The Analysis of the Addition of Stochasticity to a Neural Tree Classifier

W. Pensuwon[1,2], R. G. Adams[1] and N. Davey[1]

[1]Department of Computer Sciences
University of Hertfordshire
Hatfield, Herts, AL10 9AB, United Kingdom.

[2]Department of Electrical&Electronics Engineering
Ubonratchathani University
Ubonratchathani, 34190, Thailand.

W.Pensuwon@herts.ac.uk, R.G.Adams@herts.ac.uk and N.Davey@herts.ac.uk

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.

Finally, deliberately changing the tree shape by varying key parameters was investigated, illustrated and systematically analysed.

Keywords: clustering measure, competitive learning, genetic algorithms, neural tree networks, stochasticity.

1. Introduction

The standard Neural Network Competitive Learning algorithm [6] 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]. 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.

2. 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.

2.1. Top-Level 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.

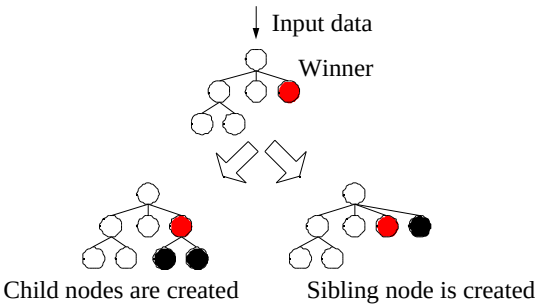
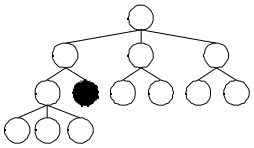
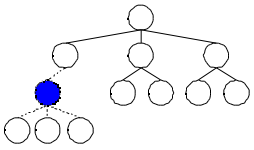


Figure 1. Process of growing a tree. Child node creation
is shown on the left whereas sibling node creation is shown on the right.

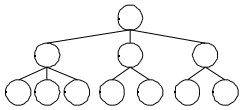
To improve the tree two pruning algorithms, short and long term, are applied to delete the insufficiently useful nodes. The short-term pruning procedure deletes nodes early in their life, if their existence does not improve the classificatory error. The long-term pruning procedure removes a leaf when its *activity* (the rate of classifying input) is not greater than a *threshold*. See Figure 2 for the pruning process.



(a) Node to be pruned.



(b) Singleton is removed, the tree is reconstructed.



(c) Final tree after pruning process.

Figure 2. Pruning process of an inactive node from the tree. The final tree is restructured so that a singleton is removed.

2.2. Parameter Setting

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 [8].

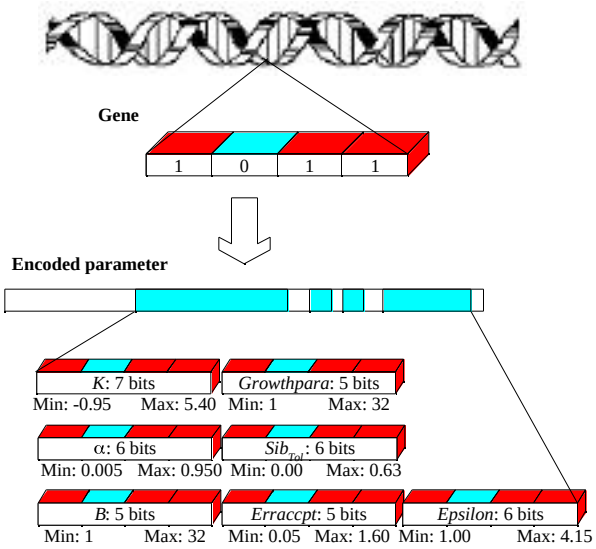


Figure 3. A Genome picture of 7 encoded parameters. Each parameter was encoded into a binary chromosome.

