CSC 535 Data Mining

Assignment #4 Report Collection

Submitted to:

Dr. Jamil Saquer

Author(s):

*Ryan Bagby*

*Michael Knapp*

**Report**

**Introduction**

*This data set was a lot easier to deal with than the previous data sets. This was just the boards that were given to us. The game is easy it has a set number of cells with a value in each of them. The player can either move by taking on the value of each cell or choose to jump the cell and take on that value. The fitness is chosen by the sum of value taken to reach the last cell on the board (which you must end with). This was a little easier for me to get my head wrapped around the logic this problem because this was given as a homework assignment for CSC 131 with Dr. Saquer.*

*Applying the genetic algorithm wasn’t’ to hard given the pseudo code to transfer to python code. Deciding what the appropriate cross over probability and the mutation probability was hard for us to come up with. We didn’t have any information other than crossover probability needed to be high and the mutation probability needed to be low. This left us with a lot of questions. Both we understand to be importation in the processing of the data.*

**Background**

*The genetic algorithm we used is just like it explains in the slides given to us on trace. The GA’s attempt to computationally mimic the process of nature in the fact that evolution uses mutations in natural selection to find the traits that might be best suited to ensure the species reproduces. Mutations will either make the specimen more fit or not. Fit specimens are more likely to reproduce and therefore slowing change things. How can anyone really argue the logic of this now days. BUT PEOPLE DO???? The five basic steps in this process are Selection, Recombination or Crossover, Mutation, evaluation, or Fitness more likely to reproduce and then selection all over again.*

*One thing that we did decide to pick in our program was the Crossover Probability Rate at 0.09. This was picked as an arbitrary number the only direction we had was it needed to be larger than the Mutation Probability Rate. The Mutation Probability Rate we choose to be 0.05*

**Implementation**

*I guess the first thing I want to talk about is the termination condition is set to 500 with the max\_gens variable. The slides talk about replacement being based on fitness or age, but our mutations are picked at random, so I think that is something to take notice of. Cost was chosen by the amount of moves or steps it took to get from the 0 value cell on the left to the last cell on the right. The fitness was calculated by the total of all the cells visited.*

*Talk about your implementation of the algorithm here. If there is anything nifty that you tried talk about it here too. If you had problems then talk about that here as we*

*You can use Notepad++ to copy in code snippets into your document if you want. Just highlight some code, right click and select* ***Plugin Commands -> Copy text with Syntax Highlights.***

Figure 1: Some Algorithm implementation

**def** someAlgorithmFunction**(**obj1**,** obj2**):**

#some descriptive comment

A **=** obj1**.**getArg**()** **+** obj2**.**getArg**()**

B **=** obj1**.**getArg**()** **-** obj2**.**getArg**()**

#some comment about the formula

result **=** **(**A**+**B**)/((**A**-**B**)\*(**A**+**B**))**

**return** result

**Experimental Setup and Results**

*Talk about how you setup your experiment. Give any performance results using standard performance metrics here. If you have graphs to illustrate your performance then put them here as well. Show output from the console or from your application here if necessary (as a picture or a table).*

**Conclusion**

*Give any concluding remarks here. If you learned anything talk about that here as well. If you discovered anything interesting (extra credit) then talk about it here too.*

**Code**

*You can use Notepad++ to copy your code right into your word document. Just* ***Ctrl-A****,* ***right-click*** *and select* ***Plugin Commands -> Copy text with Syntax Highlights.*** *Then in the word document:* ***right-click*** *and select* ***paste -> Keep Source Formatting****.**Otherwise just print out your code and staple it here. Please follow common programming conventions in your code: use descriptive variable names, don’t overwrite built-ins or keywords, write your code in a modular format (use classes and functions), etc. In the doc string at the top include the trace folder where we can find your code. If in a group: only one person should put the code on trace and list their trace folder here.*

"""

Program: HW\_Sample.py

Programmed By: Jared Hall

Description: A sample homework file containing some programming styles.

Trace Folder: Jared007

"""

#---------------------------------Imports--------------------------------------

#imports go here

**import** numpy **as** np

**import** math

#------------------------------------------------------------------------------

#---------------------------------Variables------------------------------------

#global variables go here

**global** globalVar

moduleVar **=** "SomeValue"

#------------------------------------------------------------------------------

#---------------------------------Classes/Functions----------------------------

**class** **SomeClass():**

"""A descriptive statement about the class"""

**def** \_\_init\_\_**(**self**,** arg1**):**

#some descriptive comment about the method

self**.**arg **=** arg1

**def** \_\_str\_\_**(**self**):**

**return** str**(**self**.**arg**)**

**def** setArg**(**self**,** newArg**):**

self**.**arg **=** newArg

**def** getArg**(**self**):**

**return** self**.**arg

**def** someAlgorithmFunction**(**obj1**,** obj2**):**

#some descriptive comment

A **=** obj1**.**getArg**()** **+** obj2**.**getArg**()**

B **=** obj1**.**getArg**()** **-** obj2**.**getArg**()**

#some comment about the formula

result **=** **(**A**+**B**)/((**A**-**B**)\*(**A**+**B**))**

**return** result

#------------------------------------------------------------------------------

#---------------------------------Program Main---------------------------------

**def** main**():**

**if(**\_\_name\_\_ **==** "\_\_main\_\_"**):**

#some subsection

object1 **=** SomeClass**(**2**)**

object2 **=** SomeClass**(**10**)**

#some processing comment

algorithmResult = someAlgorithmFunction(object1, object2)

object1.setArg(algorithmResult) #some descriptive comment

object2.setArg(someAlgorithmFunction(object1, object2))

print("Object 1 content: ", object1)

print("Object 2 content: ", object2)

main()

#---------------------------------End of Program-------------------------------