Stochasticity applied to a Neural Tree Classifier

W. Pensuwon, R.G. Adams, and N. Davey

Abstract— **This paper describes various mechanisms for adding stochasticity to a dynamic hierarchical neural clusterer. Such a network grows a tree-structured neural classifier dynamically in response to the unlabelled data with which it is presented. Experiments are undertaken to evaluate the effects of this addition of stochasticity. These tests were carried out using two sets of internal parameters, that define the characteristics of the neural clusterer. A Genetic Algorithm using appropriate cluster criterion measures in its fitness function was used to search the parameter space for these instantiations. It was found that the addition of non-determinism produced more reliable clustering performances especially on unseen real world data.**

Keywords— **Competitive learning, Dynamic Neural Tree, Genetic Algorithm, Hierarchical Clustering, Neural Network Classifier, Stochastic Neural Networks.**

I. Introduction

The standard Neural Network Competitive Learning algorithm [7] may be modified by the addition of dynamic node creation and the imposition of a tree structure on the classificatory ordering of the nodes. This brings two main advantages: the number of clusters that the neural network will identify does not need to be predefined, and the hierarchical tree structure improves the interpretability of the results. In addition, the use of a tree structure allows a more efficient search for the classifying node so increasing the speed of the model. A basic neural network hierarchical clusterer has been introduced in [1,2,3]. The latest version of which is called CENT II.

In this paper, we introduce stochasticity to the basic competitive hierarchical clusterer. The main goal of this addition is to make the performance of the basic model more robust to internal parameter settings and able to produce a suitable classification over a large variety of data sets. The basic competitive evolutionary neural tree model is described in Section 2. Three different forms of stochasticity that can be added to the CENT II model are introduced in Section 3. The experiments performed are described in Section 4, and the results are reported and illustrated in Section 5. Finally, some discussion and conclusions are given.

II. Competitive Evolutionary Neural Tree (CENT II)

In CENT II, the tree structure is created dynamically in response to structure in the data set. The neural tree starts with a root node with its *tolerance* (the radius of its classificatory hypersphere) set to the standard deviation of input vectors and its position is set to the mean of input vectors. It has 2 randomly positioned children. Each node has two counters, called *inner* and *outer*, which count the number of occasions that a classified input vector is within or outside tolerance, respectively. These counters are used to determine whether the tree should grow children or siblings once it has been determined that growth is to be allowed.

A. Top-Level of Algorithm

At each input presentation, a recursive search through the tree is made for a winning branch of the tree. Each node on this branch is moved towards the input using the standard competitive neural network update rule.

Any winning node is allowed to grow if it satisfies 2 conditions. It should be mature (have existed for an epoch), and the number of times it has won compared to the number of times its parent has won needs to exceed a threshold. A finite limit is put on the number of times a node attempts growth.

When a node is allowed to grow, if it represents a dense cluster, then its inner counter will be greater than its outer counter and it creates two children. Otherwise, it produces a sibling node. The process of growth is illustrated in Figure 1.

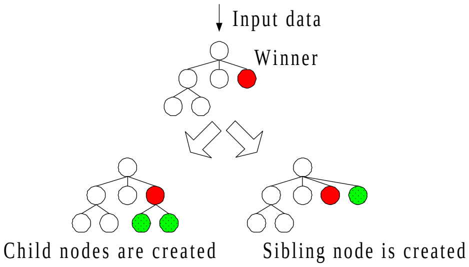

Figure 1. Process of growing a tree

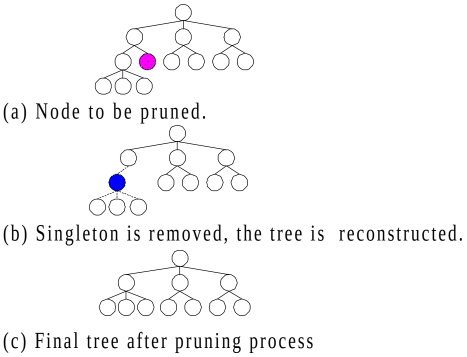
(a) Node to be pruned.
(b) Singleton is removed, the tree is reconstructed.
(c) Final tree after pruning process

