HIGH ENERGY PHYSICS EVENT SELECTION WITH GENE EXPRESSION PROGRAMMING

Liliana Teodorescu, Brunel University, West London, UK

Abstract

Gene Expression Programming is a new evolutionary algorithm that overcomes many limitations of the more established Genetic Algorithms and Genetic Programming. Its application to event selection in high energy physics data analysis is presented. The algorithm allows signal/background separation with an accuracy of $92 - 95\%$ for data samples with a signal to background ratio between 0.25 and 5.

INTRODUCTION

Evolutionary Algorithms, such as Genetic Algorithms (GA) [1] and Genetic Programming (GP) [2], are inspired by biological evolutionary theories. In these algorithms the solutions to a problem are represented as individuals which evolve throughout generations due to the interactions with other candidate solutions and the application of genetic operators that create genetic variation. Individuals are entities which encode candidate solutions to a problem (GA), or computer programs as candidate solutions to a problem (GP).

Gene Expression Programming (GEP)[3] is a new Evolutionary Algorithm that overcomes some of the limitations of GA and GP. The individuals of a population are non-linear entities of different size and shape (expression trees) which are encoded as strings of fixed length (chromosomes). This separation and the structural organisation of the chromosomes allow unconstrained genetic modifications, while always producing valid expression trees. These characteristics allow GEP to outperform GP by more than two orders of magnitude for symbolic regression problems, and more than four orders of magnitude for classification problems [4].

The first application of GEP to particle physics data analysis was presented in [5]. The present study is an extension of that analysis for more complex datasets and more complex chromosome configurations.

GEP FUNDAMENTALS

The GEP algorithm is described in detail in [3] and [4]. Here only the main ideas are summarised.

The algorithm starts with the problem definition, the encoding of the candidate solution of the problem into a chromosome and the definition of the fitness function that describes how good the candidate solution is for the problem at hand. Then an initial population of chromosomes is randomly generated, the chromosomes are translated into expression trees, and the fitness function is evaluated for each chromosome. If a solution of adequate quality is not found, a set of chromosomes is selected and reproduced, creating a new generation of chromosomes. The process is repeated until an optimal solution to the problem is found or a given number of generations is produced.

The candidate solution is encoded into a chromosome composed of one or more genes of equal length. A gene is divided into a head composed of terminals (variables and constants) and functions, and a tail composed only of terminals. The length of the head (h) is an input parameter of the algorithm while the length of the tail (t) is given by:

$$t = h(n-1) + 1 \tag{1}$$

where n is the largest arity of the functions used in the gene head.

The list of functions and variables to be used in a gene is input information for the algorithm, while the constants are created by the algorithm itself in a range specified by the user.

Each gene is translated into an expression tree using simple rules as described in [3]. In the case of multigenic chromosomes, the expression trees are connected with a linking function defined by the user. The selection of the chromosomes to be reproduced is made using the elitism (unchanged replication of the best fitted chromosome into the next generation) and the roulette-wheel [6] methods. The chromosomes selected with the roulette-wheel method are reproduced applying on them a set of genetic operators:

- replication (cloning) - copies exactly the chromosome into another chromosome;
- mutation - randomly changes a symbol of a chromosome into another symbol (preserving the rule that the tails contain only terminals);
- transposition - randomly copies a part of the chromosome to another point in the gene head of the same chromosome;
- recombination (cross-over) - exchanges parts of a pair of randomly chosen chromosomes.

METHODOLOGY

In this study GEP is applied to an event selection problem using a statistical learning approach. The algorithm is used to extract selection criteria for the signal/background classification.

The study was performed using APS (Automatic Problem Solver) 3.0 [7], a Windows based commercial software

Table 1: GEP input functions - Set 1

Function	Definition
AND1	if $x < 0$ AND $y < 0$ then 1 else 0
AND2	if $x \geq 0$ AND $y \geq 0$ then 1 else 0
OR1	if $x < 0$ OR $y < 0$ then 1 else 0
OR2	if $x \geq 0$ OR $y \geq 0$ then 1 else 0
IFB1	if $x < y$ then 1 else 0
IFB2	if $x > y$ then 1 else 0
IFB3	if $x \leq y$ then 1 else 0
IFB4	if $x \geq y$ then 1 else 0
IFB5	if $x = y$ then 1 else 0
IFB6	if $x \neq y$ then 1 else 0

Table 2: GEP input functions - Set 2

Functions
$+, -, \times, /$
$<, >, \leq, \geq$
$=, \neq$
Pow, Pow10
Sqrt, Inv
Ln, Log, Exp
Abs, Neg, Mod
Sin, Cos, Tan
Asin, Acos, Atan

for function finding, classification and time series analysis with GEP.

