Evolutionary Algorithms

Ali Eltanani – 16044867

**1 INTRODUCTION**  
  
This report will explore how a classification problem can be effectively solved by applying a specific type of Evolutionary Intelligence, which in this case is Genetic Algorithms (GA). GA is a search algorithm used to find the optimum results in large data sets. It uses Darwin’s “Survival of the fittest” concept, by applying selection, crossover and mutation. Additionally, this report will also discuss various types of Evolutionary Algorithms.   
  
The task was to solve a set of classification problems by using a Genetic Algorithm. The algorithm was written to test four text files each of which contained data of the format: input variables - predicted variable. ‘Data1’ and ‘Data2’ contained binary data, while ‘Data3’ and ‘Data4’ contained real-valued data. One of the key parts of the problem was to split the given data to training and testing.   
  
**2 BACKGROUND RESEARCH**

Evolutionary Computation is a sub-field of Artificial Intelligence and is closely associated with computational intelligence. It engages in problem solving systems that use computational

models with evolutionary processes derived from biological concepts such as

‘genetic inheritance’ and ‘natural selection’ (Bäck, 1996). Techniques in this field are used on problems that have too many variables for traditional algorithms to consider and in times where the approach to solving a given problem is not well understood.  
  
Data mining is the process of finding patterns within a large data sets to predict outcomes. Using various range of techniques, these techniques can be used to increase revenues, cut costs, improve customer relationships, reduce risks ( Hand, 2007). It refers to the process of digging through data to discover hidden connections. Its sometimes referred to as “Knowledge discovery in databaseses”, the term “data mining” wasn’t coined until the 1990’s as it was known by ‘data dredging’, ‘trawling’ and ‘fishing through data’ (Hand, 2007).

However, data mining nowadays combines tools, ideas and methods from computer science, machine learning and database technology. According to studies by the EMC, the amount of data created each day is 2.5 quintillion bytes.

The main types of tools used in data mining are:

* Data Visualization
* Artificial Neural Networks
* Decision Trees
* Genetic Algorithms
* Rule Induction
* Nearest Neighbour Method

Firstly, Data Visualization uses raw data and algorithm results to produce tables and graphs using symbolic data analysis. The aim is to display the data along certain attributes and make extreme points, clusters and trends visible to the human eye (Looi, 2005).

Artificial Neural Network is an attempt to simulate the network of neurons that make up a human brain so that the computer will be able to learn things and make decisions in a humanlike manner. ANNs are created by programming regular computers to behave as though they are interconnected brain cells (Kumar, 2011).

Decision trees is decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resources costs, and utility. Its is one way to display an algorithm that only contains conditional control statements. However, the predictive performance of Decision Trees is usually weaker than other learning methods, such as Artificial Neural Nets (Apté & Weiss, 1997).

The Nearest Neighbor classifiers are based on learning by analogy. The method is mainly used when all the attribute values are continuous. The idea is to estimate an unseen instance’s classification using the classification or instance/instances that are closest to it (Phyu, 2009).

Rule induction methods attempt to find a covering or compact rule that partitions the examples into their correct classes. The compact rule set is found by searching for a single “best” rule that cover one classes’ cases. Thus, after finding the best rule for ‘Class A’, the rule is then added to the rule set. Rule Induction programs are sometimes added onto decision tree solutions, where a tree is initially generated and then translated into a rule set (Apté & Weiss, 1997).

Finally, Genetic Algorithms (GAs) which is the focus of this report. Genetic Algorithms were developed to simulate natural systems; Genetics and Natural Selection (Sampson, 1976). When GAs is applied, the solution’s variables are encoded into a string that represents a list of ‘genes’. The solution’s aim is its performance in a ‘survival competition’. Genetic Algorithms start by populating the solutions, then said solutions compete against each other and the winners (survivors) reproduce by exchanging some of their characteristics to form new solutions. After generations of the same process, the survivors become very similar to one another. By introducing a random initial population and performing a random crossover and mutation, the GAs produce a different solution each time (Koonce & Tsai, 2000).

The Ethical issues arises when mining is executed over personal data but its unlikely to lead to any consequences in mining of manufacturing data. The issues that arises because of data mining are privacy, security, Moral and how long the data collected is stored.

**3 EXPERIMANTATION**

‘Data1’ and ‘Data2’ contained binary data, while ‘Data3’ and ‘Data 4’ contained real-valued data. Each file contained conditions on the left-hand side and outputs on the right. ‘Data1’ and ‘Data2’ consisted of 60 rows of randomly generated rules (rule size: 7), 6-bit conditions and 1-bit outputs.  
  