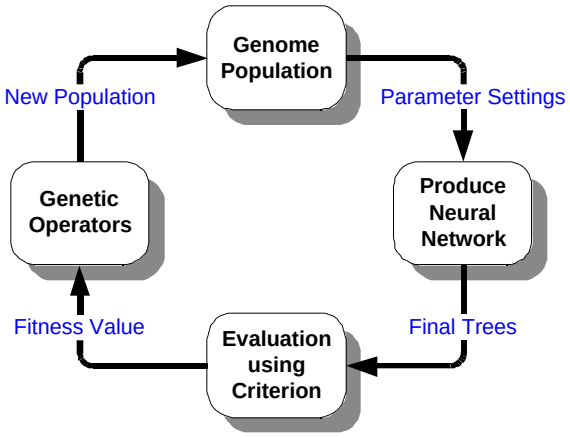


Figure 4. Process of finding parameter values using Genetic Algorithm.

In the *GA* each parameter is coded into a binary

chromosome where each parameter is given a power of 2 different values so that each binary number represent a valid parameter value. A picture of the 7 encoded parameters is shown in Figure 3. Figure 4 illustrates this process; fitness is assigned using two specific criteria which are described in Section 4.

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.

3. 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.

Secondly the attributes inherited by nodes when they are created can be calculated with a stochastic element, we call this Generative Stochasticity. 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.

3.1. Decision Based Stochasticity

There are three crucial decision making points in the model: the selection for growth, the type of 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.* Decision Based Stochasticity is generalised in the normal way to a stochastic decision, where the sharp change of decision, depending on some input, is made softer by the addition of some randomness.

Figure 5 illustrates the heaviside *threshold* function softened to a sigmoid. In the deterministic version (on the left) the decision is made at a precise value of the decision variable plotted on the horizontal axis. However, in the stochastic version (on the right) the value obtained by the sigmoid function is compared to a random number between 0 and 1, and if larger, the decision is accepted. In this way values of the decision variable less than the original *threshold* can lead to positive decisions and values greater than the precise one can lead to negative decisions.

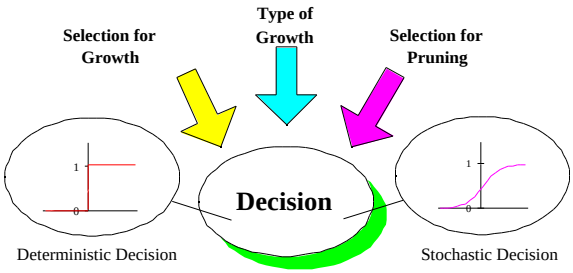


Figure 5. Decision Based Stochasticity. The probability
of accepting a decision produced in the left ellipse is
crisp whereas the probability of accepting a decision in
the right ellipse is fuzzy.

Decision Based Stochasticity is essentially the same technique as the addition of stochasticity to neural networks such as the Hopfield Network forming the Boltzmann machine [6].

The reason for adding stochasticity is that it may be useful for the network to create more tentative new growth and therefore for the pruning process to be more common. The stochasticity softens the strong decision making and allows the possibility of more chances at growth and of keeping that growth, in the hope that more correct decisions will be made for the different data sets.

3.2. Generative Stochasticity

This type of stochasticity adds noise to a generated value in the model. The major occurrence of a generated value in CENT II is during sidegrowth (Sibling creation) and downgrowth (Child creation). In both cases a new *tolerance* is required.

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. Since the network is sensitive to the value of *tolerance* a stochastic element added here could be beneficial. This technique is similar to the Soft Competition Scheme [10].

Figure 6 illustrates for the child creation process how the implementation was carried out in this new stochastic approach. The size of *tolerance*, which is computed in the deterministic model entirely in terms of certain parameters values, here has a stochastic element added to it.

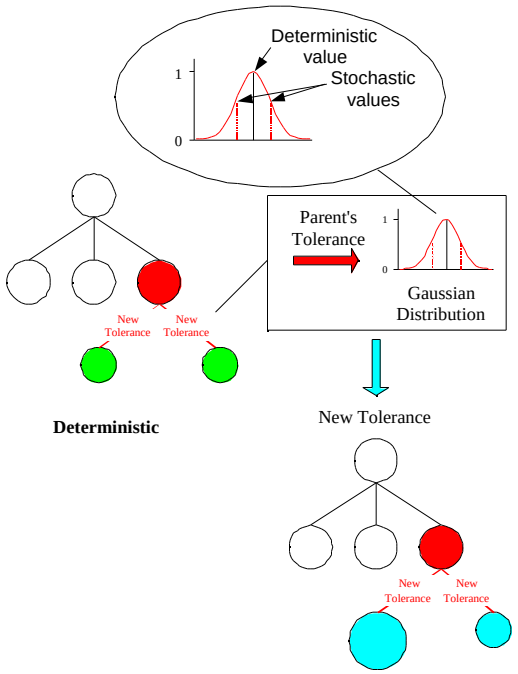


Figure 6. Generative Stochasticity process. Green circles represent equal child *tolerance* size produced in a deterministic process. Blue circles are of different sizes as a result of adding stochasticity into the computation of the *tolerance* of children.

