

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/251385790>

# Statistical and AI Techniques in GIS Classification: A Comparison

## Article

---

CITATIONS

5

READS

326

3 authors, including:



Mark Gahegan

University of Auckland

188 PUBLICATIONS 5,538 CITATIONS

[SEE PROFILE](#)

# Statistical and AI Techniques in GIS Classification: A Comparison.

*G.W.H. German<sup>1</sup>, G. West<sup>1</sup> and M. Gahegan<sup>2</sup>*

<sup>1</sup>School of Computing,  
Curtin University of Technology,  
Bentley,  
Western Australia 6102.

<sup>2</sup>Dept Geography,  
Penn State,  
University Park,  
PA 16802 USA.

Email: [gordon@computing.edu.au](mailto:gordon@computing.edu.au)  
[geoff@computing.edu.au](mailto:geoff@computing.edu.au)

## Abstract

Data classification is one of the primary tasks in Geocomputation. It is regularly used by the Geoscience professional to categorise datasets for further analysis such as land management, potential mapping, forecast analysis and soil assessment, to name a few. Traditionally, data classification tasks are based on statistical methodologies such as minimum distance-to-mean (MDM), maximum likelihood classification (MLC) and linear discrimination analysis (LDA). These classifiers have developed over the last century from the mathematical disciplines of set theory and control theory. Over the last 20 years, classification tools have also developed from the emerging fields of connectionism and inference nets within the discipline of Artificial Intelligence (AI); the more notable being the neural network based multi-layered perceptron (MLP), the decision tree and genetic algorithms (GA) such as differential evolution (DE). This paper seeks to compare the advantages and disadvantages of the various classifier types. The results show that, for simple tasks, MDM, LDA and similar classifiers are the best compromise of efficiency and classification ability, whilst for more complex datasets, variants based on the MLP and decision trees are the classifiers of choice.

## 1.0 Introduction

Within the Earth Sciences, classification is the process of identifying areas of the Earth's surface, given a particular phenomenological output domain and some different input domain (a set of attributes). Classification of raw datasets is an important step in the analysis and understanding of geographical features and their relationships. Such data can be remotely sensed, gathered from ground surveys, or even culled from some previous classification.

A classifier's function can be formulated in terms of a mapping of its input variables to its (given) output conditions. We can write:

$$\mathfrak{R}^p \xrightarrow{\Gamma(n)} \Pi^q, \quad (1)$$

where  $p$  is the number of attributes,  $q$  is the number of classes and  $n$  is the number of samples. The goal of classification is to select an output class from a different phenomenological domain ( $\Pi$ , the classification scheme) to that of the input attributes ( $\mathfrak{R}$ ) for each input vector  $x^p$ . These transformation models can be categorised as unsupervised or supervised classifiers. In the case of supervised classification, the user chooses the scheme  $\Pi$  and the classifier learns an approximation  $\Gamma'(n, p)$  to the ideal transfer function  $\Gamma(n, p)$ . This is accomplished by examining a small set (the training set) of the data for which the correct classification has already been determined (by ground survey, or a previous classification). Hence the (common) scheme of ground cover type is often derived from an attribute domain that may comprise several bands of LANDSAT data, as well as ancillary data such as digital elevation models, rainfall, etc.

Traditionally, classification of geographic datasets has been based on well-known statistical methods, as implemented in classifiers such as Maximum Likelihood Classification (MLC) or Minimum Distance to Mean (MDM, Richards, 1986). More recently, the discipline of Computer Science has developed classifiers based on machine learning techniques such as decision trees and artificial neural networks. This has lead to a greater choice of classification techniques available to the Earth Scientist. In this paper, we will examine the relative strengths and weaknesses of these various supervised classifiers, giving some comparative results.

## 2.0 Bayesian Classifiers: MLC and MDM

Many statistical classifiers are based on some approximation to the ideal Bayesian classifier, as in most practical applications the optimal Bayesian classifier can never actually be realised. The popular MLC and MDM classifiers are examples of such approximations derived from the following theory of Bayesian probability.

