Genetic Search Algorithm for Large Problems

Carl Formoso, Division of Child Support, Olympia, WA

Abstract

The limited utility of standard search algorithms in our problem of interest led to the development of a random 'genetic' type search. With the largest file that could be processed by the standard search, the genetic approach took 13 minutes while the standard search took over 4 hours for 100 iterations. While the goodness of fit was better with the standard algorithm, the difference was not large. In addition, the genetic search was able to process much larger files than the standard search.

Introduction

Modern search algorithms are fast and effective on a wide range of problems. But on some problems with a large number of parameters and a large number of observations, the manipulations of large matrices and storage and retrieval of large amounts of information may render an otherwise useful method slow or inoperable. We found this to be the case in developing a neural network simulation model for child support arrearage debt, where we have data for over 241,000 individuals and many data elements which may have an effect on arrearage debt. A standard Levenberg-Marquardt search was not able to proceed with even a 15% sample of the entire file, and was very slow with any file of reasonable size.

To circumvent this difficulty a random 'genetic' type search was devised. While the genetic approach was slower with very small files, it was much faster with files of a size necessary for this type of simulation, and it was able to process our entire data file.

Method

The logical flow of this algorithm is shown in Figure 1. A matrix **M** of parameters – the starting weight and bias values for the neural network – is randomly generated and stored. There are N 'strands' of information and each is tested with the network against the known target values. A measure of fit, matrix **E**, is obtained and stored. Through random 'mutations' matrix **M** then generates matrix **Mx**, which when tested generates matrix **Ex** as a measure of fit. 'Survival of the fittest' is the selection of the N strands from **M** and **Mx** which gave the best fit. The new 'generation' of strands are stored as matrix **M**, along with the corresponding measures of fit stored as Matrix **E**. The process is repeated until a selected criteria is met.

While each iteration is doing N parallel calculations, the calculations are not very time consuming, and only small amounts of information need to be stored from one iteration to the next.

Results

The network shown in Figure 2 was used in comparing the genetic search with a standard Levenberg-Marquardt (LM) search, with **M** having 91 parameters and 5 strands of information.

Computation times and goodness of fit were compared for the genetic search and the LM search. Figure 3 a), b), and c) show the comparisons for computing time. The records being processed are the first n in the complete file, where n is the abscissa value on the graph. For the largest file tested, with 30,000 records, the genetic search took 13 minutes for 100 iterations while the LM took over 4 hours. This file was about the largest that LM was able to process.

Figure 3 b) shows that with less than about 12,000 records, for both the genetic and LM, computing time is approximately linearly related to the number of records; but the linear region for the LM has a larger slope. The linear region for the genetic search extends to much larger files, though this was not specifically tested.

For the machine used and our particular problem*, the genetic search was faster than the LM with all file sizes above about 2,000 records, and was able to process the entire data file while the LM procedure could not work with files above about 30,000 records. Figure 3 c) shows the number of genetic iterations which could be processed in the time required for 1 iteration of the LM. For the test with 30,000 records the genetic algorithm was more than 18 times faster than the LM.

However, Figure 3 d) shows that the LM did produce better results in terms of goodness of fit for 100 iterations (except for the 10,000 record point which may be an anomaly). The last three points on the graph may indicate that as file size increases goodness of fit may be improving at a faster rate for the genetic algorithm. Unfortunately a comparison could not be achieved with larger files. But the mean square error (mse) for 100 iterations of the genetic algorithm on the complete data file *was* better than the mse for the LM with 30,000 records. In addition, if time is the criteria, the genetic approach could do up to about 1,800 iterations while the LM is doing 100, and we expect the genetic mse would be superior, though this has not been tested.

Author

Carl Formoso
Division of Child Support
PO Box 9162, Olympia, WA 98507
(360)664-5090, FAX (360)586-3274
cformoso@dshs.wa.gov

* We used an IBM 300XL, Pentium II, 4GB hard drive, 64M RAM. Results can be expected to show dramatic differences between machines.

