Using Simple Fitness and Fitness Sharing in Genetic Programming to Detect Breast Cancer

Ashkan Entezari, Brock University

St. Catharines, Ontario

Email: ae13cu@brocku.ca

Abstract **–In this work which is a challenging pattern recognition problem, given a set of parameters, Genetic Programming (GP) is being used to detect if a patient has benign or malignant breast cancer. Fitness calculation is one of the most important parts of the GP program which can highly affect the results and running time of the program. In this paper, simple fitness and fitness sharing are compared and for this application, simple fitness showed better results, although the running time of program, using fitness sharing was better.**

I. INTRODUCTION

Genetic programming (GP) is an evolutionary algorithm-based methodology inspired by biological evolution to find computer programs that perform a user-defined task. Essentially GP is a set of instructions and a fitness function to measure how well a computer has performed a task. It is a specialization of genetic algorithms (GA) where each individual is a computer program. It is a machine learning technique used to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task.**[1]** Fitness function and fitness calculation, is one of the most important parts of a GP program and as we will see in this paper, it can affect the results and the running time of the GP program.

There are three important steps in this work. In the first step, the effort was on using GP to evolve a rule that predicts whether a patient has benign or malignant breast cancer. As there are different factors and parameters in a GP program, in the second step, the effort was on improving the results using different GP parameters. In the last step, instead of simple fitness, fitness sharing was used.

Fitness sharing was introduced by Deb and Goldberg in 1989**[2]**. That form, known as explicit fitness sharing, relies on a distance metric to cluster population members. Members which are similar to each other are punished for this similarity by being required to share their fitness, while isolated individuals retain all the fitness value that they achieve. Some years later, Smith, Forrest and Perlson in 1992 **[3]** introduced implicit fitness sharing for concept learning problems. Implicit fitness sharing differs from the explicit form in that no explicit distance metric is required. Instead, all population members which correctly predict a particular input/output pair share the payoff for that pair. Implicit fitness sharing extends readily to many

genetic programming approaches, but with the increased complexity of genetic programming search spaces there is a risk that the benefits of fitness sharing may be dissipated **[4]**.

II. Experiment Description

A. Data

The dataset which is used for this work **[5]**, is a breast cancer data base, which is obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg **[6]**. Samples arrive periodically as Dr. Wolberg reports his clinical cases. The database therefore reflects this chronological grouping of the data, starting at January 1989 with 367 instances and currently it has 699 instances as donated on 15 July 1992.

Dataset has 11 attributes per entry, which attributes 2 through 10 have been used to represent instances. Each instance has one of two possible classes: benign or malignant. These attributes are listed in Table 1.

No.	Attribute	Domain
1	Sample Code Number	Id number
2	Clump Thickness	1 - 10
3	Uniformity of Cell Size	1 - 10
4	Uniformity of Cell Shape	1 - 10
5	Marginal Adhesion	1 - 10
6	Single Epithelial Cell Size	1 - 10
7	Bare Nuclei	1 - 10
8	Bland Chromatin	1 - 10
9	Normal Nucleoli	1 - 10
10	Mitoses	1 - 10
11	Class	2 or 4

Table 1. Wisconsin Breast Cancer, dataset attributes

B. GP Language

In this work, eight functions and nine terminals have been defined. GP program, uses these functions and terminals and if produce a result, greater than 0, it will be considered as malignant and if it was less than or equal with 0, it will be considered as benign. Nine terminals which are used in this work, are CT, UCS, UCSH, MA, SECS, BN, BC, NN and M. They keep the values of the attributes 2 through 10 of the Table 1. The functions are listed in Table 2.

Name	Function
ADD	Addition of two numbers
SUB	Subtraction of two numbers
DIV	Division of two numbers, returns 0 if the 2^{nd} number is 0
MUL	Multiplication of two numbers
MIN	Minimum of two numbers
MAX	Maximum of two numbers
SIN	Returns sinus of the number
COS	Returns cosines of the number

Table 2. GP functions

GP parameters can be changed in many different ways to obtain different results. As mentioned earlier, in this work GP parameters have been changed several times, in order to get better results. These parameters are listed in Table 3. Note that after this step, the effort was on comparing the results of simple fitness with fitness sharing, therefore, the parameters were the same during the both experiments.

Parameter	Value
Generations	51
Runs	20
Population size	512
Population initialization	Ramped half-and-half
Selection	Tournament (size=3)
Crossover	90%
Mutation	10%
Max tree depth	17
Size of training	383
Size of testing	300
Number of elite individuals	5

Table 3. GP parameters

C. Training and Testing Sets

The database used for this work, has 699 entries which 16 of them, have missing values, therefore, after removing the incomplete rows of data, 683 entries remained to work on. Among these data, 444 records are for patients with benign cancer and 239 for patients with malignant cancer. The most important thing to consider here is the unbalanced ratio of patients with benign and malignant cancer, which may cause the GP program to reduce biased results. As we need to split this dataset (to have one data set for training and the other for testing), and since the ratio of these two group in dataset are unbalanced, it is important that how this dataset will be split.

