Comparative study of Neural Network approach and Genetic Programming approach to Land Cover Mapping

M. G. PRASAD, S. N. OMKAR, V. MANI
Department of Aerospace Engineering,
Indian Institute of Science,
Bangalore
INDIA
{

H. HONNE GOWDA
Karnatake State Remote Sensing Application Center
Bangalore
INDIA

Abstract:- This paper explores the feasibility of applying Neural Networks and Genetic Programming to Land Cover Mapping problem. Land Cover Mapping has been done traditionally by using the Maximum Likelihood Classifier (MLC). Neural Networks (NN) and Genetic Programming (GP) classifiers have advantage over statistical methods because they are distribution free, i.e., no prior knowledge is needed about the statistical distribution of the data. Neural Network has been applied for the classification but we may not be sure of getting the optimal solution. GP has the ability to discover discriminant features for a class. GP has been applied for two-category(class) pattern classification. This idea is extended to n-class image classification problem by modeling the problem into n two-class problems, and a genetic programming classifier expression(GPCE) is evolved as a discriminant function for each class. The GPCE is trained to recognize the samples belonging to its own class and rejecting samples belonging to other classes. Experimental results are presented to demonstrate the applicability of Neural Network and GP for land cover mapping problem, and the results are found to be satisfactory.

Keywords : Neural Networks, Genetic Programming, Land Cover Mapping, Pattern Classification.

1 Introduction

Land cover mapping is the assessment of different land cover types over a certain geographic extent. These are generally obtained through satellite images. All objects emit different types of radiations. These radiations are captured by sensors from remote sensing satellite. This signal is converted into digital format and according to the spectral signatures, land cover types are found.

Remote sensing is a powerful tool for the regional mapping of natural resources. Remote sensing has become important in pattern classification from viewpoint of global environment. Pattern classification methods for remote sensing are mainly based on statistical methods such as Maximum Likelihood Classifier (MLC) or Bayesian methods [2]. In this case, classification is performed in a digital way under the guassian assumption and any pixel is classified into one of patterns. But in general, it may be possible that the distribution of pixels in a particular class may not be guassian.

This drawback can be overcome by using intelligent algorithms like NN, Fuzzy systems, Genetic Programming etc. [5][3]. Neural Network does not require any prior knowledge about input data as in the case of conventional algorithms. Neural Networks have strong power to classify various patterns as the human brains can do[1]. Especially the ability of image processing or pattern recognition is excellent than that of the conventional algorithms [4]. Also, Neural Network has ability to resolve non-linear separability between patterns. This makes Neural Network a better alternative for Land Cover Mapping. Yoshida et.al. [5] used Self Organizing Map(SOM) in conjunction with Multi-Layer Perceptron(MLP). But we found that conventional K-means algorithm in conjunction with MLP is also effective so we used K-means algorithm instead of Kohonen's algorithm.

The feasibility of using Genetic Programming to land cover mapping problem is also explored in this paper. GP has been formulated originally as an evolutionary method for breeding programs using expressions from the functional programming language LISP [6]. Genetic Programming has the ability to learn the underlying data relationship between the inputs and outputs, and express them in a mathematical manner. Although, GP uses the same principles as genetic algorithms(GAs) [7], it is a symbolic approach to program induction, i.e., it involves the discovery of highly fit computer program, from the space of computer programs that produces a desired output, when

