Available online at www.sciencedirect.com
SCIENCE DIRECT®
Remote Sensing of Environment 86 (2003) 554–565
Remote Sensing of Environment
www.elsevier.com/locate/rse

An assessment of the effectiveness of decision tree methods for land cover classification

Mahesh Pal, Paul M. Mather*

School of Geography, University of Nottingham, Nottingham NG7 2RD, UK

Received 13 May 2002; received in revised form 18 April 2003; accepted 3 May 2003

Abstract

Choice of a classification algorithm is generally based upon a number of factors, among which are availability of software, ease of use, and performance, measured here by overall classification accuracy. The maximum likelihood (ML) procedure is, for many users, the algorithm of choice because of its ready availability and the fact that it does not require an extended training process. Artificial neural networks (ANNs) are now widely used by researchers, but their operational applications are hindered by the need for the user to specify the configuration of the network architecture and to provide values for a number of parameters, both of which affect performance. The ANN also requires an extended training phase.

In the past few years, the use of decision trees (DTs) to classify remotely sensed data has increased. Proponents of the method claim that it has a number of advantages over the ML and ANN algorithms. The DT is computationally fast, make no statistical assumptions, and can handle data that are represented on different measurement scales. Software to implement DTs is readily available over the Internet. Pruning of DTs can make them smaller and more easily interpretable, while the use of boosting techniques can improve performance.

In this study, separate test and training data sets from two different geographical areas and two different sensors—multispectral Landsat ETM+ and hyperspectral DAIS—are used to evaluate the performance of univariate and multivariate DTs for land cover classification. Factors considered are: the effects of variations in training data set size and of the dimensionality of the feature space, together with the impact of boosting, attribute selection measures, and pruning. The level of classification accuracy achieved by the DT is compared to results from back-propagating ANN and the ML classifiers. Our results indicate that the performance of the univariate DT is acceptably good in comparison with that of other classifiers, except with high-dimensional data. Classification accuracy increases linearly with training data set size to a limit of 300 pixels per class in this case. Multivariate DTs do not appear to perform better than univariate DTs. While boosting produces an increase in classification accuracy of between 3% and 6%, the use of attribute selection methods does not appear to be justified in terms of accuracy increases. However, neither the univariate DT nor the multivariate DT performed as well as the ANN or ML classifiers with high-dimensional data.

Keywords: Decision tree; Land cover; Classification algorithm

1. Introduction

The past three decades have seen continuing developments in the area of pattern recognition. Research into algorithmic aspects of pattern recognition has proceeded alongside the development of instruments that are capable of producing high volumes of data, including images with increasingly finer spatial and spectral resolution. After 30 years of satellite remote sensing of the Earth's land surface, users of remotely sensed data now have access to sophisticated statistical and neural/connectionist algorithms for both fuzzy and hard classifications of their data (Mather, 1999; Schowengerdt, 1997).

Both the statistical and neural/connectionist approaches have limitations. Statistical methods rely on the assumption that the probabilities of class membership can be modelled by a specific probability density function. In most cases, the Gaussian distribution is chosen, as it is characterised by first- and second-order statistics, that is, the class mean vectors and class covariance matrices. If training set size is fixed, then the precision of the estimates of the elements of the sample class mean vector and sample class covariance

* Corresponding author. Fax: +44-115-951-5249.
E-mail address: paul.mather@nottingham.ac.uk (P.M. Mather).

0034-4257/03/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/S0034-4257(03)00132-9

matrix declines as the number of features (dimensions) increases, so that one might expect the performance of the classifier to degrade as the number of features increases. The assumption that the data in each class follow a multivariate normal model restricts the analysis to interval or ratio scale data.

Neural/connectionist methods appear to work well with training data sets that are smaller in size than those required for statistical procedures. On the other hand, network training times can be lengthy, while choice of the design of network architecture (in terms of numbers of hidden layers and neurons per layer) and the values of the learning rate parameters is not straightforward (Foody & Arora, 1997; Kavzoglu, 2001; Wilkinson, 1997). Unlike statistical methods, the neural/connectionist approach makes no assumptions concerning the statistical frequency distribution of the data or the measurement scales of the features that are used in the analysis. The most commonly used neural/connectionist algorithm is the back-propagating multi-layer perceptron (Wilkinson, 1997), which is used in this study.

Decision tree (DT) classifiers have not been as widely used within the remote sensing community as either the statistical or the neural/connectionist methods. The advantages that decision trees offer include an ability to handle data measured on different scales, lack of any assumptions concerning the frequency distributions of the data in each of the classes, flexibility, and ability to handle non-linear relationships between features and classes (Friedl & Brodley, 1997). In contrast to neural networks, decision trees can be trained quickly, and are rapid in execution (Gahegan & West, 1998). They can be used for feature selection/reduction as well as for classification purposes (Borak & Strahler, 1999). Finally, the analyst can interpret a decision tree. It is not a 'black box', like the neural network, the hidden workings of which are concealed from view.

Overall classification accuracy is used here to measure the performance of the different methods. The level of classification accuracy that is achieved in a particular case depends on a number of factors, including the nature of the classification problem in terms of the complexity of the decision boundaries that separate the classes in feature space (assuming that the classes are separable), the training sample size, the adequacy of the training data in characterising the properties of the chosen classes, the dimensionality of the data, and the properties of the classifier used (Raudys and Pikelis, 1980). We do not consider all of these problems in this paper. However, the results of our analyses are internally comparable, as the same training and test data sets are used for all three classifiers, for two dissimilar study areas (Section 2). Thus, it is possible to examine both the relative performance of the different classifiers and the consistency of these comparisons between data sets with dissimilar characteristics in terms of the terrain of the study area and the nature of the imaging system used.