Figure 2. Pruning process

To improve the tree two pruning algorithms, short and long term, are applied to delete the insufficiently useful nodes. The short-term pruning procedure deletes nodes

Department of Computer Science, University of Hertfordshire, Hatfield, Herts, AL10 9AB, U.K., Tel:+44 01707 284321 Fax:+44 01707 284361 Email:{W.Pensuwon,R.G.Adams,N.Davey}@herts.ac.uk

early in their life, if their existence does not improve the classificatory error. The long-term pruning procedure removes a leaf when its activity is not greater than a threshold. See Figure 2 for the pruning process.

B. Parameter Settings

The behaviour of CENT II is determined by a set of parameters, that specify for example the growth/pruning thresholds. In order to measure the effects of adding stochasticity to the basic model, a good instantiation of the parameters is needed. A *Genetic Algorithm (GA)* was used to search for such a set of parameters[9]. Figure 3 illustrates this process; fitness is assigned using two specific criteria which are described in Section 4.

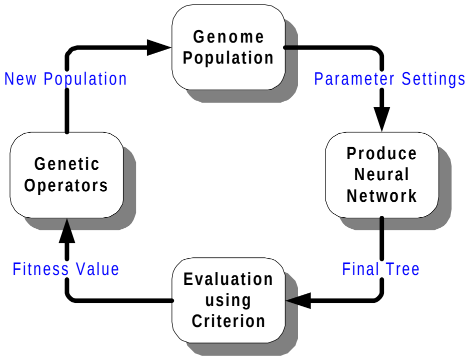

Figure 3. Process of finding parameter values using GA

In order to investigate the robustness of the model to its parameter settings, we found a second instantiation of the parameters deliberately designed to be non-optimal. This was achieved by modifying the fitness function of the GA so that it had a strong preference for small trees.

III. Stochasticity and CENT II

We anticipated that the addition of randomness to CENT II could have some benefit in helping the model avoid local minima in its implicit cost function. This approach is well known in the field of optimisation.

There are two different ways in which stochasticity can be added to the model. Firstly the deterministic decisions relating to growth and pruning can be made probabilistic, we call this Decision Based Stochasticity (DB). Secondly the attributes inherited by nodes when they are created can be calculated with a stochastic element, we call this Generative Stochasticity (G). To both of these approaches a simulated annealing process can be added to mediate the amount of non-determinism in a controlled way, so that a decreasing temperature allows for less randomness later in the life of the network.

A. Decision Based Stochasticity (DB)

There are two crucial decision making points in the model: selection for growth and selection for pruning. These decisions are made deterministically in the basic model, a relevant scalar value is calculated and compared to the appropriate threshold. This is generalised in the normal way to a stochastic decision, with the heaviside threshold function softened to a sigmoid, as shown in Figure 4.

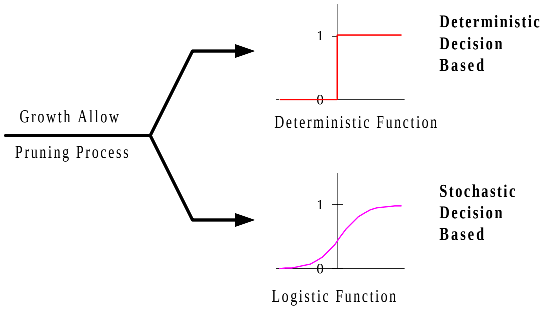

Figure 4. Decision Based Stochasticity

B. Generative Stochasticity (G)

The key property of a newly created node, calculated from its parent, is its tolerance size. Here, some randomness is added to this calculation. To achieve this, a Gaussian centred on the deterministic value gives the probability distribution of the new value. Figure 5 illustrates this process.

