A FEATURE EXTRACTOR OF SEISMIC DATA USING GENETIC ALGORITHMS

A.V. Adamopoulos[1] *S.D. Likothanassis*[1,2] *E.F. Georgopoulos*[1,2]

[1] Department of Computer Engineering and Informatics
University of Patras, Rion, 265 00 Patras, GREECE
Tel: +30 61 997755, fax: +30 61 997706

[2] Computer Technology Institute
Kolokotroni 3, 262 21 Patras, GREECE

e-mail: adamopul@ceid.upatras.gr, likothan@cti.gr, georgops@ceid.upatras.gr

ABSTRACT

A novel signal analysis technique is presented and applied for the analysis of seismic data. The main task of the method is the extraction of feature information of a given timeseries. This task is accomplished by the implementation of a specifically designed Genetic Algorithm that is utilised for the evolution of conditional sets. The term conditional set refers to a set of boundary conditions. These boundary conditions must be fullfilled by subsequent samples of the timeseries. The implemented algorithm was applied on seismic data in order to investigate for the existence of conditional sets that appear more frequently than others. It was found that there exist conditional sets with high probability of appearance. Furthermore, it was found that some of the conditional sets are followed by data samples that apper small deviations from their mean value. It is proposed that conditional sets that combine these two properties (high probability of appearance and small deviation of the next data sample) can account for short-term prediction of seismic events.

1 INTRODUCTION

Real world signal analysis is a difficult task due to a number of reasons. Among them one has to consider the dynamical complexity of the physical processes underlying the experimental measures, as well as the presence of noise, either external or ambient. Thus, although there have been major efforts for the development of automatic methods for finding significant and interesting patterns in complex data, in general, this problem remains open. A novel technique to face that kind of problems was presented by Packard and co-workers [5, 2]. This method utilises Genetic Algorithms (GA) that evolve conditional sets. The main concept of this method is the GA to find out conditional sets that appear more frequently than others in the data set under consideration. The investigation of this task will allow to extract the existence of any underlying dynamical features and characteristics of the timeseries. In addition, for any existing conditional set the average and the standard deviation of the next data samples (successors of the conditional set) were calculated. If the standard deviation of the successors is small compared to the average of these samples, (i.e. the relative error is small), then the particular conditional set can account as a good short-term predictor of the next value of the time series. The accuracy of this short-term prediction can be quantified by means the relative error. In this manner, it is possible to obtain some information on the structural characteristics and dynamical features of the underlying physical system. The method can easily be implemented on a parallel environment.

2 MATERIAL

The original data that were considered in the present work refer to recordings of seismic activity in the Hellenic region. These recodings (raw data) were preprocessed in such a way that the time intervals (time gap) between consequitive seismic events were estimated. Thus, the data sets that were examined in the present work refer to the time intervals of recordings of consequent seismic events in the Hellenic region. An extract of the generated time interval data set is shown in Figure 1.

3 THE GENETIC ALGORITHM

3.1 Conditions and Conditional Sets

Considering a data set of N samples $x_i, i = 1, \ldots, N$, a condition C is of the form:

$$C_i = (l_i < x_i < u_i) \tag{1}$$

where l_i and u_i are the lower and upper bounds of the condition respectively.

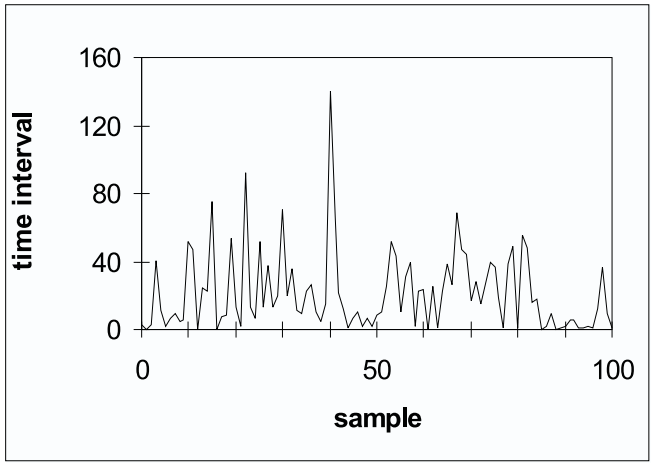

Figure 1: An extract of 100 samples of the time-intervals time series.

A conditional set S of length L consists of a number of L conditions that are coupled with the logical operator $\wedge$ (Boolean AND). Thus, a typical form of a conditional set consisting of four ($L = 4$) conditions is :

$$S = C_1 \ \wedge \ C_2 \ \wedge \ C_3 \ \wedge \ C_4 \tag{2}$$

An numerical example of a conditional set is the following:

$$S = (340 < x_1 < 470) \ \wedge \ (580 < x_2 < 610)$$
$$\wedge \ (782 < x_3 < 944) \ \wedge \ (210 < x_4 < 250) \tag{3}$$

3.2 Overview of the Genetic Algorithm

A GA consists of a population of individuals and some genetic operators that apply on that population. Mainly, two types of genetic operators are implemented: reproduction (crossover) and mutation operators.