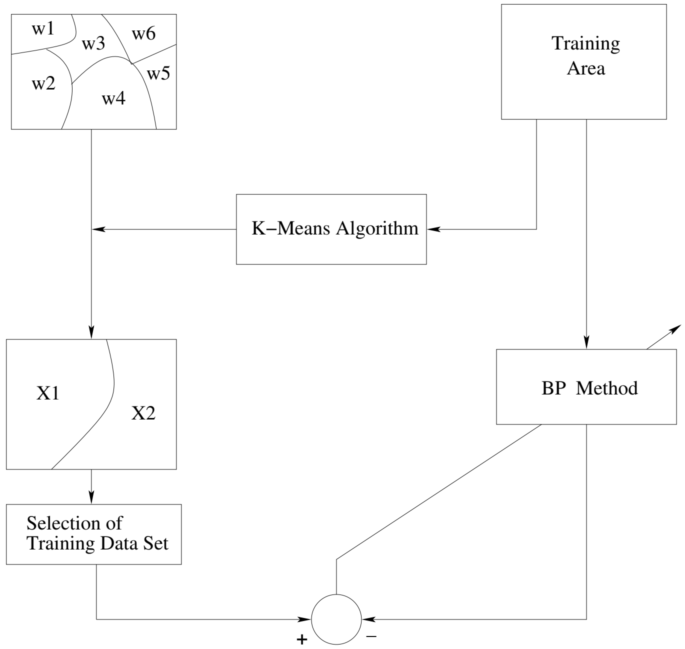

Figure 1: Classification Algorithm

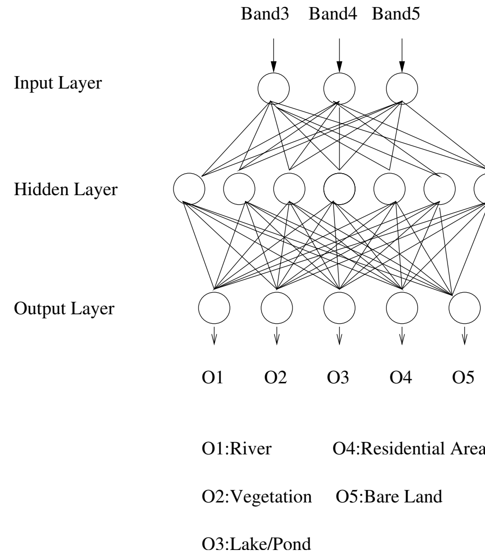

Figure 2: Structure of Multilayer Perceptron used in Back-propagation algorithm. Number of hidden Layer neurons can vary from 9 to 14.

presented with a particular input. Hence GP is used for getting the optimal solution. This paper is divided in two parts. In first part, the Neural Network based approach for land cover mapping is discussed. In this approach unsupervised classification using K-Mean's algorithm and supervised classification using Back Propagation (BP) algorithm are combined. In second part GP based approach is discussed in which Land cover Mapping problem is converted to n two-class problems and then each problem is solved using GP [8].

In this paper, we use IRS-1D LISS-III data for remote sensing data analysis. It possesses four bands from Band 2 to 5 and spatial resolution is $23 \times 23\, m^2$ for all bands. We use Bands 3 to 5 among LISS-III data.

2 PART I :Neural Network Based Approach

The classification algorithm using neural network is illustrated in Fig.1. Remotely sensed data contain a number of categories, but we focus on a very few which are prominent ones. So by applying K-Means algorithm for unsupervised classification, clusters of prominent categories are found out. Using ground truth information and cluster information, training data set is formed. This data set is used for training of supervised network. The remote sensing data does not include perfect geographical information and image data given by remote sensing is subject to noise, such as path radiance or back scattering effect. Therefore after training of MLP, some of the training samples may not be correctly classified. Such data samples will be removed from the original training data set and new training data set is built. This new training data set is used for further training of supervised network.

Input for land cover mapping problem is Correlated Color Temperature (CCT) levels of LISS data from Bands 3 to 5. Output should be the correct class corresponding to input pixel.

2.1 Unsupervised Classification

In Unsupervised Classification, simple K-Means algorithm [2] is used to find out the clusters in given satellite image. In this problem input data will be in the form of 3-dimensional vector corresponding to intensity level in three bands. Using clustering information and ground truth, input data set is formed. Input data set is randomly divided in two sets, one is used for training of MLP while other is used for testing.