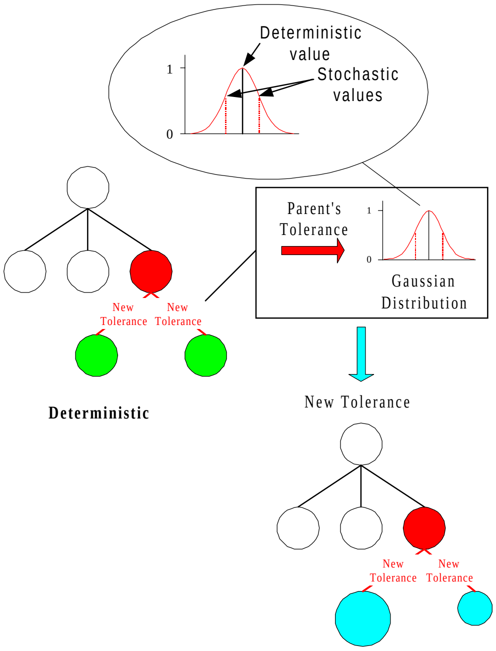

Figure 5. Generative Stochasticity

C. Integration of Stochastic Decision Based and Simulated Annealing (DB+SA)

The degree of randomness in the decision making process (DB) can be varied by changing the steepness of the sigmoid, parameterising it with a temperature, as in the standard Simulated Annealing approach. A high temperature corresponds to a large amount of randomness, and this is reduced over time. When the temperature is reduced to zero, the decision will become deterministic.

IV. Experiments

A. Data Sets

The data sets used in this investigation have been chosen to test many different aspects of the performance of the CENT II model. Three 2-dimensional and two higher dimensional artificial data sets are used, these have a variety of numbers of clusters (2 to 27) and hierarchical structure (1 to 4/5 levels). In addition, three real-world data sets are used: the well known IRIS data set which consists of 150 instances of 4 attributes describing sepal length, sepal width, petal length and petal width of the iris flower. The WINE data set contains 178 instances describing the results of a chemical analysis of wines grown in the same region in Italy in terms of 13 attributes. The ZOO data contains 59 instances of animals in terms of 18 attributes: a naming label, 15 boolean attributes, the number of legs and a type label.

B. Measurement of clustering performances

The general goal in many clustering applications is to arrive at clusters of objects that show small within-cluster variation relative to the between-cluster variation [6]. Clustering is difficult as many reasonable classifications may exist for a given data set, moreover it is easy for a clusterer to identify too few or too many clusters. Suitable cluster criterion measures are therefore needed [4].

There are two types of clustering measures, ones that grade the flat clustering performance of the leaf nodes and ones that grade the hierarchical structure.

An initial investigation concentrated on 10 non-hierarchical clustering methods from Milligan and Cooper [8] and another 2 hierarchical methods from Gordon [5]. As a result of this study, the best method of each type was chosen: the Gamma method [8], which measures the flat partitioning performance, and the Hierarchical Correlation method [5], that assesses the hierarchy structure in a network. The methods are as follows:

Gamma: $s(+)$ is the number of times when two points not clustered together are further apart than two points which are in the same cluster and $s(-)$ is the number of times when two point not clustered together are closer than two points which are in the cluster.

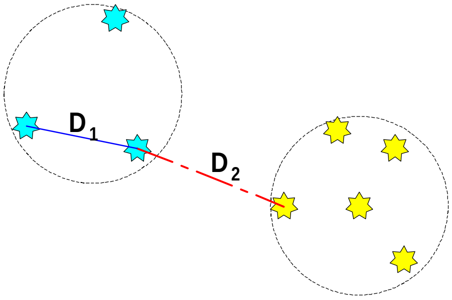

Figure 6. The distances within and between cluster used in calculating Gamma

In Figure 6, if D1 is less than D2 then $s(+)$ is incremented, otherwise $s(-)$ is incremented. Gamma is calculated from these two values using the following formula.

$$\Gamma = \frac{s(+)-s(-)}{s(+)+s(-)}$$

This gives a value between -1 and +1, where +1 is optimal. For comparison with the second measure we rescale this to the range 0 to +1.