3.3. Control of Stochasticity

The degree of randomness in the stochasticity can be controlled into 2 different ways. The first method has a *fixed temperature* (degree of randomness) during the whole run whereas for the second method the *temperature* is reduced every epoch by a *temperature* decrease factor. The second method is known as *Simulated Annealing*, 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.

4. Experiments

4.1. 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, nine real-world data sets are used: contraceptive prevalence survey, male Egyptian skulls, glass identification database, synthetic control chart time series, teaching assistant (TA), vowel recognition, IRIS, wine and zoo. Table 1 represents details of each real world data set, and full descriptions of these data sets are described in [9].

Table 1 Nine real-world data sets details

Data sets	Classes	Attributes	Total vectors
Contraceptive	3	9	1,453
Egypt Skull	5	4	150
Forensic Glass	6	9	214
Time Series	6	60	600
TA	3	5	151
Vowel	10	7	1,520
IRIS	3	4	150
Wine	3	13	178
ZOO	7	16	118

4.2. 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 [5]. 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 [3].

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 [7] and another 2 hierarchical methods from Gordon [4]. As a result of this study, the best method of each type was chosen: the Gamma measure method [7], which measures the flat partitioning performance, and the Hierarchical Correlation method [4], that assesses the hierarchy structure in a network. The methods are as follows:

Gamma measure: 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.

In Figure 7, if D1 is less than D2 then s(+) is

incremented, otherwise s(-) is incremented. *Gamma measure* 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.

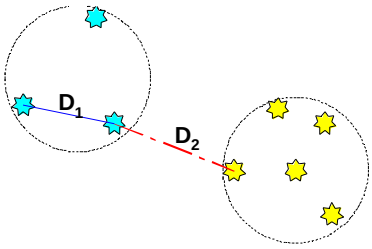


Figure 7. The distances within and between cluster used
in calculating *gamma measure*.

Hierarchical Correlation: 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.

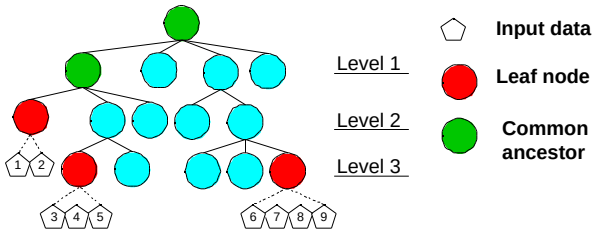


Figure 8. Examples of calculating the relative height.

Figure 8 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. *Hierarchical correlation* gives a value between 0 and +1, where +1 is optimal.

5. Results

In this section, we present the comparative results for the deterministic version of our model (CENT II) and two non-deterministic versions over fourteen data sets using two different sets of parameters.

The two different forms of stochasticity: decision based stochasticity (selection for growth, the type of growth and selection for pruning.) and generative stochasticity (sidegrowth and downgrowth) can be added separately or *tolerance* value calculation in together. There are therefore 32 different stochastic modes in both a *fixed temperature* regime and in a *simulated annealing* regime.

Initial investigations [9] were carried out on all combinations of the stochastic procedures and the networks tested with existing data sets. It showed that the effect of any individual stochastic mode was the same regardless of whether it was combined with the other modes. It was therefore decided to use just two variations. The first is a mixture of all modes for a *fixed temperature*. The second is a mixture of all modes with *SA*. In this paper, these two mixed modes are considered together with the original deterministic CENT.

The results are divided into three sections, the first two sections are comparative results between the deterministic and the non-deterministic models corresponding to the two parameter settings described earlier in Section 2. The results are presented using the 2 modes of stochasticity defined above, compared with the deterministic version. All test results are obtained by running the model for thirty epochs.

The third section presents the method of changing tree shape by tuning the parameters so that different interpretation of data can be achieved.

