Classification of Breast Cancer with Genetic Programming

Michael Wisnewski*
Aidan Larock*
mw17an@brocku.ca
al16my@brocku.ca
Brock University
St. Catharines, Ontario, Canada

Abstract

This paper will discuss the use of a genetic program in order to classify breast cancer x-rays into malignant and benign cases. Genetic programs allow for evolutionary processes to be incorporated into machine learning. By using an evolutionary process (GP), this paper will analyze the performance and applicability of using such systems to classify and determine breast cancer in x-rays.

Keywords: genetic programming, classification, Breast Cancer

1 Introduction

Determining breast cancer in patients is critical in health care. By using x-rays and specific classifiers, genetic programs are capable of determining whether an x-ray shows signs of breast cancer in a patient. A genetic program will accept a number of classifiers, and over a number of generations, adapt and alter individual trees within a population such that the final best individual will be able to accurately predict a diagnosis (malignant or benign).

2 Experiment Description

The algorithm was trained and tested on the Wisconsin breast cancer data set in two distinct sets. Each set consists of 10 runs on different seeds, averaged out, to allow for consistent and replicable results. The genetic program was built using DEAP TEX [2]. , an evolutionary framework in python (version 3.7).

DEAP allows for individual trees to be created and populated in python, while also allowing for evaluation and other genetic programming parameters and functions to be altered. The variables, parameters, and fitness are further discussed and explained below.

2.1 Parameters

The parameters that were selected for configuration can be seen below in Table 1: GP Parameters. There are nine distinct parameters used in the algorithm.

Tournament size represents how many individuals are selected to repopulate the next generation. In this paper, the 3 individuals are selected from a tournament selection. Max depth alters the maximum height of the tree. This makes sure that bloat can be controlled, max depth is set to 17 as recommended by Dr. John Koza TEX [3]. Population size controls how many individuals (trees) are created per generation.

The Terminals parameter represents how many input(s) are used. As per this data set, there are 32 control points, 30 of these points are numerical and to be regressed. Therefore, the terminal parameter is set to 30.

The crossover and mutation parameters are decimal of percentages (i.e Crossover = 0.9) these rates can be altered to affect the probability that an individual's sub-tree will be swapped or mutated. While crossover allows for individuals to share genes, mutation allows for new and otherwise unreachable genes to enter the population.

The number of generations is a stopping point, this point reflects the total number of population evaluations and evolution's to be performed.

Table 1. GP Parameters

Parameter	Set 1	Set 2
Tournament Size	3	3
Max Depth	17	7
Population Size	300	400
Terminals	30	30
Crossover	90%	90%
Mutation	10%	10%
Number of Generations	20	20
Number of Runs	10	10

2.2 Data Format

The Wisconsin Breast Cancer (Diagnostic) Data Set was created by the doctors at the Wisconsin University, this data set was created and to be used for Machine Learning applications. There are twelve unique data points of which are

[*] Both authors contributed equally to this research.

further broken down into more measurements, totalling 32 attributes. The format of the data is compiled as a comma separated text file (CSV), with 32 columns and 569 rows.

The attributes accounted for are named as such; ID, Diagnosis (M or B), radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, isymmetry, and fractal dimension numbered 1 through 12 respectively TeX [1].

The first attribute is the ID number of the case. Where the second tells us if it is Malignant or Benign as an array of characters. Attributes 3 through 12 are repeated 3 times, for "mean","standard error", and "largest" values respectively. For example selecting the first patients x-ray information looks as follows:

Looking at the first attribute gives us the patient case ID, followed by a character of B or M, since there is an M, this means this patient has a malignant tumor (Cancerous). Looking at the next three points (17.99,10.38,122.8) gives us information about the radius of the tumor including "mean","standard error", and "largest" values respectively.

When data is read, it is placed into a loosely typed two-dimensional array of size 32x569. The data is then split into training and testing as a two-dimensional array. After which a shuffle is completed on the data prior to splitting. The first $3/4^{ths}$ are added into a training set, while the remaining $1/4^{ths}$ of the data is stored for later testing. Case ID is ignored as it is not relevant for computations. Data points 2-32 are used for the genetic program the second attribute is further stripped and used in the fitness function as explained in the next subsection. The remaining 30 points are evaluated and used as training for each individual in the population within the genetic program.