Hierarchical Correlation (HC): measures the correlation between the dissimilarity matrix (d_{ij}) and hierarchical separation matrix (h_{ij}). A measure of the quality of the hierarchy structure in the tree is then given by:

$$HC = \frac{\sum(d_{ij}-\bar{d})(h_{ij}-\bar{h})}{\sqrt{[\sum(d_{ij}-\bar{d})^2 \sum(h_{ij}-\bar{h})^2]}}$$

d_{ij} is a dissimilarity between i and j objects. In this study, the dissimilarity is computed using the Euclidean distance.

h_{ij} is the relative height, which is the number of steps in the tree to the closest node that has i and j as descendants.

Figure 7 shows two examples of computing this relative height. The relative height of the node that classifies the 2nd input and the node that classifies the 5th input is 2. However, the relative height of the node that classifies the 1st input and the node that classifies the 6th input is 3. HC gives a value between 0 and +1, where +1 is optimal.

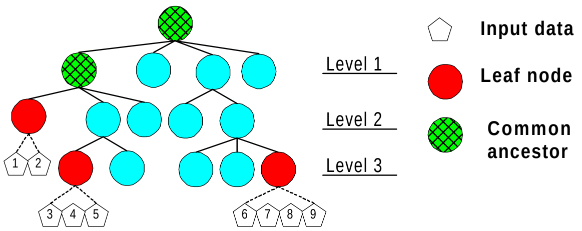

Figure 7. Examples of calculating the relative height

V. Results

In this section, we present the comparative results for the deterministic version of our model (CENT II) and a variety of non-deterministic versions over eight data sets using two different sets of parameters. The two different forms of stochasticity, decision based (DB) and generative (G) can be added separately or together.

In addition, Simulated Annealing was added to the decision based method to give five different versions of non-determinism. The results are divided into two sections, corresponding to the two parameter settings described earlier in Section 2. All test results are obtained by running the model for thirty epochs.

A. Optimal parameters

Table 1 represents the average and standard deviation of the two different clustering measures, over all eight data sets, tested for the deterministic version, and five different stochastic versions of the model, using the first set of parameter values. This set of parameters is designed to be optimal for deterministic CENT II. Unsurprisingly Table 1 clearly shows that deterministic version performed well, and that non-determinism was of limited value.

TABLE I

Clustering measures of 8 data sets using the optimal parameter settings; average values range from 0-1 where 1 represents the best results

	Gamma measure		Hierarchical correlation	
Modes	Average	Standard Deviation	Average	Standard Deviation
Deterministic	0.891	0.091	0.652	0.069
Stochastic-DB	0.865	0.091	0.693	0.035
Stochastic-DB+G	0.890	0.094	0.657	0.078
Stochastic-G	0.898	0.090	0.674	0.063
Stochastic-DB+SA	0.891	0.108	0.687	0.035
Stochastic-DB+SA+G	0.903	0.067	0.636	0.092