The paper is structured as follows. Section 2 describes the two test data sets that are used in this study. Brief details of the three classifiers are provided in Section 3. The effects of training set size, data dimensionality, attribute selection methods, pruning and boosting on the performance of the DT classifier are considered in Section 4. A short comparative analysis of the relative performance of the DT, artificial neural networks (ANNs), and maximum likelihood (ML) classifiers is given in Section 5, which is followed by a summary of conclusions.

2. Test data sets

Two contrasting data sets are used. The first is a medium-resolution (Landsat ETM+) image of part of Eastern England near the town of Littleport. This area is relatively flat and low-lying, and is mainly devoted to intensive arable agriculture. The image was collected on 19 June 2000. Seven main land cover types are identified, namely, wheat, potato, sugar beet, onion, peas, lettuce, and beans. Official field data printouts (which record the crop or crops grown in each field) for the year 2000 were collected from farmers and their representative agencies, and other parts of the area were surveyed on the ground to assemble the ground reference data. A subimage consisting of 307 pixels (columns) by 330 pixels (rows) covering the area of interest was extracted from the ETM+ image for subsequent analyses. ETM+ band 6 (the thermal band) was omitted.

Hyperspectral data acquired by the DAIS 7915 airborne imaging spectrometer on 29th June 2000 form the second test data set. The spatial resolution of DAIS data is 5 m, and measurements are made in 72 spectral bands in the visible and short-wave infrared regions of the spectrum. The study area is located within the region of La Mancha Alta, which covers an area of approximately 8000 km^2 and is located to the south of Madrid, Spain. The region contains semi-arid wetland with some irrigated and dryland agriculture. Eight different land cover types (wheat, water body, salt lake, hydrophytic vegetation, vineyards, bare soil, pasture lands, and built-up area) were identified. A subimage of size 512×512 pixels covering the area of interest was extracted. Of the 72 bands available in the visible and short-wave infrared region, a subset comprising 65 bands was selected, as visual inspection showed that seven bands suffered from severe horizontal striping effects.

Random sampling methods were used to collect separate training and test data sets in both study areas using ground reference data generated from field observations and, in the case of the Littleport study area, from official farm records. The pixels collected by random sampling were divided into two subsets, one of which was used for training and the second for testing the classifiers, so as to remove any bias resulting from the use of the same set of pixels for both training and testing. Also, because the same test and training data sets are used for each classifier, any differences resulting from sampling variations are avoided.

3. Methods

The maximum likelihood, multi-layer back-propagation neural network, and decision tree procedures are used in this study. A brief summary of the properties of each of these classifiers is given in this section.

3.1. Maximum likelihood classifier

As usually implemented, the ML procedure is based on the assumption that the members of each class follow a Gaussian frequency distribution in feature space. ML is a pixel-based method, and can be defined as follows: a pixel with an associated observed feature vector **x** is assigned to class c_j of N classes if

$$g_j(\boldsymbol{x}) > g_k(\boldsymbol{x}) \text{ for all } j \neq k, \quad \text{with } j,k = 1,\ldots,N.$$

For the multivariate Gaussian distribution, the discriminating function $g_k(\boldsymbol{x})$ is given by:

$$g_k(\boldsymbol{x}) = \ln(p(\boldsymbol{x} \mid c_j)) = \ln \hat{\boldsymbol{\Sigma}}_k + (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_k)^T \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_k)$$

where $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$ are the sample mean vector and sample covariance matrix for class k.

Implementation of the ML algorithm involves the estimation of class mean vectors ($\hat{\boldsymbol{\mu}}_k$) and covariance matrices ($\hat{\boldsymbol{\Sigma}}_k$) from training data chosen from known examples of each particular class. The function $g_i(\boldsymbol{x})$ is used to evaluate the membership probability of an unknown pixel for class j. The pixel is assigned to the class for which it has the highest membership probability value.

3.2. Artificial neural network classifier

The most widely used artificial neural network model in remote sensing applications is the back-propagating multilayer perceptron. Its design consists of one input layer, at least one hidden layer, and one output layer. The hidden and output layers are made up of sets of non-linear processing units, or neurons, and the connections between neurons in successive layers carry associated weights (Bishop, 1995). Information is carried only in the forward direction, that is, from input layer to the first hidden layer, or from a hidden layer to a subsequent hidden or output layer. Non-linear processing is performed by applying an activation function to the summed inputs to each neuron. The network is trained using back-propagation, which uses a gradient-descent algorithm to minimise the error between the known label of the training pixel and the label output by the network for that pixel. Each member of a set of training pixels is repeatedly presented to the network, and the error (measured by the difference between the network output and the known label of the training pixel) is propagated from the output layer back to the input layer. The weights on the backward path through the network are updated according to an update rule and a learning rate. ANNs are not specified solely by the characteristics of their processing units and the selected training or learning rule (Paola & Schowengerdt, 1995). The network topology, that is, the number of hidden layers and the number of neurons per layer, has a considerable influence on performance. There is no clear guide to the determination either of the network architecture or to the choice of the initial values of user-supplied parameters that control, for example, the performance of the error minimisation procedure. The settings recommended by Kavzoglu (2001) are used here.

3.3. Decision tree classifiers

Unlike conventional statistical and neural/connectionist classifiers, which use all available features simultaneously and make a single membership decision for each pixel, the DT uses a multi-stage or sequential approach to the problem of label assignment. The labelling process is considered to be a chain of simple decisions based on the results of sequential tests rather than a single, complex decision. Sets of decision sequences form the branches of the DT, with tests being applied at the nodes. The leaves (or branch termini) represent labels (Fig. 1).