2.3 Fitness

In order to evaluate the individual trees on the data set, so that the algorithm may properly evolve, a maximizing function must be used. For this paper, a hit counter was implemented with correct calculations of malignant or begin being recorded as a correct hit (Algorithm 1.). The individuals with the most correct hits were deemed most fit in the population. Each individual is tested over the entire training data set and compared to their expected result. This process was repeated for each individual in the population for each generation.

Algorithm 1: Fitness Evaluation

```
for $x, y \in input, results$ do
    if $x \geq 0.0$ & $y = "M"$ then
        $hits \leftarrow hits + 1$
    end
    if $x < 0.0$ & $y = "B"$ then
        $hits \leftarrow hits + 1$
    end
    return $hits$
end
```

3 Results

Each set consisted of 10 runs, with training and testing. Immediately after training, the best individual would be picked from the population and tested on new data using the testing data that had been separated from training data. Ensuring that the data was new to the algorithm is vital in ensuring that the algorithm was not simply picking from data it had previously seen and would determine if the algorithm was over trained.

Table 2. Max Depth: 17, Population Size: 300

Gen.	Avg. Fitness	STDEV. of Fitness	Best Fitness
0	198.52501	42.936804	328.8
4	257.09066	46.609044	359.9
8	290.56234	53.977329	370.8
12	316.199	64.0968	373.1
16	326.50867	67.103893	378.5
20	334.76233	69.099187	387

Table 3. Max Depth: 7, Population Size: 400

Gen.	Avg. Fitness	STDEV. of Fitness	Best Fitness
0	198.2915	44.438409	334.7
4	251.1255	45.946195	366
8	268.802	54.48405	370.4
12	284.593	73.977698	378.3
16	301.6525	78.211174	384.3
20	308.36825	82.86652	387.3

3.1 Figures

A graph is created for each set. With the first set Figure 1: Fitness Calculations of GP Algorithm, Using the Wisconsin

Table 4. Set 1 Average Confusion Matrix

	True Positive	True Negative
Predicted Positive	431	49
Predicted Negative	88	852

Table 5. Set 2 Average Confusion Matrix

	True Positive	True Negative
Predicted Positive	459	70
Predicted Negative	84	807

Figure 1. Variations of Elitism and non-Elitism)

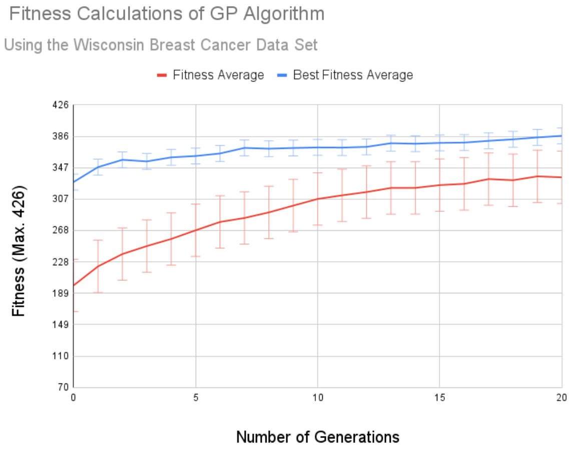
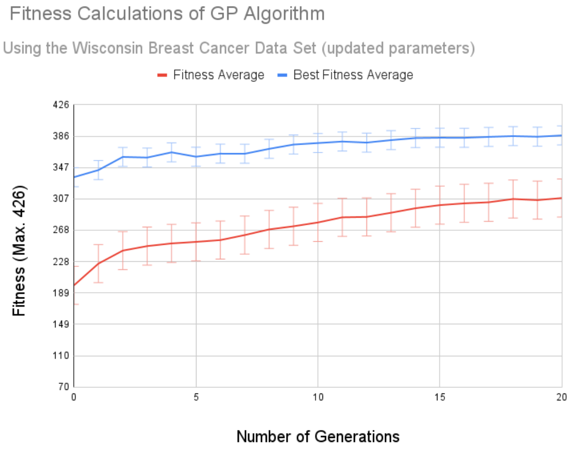


Figure 2. Variations of Elitism and non-Elitism)



Breast Cancer Data Set, It can be seen that this graph has a tighter fit of Fitness Average vs Best Fitness Average as compared to the second graph (Figure 2) meaning, we see the fitness average closer to the best fit average. Similarly, Set 1 had a greater standard deviation for fitness average as compared to Set 2. Figure 2 the best fitness average appears to be found quicker than Figure 1 (by the flattening of the curve). In Figure 1 it can also be seen that the error bars that represent standard deviation begin to overlap with the average best fitness trend line.