**3.1 Data Set 1**

The strategy for ‘Data1’ was as follows:

* Creating an initial population
* Repeating the following until the correct solution is found:

- Evaluate each individual’s fitness  
- Select an individual for reproduction(parent)  
- Recombine pairs of parents

- Apply crossover and mutation to create a new individual

- Replace the individuals in the population (New Generation)

The Genetic Algorithm imports the Data file and reads each line. The program then combines all the lines into one string. Next, it extracts every 6th bit into a separate array.

After initializing the population, we apply a selection technique. While there are many selection methods used with Genetic Algorithms, two of the most popular methods are Roulette Wheel Selection and Tournament Selection. The Roulette wheel selection is a proportional selection strategy where the wheel is partitioned into several sectors by individual. The selection method is completely random, but it is easy to predict as the fact that it has more fitness means that there are more probabilities.

Tournament selection (the selection method used in this task) on the other hand, randomly selects two individuals from the population and the one with the highest fitness will be added to the next generation.

In Comparison with Roulette Wheel selection, Tournament selection is easier to use and is more efficient, however, the Roulette Wheel selection offers more diversity (Razali & Geraghty, 2011).

After selecting two individuals, the next step is applying Crossover. This is done by looping through the genes, picking a random point from the set and swapping it over. This creates a new solution (gene set).

Finally, we apply Mutation, after setting a mutation rate, some individuals mutate into the new gene set. This is done to avoid convergence and to explore more solutions.

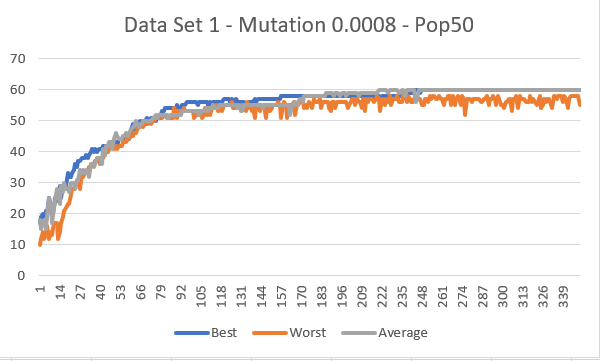
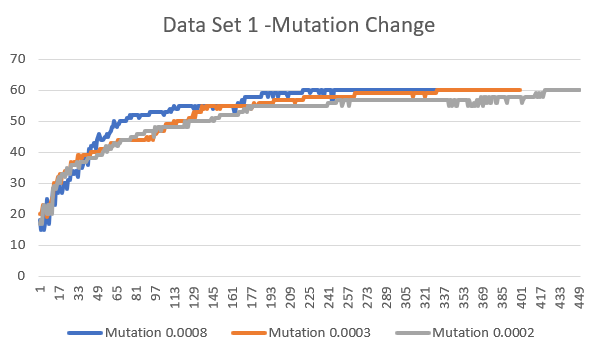


Figure 1: Data Set 1 - Initial Performance.

Figure 1 displays the result of a population of 50, with a mutation rate of 0.0008, over 400 generations. The fitness increases gradually until it reaches 60 and then it gets stuck at the local optima.



#

Figure 2: Data Set 1 – Mutation Change.

Figure 2 shows the effect or different mutation rates. Firstly, Mutation rate 0.0002, the fitness increases up to approximately 60, then there is no more evolution. This is because the algorithm is no longer exploring new solutions, thus prematurely converging to a solution (local optima). However, the best fitness was reached faster than the compared mutation rates (0.0003 and 0.0008). Next, Mutation rate 0.0003, the fitness increases up to approximately 60, then there is no more evolution. Finally, Mutation rate 0.0008 reached higher fitness values in less generations than the compared mutation rates, making it the ideal mutation rate in this scenario.

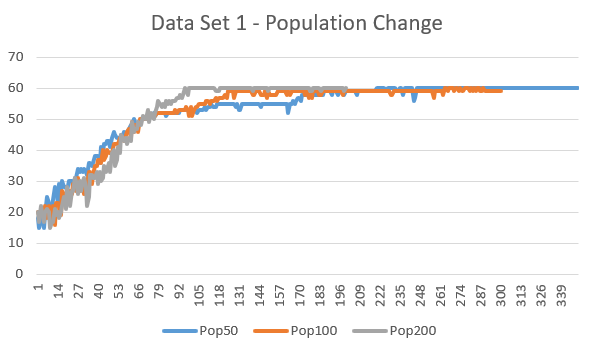


Figure 3: Data Set 1 – Population Change.

Figure 3 displays the effect of different population sizes (50, 100, 200). Population size 200 had the Highest fitness value of 60 in less generations. Both size 50 and 100 had reached the Maximum Fitness of 60, in more generations.