Schematically, in C++ pseudo-code, the GA operates as follows:

```
Population.Setup();
while (not termination_condition)
{
Population.Fitness_Estimation();
Population.Reproduction();
Population.Mutation();
}
Population.Report_Results();
```

In brief, each conditional set S is considered as an individual for the GA population. Individuals are randomly setup. The lower and upper bounds of each one condition of any conditional set are randomly selected to be in the range of the minimum and the maximum of the time series. In the following, the fitness of each individual is calculated. The fitness of each individual is an estimation of how good solution is this particular individual of the problem under consideration. A more detailed mathematical description of the fitness function is presented in the following section (3.3). After the setup procedure the individuals of the original generation are sorted in descending order with respect to their fitness. The higher the individual's fitness, the more probable that this particular individual will survive and reproduce, i.e. it will mate with another individual. Two selected mating individuals (parents) produce an offspring (child) for the next generation (reproduction procedure). In the GA under study, reproduction is realised by means of two different crossover operators: the *one-point crossover* and the *uniform crossover*. Description of the two crossover operators is given in section 3.4. After the termination of the reproduction phase and the generation of a new population of individuals, the mutation operators are applied on each one condition of each one offspring. Description of the four mutation operators implemented here is given in section 3.5. With the end of the mutation procedure a new generation of individuals has been created and the whole procedure can be re-applied, begining with the estimation of the fitness of the individuals of the new population.
As termination condition it can be considered the evolution of the GA up to a given number of generations, a convergence condition of the individuals, etc. It is believed that after a sufficient number of generations the GA converges to the optimal solution of the given problem [1, 3, 4].

3.3 Fitness Function

A number of different fitness functions were tested using a number of different GAs. The first of them refer to the number of appearances of a given conditional set in the timeseries. On the termination of that GA it was found the conditional sets that appear more often than others in the timeseries. Thus, considering $F(S_i)$ as the frequency of appearance of a conditional set S_i, the first fitness function f_1 is:

$$f_1(S_i) = F(S_i) \tag{4}$$

A second GA was used in order to find out the conditional sets that are followed by recordings with the smaller standard deviation. To obtain this task, an addition was made to the previous methodology: each time a conditional set was found in the timeseries the next value (successor) in the timeseries was saved; the standard deviation of all these successor values were calculated, and as fitness function was considered the inverse of the standard deviation. Thus, considering $\sigma(S_i)$ as the standard deviation of the successor values following the conditional set S_i the second fitness function considered here is:

$$f_2(S_i) = \sigma(S_i)^{-1} \quad (5)$$

This second GA was able to reveal conditional sets that are followed by data values with the smaller fluctuations.

3.4 Crossover Operators

The one-point crossover
In the application of the one-point crossover a number called *crossover_point* is randomly selected, sharing each parent in two parts. For the *crossover_point* stands that:

$$0 < crossover\_point < L \quad (6)$$

where L is the length of the conditional set, i.e. the number of conditions that constitute this individual. The offspring is generated by mixing the first part of the first parent with the second part of the second parent. To give a numerical example, assume S_1 and S_2 as parents with $L = 4$:

$$S_1 = (340 < x_1 < 470) \wedge (580 < x_2 < 610)$$
$$\wedge(782 < x_3 < 944) \wedge (210 < x_4 < 250)$$
$$S_2 = (248 < x1 < 311) \wedge (461 < x2 < 503)$$
$$\wedge(218 < x3 < 237) \wedge (512 < x4 < 561)$$

and $crossover\_point = 3$, the generated offspring is:

$$(340 < x_1 < 470) \wedge (580 < x_2 < 610)$$
$$\wedge(782 < x_3 < 944) \wedge (512 < x_4 < 261)$$

The uniform crossover
On the other hand, using the uniform crossover, the offspring is generated by mixing the even-numbered conditions of the first parent and the odd-numbered conditions of the second parent (counting of conditions starts from zero). Thus, assuming again S_1 and S_2 as parents the generated offspring is:

$$(340 < x_1 < 470) \wedge (461 < x_2 < 503)$$
$$\wedge(782 < x_3 < 944) \wedge (512 < x_4 < 261)$$

3.5 Mutation Operators

Four are the mutation operators used in the implemented GA. The *up-shift operator*, the *down-shift operator*, the *expand operator* and the *shrink operator*.

The up-shift operator
The up-shift operator increases the lower and the upper bounds of a condition by the same amount, so that the mean value of the condition is shifted to a higher value, but the interval of the condition (i.e. the distance of the upper bond from the lower bound) remains unchanged. An example of the up-shift mutation is the following:

$$(580 < x_i < 610) \rightarrow (650 < x_i < 680)$$

where the bounds and the mean value of the condition were increased by 70.

