Category: Genetic Algorithms

Comparing Performance of the Learnable Evolution Model and Genetic Algorithms Applied to Digital Signal Filters

Mark Coletti
13332 Jasper Court
Fairfax, VA, 22033
mcoletti@clark.net
703 818 9511

Tom Lash
12106 Green Ledge Court #202
Fairfax, VA 22033
tlash@rocketmail.com
703 691 9025

Craig Mandsager
10207 Willow Mist Ct. #D
Oakton, VA 22124
craig@thegame.com
703 263 7685

Ryszard S. Michalski
George Mason University
Fairfax, VA 22030
Michalski@gmu.edu
and
Institute of Computer Science
Polish Academy of Sciences
Warsaw, Poland

Rida Moustafa
George Mason University
Fairfax, VA 22030
rmoustaf@galaxy.gmu.edu
703 323 9406

Abstract

This paper describes an application of the Learnable Evolution Model (LEM) to a digital signal filter parameter identification problem, and compares its performance with that of two canonical genetic algorithms, GA1 and GA2. The LEM method integrates symbolic learning and evolutionary computation. When the average fitness of a population of solutions has not improved sufficiently during a Darwinian-type evolutionary computational process, LEM generates a hypothesis as to what type of individuals may perform well. This hypothesis, in the form of attributional rules, is then used for generating a new population of solutions. In the experiments on digital signal filter parameter identification LEM significantly outperformed genetic algorithms GA1 and GA2.

Keywords: Genetic Algorithms, Symbolic Learning, AQ18, Lamarckian Evolution, Learnable Evolution Model, Function Optimization, Digital Filters.

Introduction

Parameter estimation problems solved by steepest descent methods require the gradient of the error surface. Computation of the gradient of a multi-modal error surface can be extremely computationally intensive for a problem with many variables. Derivatives may not be readily available or defined over the search space. Genetic algorithms can be used to solve complex parameter estimation problems. (Yao, Sethares, 1994) With genetic algorithms, unlike the steepest-descent approach, computation of the gradient of the error surface is not necessary. The complexity of the fitness function of such a system implemented with a genetic algorithm grows only linearly with additional variables. Also, the inherent random nature of genetic algorithms makes them less susceptible to finding local minima.

A major difficulty with genetic algorithms is that they are computationally inefficient due to their stochastic nature. A new approach is offered by the Learnable Evolution Model (LEM), which was recently introduced by Michalski (1998, 1999). LEM combines symbolic learning with evolutionary computation in a novel way and offers a computational advantage over traditional genetic algorithms.

The symbolic learning stage is used to learn characteristics of advantageous populations in order to generate populations that minimize the cost function efficiently. The symbolic learning phase can be performed by different methods. LEM uses the AQ-type symbolic learning method (Michalski, 1973) in its implementation. Michalski and Zhang applied LEM and two genetic algorithms, GA1 and GA2, to DeJong's problem set (DeJong, 1975) and their inverse (Michalski, Zhang, 1998-1999). In each test case, LEM outperformed GA1 and GA2.

This paper seeks to expand on this result using a newer version of AQ15, AQ18 (Michalski, Kaufman, 1998), on a different problem. LEM will be applied to the parameter optimization of two digital filters. We show that the LEM algorithm outperforms the genetic algorithms GA1 and GA2 in solving these digital filter parameter identification problem.

Methods

The LEM algorithm works by alternating between genetic algorithm mode and symbolic learning mode. The switch is made when the fitness of the population is no longer improving; that is, when either mode stagnates around the same fitness value.

The LEM algorithm is defined as follows (Michalski, 1998a,b):

1. Randomly, or according to certain prior rules reflecting domain knowledge, generate the starting population of solutions (in the case of symbolic learning, solutions would be concept descriptions).

2. Execute the *genetic algorithm mode* (using standard selection, crossover and mutation operators), as long as the best solution in a sequence of *gen-length* iterations is better by the *gen-threshold* than the best solution found in previous generations.

3. Execute the *symbolic learning mode*:

   - Determine HIGH (high-performance) and LOW (low performance) solutions in the current population, on the basis of the value of the fitness function for a given task or problem.

   - Apply a machine learning method for characterizing differences between HIGH and LOW solutions.

   - Generate a new population of solutions by replacing not-HIGH individuals by those satisfying the learned description of HIGH solutions; the selection of new solutions among those satisfying the description is random or according to the predefined selection rules.

   - Continue the process as long as the best solution in a sequence of *learn-length* iterations is better by the *learn-threshhold* than the previously found best solution.

1. Switch to (2), and repeat the process. Continue switching between (2) and (3) until the *termination condition* is met

(the generated solution satisfactory, or the allocated computation resources are exhausted).

The values of high and low fitness individuals are given to AQ18, which will then identify the gene patterns of the high performing individuals. This information is used to generate a new population of individuals which are then evaluated using the single fitness function.

The two equations to be solved by LEM are from digital signal processing and were used in a paper "Nonlinear Parameter Estimation via the Genetic Algorithm" (Yao and Sethares, 1994).

Linear:

$$\boldsymbol{y(k) = -0.3y(k-1) + 0.4y(k-2) + 1.25u(k-1) - 2.5u(k-2) + n(k)}$$

Non-linear:

$$y(k) = \left[\frac{3 - 0.3y(k-1)u(k-2)}{5 + 0.4y(k-2)u^2(k-1)}\right]^2 + (1.25u^2(k-1) - 2.5u^2(k)))\ln(\left|1.25u^2(k-2) - 2.5u^2(k)\right|) + n(k)$$

Where:
k – is the sample index (or time)
n() – is a noise component ranging from -.25 to .25
u() – represents an inserted function (sin, step, etc.)