5.1. Optimal Parameters

Table 2 represents the average and standard deviation of the two different clustering measures, over all fourteen data sets, tested for the deterministic version, and two 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 with respect to a set of artificial data sets.

Figure 9 and Figure 10 give the *gamma measure* and the *hierarchical correlation* respectively for each

individual data set using CENT II and the two stochastic variants.

Unsurprisingly Table 2 clearly shows that the overall performances of the network over real world data sets are worse than artificial data sets for all three models. Besides, the real world data sets produce more variation in clustering performance and are particularly worse for the *gamma measure*. Additionally, when comparing amongst the models, the deterministic version performed well and non-determinism appeared to be of limited value. A good performance by the deterministic model is illustrated for the IRIS data set in Figure 11.

The true structure of the IRIS data consists of 3 different classes shown in Figure 11(a), each of 50 inputs, where each class represents a type of IRIS plant. Figure 11(b) depicts that 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 right subtree.

Table 2 Clustering measures of 14 data sets using the optimal parameter setting; average values range from 0-1 where 1 represents the best results

Artificial	Gamma measure		Hierarchical correlation	
Modes	Average	Standard Deviation	Average	Standard Deviation
Deterministic	**0.940**	**0.047**	**0.714**	**0.057**
Stochastic	0.945	0.042	0.711	0.043
Stochastic (SA)	0.947	0.037	0.684	0.061
Real world data	Gamma measure		Hierarchical correlation	
Modes	Average	Standard Deviation	Average	Standard Deviation
Deterministic	**0.746**	**0.164**	**0.562**	**0.098**
Stochastic	0.709	0.200	0.532	0.072
Stochastic (SA)	0.724	0.201	0.534	0.130

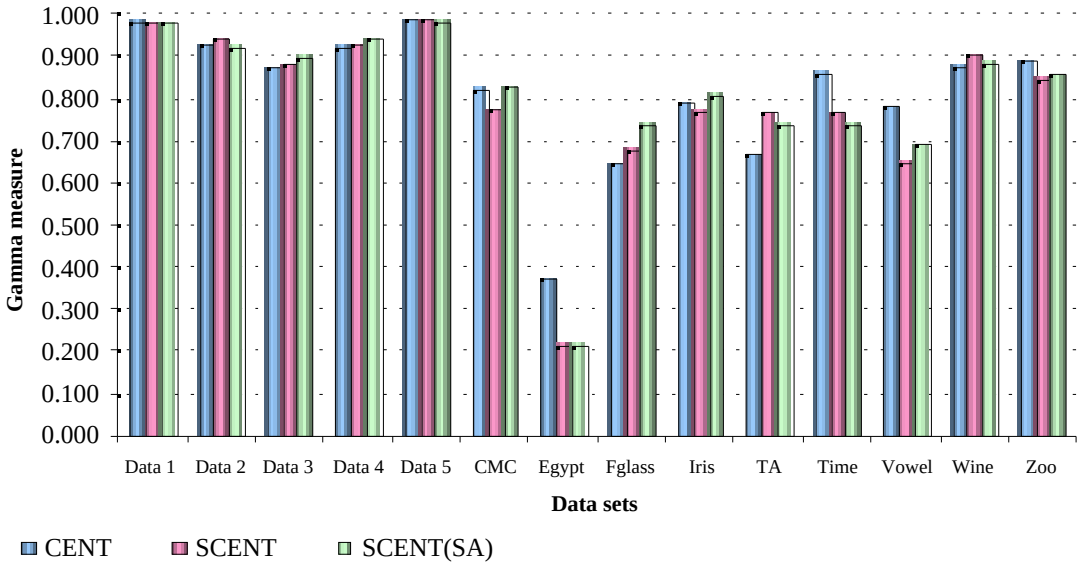


Figure 9. Gamma measure of 14 data sets using the optimal set of parameter values