DT construction involves the recursive partitioning of a set of training data, which is split into increasingly homogeneous subsets on the basis of tests applied to one or more of the feature values. These tests are represented by nodes. The univariate DT applies a test to a single feature at a time, whereas the multivariate DT uses one or more features simultaneously. Labels are assigned to terminal (leaf) nodes by means of an allocation strategy, such as majority voting. At one time, DTs were designed manually, using spectral plots. In the past decade, automatic methods of decision tree design have been developed. In this study, two univariate DT algorithms, *C4.5* and *See5.0* (Quinlan, 1993, 1996), and

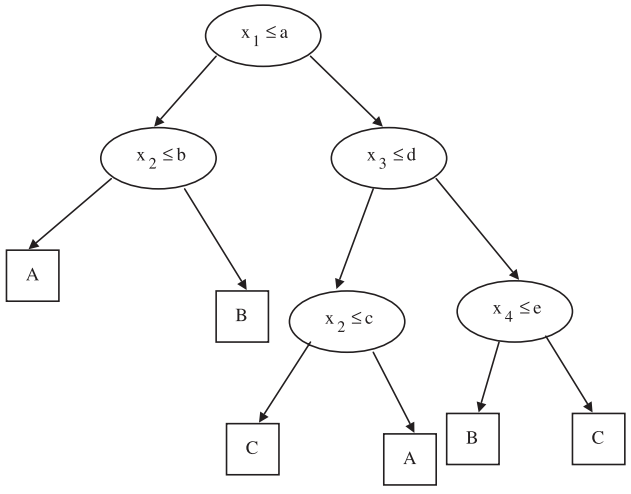


Fig. 1. A classification tree with four dimensional feature space and three classes. The x_i are feature values; a, b, c, d, and e are the thresholds and A, B, and C are class labels.

one multivariate algorithm, *QUEST* (Loh & Shih, 1997), are used.

3.3.1. Univariate decision tree

A univariate DT is one in which the decision boundaries at each node of the tree are defined by the outcome of a test applied to a single feature that is evaluated at each internal node (Swain & Hauska, 1977). On the basis of the test outcome, the data are split into two or more subsets. Each test is required to have a discrete number of outcomes. A univariate DT classification proceeds by recursively partitioning the input data until a leaf node is reached, and the class label associated with that leaf node is then assigned to the observation. The characteristics of the decision boundaries in a univariate DT are estimated empirically from the training data. In the case of continuous data, a test of the form $x_i > c$ is performed at each internal node of the DT, where x_i is a measurement in the feature space and c is a threshold estimated from the distribution of the x_i. The value of c is estimated by using some objective measure that maximises the dissimilarity or minimises the similarity of the descendant nodes, using one feature at a time (Fig. 2). A number of attribute selection methods are described in the literature. The most frequently used of these are the information gain, the information gain ratio (Quinlan, 1993), the Gini index (Breiman, Friedman, Olshen, & Stone, 1984), and the chi-square measure Mingers (1989b). As each test in univariate DT is based on a single feature, it is restricted to a split through the feature space that is orthogonal to the axis representing the selected feature.

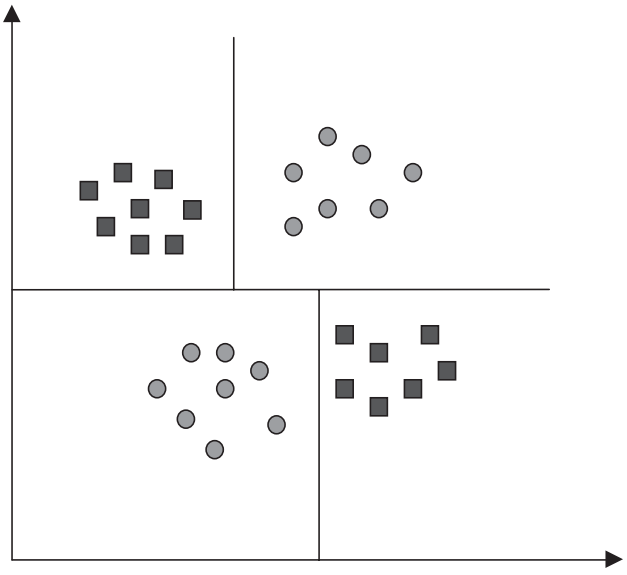


Fig. 2. Axis-parallel decision boundaries of a univariate decision tree.

3.3.2. Multivariate decision trees

If the locations of decision boundaries in feature space can be properly defined only in terms of combinations of features rather than sequences of single features, then the univariate DT will perform poorly (Breiman et al., 1984; Utgoff & Brodley, 1990). In such cases, the set of allowable splits can be extended to include linear combinations of features (Fig. 3). A set of linear discriminant functions is estimated at each interior node of a multivariate DT, with the coefficients for the linear discriminant function at each interior node being estimated from the training data. The splitting test at each node has the form $\sum_{i=1}^{n} \boldsymbol{a}_i \boldsymbol{x}_i \leq c$, where $\boldsymbol{x}_i$ represents a vector of measurements on the n selected features, $\boldsymbol{a}$ is a vector of coefficients of a linear discriminant function, and c is a threshold value. Brodley and Utgoff (1992) find that multivariate DTs are more compact and able to produce more accurate classifications than univariate DTs. The greater complexity of multivariate relative to univariate DT algorithms introduces a number of factors that affect their performance. First, any of a number of different algorithms can be used to estimate the splitting rule at internal nodes, and the relative performance of these methods can differ depending on the nature of the data and the complexity of the classification problem. Second, as the split at each internal node of a multivariate DT is based on one or more features, so several different algorithms may be used to perform feature selection at each internal node within a multivariate DT (Friedl & Brodley, 1997). These algorithms choose the features to include in each test on the basis of the data observed at a particular node, rather than selecting a uniform set of features on which tests for the entire tree are based.