3.2 Discussion of Results

As per the results in Graph 1 it can be seen that the error bars for both fitness average and best fitness average have a minimal absolute difference as compared to graph 2 where the error bars in graph seem very distant to one another. We can also see that decreasing the tree depth had a significant result on the standard deviation of the resulting averages (Graph 2) as compared to graph 1. From set 1 and 2 confusion matrices the calculated accuracy is 90.35% and 89.15% respectively. From this we can infer that set 1 performed better than set 2. It can be seen that decreasing the tree depth has no significant difference on the accuracy of the testing data; this means that it is possible that the final solution does not need a very deep tree.

It is also important to note that even though set 1 had a lower best fitness, it still performed better than set 2 on the testing data. While there are many factors that have an effect on this, over-training and max depth is thought to be the main proprietor. Set 1 was allowed to have larger trees than the runs in set 2. These larger trees allowed for set 1 to contain more information and key components than that of set 2. This tree depth also contributed to set 1 having a larger standard deviation than set 2. With more terminals and factors to change, the individuals in set 2 would be affected more by mutations and crossovers and would therefore see larger changes over generations than runs in set 2. Even though both sets approached 384, and set 2 was able to reach a higher value than set 1, set 1 had better results in the testing data. The difference in between functions in set 1 was smaller than that of set 2, as seen in the standard deviation. This difference as well as tree depth between set 1 and 2 can be seen as the cause of difference of accuracy between the testing of both sets.

4 Conclusion

The algorithm performed remarkably over the testing data. It was able to perform with a 90% accuracy on new data. While the accuracy of the program is superb for testing, real life applications would require more definitive answers, especially to such an important question.

More accurate results would be preferred, as to improve upon the results, large parameters such as larger population sizes and number of generations should be used to achieve such results. While the algorithm was very good for testing a genetic program for such a use and showing the merits of such a system, the algorithm does have room to improve.

5 Best Trees

The following best trees can be seen bellow. Over the course of each set 1 individual from set 1 and 1 individual from set 2 had the best overall fitness. The individual from set 1 had a fitness of 403 while the individual from set 2 had a fitness of 401.

5.1 Best Tree Set 1

sub(cos(neg(cos(ARG28))), add(mul(sin(protectedDiv(sin(ARG4), ARG25)), (protectedDiv(ARG24, ARG23), protectedDiv(cos(mul

(ARG21, ARG7)), ARG6))), protectedDiv(ARG6, protectedDiv(sub(ARG1, ARG6), cos(ARG18)))))

5.2 Best Tree Set 2

neg(sin(add(mul(cos(mul(ARG20, ARG24)), sin(sin(sin(ARG7)))), protectedDiv(add(neg(neg(1)), add(sub(-1, ARG27), protectedDiv(ARG22, ARG3))), mul(add(protectedDiv(ARG11, 1), sub(ARG14, 0)), add(protectedDiv(ARG28, add(mul(sub(sub(cos(cos (neg(mul(protectedDiv(ARG15, ARG22), sin(add(ARG7, ARG28)))))), sub(ARG15, sin(ARG6))), sin(ARG6)), sin(sin(sin(neg(ARG12))))), add(ARG9, 1))), mul(ARG1, ARG18)))))))

References

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Félix-Antoine Fortin, Fran-Michel De Rainville, Marc-André Gardner, Marc Parizeau, Christian Gagné (2012). DEAP: Evolutionary Algorithms Made Easy . Journal of Machine Learning Research , 13 , 2171–2175 .

[3] John R. Koza. Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, MA,USA, 1992.