Biocomputation Assignment

7th August 2021

Course
BSc (HONS) Computer Science

Module
Biocomputataion - UFCFY3-15-3

Lecturer
Mohamed Naufal Abdul Hadee

Institute
Villa College

Author
Mohamed Athfan Khaleel- S1900190

Introduction

The two steps that make up a genetic algorithm (GA) are usually quite simple. Firstly, individuals are selected for the development of the next generation, and afterwards, selected people are manipulated using crossover and mutation techniques to produce the following generation. Individuals are selected for mating (reproduction) based on the selection process, which defines how each individual reproduces. According to the selection approach, "the better a person is, the better his or her chances of becoming a parent" (Ao, 2011).

Using three datasets, this work attempts to answer a set of classification tasks given as an assignment. In order to tackle these problems, we shall use a GA. There are two algorithms that are utilized to solve the three datasets.

Research

Data mining is a process that is used to discover new data from a large data set. In other words, Data mining is the act of examining enormous datasets in order to discover previously unknown associations and summarize the data in unique ways that are both understandable and beneficial to the data owner. (Hand, Mannila and Smyth, 2001). There are several processes involved in data mining, such as classification, regression, modeling and so on. However this paper is more interested in the classification aspect of data mining.

The “IF-THEN” rule is a very popular method used when doing classification in the data mining process. The IF-THEN rule is as simple as “IF <conditions> THEN <output>”.

The IF-THEN rule is split into two parts, and as the explanation above mentions, it holds two different parts. The rule antecedent (IF part) contains a set of conditions, usually connected by a logical conjunction operator (AND). The rule consequent (THEN part) specifies the output (Parpinelli, Lopes and Freitas, 2002).

The Experiments

The assignment task was to make an algorithm for the three given datasets and get results from it. Datasets one and two consists of entirely binary data of inputs and outputs, where the only difference between the two datasets is that dataset one has five inputs and dataset one has seven six. Each of the two datasets are made in the style of [d1, d2, … d6-7, o1], where the D values are all inputs and the O value is the output.

Dataset one is specifically made up of 32 rows of 6 bits where the first 5 bits are the input bits and the last 6th bit is the output bit. Dataset two is specifically made up of 64 rows of 7 bits where the first 6 bits are the input bits and the last 7th bit is the output bit. All of the data in these two files look seemingly random and it is hard to exactly say what kind of data is encoded in these datasets.

Dataset three is much more different compared to datasets one and two. Dataset 3 comprises a whopping 2000 rows of data, and has 6 inputs and one output. The biggest difference between datasets one and two and dataset three is that the inputs in dataset 3 are not integers like in datasets one and two, but are floating point values between 0 and 1. The output value is the same as the others and can only be 0 or one.

Datasets one and two use the same algorithm with just 2 parameter changes. The dataset file path and the dataset row size. Dataset three uses a modified version of the codebase for datasets one and two, but adds support for the floating point values and adds some extra functions that are necessary for the GA to work on dataset three properly.

Dataset testing

The main purpose of the dataset testing is to make a GA that will run the processes of selection, crossover, mutation and making the next generation. The main aim of all GAs is to get the highest score possible.

Dataset One

As mentioned before, dataset one contains 32 rows of 6 parameters. The largest possible score that can be obtained for this is 32.

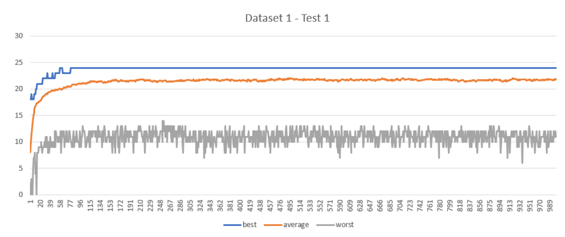

Fig 1: Dataset 1 - Test 1

The above line graph is the result of running the GA for dataset one. It displays the results of the GA over 1000 generations and displays the results of the best, average and worst performing individuals of the population. In the first run, none of the variables such as mutation rate was changed (default 0.05). It can be observed that the GA had reached a maximum fitness score of 24.

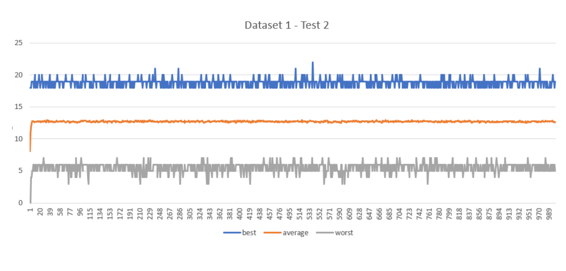

Fig 2: Dataset 1 - Test 2

Figure 2 above is the same dataset, but this time with a much higher mutation rate. The higher mutation rate meant that the individuals almost always mutated and it would always create a brand new Individual. When this happens, the rate of evolution in the algorithm is not that noticeable.

It is also important to note that the best individual in a high mutation rate scenario never settles to a maximum, but instead jumps between 17 and 19, much lower than the initial test.

I ran the dataset again with no mutation, and nothing interesting happened. There did not seem to be any exciting changes in the best individual, other than the fact that GA quickly found the best individual within the first hundred generations and got stuck into an expected optima of score 18.