It is also possible to use different classification algorithms at different nodes of a DT classifier. This type of tree is called a hybrid DT (Friedl & Brodley, 1997). Another approach to the design of DT using Support Vector Machines has also been proposed by Bennett and Blue (1998).

4. Results

The aim of the present study is to evaluate the effect of the following factors on the level of overall classification accuracy achieved by the three classification algorithms selected for this study:

- training data set size,
- dimensionality of the data set,
- attribute selection measures,
- pruning methods, and
- boosting techniques.

The effects of data dimensionality are evaluated using DAIS hyperspectral data set for a test area in La Mancha (test data set 2), while the multispectral Landsat ETM+ for the Littleport test area (test data set 1) was used to assess the impact of variations in the other factors. The salient char-

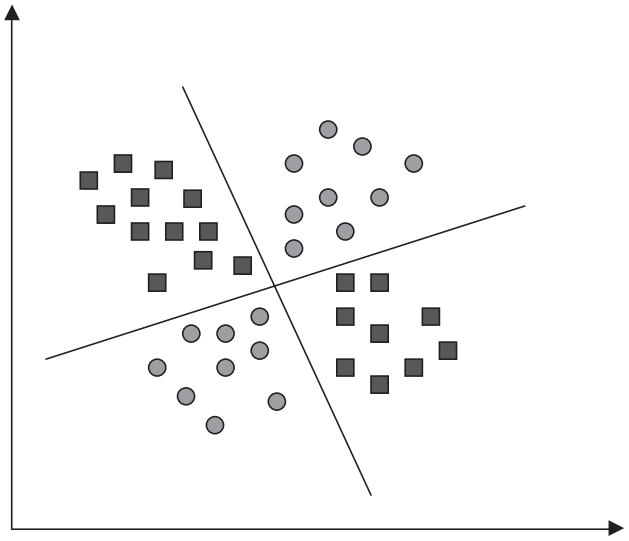


Fig. 3. Decision boundaries for a multivariate decision tree classifier.

acteristics of the two test data sets are summarised in Section 2.

4.1. Effect of training set size

The characteristics of the data used to train a supervised classifier have a considerable influence on the accuracy of the resulting classification (Campbell, 1981). It is essential that the number of classes is adequate to describe the land cover of the study area, and that the training data provide a representative description of each class. For the ML classifier, an important requirement is that the number of pixels included in the training data set for each class should be at least 10–30 times the number of features (Mather, 1999). The required training set size may therefore be large, and will increase rapidly as the number of features increases to avoid the so-called "Hughes phenomenon" (Hughes, 1967), which shows decreasing classifier performance as the number of features increases for a constant training data set size. Acquiring such large training sets may be difficult and costly where a large number of classes is involved, or where hyperspectral data are used. Consequently, some investigations may use a sample size that is smaller than the generally accepted guideline for statistical classifiers such as ML. This implies that the standard errors of the estimates of the required parameters are larger than the recommended level, and therefore, decision boundaries may be located incorrectly or imprecisely. Landgrebe (2000) considers further the relationship between classification accuracy and the problem of adequate class definition.

It has been suggested that ANN-based classifiers can perform successfully using training data sets that are smaller than those required to train statistical classifiers (Hepner et al., 1990; Foody, McCulloch, & Yates, 1995). Nevertheless, investigations of the effects of training set characteristics on the performance of ANNs indicate that training data set size has a substantial effect on classification accuracy (Foody & Arora, 1997; Foody et al., 1995; Kavzoglu, 2001).

To evaluate the effects of training set size on classification accuracy using a DT classifier, seven subsets of training data for the first test area (Littleport, eastern England) were formed by randomly sampling the set of available training data. The numbers of pixels in each of these training data subsets are 700, 1050, 1400, 1750, 2100, 2400, and 2700 pixels, respectively, with an equal number of pixels per class for the seven classes, giving 100, 150, 200, 250, 300, 350, and 400 pixels per class, respectively. A separate set of 2037 pixels was used for testing the classifier. The test set did not include any pixels from the training data sets.

Fig. 4 shows the relationship between accuracy and training set size using a univariate DT classifier. These results indicate that the level of accuracy increases with the size of the training set, and that the rate of increase in classification accuracy with increasing training set size is linear up to the fifth training data set, which contains 2100 pixels. As the training set size increases from 700 to 2100 pixels (i.e., 100–300 pixels per class), there is an increase in classification accuracy, from 78.3% to 84.1%. However, further increases in training set size, using the sixth and seventh data sets, produced anomalous results, with the sixth training data set producing a slight decrease in accuracy. These results indicate that (i) the accuracy of a univariate decision tree classifier improves as the size of the training set is increased, but only up to a point, and (ii) these classifiers do not require very large training sets to be effective. It should be noted that our results do not concur with the findings of Oates and Jenson (1997), who suggest that the size of the training data set has no effect on classification accuracy. Our findings indicate that—for the test problem that we addressed—a training data set size of 300 samples per class provided an adequate description of land cover variations. The figure of 300 samples is problem-specific, and should not be used as a guide for other applications.

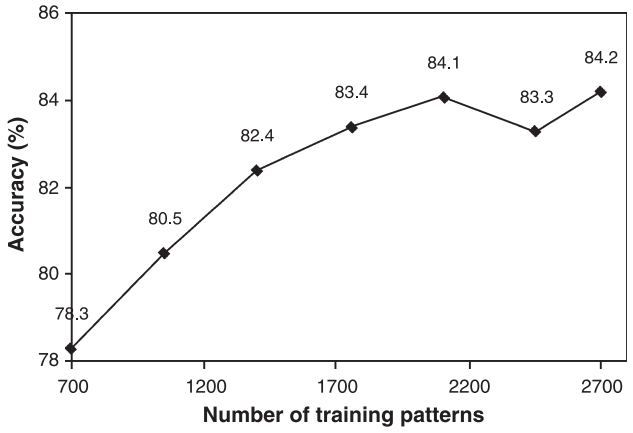


Fig. 4. Variation of classification accuracy with increasing number of training patterns using univariate decision tree classifier and ETM + data (test data set 1, Littleport).