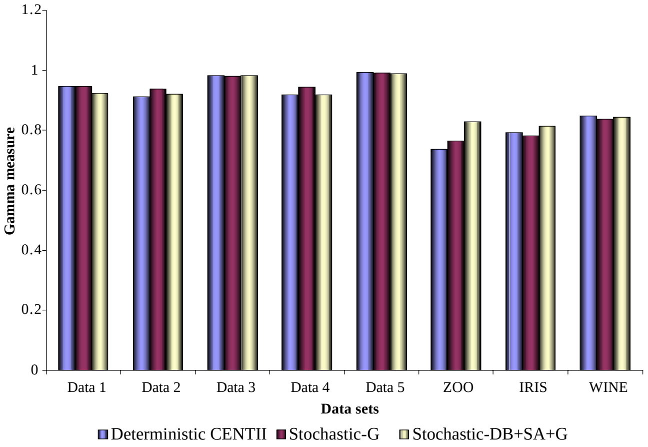

Figure 8. Gamma measure of 8 data sets using the optimal set of parameter values

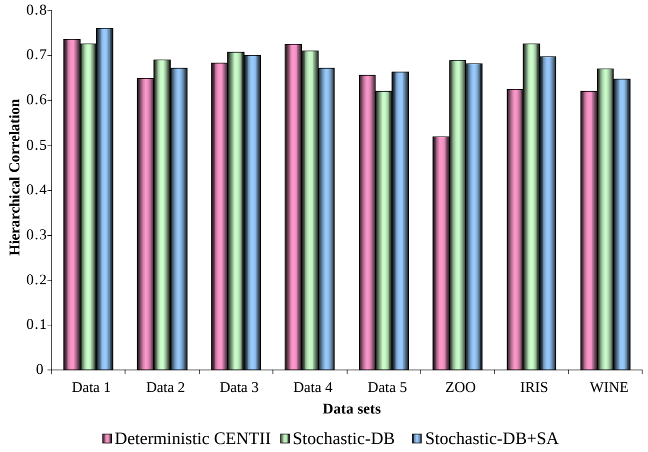

Figure 9. Hierarchical correlation of 8 data sets using the optimal set of parameter values

Figure 8 and Figure 9 give the Gamma measure and the Hierarchical Correlation respectively for each individual data set using CENT II and two of the best stochastic variants. Further work is being undertaken to determine the best combination of stochastic mechanisms.

The inter-version Hierarchical Correlation variability is greater than for the Gamma Measure, which means that there was considerable variation in tree structures produced. Interestingly Figure 9 shows the stochastic versions of the model significantly out perform the deterministic version for all the real world data sets. This can be illustrated for the IRIS data set, where the Stochastic-DB version produced the best tree which is shown in Figure 10.

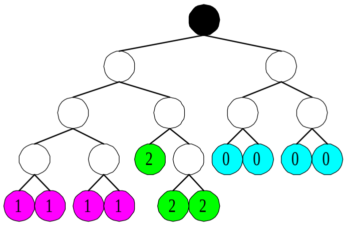

Figure 10. Tree structure produced by Stochastic-DB model with the IRIS data set

The true structure of the data consists of 3 different classes, each of 50 inputs, where each class represents a type of iris plant. One class is linearly separable from the other two, and this division is made at the top level of the tree. The other two classes are then immediately separated in the left subtree.

TABLE II

Clustering measures of 8 data sets using the non-optimal parameter values; average values range from 0-1 where 1 represents the best results

	Gamma measure		Hierarchical correlation	
Modes	Average	Standard Deviation	Average	Standard Deviation
Deterministic	0.772	0.142	0.671	0.161
Stochastic-DB	0.849	0.148	0.703	0.063
Stochastic-DB+G	0.884	0.130	0.690	0.058
Stochastic-G	0.719	0.166	0.690	0.135
Stochastic-DB+SA	0.881	0.138	0.708	0.063
Stochastic-DB+SA+G	0.867	0.128	0.677	0.086

B. *Non-optimal parameters*

Table 2 gives the results using the second parameter set. These parameters were obtained in the GA where the fitness function deliberately restricted the size of the tree for the CENT II model, which means that the trees produced using this parameter set may be non-optimal for data with many clusters. As expected the deterministic version did not perform as well here, and the high standard deviation of the Hierarchical Correlation measure shows that there is large variation in performance over the data sets. On the contrary, the stochastic model performed well in this situation. In particular, it improved the performance over the real world data. As an example of the improvement the non-deterministic version offers consider Figures 11 and 12. They illustrate the performance of both versions of the model, with non-optimal parameter settings, on a data set of 27 clusters.

CENT II produced an inappropriate tree, with only 11 nodes for the 27 clusters and the nodes were not well distributed. However the stochastic model produced enough nodes to represent the data. At least one leaf node appears in each cluster position and the four main cluster areas were separated by the second level of the tree. Admittedly there is a slight overproduction of nodes but this is preferable to not finding all the clusters.