For acceptable classification, a classifier must contain sufficient complexity to enable encoding of the approximated transformation function  $\Gamma'(n, p)$  of Eqn 1. Bayesian estimation is a process of determining the probable outcome of an event (the *a posteriori* probability) given some new piece of evidence and the original (*a priori*) probability of that outcome. The Bayes Theorem can be restated in terms of classification of data as (Dunteman, 1984):

$$\rho(i | x) = \frac{\rho(x | i)\rho(i)}{\rho(x)} \quad i = 1, \dots, q, \quad (3.2)$$

where

$q$  = the number of classes,

$\rho(i | x)$  = the probability of class  $i$  given the input vector  $x$ ,

$\rho(x | i)$  = the probability of an input vector with characteristics of  $x$  given class  $i$ ,

$\rho(i)$  = the probability that class  $i$  is present in the dataset,

$\rho(x)$  = the probability of an input vector with characteristics of  $x$  given *any* class.

Intuitively, to assign a class membership for a given  $x$ , we would calculate  $\rho(i | x)$  for all classes and assign  $x$  to that class  $i$  for which  $\rho(i | x)$  is a maximum. However for real-world data, it is generally the case that the prerequisite  $\rho(x | i)$  is not known and is therefore estimated from the training set as a probability density function (*pdf*). The specific form of the *pdf* used for this estimation of  $\rho(x | i)$  defines the type of approximation model. The *pdf* is used as a *discriminant rule* to identify a given vector  $x$  as belonging to a particular class  $i$ .

If we make the assumption that the cost of misclassifying class  $i$  as class  $j$  is the same for all  $i$  and  $j$  ( $i \neq j$ ), we can rewrite Eqn 2 as:

$$x \in i \quad \text{if} \quad \rho(x | i)\rho(i) > \rho(x | j)\rho(j), \quad \text{for all } j \neq i. \quad (3)$$

This can be alternatively expressed, by taking the logarithm of both sides of the inequality, as:

$$x \in i \quad \text{if} \quad \ln \rho(x | i) + \ln \rho(i) > \ln \rho(x | j) + \ln \rho(j), \quad \text{for all } j \neq i \quad (4)$$

and for the normal case where the priors are unknown, or assumed equal, reduces to:

$$x \in i \quad \text{if} \quad \ln \rho(x | i) > \ln \rho(x | j), \quad \text{for all } j \neq i \quad (5)$$

The Maximum Likelihood discriminant rule uses Eqn 5. The term "maximum likelihood" may be seen as more appropriate if we rewrite Eqn 5 as:

$$x \in i \quad \text{if} \quad \ln \rho(x | i) = \max_j (\ln \rho(x | j)). \quad (6)$$

Consider the case where  $\rho(x | i)$  is estimated as a multivariate normal (Gaussian) distribution:

$$\rho(x | i) \cong 2\pi^{-p/2} |\Sigma_i|^{-1/2} e^{\left\{ -\frac{1}{2}(x - u_i)^T \Sigma_i^{-1} (x - u_i) \right\}}, \quad (7)$$

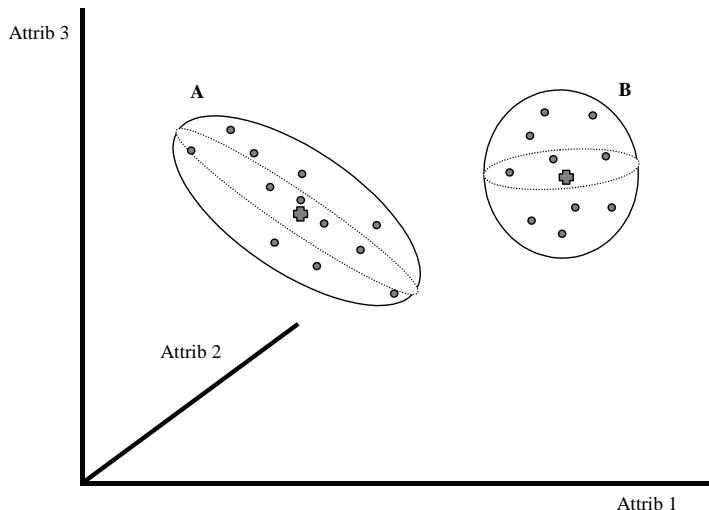
where  $p$  is the dimensionality of the input vector  $x$ , with  $\Sigma_i$  and  $u_i$  the sample covariance matrix and sample mean vector, respectively, for class  $i$ . This assumption of normality underlies the Maximum Likelihood classifier (MLC, more correctly the *Gaussian-based* Maximum Likelihood Classifier, see Mardia *et. al.*, 1979), the most common statistical classifier for GIS/RS datasets (Richards, 1986). Substituting Eqn 7 into Eqn 6 and tidying terms gives the MLC discrimination rule:

$$d_i(x) = -\ln|\Sigma_i| - (x - u_i)^T \Sigma_i^{-1} (x - u_i), \quad (8)$$

From Eqn 8 it is easier to see how the methodology implicitly sets a lower limit to the sample class size for each  $i$ . To ensure that the inverse of  $\Sigma_i$  remains non-singular, the number of representative patterns in each class  $i$  (denoted  $\text{size}(i)$ ) must exceed  $p$ . In practice, it is recommended to maintain  $\text{size}(i) > 10p$ , for all  $i$  (Swain & Davis, 1978), so as to provide a minimal set of construction points in each dimension for the Gaussian curves. The Minimum-Distance-to-Mean (MDM) classifier simplifies the discrimination rule of Eqn 8 by dropping the covariance term  $\Sigma_i$  and implementing a simpler Euclidean distance-to-mean metric to give a discriminant function:

$$d_i(x) = -(x - u_i)^T (x - u_i). \quad (9)$$

This produces spheroid decision boundaries in (Euclidean) feature space, rather than the ellipsoid boundaries of the MLC (see Figure 1). This is obviously less flexible; for example, classes with a long, spread-out scatter-pattern in feature space will be poorly modelled. However, because there is no need to calculate the covariance matrix, it requires fewer data points in each sample class to construct the decision surfaces, the restriction being relaxed to  $\text{size}(i) > p$ .



*Figure 1: MLC and MDM representations. Distributions A and B can be modelled by the MLC as spherical or ellipsoidal. The MDM can only model distributions as spherical.*

### 3.0 Linear Discriminant Analysis

The preceding Bayesian approximation functions are quadratic in nature, as can be seen by examining Eqns 8 and 9. They model a *volume* – that of the particular class distribution. Linear discrimination, as the name suggests, looks for linear combinations of the input variables that can provide an adequate separation for the given classes. Rather than look for a particular parametric form of distribution, LDA uses an empirical approach to define linear decision planes in the attribute space i.e. it models a *surface*. The discriminant functions used by LDA are built up as a linear combination of the variables that seek to somehow maximise the differences between the classes:

$$z = a_1 x_1 + a_2 x_2 + \dots + a_p x_p = \mathbf{a}' \mathbf{x}. \quad (10)$$

The problem then reduces to finding a suitable vector  $a$ . There are several popular variations of this idea, one of the most successful being the Fisher pairwise Linear Discriminant Rule<sup>1</sup>.

Fisher's Rule is considered a 'sensible' classification, in the sense that it is intuitively appealing. It makes use of the fact that distributions that have a greater variance *between* their classes than *within* each class, should be easier to separate. Therefore, it searches for a linear function in the attribute space that maximises the ratio of the between-group sum-of-squares ( $B$ ) to the within-group sum-of-squares ( $W$ )<sup>2</sup>. This can be achieved by maximising the ratio

$$\frac{a'Ba}{a'Wa}, \quad (11)$$

and it turns out that the vector that maximises this ratio,  $a$ , is the eigenvector corresponding to the largest eigenvalue of  $W^{-1}B$  i.e. the linear discriminant function  $z$  is equivalent to the first canonical variate (see Mardia *et. al.*, 1979 for more detail). Hence the discriminant rule can be written as:

$$x \in i \quad \text{if} \quad |a^T x - a^T u_i| < |a^T x - a^T u_j|, \quad \text{for all } j \neq i. \quad (12)$$

The standard LDA can only form linear decision surfaces, although there is no restriction on the orientation of these in the feature space. In the case where the class distributions are unknown, or we have reason to believe they are not normally distributed, we can expect more satisfactory results than the MLC or MDM methods, as it is unconstrained by any prior statistical model.