A second experiment was carried out to study the response of a multivariate DT classifier to increasing training data set size. The same set of training and test data as used in the previous example was employed, to ensure compatibility (Fig. 5). Again classification accuracy increases with training set size. This increase is almost linear up to fourth data set (700–1750 pixels). Accuracy then starts to fall with the fifth and sixth data sets, but rises again so that the highest classification accuracy is achieved by the seventh data set.

These results indicate that the level of classification accuracy achieved by a multivariate DT increases with the size of the training set, but not in a systematic way. The behaviour of the multivariate classifier was found to be rather less predictable than that of the univariate DT as the number of training patterns increases beyond a certain limit. It is evident also that the level of classification accuracy associated with the multivariate classifier is no higher than that of the univariate classifier for this data set. As training time is always greater with a multivariate decision tree classifier, we conclude that the univariate DT classifier is adequate for this type of data.

4.2. Dimensionality of the feature space

Hyperspectral data are characterised by their high dimensionality. An important characteristic of statistical classifiers is that the properties of each class are modelled using a probability density function. Usually the Gaussian density is chosen, as it can be described in terms of a mean vector and a variance–covariance matrix, both of which are estimated from the sample data for each class. The standard errors of the k estimates of the elements of the class mean vector and the $k(k-1)/2$ elements of the variance–covariance matrix for each class depend on the ratio between the number of dimensions, k, and the number of pixels included in the training data set for that class. As the dimensionality of the data increases, so more training data are required to provide acceptable estimates of the statistical parameters. If the number of training data pixels is inadequate, which may be the case with hyperspectral data, then parameter estimation becomes inaccurate as standard errors of the estimate become larger (Hsieh & Landgrebe, 1998). Increasing the number of spectral bands provides more information to be used in discriminating between classes but, for statistical classifiers at least, this information is only useful if the number of training data increases proportionately.

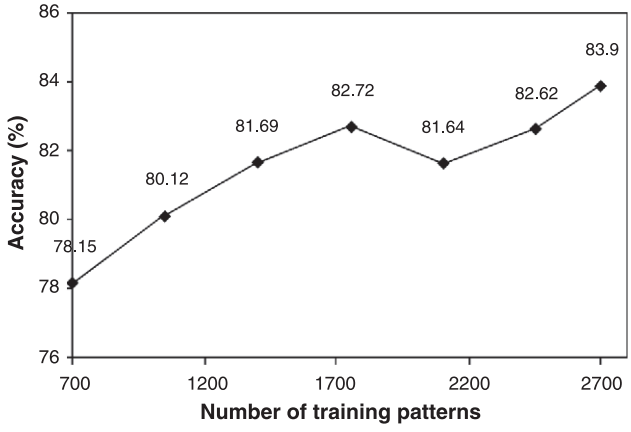
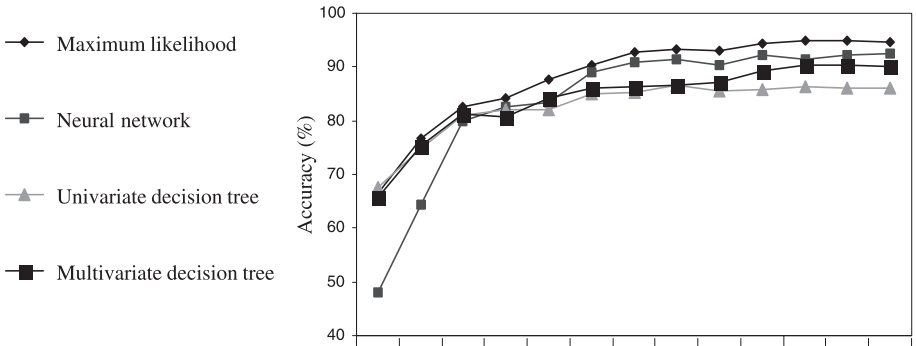


Fig. 5. Variation of classification accuracy with increasing number of training patterns using a multivariate decision tree classifier and ETM + data (test data set 1, Littleport).

The purpose of this part of the study is to assess the behaviour of univariate and multivariate DT classifiers as the number of features increases while training data set size is kept constant. As DT classifiers do not use all features simultaneously for training, and also because class separability in high-dimensional data may be a function of a combination of features rather than a single feature, the performance of the DT classifier is compared with that of the ML and NN classifiers, which use all available features simultaneously in the labelling process. A fixed-size training set composed of 2000 pixels (giving 250 pixels per class) and a test data set of 3800 pixels were employed. Both were drawn from the second (La Mancha) hyperspectral test data set.

Fig. 6 shows the levels of overall classification accuracy obtained using the ML, NN, and DT classifiers. The number of features was initially set to 5 (the first five DIAS bands) and then increased by 5 at each iteration, so that the first experiment is based on DAIS bands 1–5, the second on bands 1–10, and so on. The level of accuracy associated with the univariate DT classifier is higher than the corresponding values for the NN, ML, and multivariate DT classifiers for the first data set in which five features were used, but the classification accuracy of the univariate DT declines as the number of feature increases. A possible reason for this behaviour may be that the performance of the univariate DT classifier is affected by the number of training samples, and the use of a large training sample to subdivide feature space may result in a very large and complex decision tree. The univariate DT classifier uses a test applied to the value of a single attribute at each branch or node in the tree. As the number of features increases, it becomes more likely that two or more features are interrelated. This is especially true of hyperspectral data, with contiguous and narrow wavebands. Class structure becomes more dependent on combinations of features as a result of these correlations, thus making it difficult for a univariate DT classifier to perform well.