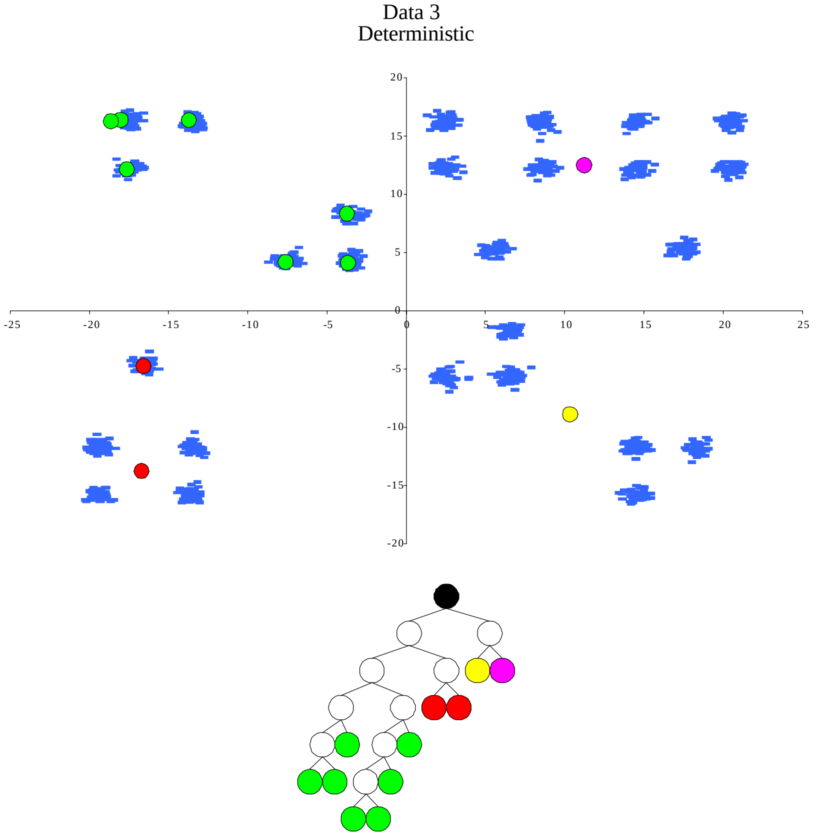

Figure 11. Positions of leaf nodes and a tree structure produced by CENT II with data 3 which contains 27 clusters

