Feature Selection of Augmented GIS Data Using Genetic Algorithms

Siwei Wu
School of Engineering and Computer Science
Australian National University

Abstract. Classification of Tree types based on data collected from a combination of Satellite, Photography and Soil maps is an efficient alternative to traditional ground surveys. This offers an opportunity for improvement in the field of Forrest Management. Studies have been conducted using traditional Neural Networks for Binary Class Tree Classification [2] and encoding the same data for multi-class Classification [1]. In this study, Evolutionary Algorithm technique is applied to select optimal features to be tested against a traditional back-propagation Neural Network based on [1] and encoded with technique described in [3]. The test involves several selection techniques to explore the effect on feature selection.

Keywords: Evolutionary Algorithms, Genetic Algorithm, Feature Selection, GIS Classification, GIS Encoding

1 Introduction

The right application of ML techniques in conjunction with the right dataset can lead to useful tools that could automate processes and improve efficiencies. In this example, an automated Tree species classifier based on data gathered using Geospatial Information System (Satellite imagery) enhanced with Photography and Soil maps, could replace the need for costly manual ground surveys and vastly reduce costs of Forest Management. The original dataset as defined in [2] includes 16 inputs and when encoded using [3], yields 22 inputs. Given that both [1], [2] postulated that the dataset may have redundant/non-independent data, it would be a useful exercise to identify an optimal subset of features.

In this study, I present the experience on applying Genetic Algorithm to the set of raw data collected from a forest in New South Wales, Australia [2] to identify subset (Chromosome) of input feature (Gene). The fitness function will be a standard feedforward Neural Network five category classifier as defined in [1] (with minor modification, see Fitness Function). In terms of the selection algorithm, several methods were tested and results compared.

The GA test were hindered by the significant run time required and though attempts were made in optimizing runtime using cuda, the small size of the dataset meant this was ineffective. Therefore, the test was conducted with a relatively low number of generations and population size in order to assess different GA Selection techniques.

Overall the test yielded promising results and interesting observations could be made on the input dataset. However, objectively, the application of GA was not able to significantly improve the performance of the NN classifier. This will be analyzed further in the Discussion Section.

2 Method

2.1 Data Encoding

The raw data input type and its encoding is detailed in [1] with a few minor exceptions. The goal of the input encoding is to ensure all datapoints are continuous and squashed between 0 and 1 with outliers removed using Z score. Features that were not included in [1] were encoded as described in Table 1 below. In total 22 Input parameters were encoded from 16 raw data parameters.

Table 1 Input Encoding (additional to [1] in Red)

Parameter Name	Para Code	Raw Code Description	Processing	Included
Aspect	AS	0: flat, 10: North, 20: NE, 30: East, ..., 70: West, 80: NE	Convert to A1-A4 IAW [2] representing aspects of a compass	
	A1	Nil, Generated Data		Yes

	A2			Yes
	A3			Yes
	A4			Yes
Sine Aspect	*SA*	*Unknown encoding of the Aspect data AS*	*Linear Squashing (with Z function bias clipping)*	*Yes*
Cos Aspect	*CA*			*Yes*
Altitude	AL	Metres above sea level	Linear Squashing (with Z function bias clipping)	Yes
Topographic Position	TP	One of five positions associated with numerical value -e.g. 32 = gully	Since only 6 position figures exist, this was manually mapped to 0.0 to 1.0 at 0.20 interval[1]	Yes
Slope	SL	% slope Encoded to numerical 10-80	Linear Squashing (with Z function bias clipping)[2]	Yes
Geology	*GE*	*Unknown encoding of unknown information*	*Convert to G1-G4* IAW [3]	
	G1	*Nil, Generated Data*		*Yes*
	G2			*Yes*
	G3			*Yes*
	G4			*Yes*
Rainfall	RA	(mm - 801)/5	Linear Squashing (with Z function bias clipping)	Yes
Temperature	TE	(degrees - 11)*30	Linear Squashing (with Z function bias clipping)	Yes
Landsat tm band	T1-T7	Landsat Thematic Mapper Wavelength Band 1-7[3]	Linear Squashing (with Z function bias clipping)	Yes