As a solution to the aforementioned problem, and in order to have two datasets with fairly distribution of benign and malignant patients, two hand-selected datasets was used. The first dataset which is used for training, has 383 entries with 244 records for patients with benign cancer and 139 for patients with malignant cancer. The second one has 300 entries with 200 and 100 records, for the benign and malignant patients respectively.

D. Fitness Calculation

In this work, two different methods for calculating fitness were used: simple fitness and fitness sharing. One of the advantages to using fitness sharing is that unique behavior is rewarded. This leads to populations with higher diversity, and fitness sharing has even been shown to be an effective niching technique **[7]**.

For simple fitness, the procedure is that an individual will be tested over all examples and after that a fitness value will be computed and then assigned to that individual and then the same process for the next individual. This will continue until all individuals are tested toward all examples and have been assigned a fitness value. In this part, the reward given for each training example is equal to the number of individuals who correctly classified it. Equation 1 shows the way of calculating fitness value for training examples:

$$\text{fitnessValue} = \text{hits} / 383 \quad \textbf{(1)}$$

As mentioned in section C, 383 is the number of training examples. For testing, in order to make sure that both positive examples and negative ones have the same effect, the fitness value is computed as is shown in Equation 2:

$$\text{fitnessValue} = (tp/100+tn/200)*50 \quad \textbf{(2)}$$

Here tp stands for number of true positives and tn stands for number true negatives. In this way, the fitness is normalized over positive and negative examples and the final value is between 0 and 100.

The procedure for fitness Sharing is that all individuals will be tested with all examples and after that, for each example a specific reward (which is the same for all examples) will be shared equally among the individuals who have successfully classified it. Finally, for all individuals, their rewards will be added together and in this way their fitness value will be computed (Equation 3). One thing to consider in fitness sharing is that examples which were classified correctly less often, worth more than those which were classified correctly very often.

$$fitnessValue(i) = \sum_{j=1}^{e} reward(i,j) \quad (3)$$

In the above equation, *e* is the number of examples and *reward(i,j)* denotes the reward that individual i has obtained from example j. This reward will be 0 if individual i was not successful in classifying the example j, otherwise it will be computed as shown in equation 4:

$$reward(i,j) = 1/num(j) \quad (4)$$

In the above equation, num(j) is the number of individuals who successfully have classified the example j.

III. Results And Discussions

In the first step of this work, the effort was on using GP to work on the given dataset. The fitness which is used in this step is a simple fitness in which individuals will be rewarded according to the total number of hits. Several experiments have been done in order to achieve better results. The parameters shown in Table 3, got the best results so far. In this case, the best result was for the second run, with the fitness value of 97.5 out of 100. Refer to Table 4 to see the best five results, which were achieved during the 20 runs of this experiment:

Run Number	Fitness Value
2	97.5
13	96.25
19	96.25
7	95.0
18	94.5

Table 4. Experiment 1 - Simple Fitness: Best 5 results of 20 runs, according to their fitness value.

GP tree of the best individual in simple fitness mode is shown in Figure 1.

```
Tree 0:
(- (+ (- (+ (- (* CT BN) (MIN SECS M)) (-
(- (- (* (- (+ (- (+ (- MA (* BN SECS)) (-
(MIN SECS M) CT)) (MAX (SIN BC) (COS UCS)))
(- (- (- (* (* BN SECS) (MAX M UCS)) (+ (+
(- BC NN) BN) CT)) SECS) (/ BN M))) (- (+
(* BN SECS) (- NN CT)) (+ M CT))) BC) (+ M
CT)) SECS) (MAX BC MA))) (SIN (SIN (MAX (*
UCSH (* MA BC)) (+ MA MA))))) (- (- (-  (*
(- (+ (- (+ (- MA BN) (- (MIN SECS M)  CT))
(MAX (SIN BC) (COS UCS))) (- (- (- (* (* BN
SECS) CT) (+ (+ (- BC NN) BN) CT)) SECS) (/
BN M))) (MAX (* CT BN) CT)) BC) (+ M CT)) SECS)
(* BN SECS))) (SIN (SIN (MAX (* UCSH (* MA BC))
(+ MA MA)))))
```

Figure 1. GP tree of best individual – Simple Fitness.

Table 5 shows the confusion matrix of the best run (i.e. the second run):

	Positive	Negative
True	32.33%	65.33
False	1.33%	1.00%

Table 5. Confusion Matrix of the best run.

Now to have a better understanding of the whole experiment, refer to the Figure 2 which shows the convergence plot of these runs. This convergence plot is based on the average of the average raw fitness per generation over 20 runs as well as the average of the best raw fitnesses per generation over 20 runs.