## 4.0 Decision Trees

Decision trees have evolved from both a statistical consideration (Hunt *et. al.*, 1966) and from development in the field of AI (Quinlan, 1986). Decision trees are an example of inductive learning and, as such, implement a rule-based classifier. They involve a recursive partitioning of the feature space, based on a set of rules that are learned by an analysis of the training set. A tree structure is developed where, at each branching, a specific decision rule is implemented, which may involve one or more combinations of the attribute inputs. A new input vector then "travels" from the root node down through successive branches until it is placed in a specific class. In essence then, the classification is determined by describing the path from the root node of the tree to a leaf node - each nodal set of rules progressively refining the classification in a hierarchical manner. The tree encodes high levels of complexity where necessary and more simplistic rule combinations when appropriate, so that the tree only becomes complex (deep) where class separation is difficult. Likewise, only attributes that appear to aid the classification problem are considered when rules are defined; other attributes are simply ignored.

The thresholds used for each nodal decision are chosen using minimum entropy or minimum error measures. The minimum entropy method was originally proposed by Hunt (1966) and used by Quinlan (1993) in C4.5. It is based on using the minimum number of bits to describe each decision at a node in the tree based on the frequency of each class at the node. Alternatively, some minimum error function based on statistics or algebraic distance can be used, although this is not popular in decision trees. This threshold is set by the user, again by experimentation. At some stage the process must be terminated and the criterion used to determine when a class is adequately described has been the subject of much research. With minimum entropy, the stopping criterion is based on the amount of information gained by a rule (the gain ratio).

There are several well-established decision tree classifiers e.g. C4.5 (Quinlan, 1993), CART (Breiman, 1984; Buntine, 1992) and FACT (Loh & Vanichsetakul, 1988). The partitions that are built in the feature space are orthogonal to the attribute axes, which somewhat limits the generalisation ability, although some decision trees, such as OC1 can use a non-orthogonal splitting rule (oblique), which is based on perturbing the best orthogonal cut to see if the performance of the rule can be improved.

<sup>1</sup> Fisher, R.A., (1936)

<sup>2</sup>  $W = \sum n_i S_i$ ,  $B = \sum n_i (x_i - \bar{x})(x_i - \bar{x})'$ , where  $n_i$  is class  $i$  sample size,  $S_i$  is class  $i$  covariance matrix,  $x_i$  is the class  $i$  mean sample value and  $\bar{x}$  is the population mean.

Decision trees are not constrained by any lack of knowledge of the class distributions, as they do not try to model them in any way. Sample size is therefore not restricted (unlike the case of Bayesian approximators) and multi-modal distributions are easily handled. In general, decision trees are fast and good at interpolation of data, but generally poor at extrapolation and importantly, are noise intolerant as they depend on the specified sets of data being available at each node for rule resolution. Lim *et. al.* (1998) gives a good comparison of the various decision tree algorithms, based on 32 different datasets. The C4.5 classifier (which Lim rates among the best for both accuracy and speed on geographical datasets) is used for comparative purposes in this paper.

## 5.0 Artificial Neural Networks – the MLP

MLP's have become increasingly popular as classification tools in a number of fields. They are highly parametric, in the sense that they must be fitted with a large parameter set (their *weights* and *biases*) but, similar to decision trees, they do not depend on knowledge of the class distributions, as they use an inductive, data-driven approach to modelling class discrimination. Like the LDA approach, they model decision surfaces, not class distribution volumes. Their main perceived drawback is the complexity and time involved in choosing and setting up the initial network. The software package DONNet (Discrete Output Neural Net), developed at Curtin University, shortcuts these problems by using an automated procedure to calculate the initial values and architecture, hence removing the need for multiple *ad hoc* testing to select these parameters. There are many variants, but the most popular for supervised classification is the feed-forward backpropagation network, of which DONNet is an example (German & Gahagan, 1996).