The output raw data consists of 5 parameters representing the 5 categories of trees. For each parameter a value of 90 indicates the pattern belongs to that category, and 10 indicates it does not. As suggested by [3], to avoid sparse output, Equilateral coding was utilized so the 5 parameters are converted into 4 and all patterns have some values in all 4 parameters. A classification function is then applied to the output where the Euclidean distance between the output and all 5-category patterns is calculated and is classified to the 'closest' category pattern. An example of this is shown below:

Table 2 Output Classifier Example[4]

Category	**Output**	**Scrub**	**Dry scler.**	**Wet-dry scler.**	**Wet scler.**	**Rain Forest**
Unit 1	0.2	0.1838	0.8162	0.5	0.5	0.5
Unit 2	0.2	0.3174	0.3174	0.8651	0.5	0.5
Unit 3	0.2	0.3709	0.3709	0.3709	0.8872	0.5
Unit 4	0.2	0.4	0.4	0.4	0.4	0.9
Distance from Output	Classified as Scrub	**0.2885 (**Min distance)	0.6802	0.7756	0.8320	0.8718

2.2 Standard Neural Network (Fitness Function) Design

The Fitness Function utilizes Standard Multilayered Perceptron Neural Network with a variable number of neurons input layer (dependent on the GA output), 10 neurons hidden layer and 4 neurons output layer. For GA fitness determination, 20 epochs will be performed using Adam algorithm for optimization with a learn-rate of 0.01. Mini batching is utilized (batch size 10) to reduce computation cost.

[1] Note. [3] specified that Topology encoding only ranges from 32 to 96 while the GIS data provided ranges from 16 to 96
[2] Z clipping has no effect on Slope data since the encoding constrains the data to 10 to 80 with no Z score above 3.0
[3] See https://www.usgs.gov/faqs/what-are-band-designations-landsat-satellites?qt-news_science_products=0#qt-news_science_products for detailed description of the band wavelengths and resolutions
[4] Table based on Table 5, [3] transposed

The fitness score will be the average accuracy of performing a 5-fold (in keeping with the same 80-20 training/test ratio test of [1]) cross-validation procedure using the NN described above against the full dataset.

The number of epochs was based on the findings in [1] (see [1], Figure 2) where it was observed that accuracy converged relatively quickly at around 50 epoch level. This was further reduced to 20 epochs due to runtime constraints (see 2.3.3.2).

2.3 Genetic Algorithm

2.3.1 Feature representation

The Chromosome design is based on [4] with a string of binary representation of the input feature selection, where 0 means it will not be used in the NN and 1 means it will be used.

For example, a Chromosome with a value of: "10101110000011100011", where the NN will be trained with 10 Inputs as specified below:

Table 3 Example Chromosome Representation (full set)

A1	A2	A3	A4	G1	G2	G3	G4	T1	T2	T3	T4	T5	T6	T7	SA	CA	AL	TP	SL	GE	RA	TE
1	0	1	0	1	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1

However, because the test is using relatively small population and generations (due to runtime constraints), the search space should be reduced. Therefore, the DNA Size was reduced by 6 to 16. This was achieved by considering Aspect (A1-A4) and Geology Encoding (G1-G4) as one bit each (see Table 4). The rationale is that since those two were each encoded from a single feature, using only a subset (e.g. A1, A3) of either would not make sense.

Table 4 Example Chromosome Representation (reduced set)

A1	A2	A3	A4	G1	G2	G3	G4	T1	T2	T3	T4	T5	T6	T7	SA	CA	AL	TP	SL	GE	RA	TE
1				1				0	0	0	0	0	1	0	0	0	0	1	1	0	1	1

Therefore, the GA will utilize 16-bit DNA and when inputted into the Fitness Algorithm, a Padding function is performed to create a 22-bit mask to select input features.

$$Padding([b_0, b_1 .... b_{15}]) = [b_0, b_0, b_0, b_0, b_1, b_1, b_1, b_1, b_2, b_3 .. b_{15}]]$$

2.3.2 Selection Techniques

Four Selection Techniques were utilized in the trial:

1. Proportional Selection –sampling based on Probability distribution proportional to the fitness (f(x)).

$$P_{selection}(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

2. Rank-Based Selection (specified in [5]) – sampling based on Probability distribution proportional to the Ranking of the fitness.

$$P_{selection}(x_i) = \frac{Rk(x_i)}{\sum_{j=1}^{N} Rk(x_j)}$$

3. Non-linear Rank Based Selection – sampling based on Probability distribution of a non-linear Ranking function:

$$P_{selection}(x_i) = \frac{1 - e^{-Rk(x_i)}}{\sum_{j=1}^{N} Rk(x_j)}$$

4. Elitism – The Top 10 of each generation are selected and subjected to crossover and mutation.

In addition, for all Selection methods, Hall-of-Fame technique is used to capture the fittest feature set for each generation.

2.3.2 Crossover and Mutation

Uniform crossover was utilized along with random mutation. The crossover rate was 0.8 and mutation rate 0.02. The rationale of utilizing Uniform over One-Two point crossover is that, the features have no proximity relations to each other except for T1-T7. However, since as demonstrated in [1], there is a level of linear dependence between those features, the assumption is that a subset of the 7 features selected would provide some level of representation of all 7 features.

2.3.3 Training Optimization

Early test showed that significant training time was required. Using 5-fold cross validation, during 50 epochs, it took approximately 2.2 seconds to run the fitness function. For a population of 100, it will take approximately 12 hours to run through 200 generations. Factoring the need for trial and error by tweaking hyperparameters, optimization was required to reduce training time. Two methods were pursued with varying success.

2.3.3.1 GPU acceleration

An attempt was made using CUDA acceleration to reduce training time. However, this experiment showed that instead decreasing training time, it increased it (see Table 3).

Table 3 –Running Time - 10 iterations of 100 epoch NN Run

Method	Processing Time
Cuda	37.62 seconds
CPU	13.74 seconds

The likely causes of this could be:

- The data size is not big enough for GPU to have any real performance advantage.
- The time to move data between GPU and CPU resulted in the additional processing time.

2.3.3.2 Reduced Epoch

Another way to improve processing time was to utilize a much smaller epoch size. The rationale is that the more optimal features will also converge faster and therefore for the purpose of generating comparable accuracies for selection, lower epoch size will still generate varying accuracy rates to enable selection. When considering when Rank selection is utilized, then the relative accuracy would be even less of a factor (i.e. 0.21 vs 0.22 is the same as 0.61 vs 0.62 in terms of rank).

3 Result

As shown in Table 4 and Figure 1 below, the overall performance did not vary much over the 200 generations for all selection methods except for Elitism. In general, Non-linear Rank, Rank and Proportional had similar performance with no noticeable improvement in performance over generations (See Figure 1,2) for both fittest of each generation or average fitness of each generation.

Elitism however, quickly converged so that the top 10 were all the same and the population only changed due to mutation. By Generation 18, the test was terminated.

Table 4 GA Highest Accuracy with Selection Methods

Selection Method	Gen 1	Gen 100	Gen 200	Highest Accuracy
Non-Linear Rank	0.6128	0.6289	0.6289	0.661
Rank	0.6236	0.6449	0.65	0.667
Proportional				
Elitism	0.6245 (Max at gen 18 – 0.565)			

Table 5 GA Average Accuracy with Selection Methods

Selection Method	Gen 1	Gen 100	Gen 200	Highest Accuracy
Non-Linear Rank	0.523	0.538	0.526	0.574
Rank	0.533	0.609	0.611	0.615
Proportional				
Elitism	0.61337 (Max at gen 18) 0.645~0.656	N/A (stopped in Gen 18 due to population convergence)	N/A	0.656

In terms of determining the optimal feature set, the raw output from the EA was inconclusive. Purely assessing highest accuracy was not suitable since the side-effect of using a small sized test set (only 38) means that the highest accuracy score is associated with many different feature sets. This is due to there being only 190 (5-cross variance x 38 datapoints) possible accuracy scores. Therefore, an alternative means must be found to determine the optimal feature set.