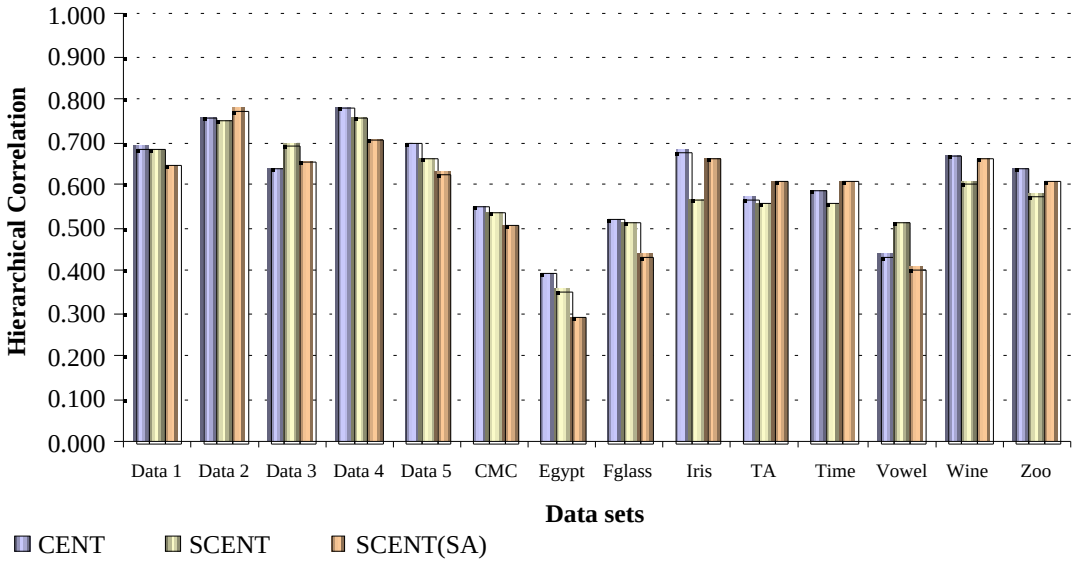
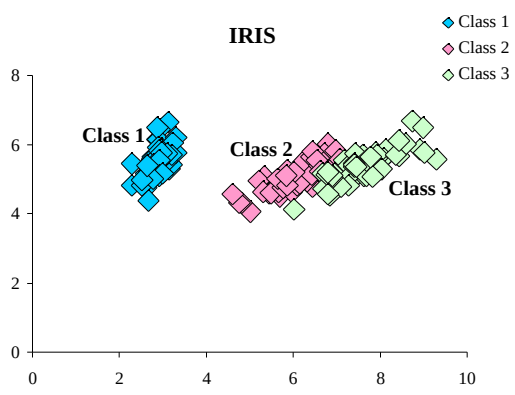
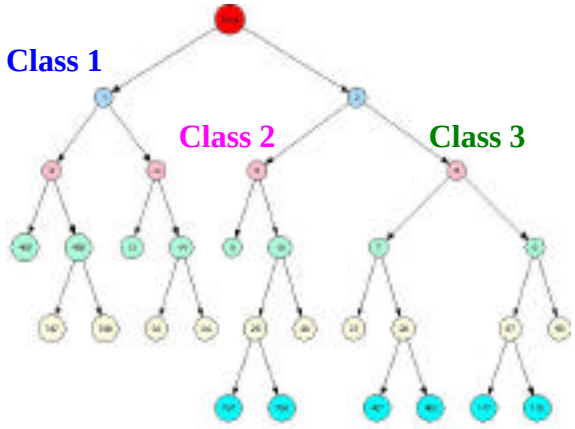


Figure 10. Hierarchical correlation of 14 data sets using the optimal set of parameter values



(a) Actual classes in IRIS data set



(b) Tree structure presenting 3 classes in IRIS

Figure 11. Tree structure produced by deterministic model with the IRIS data set.

5.2. Non-Optimal Parameters

Table 3 gives the results using the second parameter set. These parameters were obtained in the Genetic Algorithm 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, as can be seen by comparing Tables 2 and 3.

The performance of the three networks using this parameter setting produce a smaller *gamma measure* and large variation in performance especially over real world data sets. However, the *hierarchical correlation* is more resilient

Interestingly, the stochastic model performed well in this situation especially the mode with the addition of *simulated annealing*. In particular, it improved the performance over the real world data.

Results in Table 2 and 3 shows that the macro analysis of the performance using these two clustering measures did not show much different in performance among the three networks. Therefore, we used a micro analysis, that is the consideration of details of the actual tree structure produced by the network. From this micro analysis it is apparent that stochasticiy did bring some benefit.