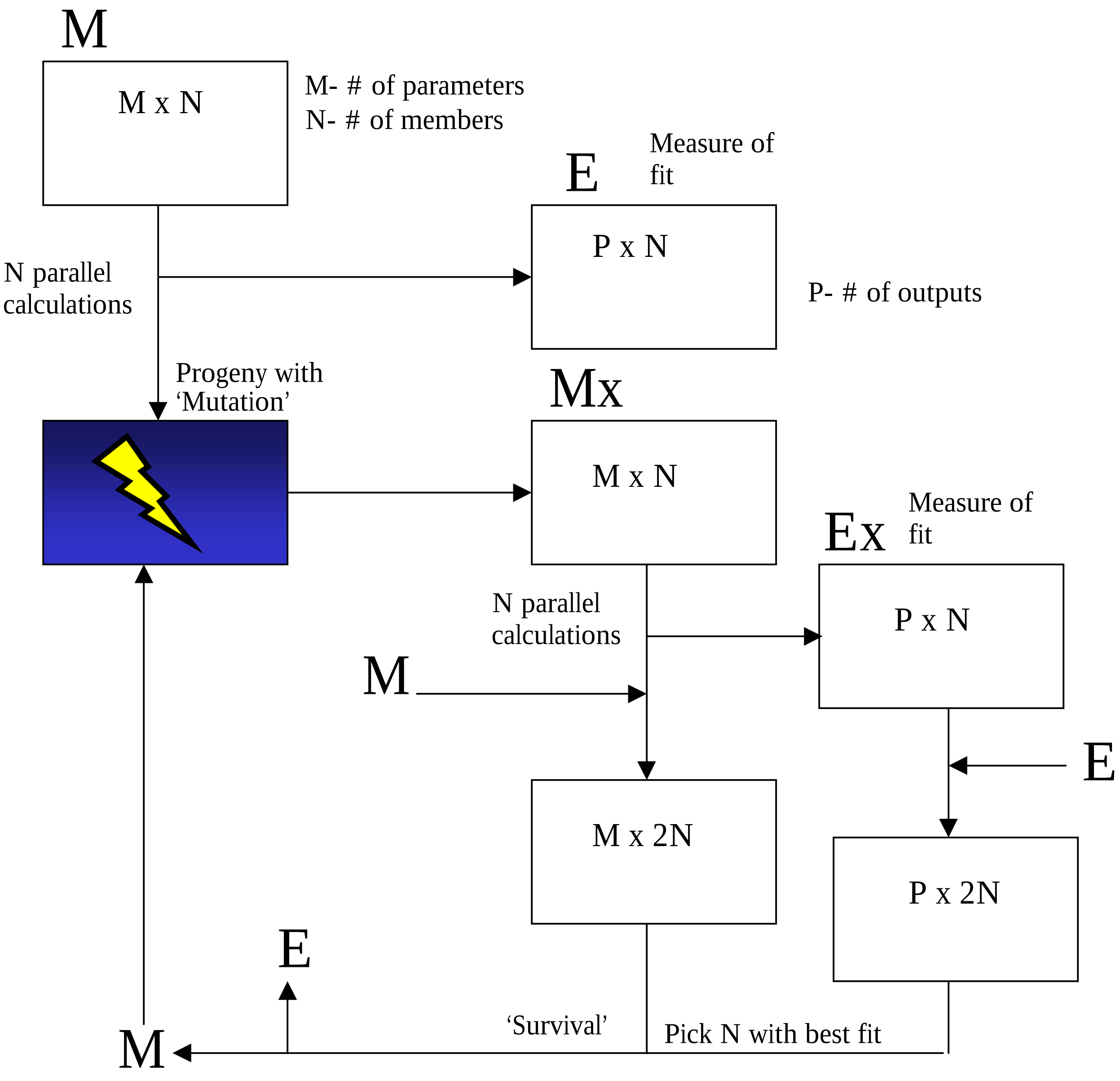

Figure 1
Flow of Genetic Search Alogrithm

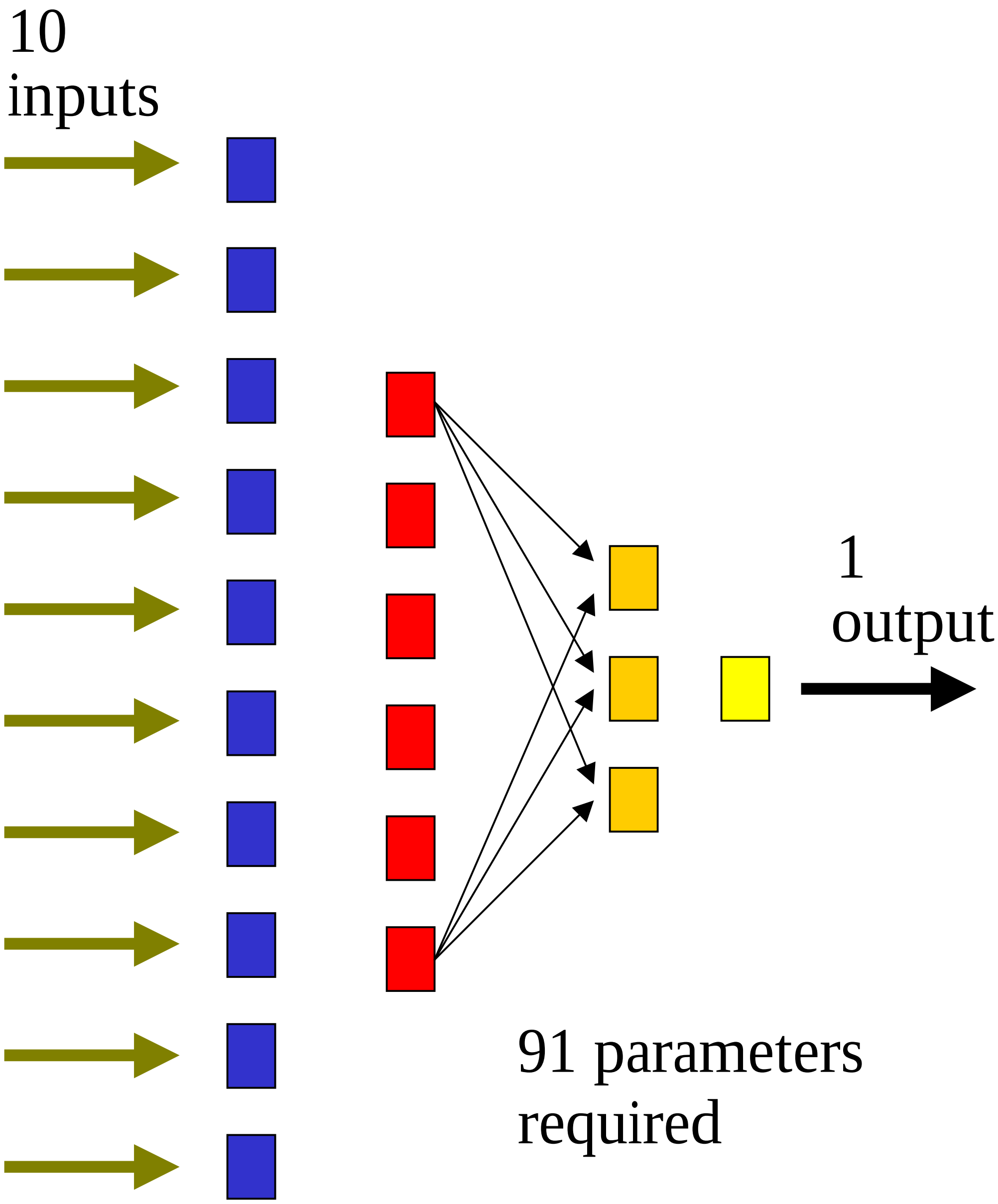

Figure 2
Neural Network Simulation for Child Support Arrearage Debt

The network operates by combining and transforming inputs. Thus in the first hidden layer (column of six blocks) each cell, represented by a block, would receive 10 weighted inputs. These