Figure 1 Max Accuracy per generation

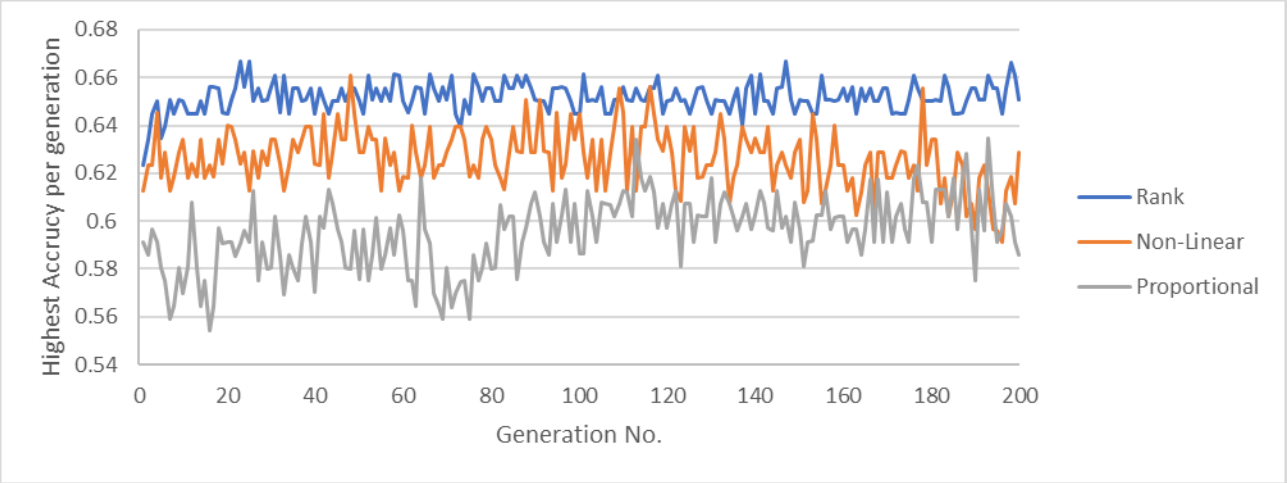
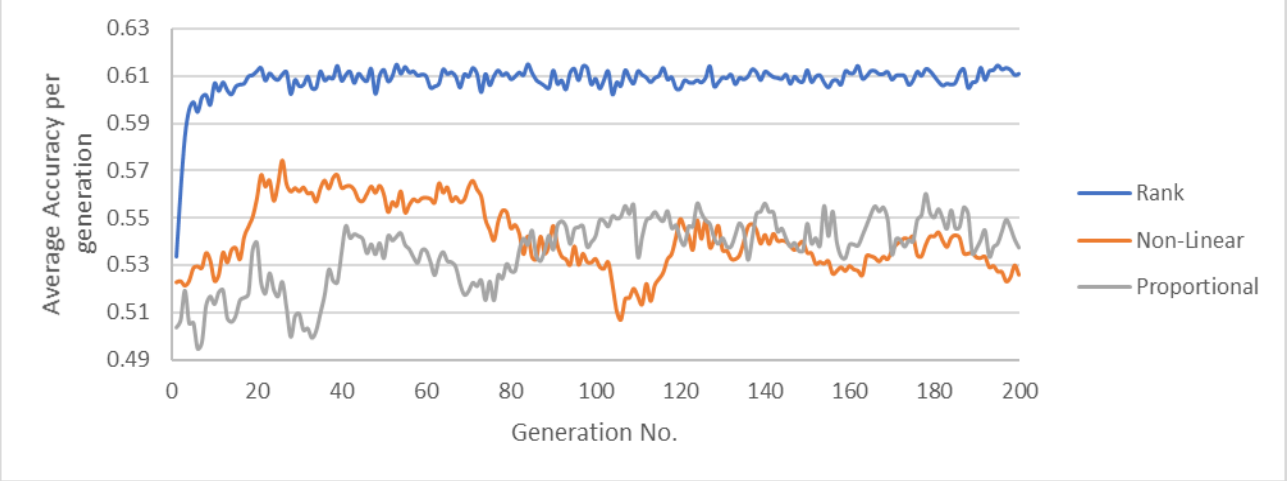


Figure 2 Average Accuracy per Generation



3.1 Statistical Analysis

A potential method for identifying optimal feature set is to calculate the average fitness score of each feature. The result is shown in figure 3 below. Due to runtime constraints and the need to re-run, only Rank selection criteria is tested with a reduced population size of 50. The results show that all features, when averaged over the 10,000 feature sets, did not exhibit significant variance in average accuracy.