2.2 Supervised Classification

Feed Forward Network used for the supervised classification is given in Fig.2. At input Layer of MLP, Correlated Color Temperature(CCT) levels of LISS data from bands 3 to 5 are applied to each neuron. At the output layer, the neuron's outputs are compared with desired values that represent a set of classification patterns. The classification patterns adopted here are River(O_1), Vegetation(O_2), Lake/Pond(O_3), Residential Area(O_4) and Bare Land(O_5). Training data set is formed from clustering information and ground truth. Generalized Back Propagation algorithm[4] is used for the weight updating. In hidden layer *tangential sigmoid* function is used as activation function while in output layer *purely linear* function is used as activation function. Even after sufficient number of epochs, if MSE is not reduced to a acceptable limit then the some of pixels which are being incorrectly classified are deleted from the original training data set and new training data set is formed. Using new training data again MLP is being trained. The process is continued until the MSE error come sufficiently small. This newly trained network is used to classify the testing data set.

3 PART II : Genetic Programming Based Approach

In the GP framework of program discovery, the GP algorithm is supplied with training data, a set of primitives and a fitness function[6]. The training data consist of samples, in which each data describes the specific problem, in terms of desired inputs and desired (known) outputs.

Primitives are functions or variables that can be used by the algorithm to compose a program. Each composition of primitives (or program) is a candidate solution, to the specific problem described by the inputs and outputs. Fitness function is a measure that can be applied to any program that is a candidate solution. It is computed by executing the program with training data, one at a time and then measuring, how similar it's computed outputs are to the desired (known) outputs. The only feedback available to the program discovery process is a program's fitness value.

A program that meets all the functional requirements by virtue of learning all the training data has perfect fitness. The objective of the GP algorithm is to search for a program of perfect fitness in the space of computer programs. The only information used in the search is the fitness values. Now, we will describe how GP works.

Let F = f_1, f_2..., f_n be the set of functions, T =x_1, x_2, ... ,x_n be the set of terminals. The functions in the function set may include,

- Arithmetic operations (+, -, *, /)
- Mathematical functions (sine, cos, exp, log)
- Boolean operators (such as AND, OR, NOT)
- Conditional operators (such as IF THEN ELSE)

The set of possible structures, i.e., computer program in the GP, is the set of all possible composition of functions that can be composed from F to T.

GP begins with a population of randomly created computer programs. Each computer program represents a possible solution. GP maintains a population of solutions to the given problem. During every generation, fitness of each solution is evaluated, i.e. execute each program and assign a fitness value. Fitness is the measure that shows how well the computer program solves the problem.

For the next generation, the computer programs (or solutions) are selected based on their fitness. In the next generation, the populations of computer programs are obtained by keeping the best existing programs, and through the application of mutation and crossover. Termination of the GP is based on specific value of the fitness of the best computer program that appeared in any generation or based on number of generations.

3.1 Multicategory Pattern Classification

GP has been used for a two-class pattern classification problem [6]. In a two-class problem, a single GP expression is evolved. While evaluating the best GP expression, if the result is 0, the input data is assigned to one class (say class-1); else they are assigned to the other class (class-2). We call this evolved GP expression as GPCE(Genetic Programming Classifier Expression) for pattern classification problem. So one GPCE is sufficient for two class problem.

To classify n-class data set, we extend the problem to n two-class problems [8]. For the sake of illustration consider 3-class problem. Let n_j be the number of samples that belongs to class j and N_j be the number of samples that do not belong to class $j(j = 1, 2, 3)$.

Thus,

$$\begin{aligned} N_1 &= n_2 + n_3 \\ N_2 &= n_1 + n_3 \\ N_3 &= n_1 + n_2 \end{aligned} \tag{1}$$

When three class problem is formulated as three two-class problem, we need three GPCE to discriminate n_1 and N_1, n_2 and N_2, n_3 and N_3. Thus, each of these three two-class problem are handled as three separate two-class problem as discussed above. Each GPCE partitions the feature space differently into two regions. Thus, for an n-class problem, n GPCE's are evolved.