**3.2 Data Set 2**

Data Set 1 and Data Set 2 were very similar and used the same Rule Set.

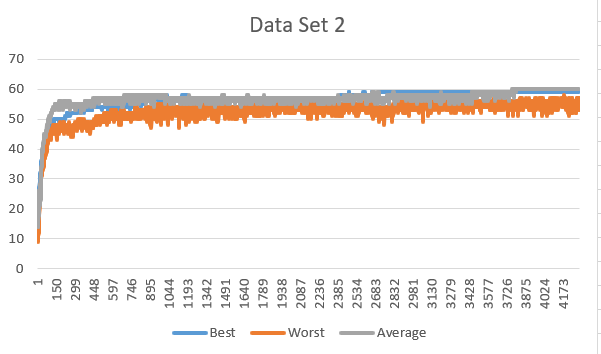


Figure 4: Data Set 2 – Initial Performance.

Figure 4 shows the result of Data Set 2’s initial performance from 4000 generations, while using a population size of 50 and a mutation rate of 0.0009. The Algorithm reaches the highest fitness (local optima) of about 60.

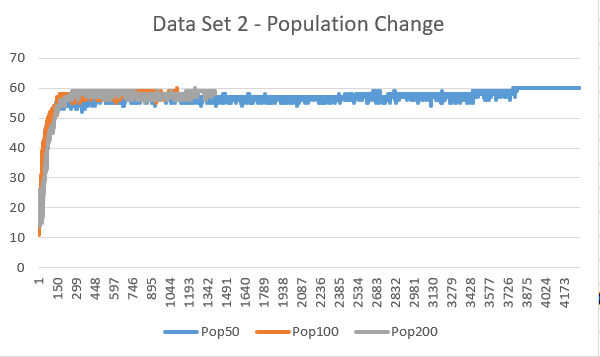


Figure 5: Data Set 2 – Population Change.

Figure 5 shows the effect different population sizes (50, 100, 200) have of Data Set 2. Similar to Data Set 1, lower population sizes resulted in lower accuracy, whilst higher population sizes did not necessarily result in higher accuracy. Additionally, Population size 100 reaching the highest fitness value, in less time.

4 CONCLUSIONS  
  
To conclude, while there may be many Evolutionary Search Algorithms, Genetic Algorithms remain one of the best options so solve classification problems. The Crossover and Mutation functions offer more diversity that helps avoid getting stuck at the local optima. Furthermore, the results discussed in this report demonstrated the effects of the Population Size and the Mutation Rate on the Data Sets. A population size that is too high does not guarantee higher accuracy, while low population sizes will have low accuracy. A mutation rate that is too low will result in the algorithm getting stuck at the local optima, while a mutation rate that is too high prevents the algorithm from converging to any optimal solution.

REFERENCES

- Ahmed, S. (2018) *Applications of data mining in retail business IEEE Conference Publication*. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1286695 [Accessed 25 November 2018].  
  
- Apté, C. & Weiss, S. (1997) Data mining with decision trees and decision rules. *Future Generation Computer Systems*. 13 (2-3), pp. 197-210.- Bäck, T. (1996) *Evolutionary algorithms in theory and practice*. New York: Oxford Univ. Press.  
  
- Dean, J. (2014) *Big data, data mining, and machine learning*. Hoboken (N.J.): J. Wiley.  
  
- Hand, D. (2007) Principles of Data Mining. *Drug Safety*. 30 (7), pp. 621-622.  
  
- Kumar, A. (2011) Artificial Neural Networks for Data Mining. *I.A.S.R.I.*. [Accessed 27 November 2018].  
  
- Looi, C. (2005) *Artificial intelligence in education*. Amsterdam [etc.]: IOS Press.  
  
- Phyu, T. (2009) Survey of Classification Techniques in Data Mining. *Proceedings of the International MultiConference of Engineers and Computer Scientists*. 1. [Accessed 27 November 2018].

- Raorane, A. & Kulkarni, R. (2011) Data Mining Techniques: A Source for Consumer Behavior Analysis. *International Journal of Database Management Systems*. 3 (3), pp. 45-56.  
  
- Raval, K. (2012) Data Mining Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2 (10), pp. 441. [Accessed 27 November 2018].  
  
- Razali, N. & Geraghty, J. (2011) Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. *Proceedings of the World Congress on Engineering*. 2. [Accessed 28 November 2018].  
  
- Sampson, J. (1976) Adaptation in Natural and Artificial Systems (John H. Holland). *SIAM Review*. 18 (3), pp. 529-530.  
  
- Koonce, D. & Tsai, S. (2000) Using data mining to find patterns in genetic algorithm solutions to a job shop schedule. *Computers & Industrial Engineering*. 38 (3), pp. 361-374.