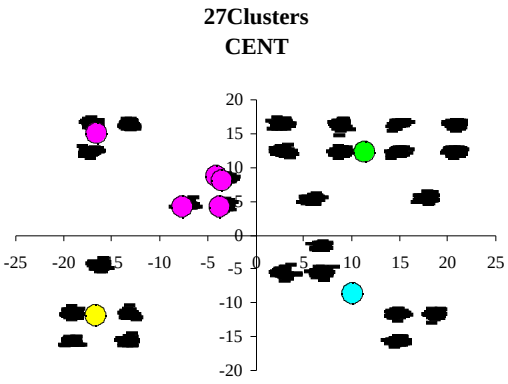
As an example of the improvement the non-deterministic version offers consider Figures 11 and 12. They illustrate the performance of both versions of the model, with non-optimal parameter settings, on a data set of 27 clusters.

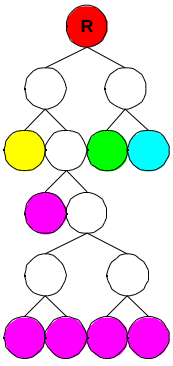
CENT II produced an inappropriate tree, with only 8 leaf nodes for the 27 clusters and the nodes were not well distributed. However the stochastic model produced enough nodes to represent the data. At least one leaf node appears in each cluster position and the four main cluster areas were separated 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.

Most data sets show similar results with the stochastic version performing better than the deterministic version.

Table 3 Clustering measures of 14 data sets using the non-optimal parameter setting; average values range from 0-1 where 1 represents the best results

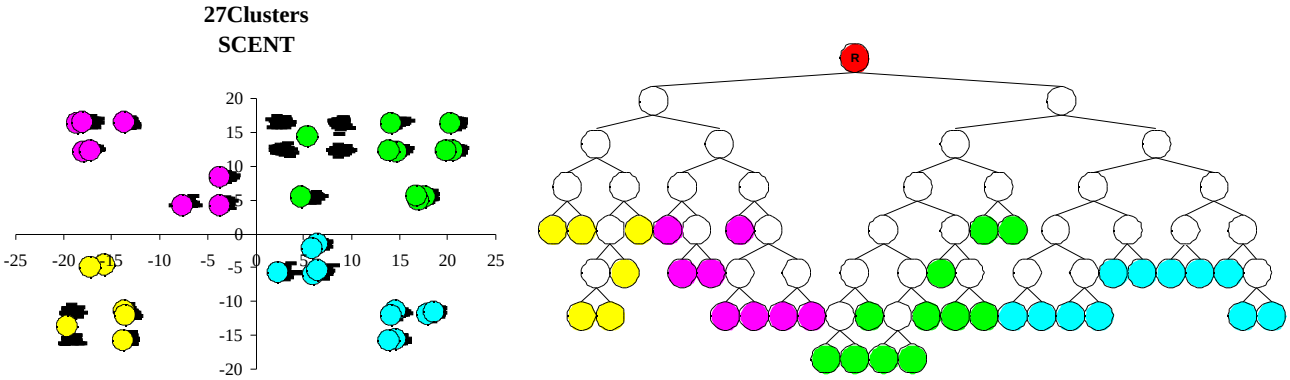
Artificial	Gamma measure		Hierarchical correlation	
Modes	Average	Standard Deviation	Average	Standard Deviation
Deterministic	**0.755**	**0.166**	**0.630**	**0.259**
Stochastic	0.927	0.075	0.702	0.093
Stochastic (SA)	0.910	0.072	0.663	0.058
Real world data	Gamma measure		Hierarchical correlation	
Modes	Average	Standard Deviation	Average	Standard Deviation
Deterministic	**0.395**	**0.293**	**0.642**	**0.130**
Stochastic	0.589	0.257	0.570	0.140
Stochastic (SA)	0.490	0.239	0.603	0.123





(Hierarchical Tree Structure)

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



(Hierarchical Tree Structure)

Figure 12. Position of leaf nodes and a tree structure produced by Stochastic (fixed amount of randomness) with data 3
which contains 27 clusters.

5.3. Tuning Tree Shape

A desirable feature of a hierarchical clusterer would be one in which the shape of the induced tree could be controlled and selected by the user in order to provide the best interpretation of the structure of the data.

