming:

1. the set of terminals,
2. the set of function,
3. the fitness function,
4. the parameters for running the algorithm,
5. the criterion for designing a result and terminating a run.

The first step in the set up of genetic programming paradigm is the selection of a set of terminals, available to construct the computer program (LISP S-expression), which will attempt to solve a given problem. Such a terminal set contains 6 elements: the first five corresponding to raw laboratory data and the sixth one being the set of integer (1 to 9).

The second step involves the selection of a sufficient set of functions to construct the required computer programs. For the real-valued domain, it may be set of the four arithmetic operations, i.e., {+, -, *, % }, where % denotes modified division operations. It is to be noticed that every possible genetic program can be effective by the LISP S-expression composed of the above-mentioned functions and terminals.

The third step is an identification of the fitness function for the problem to be solved. The required output for *i-th* genetic program $\mathbf{GP}_i$ is equal to 1 for cases number (*i-*

*1)*10+1, ..., (i-1)*10+10* and equal to 0 otherwise, so the fitness function is the sum of the squares of the distance between the real and required outputs for all the 250 cases.

The fourth step is the selection of running parameters of the algorithm. The first attempts were made with the following parameters:

- Population size = 2000,
- Max. number of generations = 50,
- Probability of mutation = 0,001,
- Selecting the parents for crossover – the roulette wheel.

Finally, the fifth step is the selection of termination criterion and of solution identification. A member with the best fitness function, out of all the generations, is the required solution. **GP21** program (LISP S-expression) is presented below as an example of the approach:

*((* (% (* 3 D) (% A D)) (* E (* (* A (+ D B)) (- (* (* (+ E A) (+ B B)) D) (% (* (* (% 9 (% C (- (% B C) (% (% (+ (% D (* (+ C D) A)) (+ B (- E B))) C) (* A (- (+ (+ (- B (- C E)) C) (* (% E C) E)) 7)))))) (- B (% (- D 0) (% D D)))) (+ A B)) 3)))))*

where A, B, C, D, E denotes pCO_2, HCO_3, Sodium, Potassium, Chloride respectively and % denotes modified division. This solution appeared in the 11th generation, its structural complexity being 83.

The testing set of the 250 cases was used to check the ability of developed genetic programs in differential diagnostics.

5. Conclusions

The first version of the diagnosing system, based on genetic programming, still indicates certain problems with the proper identification of disorder; however, it looks fairly promising for the future explorations. Some improvement of the system is probably possible by an addition of the (IF ... THEN ...ELSE...) logical expression to the set of functions. At the same time, the behawiour of the system will be tested with various genetic parameters.

In [5], the authors used an artificial neural network to solve the same problem. We hope, however, that further investigations in genetic programming may eventually provide a better outcome than the results obtained via the neural network.

References

[1] M. Goldberg, S.B. Green, M.L.Moss., et al., Computer-based instruction of acid-base disorders. JAMA, 1973, vol. 223, pp. 269-275

[2] D.L. Horn, J. Radhakrishnan, S. Saini, et al., Evaluation of a computer program for teaching laboratory diagnosis of acid-base disorders. Comp. Biomed. Research, 1992, vol. 25, pp. 562-568

[3] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA, The MIT Press, 1992

[4] A. P. Frazer, Genetic Programming in C++, Technical Report 040, University of Salford, 1994

[5] T. Bogdanik, E. Kacki, A. Malolepszy: Application of an Artificial Neural Network for the Differential Diagnosis of Acid-Base and Anion Gap Disorders, Proc. Of The Fifth Conf "Computers in Medicine", Lodz, Poland, 1999, pp. 75-80