MLP's consist of a number of layers of computationally simple units (nodes), that process their input via a non-linear activation function. The layers are attached to each other by a set of plastic weighted connections (see Figure 2). The learning phase is devoted to varying the weights in such a way as to produce a classification with minimal error. The error is calculated by implementing some error (or cost) function  $E$ . This is minimised via some routine or algorithm  $S$ . The cost function is generally of the form:

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^q (t_j - z_j)^2, \quad (13)$$

where  $t_j$  represents the output at output node  $j$  and  $z_j$  represents the expected output at that node (given by the training set).  $S$  is generally some common numerical minimisation routine, such as gradient descent or conjugate gradient descent. For more detail, please refer to German & Gahagan (1996).

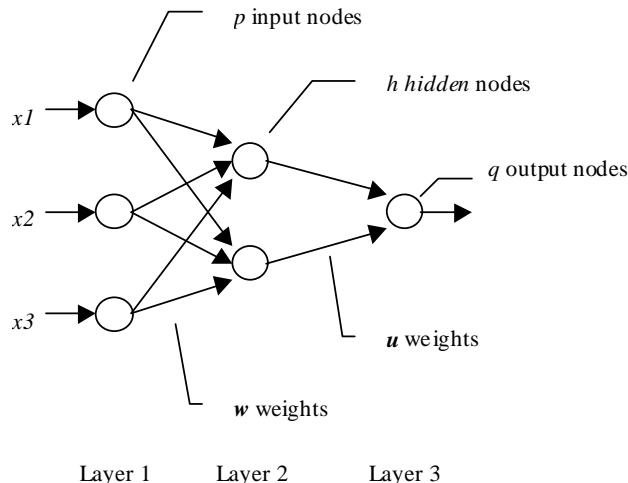


Figure 2: The multi-layered perceptron (MLP)

The use of a cost function such as Eqn 13 places a constraint on the MLP classifier that is often overlooked. This “least-squares” form of  $E$  assumes a normally distributed noise component within the data, so it is not strictly true to say that the MLP works independently of any distribution trends within the data. This can be alleviated by the implementation of a cost function of a different form (eg. Differential Learning cost functions), but many of the more effective (and efficient) minimisation routines assume such a noise distribution and are hence ineffective unless  $E$  is of a similar form to Eqn 13. Despite this, MLP’s are adept at producing acceptable classification schemas where the class distributions are unknown, sample sizes are small, or there is a high level of noise in the data. Their technique for classification can be viewed in terms of decision surfaces within the attribute space, as can both the LDA and decision tree classifiers. Each decision surface is formed/controlled by each hidden-layer node (and its associated input weights). However, unlike the decision tree, there is no constraint on the orientation of the surfaces and unlike the LDA classifier, there is no constraint on the number of surfaces or in the way they can be superimposed to produce complex, piecewise linear boundaries.

## 6.0 A Comparison of the Techniques

Comparisons of classification tools are problematic, as there are any number of ways to set-up a given classifier. For this paper, the specific classifiers used are listed in Table 1. The MDM and MLC classifiers were implemented in Splus, based on the algorithms presented here. The LDA and MLP classifiers are from the DONNet package mentioned earlier and available through the World Wide Web. The decision tree used is the popular C4.5, optimised for most generalisation. All these classifiers can be further “tuned” to provide incremental improvements in performance (for instance, DONNet can search for an optimal set of additional hyperplanes), but we are using them here in their standard mode of operation. Hence the results shown here are *indicative* only, of the performances of the particular classifiers on the presented datasets.

Type	Variant	Restrictions
MDM	In house - SPlus	Size(class $i$ ) $>= p$
MLC	In house - SPlus	Size(class $i$ ) $> 10p$
LDA	DONNet	-
Decision Tree	C4.5	-
MLP	DONNet	Gaussian noise dist

Table 1: Classifier types used for comparisons

Performance of these classifiers can be measured in terms of their:

- Learning ability.
- Generalisation ability.
- Speed.

The first criterion can be determined from measuring the performance on the *training* set. The second can be determined from measurement on some *validation* set, which is statistically independent of the training set. The speed is simply measured as the time taken to converge to the figures given by the second criterion. The datasets used are described in Table 2. Dataset 1 comprises of Landsat TM data only. The output classes are well defined crop types and form large, homogeneous regions within the image. Dataset 2 comprises of Landsat TM data plus 7 ancillary layers of data, including digital elevation, geology and flow accumulation. The output classes represent the dominant vegetation cover and, unlike dataset 1, the training sites do not represent contiguous regions in larger target objects, but are instead isolated and ‘random’ samples (pixels). As such, it provides a much “harder” classification problem and none of the classifiers can be expected to produce a high level of accuracy on this dataset. Table 2 also shows the number of samples in the smallest class within the training sets.