The problem is to identify the constant coefficients given a list of *y(k)* values for a number of *k* values. Four real valued genes represent the coefficients -0.3, 0.4, 1.25, and -2.5 for both equations, and comprise a chromosome or "individual." When substituted in the equation, the individual's genes yield a result which is compared with the correct value. The fitness is inversely proportional to the difference between the result and correct value. Therefore the individual whose gene coefficients give the lowest error is assigned the highest fitness.

The genetic algorithm parameters we used are as follows, for both the linear and non-linear equations:

Gene representation	Real
Number of genes per individual	4
Gene landscape (constraint on range)	-30 to 30
Number of individuals per generation	60
Maximum number of births	100,000
Maximum number of generations	1500

Experiments

We ran GA1, GA2, and LEM with the linear and nonlinear filters with three different input functions. Yao and Sethares used a uniformly distributed random input over the interval (-2.5, 2.5). In addition to this input, we used a unit step function 2.5u(k) and a sine wave 2.5sin(Π/10) for comparison. The landscape function generated an output array based on a 200 sample input sequence and stored it for comparison against the populations. Populations were generated, and the fitness of each individual was calculated by computing the mean-squared error between the known system and the output generated by the individual's genes. Yao and Sethares defined the fitness of the individual as the reciprocal of the mean-square error over the 200 sample window:

$$Fitness = \frac{1}{MeanSquareError} = \frac{200\,samples}{\sum_{200samplewindw}(individual - known)^2}$$

We ran each system, GA1, GA2, and LEM ten times for the linear and nonlinear filters using the sine, step, and uniform random inputs. The GAs converged for every case. Since the initial populations were generated randomly, the convergence rate varied greatly between populations and generations. We averaged the ten runs of each type together to try to get an aggregate performance comparison, but we found that the performances varied so greatly that a few runs would dominate the average. So instead of an average performance for each method, the following figures show learning curves from the three systems with different input functions.

Figure 1: GA1 learning curve, nonlinear filter, sine wave input

Figure 2: GA1 learning curve, nonlinear filter, unit step input

Figure 3: GA1 learning curve, nonlinear filter, uniform noise input

Figure 4: GA2 learning curve, nonlinear filter, sine wave input

Figure 5: GA2 learning curve, nonlinear filter, unit step input

Figure 6: GA2 learning curve, nonlinear filter, uniform random noise input

Figure 7: LEM learning curve, nonlinear filter, sine input

Figure 8: LEM learning curve, nonlinear filter, unit step input

Figure 9: LEM learning curve, nonlinear filter, uniform random noise input

LEM's symbolic learning phase is demonstrated by the dramatic drop in mean-square error when it has learned useful rules for generating the next generation. LEM toggles into symbolic learning mode roughly 10-800 times through out course of a typical experiment. A dramatic drop in mean-square error usually occurs within the first 100 generations.

Since we used four genes to represent the four weights of the filter, the error surface generated by the mean-square error is four dimensional. A four dimensional error surface can be problematic for traditional search techniques. Traditional techniques such as gradient descent and LMS are subject to finding local minima, and they would have to run parallel solutions to achieve the robustness of a GA approach. LEM alleviates much of the computational cost of the GA by accelerating the learning process with a series of symbolic learning cycles.

Summary

With LEM, we have shown that a genetic algorithm that is augmented by the use of a symbolic learning mechanism creates significant speedups when compared to more traditional GAs. This speedup comes from LEM's symbolic learning mechanism, which creates rules from the observed differences between the poor performing individuals and high performing individuals in a given population. New populations are generated based on these rules; that is, LEM will generate better solutions based on observed patterns, as opposed to populations derived from stochastically driven GAs.

We ran several experiments exercising GA1, GA2, and LEM on linear and non-linear filters for unit step, sine wave, and noise input. In each case, LEM outperformed GA1 and GA2. LEM also converged closer to the optimal solution than did the other GAs.

LEM, as we have demonstrated, is suitable for tackling complex search spaces. Finding strategies in planning problems is another area whereby LEM could potentially do well.

Acknowledgements

Dr. Kenneth DeJong supplied the source code to GA1 and GA2.

Guido Cervone provided administrative assistance.

Dr. Ken Kaufman provided the original LEM source code.

Dr. Ryszard Michalski gave project guidance.

References

Yao, Leehter and Sethares, William , *Nonlinear Parameter Estimation via the Genetic Algorithm*, IEEE Transactions on Signal Processing, Vol. 42 NO.4 p.927-935. 1994.

Kaufman, Kenneth and Michalski, Ryszard S., ELENA/AQ18: An Environment forLearning and Natural Induction: User's Guide, *Publications of Machine Learning and Inference Laboratory*, George Mason University, P98-11, October, 1998.

Michalski, R. S., "AQVAL/1 -- Computer Implementation of a Variable-Value3d Logic System VL and Examples of its Application to Pattern Recognition," *Proceedings of the First International Joint Conference on Pattern Recognition,* Washington DC, pp. 3-17, October 30 - November 1, 1973.

Michalski, R.S., Learnable Evolution: Combining Symbolic and Evolutionary Learning *Proceedings of the Fourth International Workshop on Multistrategy Learning*, (MSL'98), Desenzano del Garda, Italy, pp. 14-20, June 11-13, GMU MLI publication P98-9, 1998.

Michalski, R.S. and Zhang, Q., LEM: The Learnable Evolution Model that Speeds Up Evolution by Symbolic Learning, accepted for publication in Machine Learning Journal, 1999.

De Jong, K.A., *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems,* Ph.D. thesis, Department of Computer and Communication Sciences, University of Michigan, An Arbor, 1975.