An alternative method was utilized to graph the relative frequency of each feature within the Hall-of-Fame population (Figure 4).

Figure 3 Full GA data –Average Fitness score per Feature

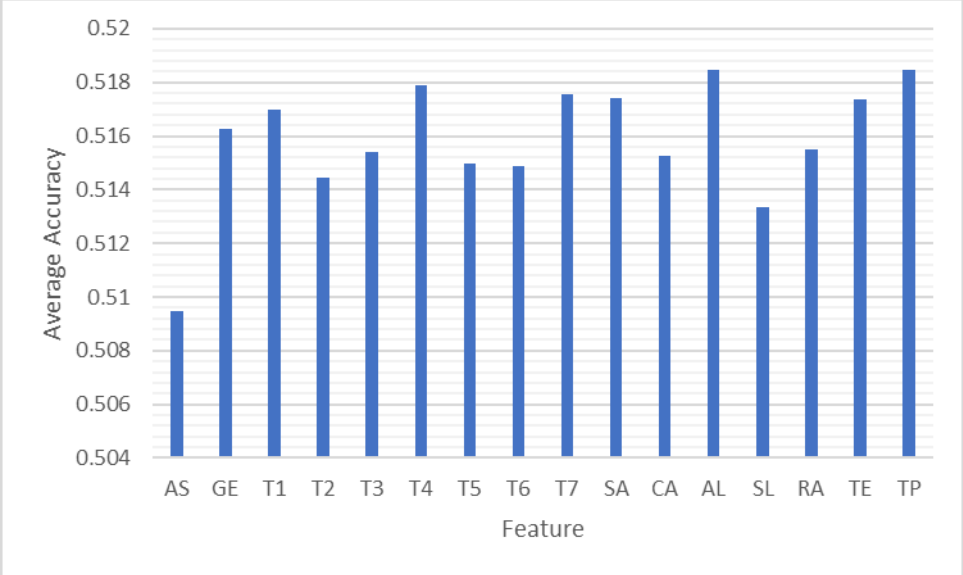
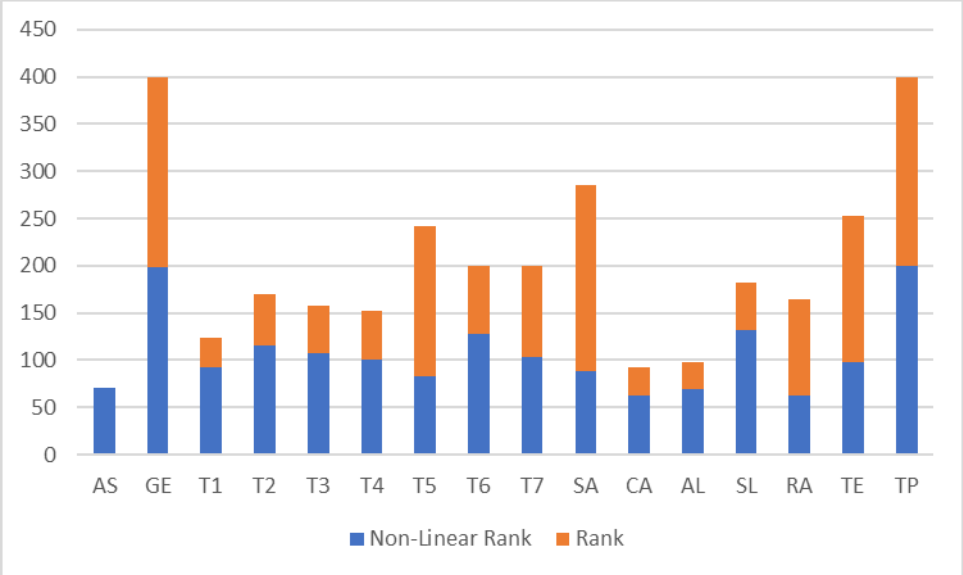