3.2 Fitness measure

GP is guided by fitness function to search for the most efficient computer program to solve a given problem. A simple measure of the fitness has been adopted for Pattern Classification problem.

$$fitness = \frac{\text{Number of samples classified correctly}}{\text{Number of samples used for training during evolution}} \tag{2}$$

3.3 Function set

In our study, for evaluating the GPCE we have used the function set with only arithmetic operations (+, -, *, /).

3.4 Termination Criterion

Koza [6] has shown that in GP, evolution is never-ending process, and hence a termination criterion is needed. The termination criterion for the GP is based on the problem or limited by the number of generations. In our case, we have used maximum number of generations and maximum fitness value.

4 Simulation

The color composite image of the study area is shown in Fig. 3. Corresponding ground truth information is given in Fig. 4.Simulation results are divided into two parts.

Table 1: The Confusion Matrix with Average Accuracy and Overall Accuracy for Neural Network Classifier where I = River, II = Vegetation, III = Pond/Lake, IV = Residential Land, V = Bare Land

		Pattern Recognition Results				
		I	II	III	IV	V
Ground Truth	I	31627	2217	28	65	3851
	II	0	34777	316	7	0
	III	7033	16	14993	60	1029
	IV	6	1465	1122	18903	71
	V	98	4200	0	59	24010

Average accuracy = 78.2% Overall Accuracy = 84.2%

4.1 NN Based Approach

Using the Neural Network classification algorithm(Fig. 1) the classification is done. The training curve before the erroneous pixels are removed from training data, is shown in Fig.5. This curve shows that the MSE is not becoming small even after sufficiently large number of epochs. Corresponding classification result is shown in Fig. 7. Erroneous pixels which are incorrectly classified are removed from training data and again network is trained. The training curve for this is shown in Fig. 6. Corresponding classification result is shown in Fig. 8. Table 1 shows the confusion matrix along with average accuracy and overall accuracy for Neural Network based classifier.

4.2 GP Based Approach

The remote sensed image considered in this paper is available in three bands: Band-3, Band-4 and Band-5. So the input for each pixel will be its intensity value in three bands. We are classifying the image in five classes as River, Vegetation, Pond/Lake, Residential Area and Bare Land. We will call these classes as Class A, Class B, Class C, Class D and Class E respectively. The given five class problem is converted into 5-two class problems. These five two-class problems are defined as below,

$$P_1 = \text{Class A} + \text{ Class(B, C, D, and E)}$$
$$P_2 = \text{Class B} + \text{ Class(A, C, D, and E)}$$
$$P_3 = \text{Class C} + \text{ Class(A, B, D, and E)}$$
$$P_4 = \text{Class D} + \text{ Class(A, B, C, and E)}$$
$$P_5 = \text{Class E} + \text{ Class(A, B, C, and D)}$$

4.2.1 Creation of Training Set

As the n-class problem has been converted into n two-class problems, n GPCEs are evolved, and so n GPCE specific training sets are needed. In each GPCE specific training data set, the number of samples belonging to one class (whose desired output is $+1$) is outnumbered by the samples belonging to all other classes. For example, in our problem, let the number of samples belonging to each class be 100. Thus, in our formulation $n_1 = 100$ and $N_1 = 400$.

For creation of these samples we have used the K-Means Algorithm as in Neural Network Algorithm.

4.2.2 Generating GPCEs

The training samples are used to obtain the genetic programming classifier expression (GPCE). We use three runs of GP with the training set and obtain the best computer program (GPCE) evolved at each run for each two-class problem. In all these runs, we have used the termination criterion as 99.5% classification accuracy or 100,000 generations. At the end of each run, the best computer program evolved is in the form of LISP s-expression This expression can be easily converted into a mathematical expression.

4.2.3 Validation of GPCE

The validation sets are used to analyze the performance of the GP classifier for each two-class problem. The confusion matrix (q_{ij}) is obtained by applying the validation set to GPCE of each two class problem. The size of confusion matrix is $n \times n$, where n is the number of classes. A typical entry q_{ij} in the confusion matrix shows how many samples belonging to class i have been classified, as class j. For a perfect classifier the confusion matrix is diagonal. However in practice, due to misclassification, we get off-diagonal elements.[9]

Classification results using GP are shown in Fig. 9. Table 2 shows the confusion matrix along with average accuracy and overall accuracy for GP classifier.