Data set	attributes	classes	samples	Min/class
1	6 – Landsat TM imagery	8 – crop cover	3630	165
2	11 – 4 Landsat TM + ancillary data	9 – floristic classification	1160	53

Table 2: Datasets used for these comparisons.

Table 3 lists the performance on dataset 1, whilst the more complex dataset 2 is used in Table 4. Rather than simply adding up the total number of correctly classified samples, all % figures are calculated as the *averaged* performance over every class in the dataset, thereby removing any bias associated with varying class sample sizes (remembering that these are real-world datasets).

Classifier	Training Set (%)	Validation Set (%)	Time (min:sec)
MDM	53.55	51.25	0:25
MLC	70.65	69.70	1:45
LDA	68.75	65.15	0:35
Decision Tree	73.35	70.05	0:15
MLP	71.60	70.30	3:20

Table 3: Classifier performance on dataset 1.

As the tables show, there is quite a difference in performance across the various types of classifiers and datasets. The decision tree produces a useful combination of speed and classification ability. Its ability to generalise is somewhat hampered by the restriction of orthogonal decision surfaces, but the computational complexity of growing the tree is only of the order of  $O(n)$ . It performs equally well on both the simple and more complex datasets.

Classifier	Training Set (%)	Validation Set (%)	Time (min:sec)
MDM	38.55	37.25	1:30
MLC	46.50	41.00	2:45
LDA	48.05	43.15	0:45
Decision Tree	65.30	52.35	0:35
MLP	70.75	63.10	6:10

Table 4: Classifier performance on dataset 2.

Of the statistical methodologies, the MLC gives the best performance on datasets where the assumption of normality can be said to be reasonable. However, in dataset 2, where some of the class samples are quite sparse (remembering that the MLC would require a minimum of 110 samples per class for this dataset) and where much of the ancillary data is either multi-modal or severely skewed, it starts to show its shortcomings and the empirical approach of statistical classifiers such as LDA can actually outperform it. In these cases, the LDA classifier is more adept at generalisation than either of the Bayesian-based classifiers and also gives a useful speed improvement. The LDA is restricted by the number of decision surfaces it can generate (as it can only generate so many covariance matrices) and the positioning of these surfaces is (generally) fixed by the distribution (these arguments also apply, to a lesser extent, to the MDM and MLC). This is more noticeable when dealing with complex, overlapping distributions such as is present in dataset 2, but its non-reliance on a known set of class distributions more than compensates, when compared to the Bayesian approximation techniques.

The MLP shows the best learning and generalising ability, but at a speed sacrifice that may be prohibitive in some instances. This is not surprising, as the network must calculate on the order of  $q^3$  derivatives and 3 function evaluations (per active node), for every iteration. This may be one reason why many researchers favour the MLC approach, although the LDA should perhaps be the real choice here. However, as the number of input variables increases significantly, the MLC (which scales at  $pq^2$  due to the covariance matrices required) may actually become less efficient (for example, on hyperspatial data).

## 7.0 Other Classification Techniques

Although the classifiers presented here are fairly representative of the types of approaches currently available, there are several other methodologies emerging that require some comment.

Logistic Discriminant Analysis is similar to LDA, but with the ability to construct non-linear decision boundaries. Several classifiers based on this technique have been developed and have been shown to

give results comparable to the decision tree approach. They are, however, quite complex in implementation, hence rather oblique to analysis (see Hastie *et. al.*, 1995).

The self-organising map (SOM) is a variant of the unsupervised Kohonen neural network that is being used with some success as an alternative neural network approach. It essentially does a vector search in attribute space to find a set of “key” vectors that represent each class and then runs a clustering routine to develop decision boundaries. It is a much faster technique than neural networks based on backpropagation, giving training times on par with optimised decision trees. It does not provide quite the same generalising ability as the MLP used here, however, so at this stage, it is hard to see any distinct advantage that it might have over an optimised decision tree technique, although current research is encouraging (see Gahegan & Takatsuka, 1999).