Figure 4 Hall of Fame - Features Frequencies



4. Discussion

Even though marginal improvement was observed over the generations and the frequency histogram approach was able to identify candidate optimal features, whether GA is necessary should be considered. The DNA length is only 16, the search space is only 65536 (2^16). For comparison, the 200 generations GA could theoretically cover a maximum of 20,000 unique feature sets. Therefore, using run time calculated in Table 3, it would take only approximately 25.5 hours to calculate the fitness scores (100 epochs NN) of all possible feature set.

By examining the result, another observation could be made that some feature sets generated different accuracies, this is likely because of a combination of small epochs size and random weight initialization. At 20 epochs, the NN has not yet

stabilized, contributing to varying results. This adds a degree of uncertainty to the veracity of the identified optimal features.

In terms of effects on the problem space, the Hall of Fame histogram (Figure 4) can be a useful aide to prioritize data collection and optimize data collection. In this example, by observing the feature frequencies, the most important satellite data would be T5 to 7, and if there is an alternative satellite source that collects data in those same spectrum range with better precision, it could be used to improve performance. If resources were allocated for collection of ground data, then Geological Encoding (GE) and Topographic Position (TP) should be prioritized.

5. Conclusion and Further Work

Even though the effectiveness of GA was constrained by the runtime, the result, especially the output histogram could still be useful in feature selection. Though the same data could be obtained through randomly selecting chromosomes and recording high accuracies rather than feature selection through EA.

Future work could utilize other Selection techniques such as Entropy-Boltzmann as specified in [6] or a modified version of GA such as the Tribe competition method outlined in [8]. In terms of optimizing runtime, instead of simply passing feature data into GPU to optimize NN based fitness function runtime, the GA could be redesigned with similar techniques described in [7]. Another potential method for improvement would be to find optimal feature sets with a fixed length by using EA against integer representation of a fixed DNA length where each Gene will present an integer value between 0 and 16 (or 0 and X number of features).

Another consideration is that any future extension of this study should utilize a dataset with a much larger feature space than the GIS data used in this study. As Discussion shows, it would not cost significantly more computation to just evaluate the entire search space and find the absolute optimal feature set.

References

1. Wu, S. Classifying Tree Types from Augmented GIS Data Comparing Unidirectional with Bidirectional Neural Network, 2020, 3rd ANU Annual Bio-Inspired Computing Conference
2. Milne, L. K., Gedeon, T. D., & Skidmore, A. K. (1995). Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. In *Proceedings Australian Conference on Neural Networks* (pp. 160-163).
3. Bustos, R. A., & Gedeon, T. D. (1995). Decrypting Neural Network Data: A GIS Case Study. In *Artificial Neural Nets and Genetic Algorithms* (pp. 231-234). Springer, Vienna.
4. Sharma, Nandita & Gedeon, Tom. (2013). Hybrid Genetic Algorithms for Stress Recognition in Reading. 7833. 117-128. 10.1007/978-3-642-37189-9_11.
5. Kumar, Rakesh & Jyotishree,. (2012). Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms. International Journal of Machine Learning and Computing. 365-370. 10.7763/IJMLC.2012.V2.146.
6. Lee, Chang-Yong. (2003). Entropy-Boltzmann selection in the genetic algorithms. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society. 33. 138-49. 10.1109/TSMCB.2003.808184.
7. Sinha, Rashmi & Singh, Satvir & Singh, Sarabjeet & Banga, V K. (2016). Accelerating Genetic Algorithm Using General Purpose GPU and CUDA. International Journal of Computer Graphics. 7. 17-30. 10.14257/ijcg.2016.7.1.02.
8. Ma, Benteng & Xia, Yong. (2017). A Tribe Competition-Based Genetic Algorithm for Feature Selection in Pattern Classification. Applied Soft Computing. 58. 10.1016/j.asoc.2017.04.042.

Appendix

A. Code Readme (see attached - README.txt)
B. Genetic Algorithm Source Code (see attached – Assignment2 – u6735397.ipynb)