5 Conclusion

In this paper, we have demonstrated the applicability of neural networks and genetic programming to land cover mapping problem. When we applied Neural Networks, we observed that MSE is not coming down sufficiently regardless of number of epochs so we removed the pixels which are being incorrectly classified, from training data. It improved the training as well as the classification.

Though the performance of neural networks is good, it is not optimal. So we applied GP for this problem. To solve this problem using GP, we converted n-class problem to n two-class problems.

In both approaches i.e. Neural Network and Genetic

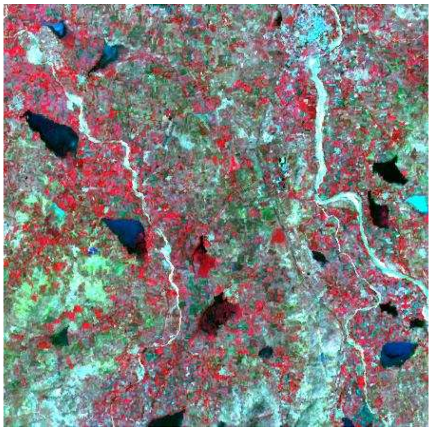

Figure 3: Color Composite image

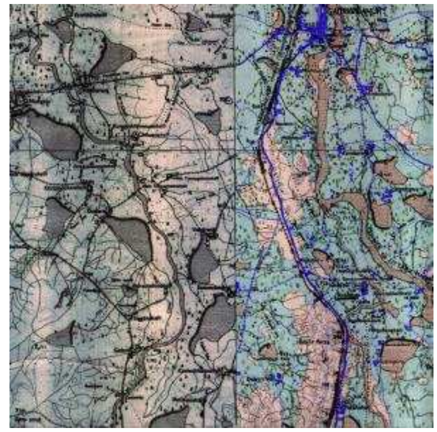

Figure 4: Ground Truth

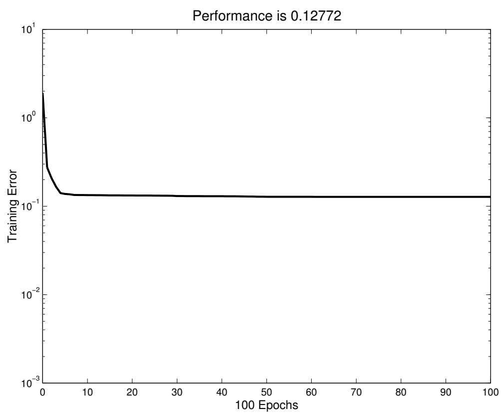

Figure 5: Training curve of the Neural Network with training data without deleting erroneous pixels Final Training Error = 0.127

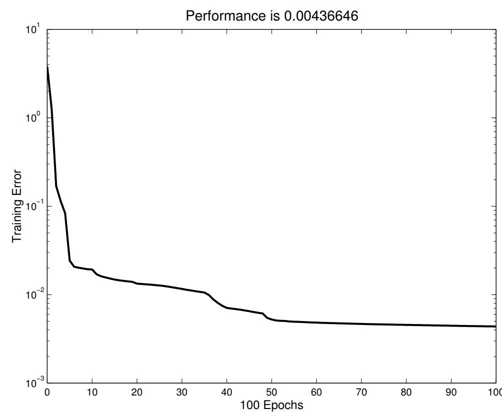

Figure 6: Training curve of the Neural Network with training data after deleting erroneous pixels Final Training Error = 0.004