The experiment was conducted by varying some key parameters using 5 selected real world data sets and CENT II. Results in this section are assessed by two values: branching factor and mean depth of tree. The branching factor is the ratio of the total number of children nodes to the nodes that have children. The mean depth of tree is the mean of the depth all leaf nodes in the tree. Figure 13 illustrates the average value of branching factor and mean depth over 5 data sets.

Figures 14 and 15 illustrate the different shape of trees that can be produced by changing parameters. The predictability of the model to certain changes in the parameter settings is of benefit to users who can choose to interpret the tree shape when interpreting data.

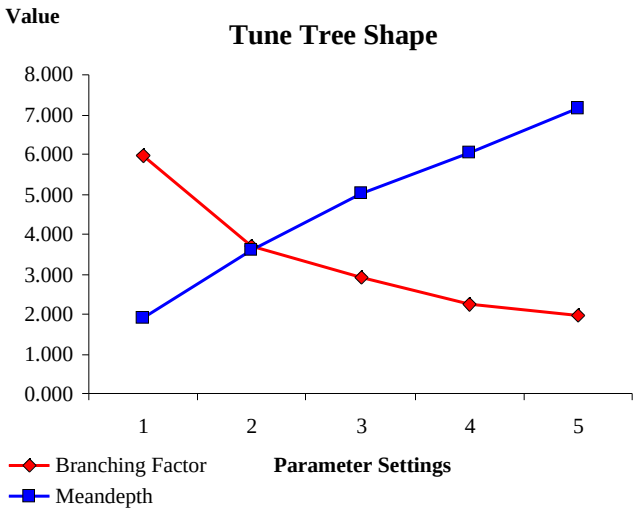
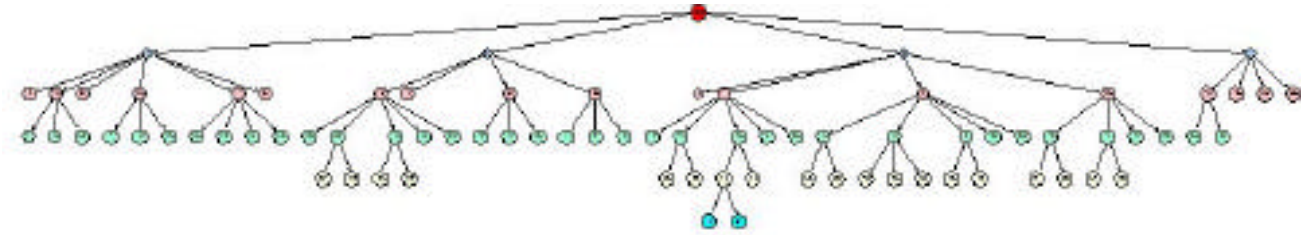
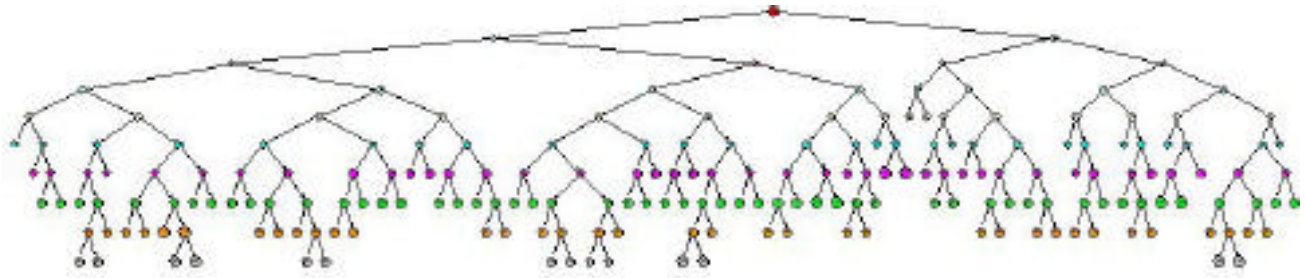


Figure 13. Average value of branching factor and mean depth produced by 5 different sets of parameters over 5 data sets.