The data sample was a set of Monte-Carlo events for K_S production in e^+e^- interaction at ≈10 GeV from the BaBar experiment [8]. The classification rules were extracted from training samples containing 5000 events and tested with other samples containing a similar number of events. The number of events was limited by the processing capabilities of the APS 3.0 software.

Each event contains a set of variables usually used in a standard cut-based analysis for $K_S \to \pi^+\pi^-$ selection:

- $doca$ - distance between the two π daughters of K_S at the point of closest approach,
- R_{XY} - radius of the cylinder that defines the e^+e^- interaction region,
- $|R_Z|$ - half length of the cylinder that defines the e^+e^- interaction region,
- $|cos(\theta_{hel})|$ - absolute value of the cosine of the K_S helicity angle,
- SFL - K_S signed flight length defined as the projection of the vector from interaction point to K_S decay vertex on the K_S momentum direction,
- $Fsig$ - statistical significance of the K_S flight length,
- $Pchi$ - χ^2 probability of K_S vertex,
- $Mass$ - K_S reconstructed mass.

These variables, together with the functions listed in Table 1 and Table 2 (given as input to the algorithm), and with floating point constants in the range of $(-10, 10)$ (range also given as input to the algorithm) were used to construct the GEP chromosomes.

Table 3: Classification accuracy of solutions found by GEP for training datasets with $S/B = 0.25, 1$ and 5

Head	Acc.(%)	Acc.(%)	Acc.(%)
	S/B=0.25	S/B=1	S/B=5
1	83.34	85.82	92.00
2	90.76	90.30	92.80
3	94.88	91.66	91.98
4	94.88	92.22	92.80
5	95.04	92.10	92.80
7	95.04	92.10	92.94
10	95.36	91.66	92.98
20	95.50	92.14	92.80

Other GEP input parameters were: the length of the gene head (between 1 and 20), the number of chromosomes per generation (100) and the maximum number of generations (between 3000 and 20000, depending on the complexity of the chromosomes). The genetic operator rates were kept as recommended in [4]: 0.044 for mutation, 0.3 for transposition and 0.1 for recombination.

The fitness function was the number of hits (the number of events correctly classified as signal or background).

The performance of the algorithm was analysed in terms of the classification accuracy parameter (Acc) defined as the the ratio of the total number of events correctly classified (signal and background) to the total number of events of the sample.

ANALYSIS AND RESULTS

Two analyses, with two sets of input functions, were performed on datasets with the signal to background ratio (S/B) equal with 0.25, 1 and 5.

Analysis with a small number of input functions

In this analysis the set of ten functions listed in Table 1 (Set 1) was used as input information for the algorithm, together with the variables and the constants range listed in the previous section.

The classification accuracy of the solutions found by GEP using chromosomes with the length of the gene head varying between 1 and 20 is presented in Table 3, for all training datasets analysed. It can be seen that the classification accuracy is high, over 90%, in almost all cases.

The solutions found by the algorithm can be interpreted as cuts, similar with those used in a standard cut-based analysis. Table 4 summarises the GEP classification criteria for the dataset with $S/B = 0.25$.

An example of a solution found by the algorithm for a chromosome with the length of the gene head equal to 10 is depicted in Figure 1 as an expression tree. The variation of the fitness of the best individual per generation as a function of the number of generations is shown in Figure 2

Table 4: Classification criteria found by GEP for the training dataset with $S/B = 0.25$

Head	Selection criteria
1	$Fsig > 9.93;$
2	$Fsig > 8.80, doca < 1;$
3	$Fsig > 3.67, Rxy \leq Pchi;$
4	$Fsig > 3.67, Rxy \leq Pchi;$
5	$Fsig > 3.63, \|Rz\| \leq 2.65, Rxy < Pchi;$
7	$Fsig > 3.64, Rxy < Pchi, Pchi > 0;$
10	$Fsig > 5.26, Rxy < 0.19, doca < 1, Pchi > 0;$
20	$Fsig > 4.10, Rxy < 0.20, SFL > 0.2,$ $Pchi > 0, doca > 0, Rxy \leq Mass;$