Further investigation using a multivariate DT suggests that it achieves a lower level of classification accuracy using high-dimensional data than either the ML or the ANN classifiers. The complexity of higher dimensional feature spaces, with intercorrelated features, may be too great to allow the DT classifier—either univariate or multivariate—to perform well in comparison to either the ML or the NN classifiers.



	5	10	15	20	25	30	35	40	45	50	55	60	65
Maximum likelihood	66.5	76.8	82.7	84.1	87.7	90.3	92.8	93.3	93	94.3	94.9	95	94.7
Neural network	48	64.4	80	82.7	83.5	89.1	91	91.5	90.4	92.3	91.5	92.3	92.5
Univariate decision tree	67.6	74.8	81.1	82.1	82.1	84.9	85.2	86.5	85.6	85.7	86.4	86.2	86
Multivariate decision tree	65.8	75.3	81.3	80.7	84.2	86.1	86.4	86.5	87.2	89.4	90.3	90.3	90.1

Number of bands

Fig. 6. Classification accuracies using DAIS hyperspectral data (test data set 2, La Mancha) with fixed training set of 250 pixels/class and increasing number of features.

4.3. Attribute selection measures

There are many approaches to the problem of selecting the set of attributes to be used for DT induction, and these approaches have been studied in detail by Borgelt, Gebhardt, and Kruse (1996), Breiman et al. (1984), Kononenko and Hong (1997), Mingers (1989b), Murthy, Kasif, and Salzberg, (1994), and Quinlan (1993). Some approaches use measures of the "goodness of split" (Brieman et al., 1984) while other approaches attempt to minimise the "impurity" of the training data.

An impurity function measures the heterogeneity of a set of observations. It records its lowest value for a pure set and its highest value for a maximally impure set. Impurity functions are used in selecting the feature to be used to further split the data at the current node of a DT. The best attribute for splitting is selected by examining how well each candidate feature separates the data into the various classes.

The purpose of this section is to examine these various attribute selection measures in terms of their comparative performance for land cover classification. A univariate DT classifier is used, together with error-based pruning (Section 4.4). Four attribute selection measures are employed: the information gain, the information gain ratio (Quinlan, 1993), the Gini index (Breiman et al., 1984), and the chi-square measure (Mingers, 1989b). A total of 2700 training patterns and 2037 testing patterns was used in this experiment. The result shown in Fig. 7 essentially confirms the findings of Breiman et al. (1984) that the level of classification accuracy is not seriously affected by the choice of attribute selection measure. Except for the information gain ratio, the accuracy level obtained for each selection measure is almost identical, and the increase in accuracy resulting from the use of the information gain ratio is less than 1%.

4.4. Pruning methods

DT classifiers attempt to divide the training data into subsets that should contain only a single class. The result of this procedure is often a very large and complex tree. In most cases, fitting a DT until all leaves contain data for a single class may overfit to the noise in the training data, as some training samples may not be members of the class that they purport to represent. If the training data contain any errors, then overfitting the tree to the data in this manner can lead to poor performance on unseen cases (Breiman et al., 1984). To reduce the impact of this problem, the original tree can be pruned.

Simplification involves the removal of those parts of the tree that do not contribute to classification accuracy on unseen cases, thus producing a less complex and more

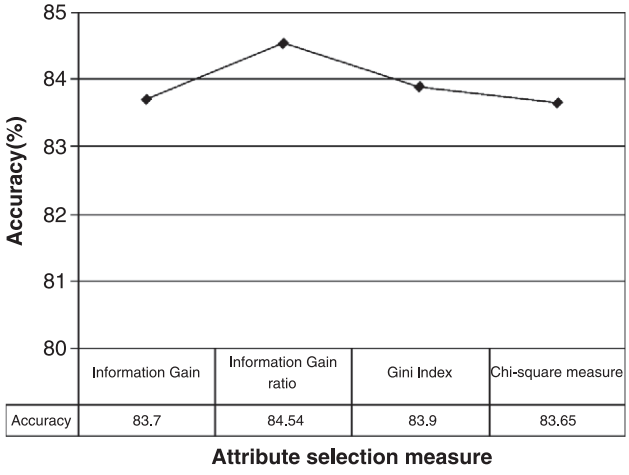


Fig. 7. Variation of accuracy with different attribute selection measures using ETM + data (test data set 1, Littleport) with 2700 training pixels.

Table 1
(a) Effect of pruning on tree size (complexity) and classification error using ETM + data (test site 1, Littleport)

Evaluation on training data				Evaluation on test data			
Before pruning		After pruning		Before pruning		After pruning	
Tree size	Error (%)	Tree size	Error (%)	Tree size	Error (%)	Tree size	Error (%)
713	1.6	231	8.6	713	17.6	231	15.7

Classification accuracy is defined as (100 − error)%. Results indicate that pruning reduces the size of decision tree as well as error on test data. In this case, the size of the pruned tree is ~ 33% of the original tree size. The error level of the independent test data set drops by ~ 2%.

(b) Classification accuracy from boosted and unboosted decision trees using ETM + data from test site 1 (Littleport) with a total of 2700 training pixels and seven classes

	Accuracy (%)	Kappa value
Unboosted decision tree	84.24	0.816
Boosted decision tree	88.46	0.865

comprehensible tree. There are two ways in which a decision tree classifier can be modified to produce a simpler tree:

- Stop the subdivision of the training data before the tree is complete, or
- Remove retrospectively some part of the tree structure by recursive partitioning.