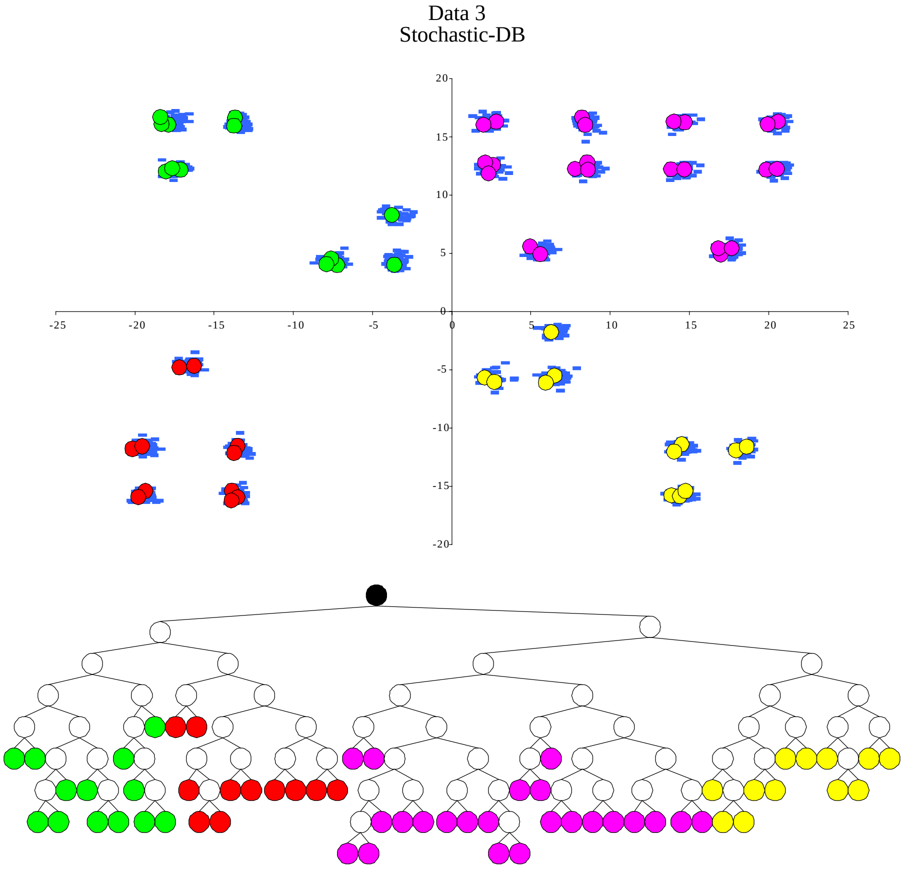

Figure 12. Positions of leaf nodes and a tree structure produced by Stochastic-DB with data 3 which contains 27 clusters

VI. Discussion and Conclusion

This paper presents a method for adding stochasticity to a dynamic hierarchical neural clusterer. We identified five different combinations of non-deterministic behaviour that can be added to the basic model. In order to investigate the benefits or costs of these additions we created two parameter sets for the deterministic model using a GA.

One set is optimal for deterministic CENT II and a particular collection of artificial data, and the other is deliberately non-optimal. As expected the deterministic version performed well on the artificial training sets for which its parameters were optimised, and adding stochasticity had little effect on performance.

However for unseen data sets with unpredictable structure the parameters would obviously not be optimal. The variability inherent in the stochastic models allows the tree growth to adapt to this new data producing reliably good tree structures to represent the data. This can be seen most clearly in the performance of deterministic CENT II with non-optimal parameter settings, compared to the stochastic version with the same parameters, where the non-determinism still allowed high quality trees to be produced.

The stochastic model has produced a consistently good performance over all of the data set presented, has maintained the quality of performance shown by the CENT II and has improved reliability. Further work is being carried out to determine the best combination of mechanisms for adding stochasticity to the basic model.

Acknowledgement 1: Part of this research is funded by a Royal Thai Government Scholarship.

References

[1] Adams, R.G., Butchart, K. and Davey, N. (1999) Hierarchical Classification with a Competitive Evolutionary Neural Tree. Neural Networks, Vol. 12, pp 541-551.

[2] Butchart, K. (1996) Hierarchical Clustering Using Dynamic Self Organising Neural Network. Ph.D. Thesis. University of Hertfordshire.

[3] Davey, N., Adams, R.G. and George, S.G. (1999) The Architecture and Performance of a Stochastic Competitive Evolutionary Neural Tree Network, Applied Intelligence, Vol. 12, No. 1/2, pp.75-93.

[4] Everitt, B.S. (1993) Cluster Analysis, Edward Arnold, London.

[5] Gordon, A.D. (1999) Classification, Chapman & Hall, London.

[6] Hartigan, J.A. (1975) Clustering Algorithms, John Wiley & Sons, USA.

[7] Hertz, J., Krogh, A. and Palmer, R.G. (1991) An Introduction to the Theory of Neural Computation, Addision Wesley. USA.

[8] Milligan, G.W. and Cooper, M.C. (1985) An Examination of Procedures for Determing the Number of Clusters in a Data Set. Psychometrika, Vol. 50, No. 2, pp 159-179.

[9] Pensuwon, W., Adams, R.G. and Davey, N. (2000) Optimising a Neural Tree Classifier Using a Genetic Algorithm, Submitted to *KES'2000 Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies.*