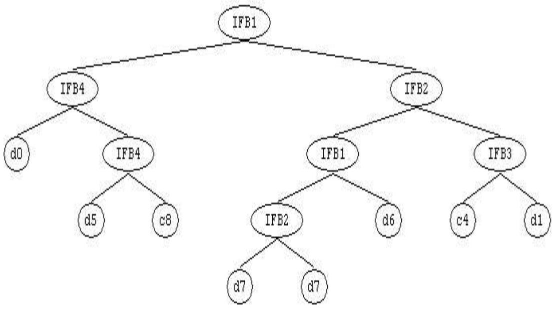

Figure 1: Expression tree corresponding to the solution found with a gene with head length= 10 and with the Set 1 input functions ($S/B = 0.25$ training dataset)

for the same solution. A high quality solution is found very early in the evolution process, in less than 500 generations. The plateau of the distribution indicates the convergence of the search process.

The highest classification accuracy is obtained with a chromosome with the length of the gene head equal to 20. The corresponding selection criteria are very similar to those used in a standard cut-based analysis [9]:

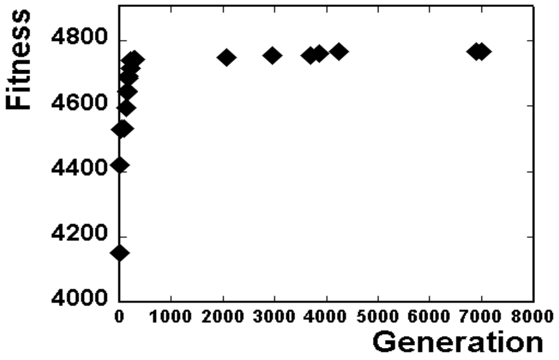

Figure 2: Fitness of the best individual per generation as a function of the number of generations (head length = 10, Set 1 input functions, $S/B = 0.25$ training dataset)

- $Fsig \geq 4.0$
- $Rxy \leq 0.2cm$
- $SFL \geq 0cm$
- $Pchi > 0.001$
- $doca \leq 0.4cm$
- $|Rz| \leq 2.8cm$

In the standard analysis the selection criteria were optimised in order to maximise the statistical significance of the signal while in GEP analysis the selection criteria were optimised to maximise the classification accuracy. This difference in the optimisation procedure explains why GEP does not find the last two selection criteria used in the standard analysis. In addition, these two last cuts do not have a major influence on the selection, reducing the background by an additional 0.3% and the signal with an additional 1%, compared to the previous four cuts.

The $doca > 0$ and $Rxy \leq Mass$ cuts found by GEP does not have any influence on the selection. They are found early in the search process and are superseded by more powerful selection rules (for example $Rxy \leq 0.2$) found later in the evolution process. They remain, however, in the final solution as the algorithm, in the current development, does not have the ability to eliminate this redundancy. Mechanisms for penalising the redundant rules can, however, be considered and developed.

It is also interesting to note that GEP finds alternative powerful selection rules, as for example $Rxy \leq Pchi$, that are not normally used in a standard cut-based analysis.

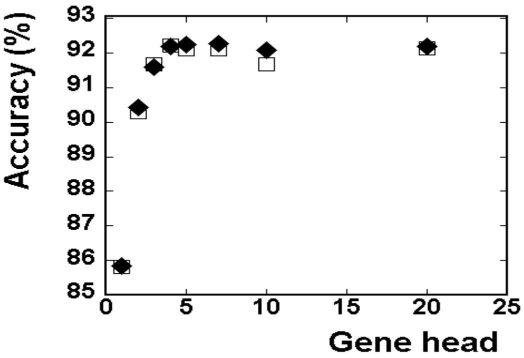

Figure 3: Classification accuracy as a function of the length of the gene head for the training (open squares) and test (full diamonds) data samples with $S/B = 1$ (Set 1 input functions)

The selection criteria found by GEP on the training data were tested on independent test data, with similar signal to background ratios. The classification accuracy obtained on the test data is very similar to that obtained on the training data, in all configurations. An example is shown in Figure 3 where the dependence of the classification accuracy on both training and test datasets as a function of the length of the gene head is presented for the datasets with $S/B = 1$. This close similarity indicates that the solution found by GEP has a good generalisation power.