are summed and transformed in the layer. The outputs from the first hidden layer are passed on as inputs to the second hidden layer. Here each cell receives 6 weighted inputs, partially demonstrated by the arrows in the Figure. Each cell in the final hidden layer receives 3 weighted inputs. By adjusting weight values the network is 'trained' so that outputs approach known target values.

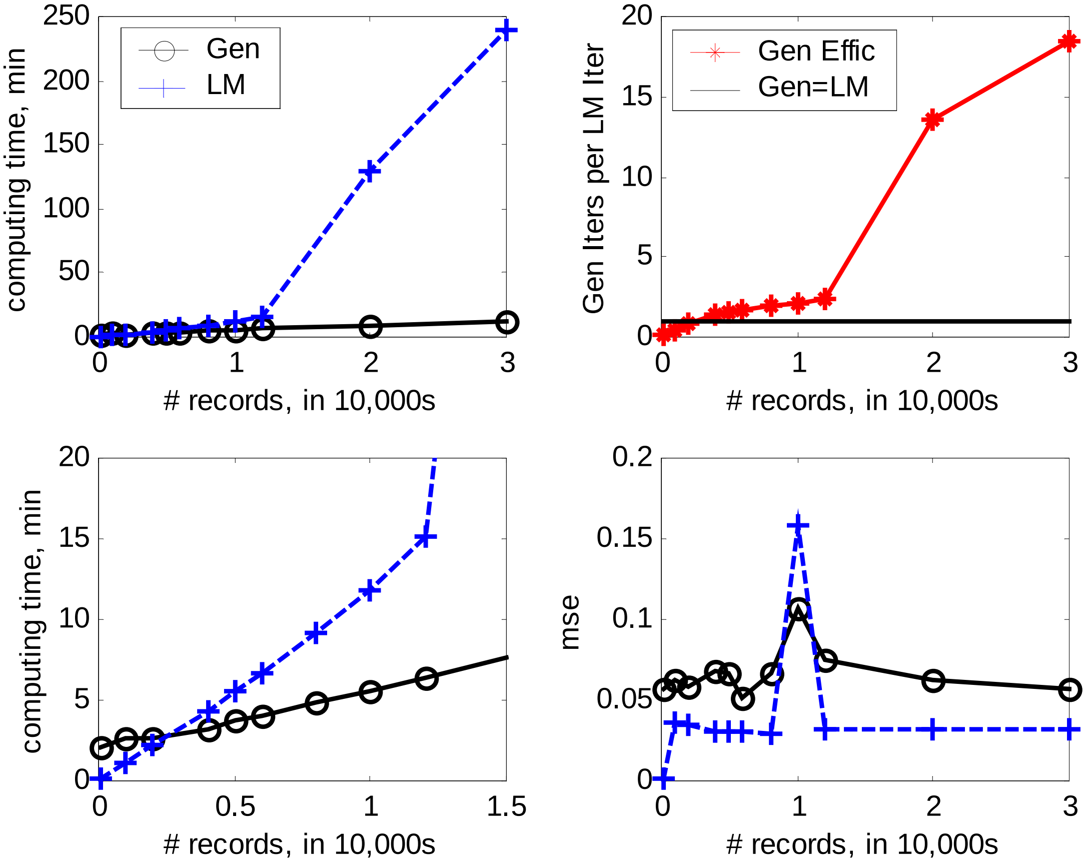

Figure 3
Genetic vs Levenberg-Marquardt Search

a) Computing time for 100 iterations; b) expanded view of 3a) for region of less than 15,000 records; c) the number of iterations of genetic search which could have occurred in the time required for one iteration of LM; d) mean square error (mse) after 100 iterations.