Branching factor=3.500
Mean depth=3.212

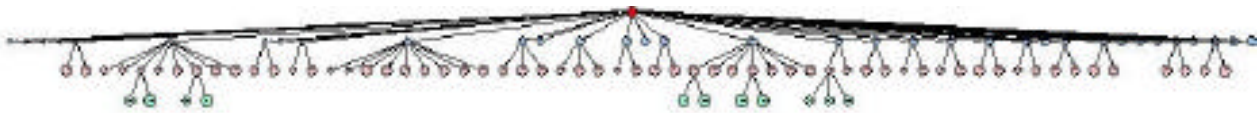
(a) Tree produced by parameter setting 1 (see Figure 13)



Branching factor=1.991
Mean depth=7.539

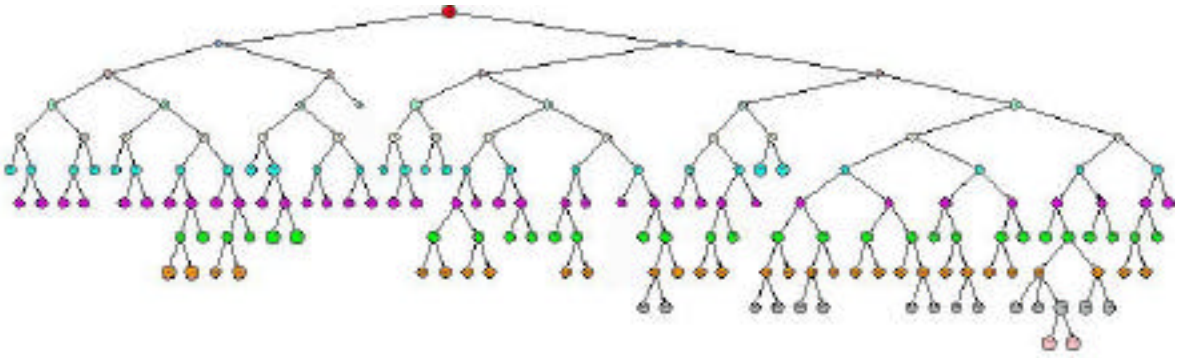
(b) Tree produced by parameter setting 5 (see Figure 13)

Figure 14. Different tree shapes produced by varying parameter settings on Wine data set.



Branching factor=5.567
Mean depth=2.756

(a) Tree produced by parameter setting 1 (see Figure 13)



Branching factor=1.988
Mean depth=8.384

(b) Tree produced by parameter setting 5 (see Figure 13)

Figure 15. Different tree shapes produced by varying parameter settings on the picture data set analysed in [2].

6. Discussion and Conclusion

This paper presents a method for adding stochasticity to a dynamic hierarchical neural clusterer. We identified two different combinations of non-deterministic behaviour that can be added to the basic model in terms of the control of stochasticity. 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.

Furthermore, the integration of *simulated annealing* with stochasticity produced better results than using a fixed amount of randomness.

By varying the parameter settings, different tree shapes can be selected. So that the most appropriate data interpretation could be achieved.

Acknowledgment

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

References

[1] R. G. Adams. K. Butchart. And N. Davey, "Classification with a Competitive Evolutionary Neural Tree," *Neural Networks*, Vol. 12, pp. 541-551, 1999.

[2] N. Davey, R.G. Adams. and S. G. George, "The Architecture and Performance of a Stochastic Competitive Evolutionary Neural Tree," *Applied Intelligence*, Vol. 12, No. 1/2, pp. 75-93, 2000.

[3] B. S. Everitt, *Cluster Analysis*, Edward Arnold, London, 1993.

[4] A. D. Gordon, *Classification*, Chapman & Hall, London, 1999.

[5] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, USA, 1975.

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

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

[8] W. Pensuwon, R. G. Adams. and N. Davey, "Optimising a Neural Tree Classifier Using a Genetic Algorithm," *Proceeding of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies (KES'2000)*, Vol. 2, pp 848-851, 2000.

[9] W. Pensuwon, *Stochastic Hierarchical Dynamic Neural Networks*, Ph.D. Thesis. University of Hertfordshire, 2001.

[10] E. Yair, K. Zeger, and A. Gersho, "Competitive Learning and Soft Competition for Vector Quantiser Design," *IEEE Transactions on Signal Processing*, Vol.40, pp.294-308, 1992.