The first approach, sometimes called stopping or pre-pruning, has the advantage that time is not wasted in assembling a structure that is not used in the final simplified tree. This method looks for the best way of splitting a data set in terms of a criterion such as information gain or error reduction. If the value of the criterion falls below some threshold, further division of the data set is rejected. The problem with this approach lies in the formulation of an appropriate stopping rule (Breiman et al., 1984). If the threshold value is set too high, then division is terminated before the benefits of subsequent splits become evident, while too low a threshold value results in little simplification of the tree.

In the second approach, the tree is allowed to grow to its full extent. This overfitted tree is then pruned. More computation time is required to build those parts of the tree that are subsequently discarded, but this cost is offset against benefits resulting from a more thorough exploration of possible partitions.

Pruning a DT will cause it to misclassify more of the training data (Table 1). Thus, the leaves of the pruned tree will not necessarily contain training data from a single class. Instead, there will be a class distribution specifying, for each class, the probability that a training data sample at the leaf belongs to that class. Two families of techniques to predict error rates of a tree are available. In the first family, the error rates of the tree and its subtrees are predicted by using a set of test data that is separate from the training data. Because these test cases were not used in the building of the tree, the estimate of classification accuracy that is obtained from them will be unbiased and, if enough data are available, the estimate will also be reliable. In the second approach, the training data themselves are used to predict these error rates.

This section describes the results of an experiment that investigates the effect of various pruning methods on classification accuracy. Five different pruning methods are used with the information gain ratio as the attribute selection measure in a univariate DT classifier (*C4.5*). The pruning methods employed are: reduced error pruning (REP), pessimistic error pruning (PEP), and error-based pruning (EBP), all proposed by Quinlan (1987, 1993); critical value pruning (CVP) proposed by Mingers (1989a); and cost-complexity pruning (CCP) proposed by Breiman et al. (1984). Fig. 8 shows the impact on classification accuracy of the different pruning methods.

The performance of the REP method is worst, with a classification accuracy of 81.4%. The reason for this could be the requirement of a separate data set for pruning, a conclusion also suggested by Esposito, Malerba, and Semeraro (1997). PEP gives the highest accuracy of 82.9%;

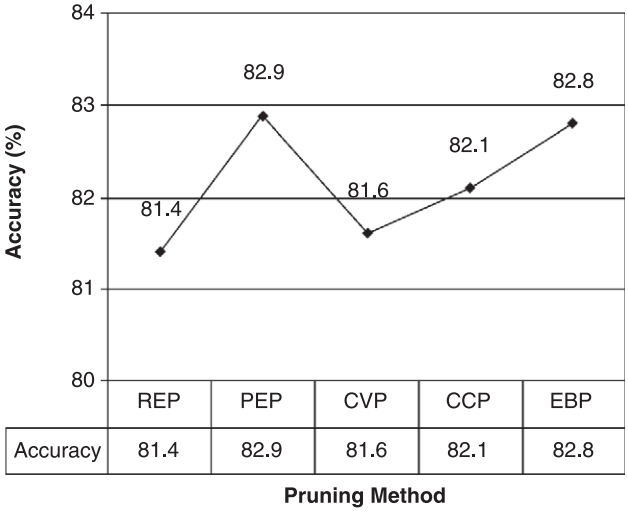


Fig. 8. Variation of classification accuracy with different pruning methods using ETM + data (test data set 1) with 2700 training pixels.

Table 2
Results from maximum likelihood and neural network classifiers using ETM+ data (test data set 1) with 2700 training pixels and seven classes

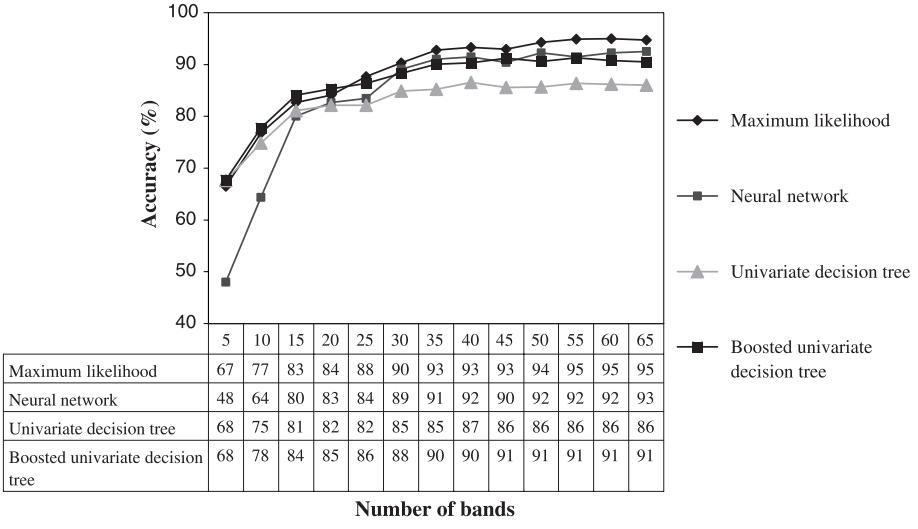
Classifier	Accuracy (%)	Kappa value
Maximum likelihood (ML)	82.9	0.80
Neural network (NN)	85.1	0.83

however, Esposito et al. (1997) note that the introduction of a continuity correction in the estimation of error rate has no theoretical justification and such a factor is improperly compared to an error rate, which may lead to either under-pruning or overpruning of the tree. The performance of CVP is affected by the choice of critical value set to prune a tree. CCP uses a separate data set or a cross-validation approach to pruning, and a pruned sub-tree is selected by minimising a complexity factor over a set of pruned sub-trees. Gelfand, Ravishankar, and Delp (1991) suggest that this set of sub-trees may not include the optimum sub-tree. Finally, EBP uses training data for pruning the tree. These results suggest that the choice of a suitable pruning method is an important factor in the design of a DT classifier in comparison to the attribute selection measures, as the availability of a sufficient number of training data elements is always a problem. Our conclusion is that EBP, which gives an overall classification accuracy of 82.8%, is the preferred method. This conclusion is not based exclusively on the relatively small increase in the level of classification accuracy that results from the use of pruning; it also takes into account the benefits deriving from the reduction in tree size and the simplification of the tree, which can improve interpretation and understanding.