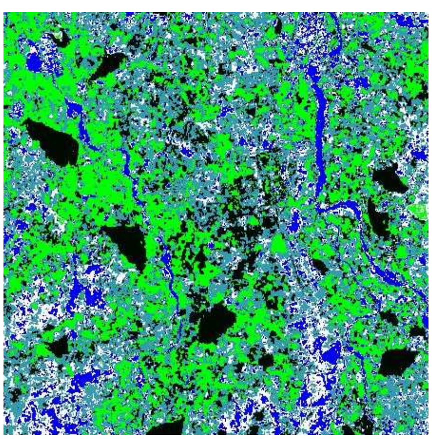

Figure 7: Pattern Recognition Result by the Neural Networks without deleting erroneous pixels

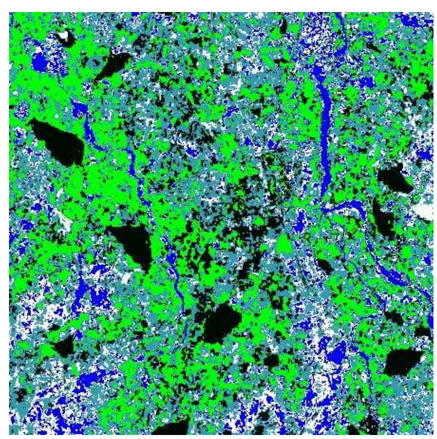

Figure 8: Pattern Recognition Result by the Neural Networks after deleting erroneous pixels

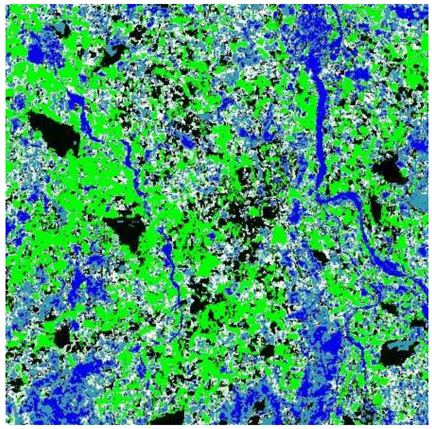

Figure 9: Pattern Recognition Result by the Genetic Programming

Table 2: The Confusion Matrix with Average Accuracy and Overall Accuracy for Genetic Programming Classifier where I = River, II = Vegetation, III = Pond/Lake, IV = Residential Land, V = Bare Land

		Pattern Recognition Results				
		I	II	III	IV	V
Ground Truth	I	37622	9	59	0	138
	II	0	56330	3	32	0
	III	3	42	15281	30	129
	IV	216	35	11	20431	0
	V	189	0	100	0	28255

Average accuracy = 99.1% Overall Accuracy = 99.3%

Programming, we are classifying the data in five areas as river, vegetation, pond/lake, residential area, and bare land. From results it is seen that GP can classify land cover types more accurately than NN.

References

[1] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Reading, MA:Addison-Wesley, 1989.

[2] Duda R. O. and Hart P. E., *"Pattern Classification and Scene Analysis"*, New York :Wiley, 1973.

[3] Du L. E., Lee J. S., and Mango S. A., *"Fuzzy Classification of earth terrain covers using multi-look polarimetric SAR image data"*, IGARSS '93;vol.4, pp. 1602-1604.

[4] Simon Haykin, *Neural Networks - A Compressive Foundation*, New York, McMillan College, 1994.

[5] Tomoji Yoshida and Sigeru Omatu, *"Neural Network Approach to Land Cover Mapping"*, IEEE Trans. on Geoscience and Remote Sensing, vol. 32, no.5, pp.1103-1109.

[6] J. R. Koza, *Genetic Programming: On the programming of computers by means of natural selection*, Cambridge MA:M.I.T. Press, 1992.

[7] D. E. Goldberg, *Genetic Algorithms in search, Optimization and Machine Learning*,Reading,MA: Addison-Wisley, 1989.

[8] J. K. Kishore, L. M. Patnaik, V. Mani, and V. K. Agarwal, *"Application of genetic programming for Multicategory pattern classification "*, IEEE Trans. on Evolutionary Computation, 2000; vol. 4, pp. 242-258.

[9] Brandt Tso, and Paul M. Mather, *Classification Methods for Remotely Sensed Data*, Taylor and Francis, London, 2001.