Analysis with a large number of input functions

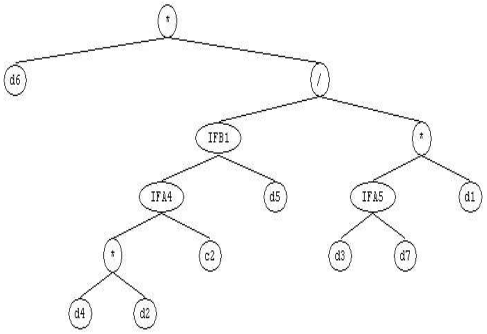

Figure 4: Expression tree corresponding to the solution found with a gene with head length= 10 and with the Set 1 and Set 2 input functions ($S/B = 0.25$ training dataset)

The analysis presented in the previous section was repeated using the 36 input functions listed in Table 1 and Table 2 (Set 1 and Set 2). The solutions found by GEP are more complex but they do not improve the classification accuracy. Figure 4 shows the solution found by the algorithm with a chromosome made of one gene with the length of the head equal to 10 on the training dataset with $S/B = 0.25$. The classification accuracy obtained with this solution was around 95.00%.

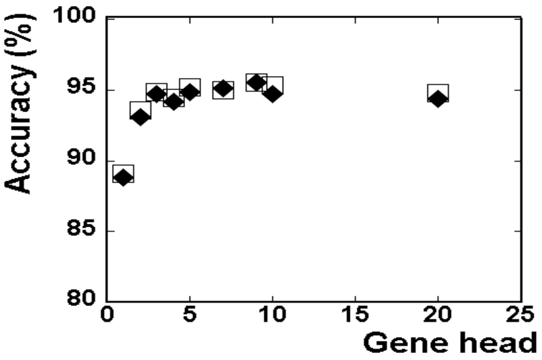

Figure 5: Classification accuracy as a function of the length of the gene head for the training (open squares) and test (full diamonds) data samples with $S/B = 0.25$ (Set 1 and Set 2 input functions)

Similar results were obtained for different chromosome configurations and for all datasets analysed. An example is shown in Figure 5 that represents the dependence of the classification accuracy, on both training and test datasets with $S/B = 0.25$, as a function of the length of gene head. The most complex chromosomes give an accuracy around 95%, similar to that found in the analysis with a small number of input functions.

CONCLUSIONS

The present study of GEP application for event selection in high energy physics data analysis indicates this algorithm is a potential powerful method for fast and automatic identification of powerful selection criteria.

A signal/background separation with an accuracy of 92-95% was obtained on datasets with $S/B = 0.25, 1$ and 5.

The algorithm seems also to have potential to discover new correlations between variables, potential that can be exploited in searches for both known and unknown physics processes.

For the problem and the data samples analysed here, the increase of the number of the input functions was found not to improve the signal/background classification accuracy.

These promising results motivate further developments, applications and software implementations of the GEP algorithm.

ACKNOWLEDGEMENTS

Support from the UK Particle Physics and Astronomy Research Council (PPARC) is gratefully acknowledged.

REFERENCES

[1] J.H. Holland, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, 1975.

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

[3] C. Ferreira, “Gene Expression Programming: A New Adaptive Algorithm for Solving Problems”, Complex Systems, 13 (2001) 87.

[4] C. Ferreira, “Gene Expression Programming: Mathematical Modelling by an Artificial Intelligence”, Angra do Heroismo, Portugal, 2002.

[5] L. Teodorescu, “High Energy Physics Data Analysis with Gene Expression Programming”, 2005 IEEE Nuclear Science Symposium Conference Record, N8-2, 143.

[6] D.E. Goldberg, “Genetic Algorithms in Search, Optimisation, and Machine Learning”, Addison-Wesley, 1989.

[7] Automatic Problem Solver - APS 3.0, http://www.gepsoft.com.

[8] B. Aubert et.al.,”The BaBar detector”, Nuclear Instruments and Methods in Physics Research, A479 (2002), 1.

[9] B. Aubert et. al. (2004), “Search for Strange Pentaquark Production in e^+e^- Annihilations at $\sqrt{s} = 10.58$ GeV and in $\Upsilon(4S)$ Decays”, http://arXiv.org/hep-ex/0408064.