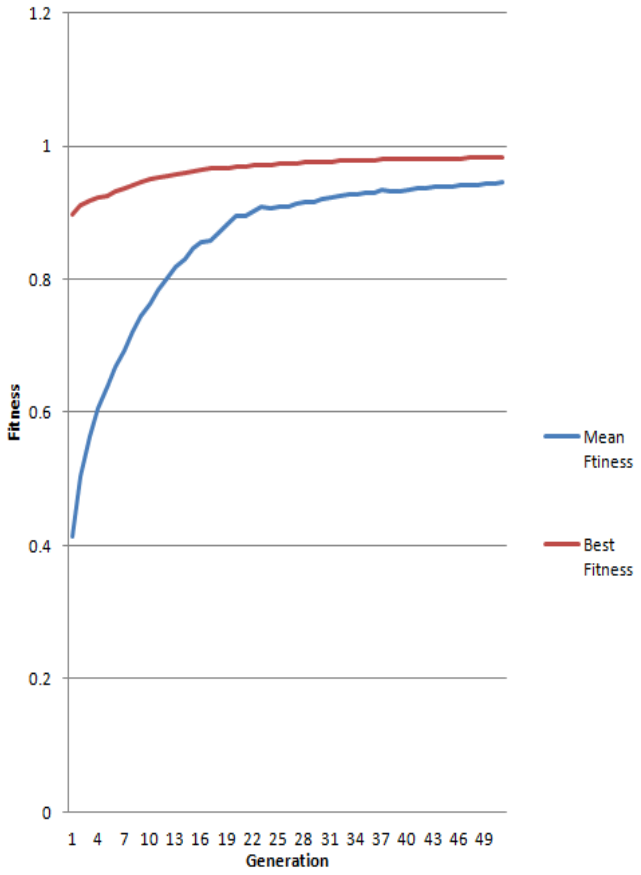

Figure 2. Convergence of the average of mean fitness and best fitness of 20 runs, over 50 generations.

This plot shows that the average of mean fitness of each run is getting better over generations and it goes from near 40% to more than 90%.

The next step, is to implement fitness sharing and replace it with simple fitness. After doing this, although the running time decreased significantly (from 7 minutes and 41 seconds, to 41 seconds) but the results were worse. Note that all the parameters are the same as simple fitness mode and the only thing different, is the way of calculating fitness. In this case, according to the fitness value, best result was for the 12th run with the fitness value of 93.5 out of 100. Table 6 shows the five best results of the fitness sharing. This table can be compared with Table 4 and it can be inferred that the results are not as good as expected.

Run Number	Fitness Value
12	93.5
8	92.25
7	88.75
3	84.0
15	84.0

Table 6. Experiment 2 – Fitness Sharing: Best 5 results of 20 runs, according to their fitness values

Figure 3, shows the GP tree of the best individual in the sharing mode. This tree is smaller and simpler, in comparison with the tree of the best individual which was obtained in the simple fitness mode.

```
Tree 0:
 (- (* (MIN UCS UCSH) (MAX NN BN))
 (MAX (+     BN CT) (/ MA BN)))
```

Figure 3. GP tree of the best individual – fitness sharing

Confusion matrix of the best run (12th run) in the sharing mode, is shown in table 7:

	Positive	Negative
True	30.67%	63.33%
False	3.33%	2.67%

Table 7. Confusion Matrix of the best run

Figure 4 shows the convergence plot of these runs in fitness sharing. This plot is based on the average of the average raw fitness per generation over 20 runs, as well as the average of best raw fitnesses per generation over 20 runs.

This convergence plot shows that after 5 generations, there is a significant reduction in fitness values. As we go through

generations, individuals which GP selects for breeding, are the better ones and because of the nature of the dataset which GP works on, there are not too many unique data which require unique individuals to classify them. Hence, after 5th generation, most of the individuals are able to classify most of the examples and because of the nature of fitness sharing, the reward will be shared among these individuals and this can be the reason of decrement in the average of raw fitness over generations.