Dataset 2

As mentioned before, dataset two contains 64 rows of 7 parameters. The largest possible score that can be obtained for this is 64.

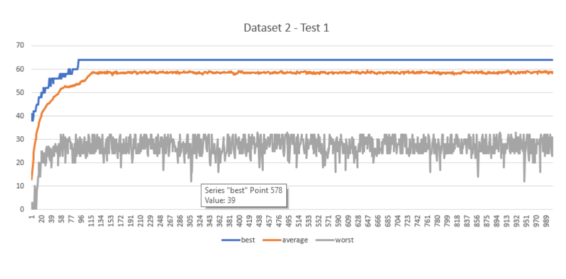

Fig 3: Dataset 2 - Test 1

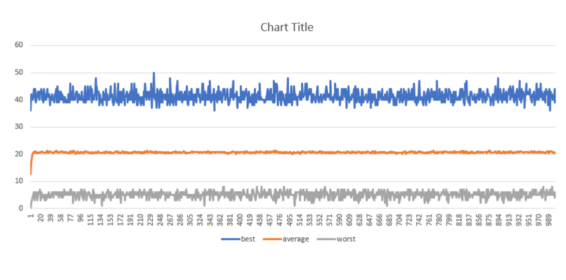

Fig 4: Dataset 3 - Test 2

The same two tests that were run on dataset one were run on dataset two as well, and here we observed similar behaviour on dataset two that we observed on dataset one. When comparing Figure 1 against Figure 3, it displays the same type of curves on each other, except the main difference being that a larger rule set has enabled the algorithm to quickly reach an optimum value. Compare Figure 2 and Figure 4 we again observe the same behaviour from dataset one on dataset two.

Dataset 3

Dataset 3 is completely different from the other two datasets. It contains entirely floating point values instead of the binary values in datasets one and two. For this dataset, to make it easy to run in the GA, each of the floating point values were encoded into 1's and 0's by finding an upper and lower bound of the data. This dataset also contains a lot more data compared to the previous datasets. Dataset 3 contains 2000 rows of data with 6 floating point inputs and 1 binary output. It would take the GA a very long time if we passed all of these 2000 rows into the population. Because of this, the dataset is split into two sets. A training set and a testing set. The GA will switch between the training and testing set every now and then so that we can check if the algorithm is working as expected.

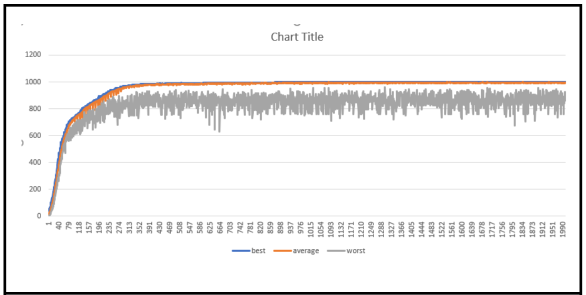

Fig 5: Dataset 3 - Test 1

Figure 5 is results of dataset 3 running on the GA with a population size of 200, gene size of 20 and a mutation rate of 0.01 In 2000 generations. It is clear that it is quite similar to the results we got for figures 1 and 3 from datasets one and two.

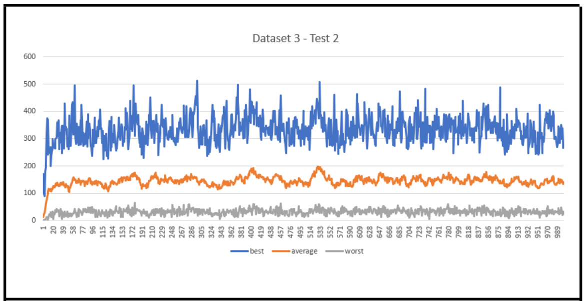

Fig 6: Dataset 3 - Test 2

For consistency, dataset 3 was also subject to a mutation rate change. The mutation rate was changed to 0.5 and we observed that the GA failed to read a stable maximum fitness like compared to figure 5 where it reached peak fitness in under a 1000 generations. With a higher mutation rate, the best fitness value was anywhere between 300 and 400 most of the time.

Conclusion

Genetic Algorithms are very suited to tackle categorization issues like those we have met in this assignment. Genetic algorithms function in such a manner that enables them to quickly handle issues like these in a more faster and efficient method. Numpy arrays instead of Python lists might be used to improve my algorithms. Large datasets would benefit from this as well as performance. As of now, the tournament selection technique appears to be functioning well. I have not tested other selection methods such as the roulette wheel selection method, but if it were used, I predict that it would have taken more generations to reach the fittest Individual, because the roulette wheel selection method may not return the best results each time.

References

Ao, S., 2011. *World Congress on Engineering*. Hong Kong: Newswood Ltd. [Accessed: 5 June 2021].

Hand, D., Mannila, H. and Smyth, P., 2001. *Principles of data mining*. Cambridge, Mass.: Bradford Book. [Accessed: 5 August 2021].

Parpinelli, R.S., Lopes, H.S. and Freitas, A.A. (2002). Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation. [online]. 6 (4). p.pp. 321–332 [Accessed: 7 August 2021].

Parpinelli, R.S., Lopes, H.S. and Freitas, A.A. (2002). Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation. [online]. 6 (4). p.pp. 321–332. [Accessed: 7 August 2021].