Genetic algorithms (GA) have recently stirred research interest, particularly a variant known as Differential Evolution (DE, Storn & Price, 1995). The technique is based on a model of the biological system of splitting and recombining chromosomal sequences. Combinations of attributes are represented as “chromosomes” and those that provide the best class separation per iteration (or “generation”) are selected to “evolve” through to the next iteration. The technique provides a truly global attribute search strategy, rather than the local strategies used in decision trees and neural networks. However, the price is efficiency. DE is *very* slow, commonly taking hours to days to train. Additionally, techniques such as LDA or DONNet, although strictly speaking only local in their search scope, start out reasonably close to the global minimum (as starting conditions are computed from the characteristic spread of the data) and therefore will usually converge on the global solution anyway, in a fraction of the time.

## 8.0 Conclusions

When generalisation ability is the dominant criterion for success, unconstrained by efficiency considerations, the MLP is a consistently superior classifier. It can work with sparse, noisy data and does not require any assumptions on the population distribution or the sampling process. Additionally, MLPs such as DONNet are simple to use and require no specialised knowledge. By contrast, the popular MLC classifier is not significantly faster, becomes rapidly more inefficient as the number of attribute dimensions increases and gives poorer classification accuracy on real-world problems due to its underlying statistical data requirements. The MDM classifier has a significant benefit in terms of speed, but if this is what is required, the LDA classifier is a better choice; it outperforms the MDM and even the MLC on the more complex dataset.

The decision tree classifier is perhaps the best all-round choice. It is as fast as LDA, approaches the MLP in terms of learning ability and still maintains useful generalising ability. As long as the noise in the dataset to be classified has been well-modelled in the training set, it is a quite robust classifier that does not require any knowledge of the data distribution.

## References

1. Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books and Software, Monterey, CA.
2. Buntine, W. (1992). Learning Classification Trees. *Statistics and Computing*, Vol. 2, pp. 63-73.
3. Duntzman, G. H. (1984). *Introduction to Multivariate Analysis*. Sage Publications, Beverly Hills, CA.
4. Gahegan, M. and Takatsuka, M. (1999). Dataspaces as an organizational concept for the neural classification of geographic datasets. Proc. *Fourth International Conference on GeoComputation*, Virginia, USA.
5. German, G. W. H. and Gahagan, M. N. (1996). Neural Network Architectures for the Classification of Temporal Image Sequences, *Computers and Geosciences*, Vol. 22, No. 9, pp 969-979.

6. Hastie, T. J., Buja, A. and Tibshirani, R. J. (1995). Penalized Discriminant Analysis. *Annals of Statistics*, Vol. 23, pp. 73-102.
7. Hunt, E. B., Marin, J. and Stone, P. J. (1966). *Experiments in Induction*. Academic Press, NY, USA.
8. Lim, T., Loh, W. and Shih, Y. (1998). An Empirical Comparison of Decision Trees and Other Classification Methods. *Technical Report No. 979*, Dept. Statistics, University of Wisconsin, Madison, USA.
9. Loh, W. Y. and Vanichsetakul, N. (1988). Tree Structured Classification via Generalised Discriminant Analysis (with Discussion). *Journal of the American Statistical Association*, Vol. 83, pp. 715-728.
10. Mardia, K. V., Kent, T. and Bibby, J. M. (1979). *Multivariate Analysis*. Academic Press, London.
11. Quinlan, J. (1986). Induction of Decision Trees. *Machine Learning*, Vol. 1, pp. 81-106.
12. Richards, J. A. (1986). *Remote Sensing Digital Image Analysis: An Introduction*. Springer-Verlag, Berlin.
13. Storn, R. and Price, K. (1995). Differential Evolution - A Simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report TR-95-012*, ICSI Berkeley University, [ftp.icsi.berkeley.edu](ftp://ftp.icsi.berkeley.edu).
14. Swain, P. H. and Davis, S. M. (1978). *Remote Sensing: The Quantitative Approach*. McGraw-Hill.