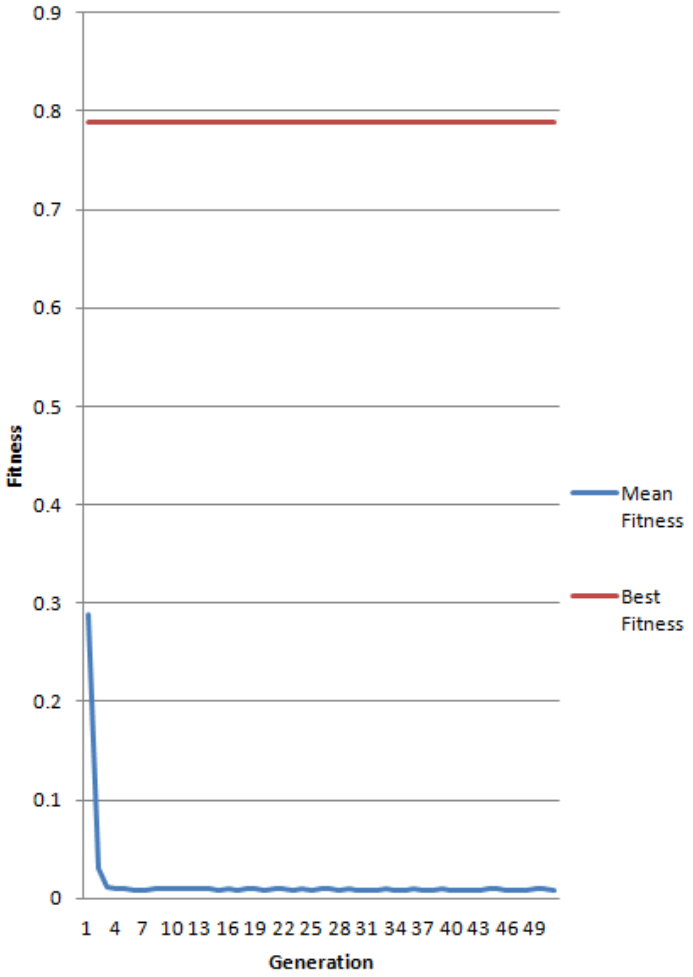

Figure 4. Convergence of the average of mean fitness and best fitness of 20 runs, over 50 generations.

In order to see the difference of these two approaches, an un-paired two sample t-test, assuming unequal variances was done. In this test null hypothesis is that the fitness sharing performs better than simple fitness and the alternate hypothesis is that the simple fitness is better than the fitness sharing.

The result of t-test is shown in Appendix. Since the value of t Stat is greater than the value of t Critical, and also since the p-value for one-tail is less than the significance in this test (which is 0.05), we should reject the null hypothesis and accept the alternate one. Hence, this t-test clearly states that in this application, simple fitness model performs better.

IV. Conclusion and Future Works

One of the most important parts of any GP program is the fitness calculation. A slight change in fitness function, can affect the running time and the results of GP program. By looking at the results section of this paper, effect and importance of fitness calculation can be understood. The results of this work showed that using the same parameters and configuration, simple fitness is performing better for detecting breast cancer. One thing to consider is applying fitness sharing is not always good and it depends on the application. If the purpose is to analyze images with tricky parts, using fitness sharing can be a great help **[8]**.

A problem that rises here is how to determine that for a specific application, fitness sharing is helpful or not. There could be some attributes in dataset (like tricky parts of data) which fitness sharing can perform well on them. Another thing is to find out the best configuration for fitness sharing. In this work all parameters were kept the same, in order to have a fair comparison between two approaches, but by changing the parameter values we may find better results.

V. REFERENCES

[1] **Definition from http://en.wikipedia.org/wiki/Genetic_programming, accessed on Jan 16, 2014**

[2] **Deb, K and Goldberg, D E: 'An investigation of niche and species formation in genetic function o`ptimization' in J D Schaffer (Ed) Proceedings of the Third International Conference on Genetic Algorithms, Pp 42-50, Morgan Kaufmann, 1989**

[3] **Smith, R E, Forrest, S and Perelson, A S: "Searching for diverse, cooperative populations with genetic algorithms", Evolutionary Computation 1(2), Pp 127-149, 1992**

[4] **R I (Bob) McKay: "Fitness Sharing in Genetic Programming". Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000). Pages 435-442, Las Vegas, Nevada, USA. 10-12 July 2000**

[5] **UCI Machine Learning Repository, Breast Cancer Wisconsin (Original) Data Set. "**http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29**" accessed on Dec 26, 2013.**

[6] **O. L. Mangasarian and W. H. Wolberg: "Cancer diagnosis via linear programming", SIAM News,Volume 23, Number 5, September 1990, pp 1 & 18.**

[7] **S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, Champaign, IL, USA, 1995, uMI Order No. GAX95-43663**

[8] **A. Bailey, "Evolving a Spacial Image Analyzer for Tree Detection Using Fitness Sharing", Brock University, St. Catharines, Ontario**

APPENDIX

This is the result of the t-test, a two sample t-test, assuming unequal variances:

	Variable 1	*Variable 2*
Mean	0.96508217	0.789113086
Variance	1.73543E-05	0.001503391
Observations	20	20
Hypothesized Mean Difference	0	
df	19	
t Stat	20.18008872	
P(T<=t) one-tail	1.35284E-14	
t Critical one-tail	1.729132792	
P(T<=t) two-tail	2.70568E-14	
t Critical two-tail	2.09302405	