The down-shift operator
The down-shift operator works in the opposite direction by decreasing the lower and the upper bound of a condition by the same value, so that the mean value of the condition is decreased, while the interval of the condition remains unchanged. In the following example the down-shift mutator decreases the bounds and the mean value of a condition by 40.

$$(580 < x_i < 610) \rightarrow (540 < x_i < 570)$$

The expand operator
The expand operator expands the interval of one condition by increasing the upper bound and decreasing the lower bound by the same value. The mean value of the condition remains unchanged, as it is shown in the folowing example:

$$(580 < x_i < 610) \rightarrow (550 < x_i < 640)$$

where the condition was expanded by 30 in both directions.

The shrink operator
On the opposite of the expand operator, the shrink operator shrinks the interval of a condition by increasing the lower bound and decreasing the upper bound by the same value. The mean value remains unchanged. Example of the action of the shrink mutation follows:

$$(580 < x_i < 610) \rightarrow (590 < x_i < 600)$$

where the condition's bounds were moved by 10 closer to the mean value.

4 RESULTS

A number of runs of the GA's were performed on the experimental timeseries. Typical results have the form:

$$(340 < x_1 < 470) \wedge (580 < x_2 < 610)$$
$$\wedge(782 < x_3 < 944) \wedge (210 < x_4 < 250) \rightarrow 5136$$

which means that the conditional set shown above was found 5136 times in the data set. A plot of the frequency of appearance of the fittest conditional set (maximum

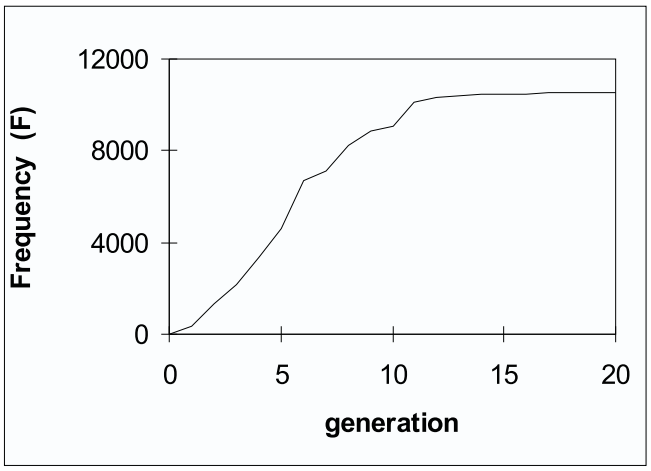

Figure 2: Evolution of f_1, the frequency of appearance F of the fittest conditional set.

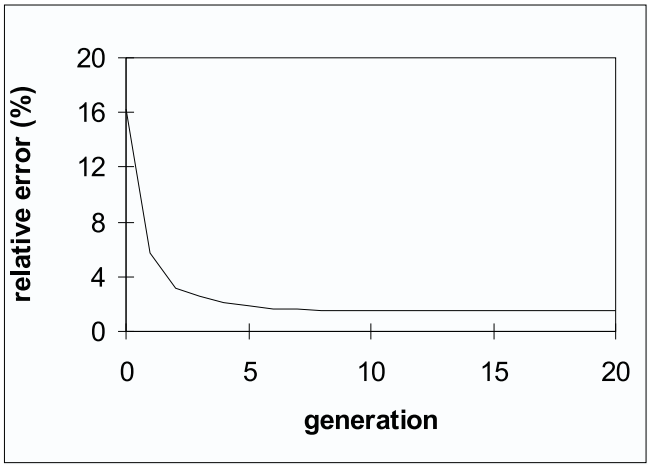

Figure 3: Evolution of the % relative error of the successors of the fittest conditional set.

fitness value of f_1) per generation is shown in Figure 2.

With respect to the GA that considers the standard deviation of the successor values as the fitness function, typical results are of the form:

$$(340 < x_1 < 470) \wedge (580 < x_2 < 610)$$

$$\wedge (782 < x_3 < 944) \wedge (210 < x_4 < 250) \rightarrow 414 \pm 7$$

which means that the average of the data values following the above conditional set was found 414, whereas the standard deviation σ of that values was 7, resulting to a relative error about 1.7 %. A plot of the % relative error ($\sigma/average$) of the successors of the fittest conditional set per generation is shown in Figure 3.

5 CONCLUSIONS

It is for the first time that a GA with the specific design presented above is applied on seismic data. GA is well understood as powerful searching tools, ideal for application on problems characterised by high complexity, as the extraction of structural and dynamical features of real world timeseries. Although the above results are preliminary, we believe that a further application of the methodology extended above, fruitful, promising and probably capable for short-term prediction.

References

[1] Goldberg D. E. (1989) "Genetic Algorithms in Search, Optimization and Machine Learning" Addison-Wesley, Reading, Mass.

[2] Meyer T.P. and Packard N.H. (1991) "Local Forecasting of High Dimensional Chaotic Dynamics. University of Illinois, Technical Report CCSR-91-1.

[3] Michalewicz Z. (1996) "Genetic Algorithms + Data Structures = Evolution Programs" Springer Verlang, N.Y.

[4] Mitchell M. (1996) "An Introduction to Genetic Algorithms" MIT Press.

[5] Packard N.H. (1990) "A genetic learning algorithm for the analysis of complex data" Complex Systems 4, 543-572.