4.5. Boosting

Boosting is a method of improving the performance of a "weak" classifier. In essence, this improvement is achieved by weighting the individual elements of the training data set. The weights are initially set to be equal. Comparison of the classifier output and the known label of each element of the training data should reveal cases in which elements of the training data have been classified incorrectly. These incorrectly classified training data elements are given an increased weight, and the classifier is run again. The increased weighting of the "difficult cases" forces the classifier to focus on these cases. A method similar to boosting is described by Jackson and Landgrebe (2001). We used the *AdaBoost M1* algorithm (Freund & Schapire, 1996) together with the *C4.5* decision tree software (Quinlan, 1993).

Classification accuracies and Kappa values obtained from unboosted and boosted DTs, estimated using sets of 2700 training and 2037 separate test data for test area 1 (Littleport) are shown in Table 2. The boosted DT classifications were estimated using 14 iterations of the base decision tree algorithm. The number of boosting iterations was varied from 2 to 20 but there was little change in classification accuracy beyond 14 iterations, and the degree of accuracy improvement achieved through the use of boosting starts to stabilise after eight iterations. It appears that 10–15 boosting iterations are sufficient to achieve an improvement in classification accuracy for this type of data. The result also concurs with the conclusions of studies using non-remote sensing data. Quinlan (1996) concludes that about 10 iterations is the optimum number, and that little is gained by performing additional boosting runs. We noted an increase of more than 4% in the level of classification accuracy following boosting, using test data set 1, which is a small increase in comparison to the results reported by Quinlan (1996) and Muchoney et al. (2000). Although the level of improvement is small, it should be borne in mind that even small percentage increases are difficult to generate when the overall classification accuracy level exceeds 80%.



	5	10	15	20	25	30	35	40	45	50	55	60	65
Maximum likelihood	67	77	83	84	88	90	93	93	93	94	95	95	95
Neural network	48	64	80	83	84	89	91	92	90	92	92	92	93
Univariate decision tree	68	75	81	82	82	85	85	87	86	86	86	86	86
Boosted univariate decision tree	68	78	84	85	86	88	90	90	91	91	91	91	91

Fig. 9. Classification accuracies using DAIS hyperspectral data (test data set 2, La Mancha) with a fixed training set of 250 pixels/class and an increasing number of features. The classification accuracies obtained with a boosted univariate decision tree are also shown.

The results shown in Fig. 9 and Table 2 suggest that boosting also improves the classification accuracy of a univariate DT by about 4% when used with hyperspectral data (test data set 2). We can, therefore, conclude that boosting is a useful technique for improving the performance of the DT classifier for land cover classification studies.

5. Comparison of decision tree, maximum likelihood (ML), and neural network (NN) classifiers

In this section, the results achieved by the univariate DT in classifying the test data described in Section 2 are compared with those produced by ML and ANN classifiers. The aim of this comparison is to determine whether the high accuracy values that we achieved with the DT are classifier dependent. We used a standard back-propagation neural classifier, with a single hidden layer having 26 nodes. The number of nodes and the values of the user-defined inputs to the network were set using the guidelines suggested by Kavzoglu (2001). Table 3 shows the accuracies achieved by the ML and NN classifiers with test data set 1 (ETM+), using the same training and test data as in previous sections.

The results presented in Tables 2 and 3 show that the DT classifier produces a higher level of classification accuracy than does the ML classifier, and its performance is comparable to that of an ANN, even without boosting. After boosting, the level of classification accuracy achieved by the DT improves by about 3.3%. This may not be a large increase, but it does indicate that the boosted DT produces approximately the same level of classification accuracy as an ANN, while both ANN and DT methods give a higher accuracy than the ML classifier. However, the DT classifier requires only the choice of attribute selection and pruning methods, while the use of the ANN involves decisions concerning the type of network, the network architecture, and the initial values of various parameters. It is usually the case also that the training time required by an ANN classifier is lengthy, as noted below.

The performance of boosted and unboosted DT trees using hyperspectral data (test data set 2) is compared with the accuracies achieved by using ML and NN classifiers in Fig. 9. The results of this part of the study suggest that, even after boosting, the performance of the DT classifier is below that of the ML and NN classifiers, indicating that—although boosting may help in increasing the performance of a weak base classifier—the performance of the boosted DT classifier depends fundamentally on the performance of the base DT classifier, which is relatively poor when a large number of features are used. The results presented in Fig. 9 show that the ML classifier consistently produces the highest level of classification accuracy for data sets including 20–25 features or more. Below this point, the boosted DT is only slightly better. To check these results, the experiment was repeated using a different random sample of training data of the same size as that used previously. The results are very close to those achieved by the first experiment, and the same conclusions are suggested, namely, that beyond a data dimensionality of 20–25, both the ML and ANN classifiers produce higher overall classification accuracies than the boosted or unboosted univariate DT.

Table 3
Calculated Z values for comparison between different classification systems

Classifier	Z value
Decision tree (WB) v. Maximum likelihood	2.13
Decision tree (WB) v. Neural network	1.01
Decision tree (B) v. Neural network	2.54

Shaded values indicate improvements in the performance of first-named classifier at the 95% confidence level (critical value of $Z = 1.96$). Unshaded value indicates that both classifiers perform equally well. WB means "without boosting" and B means "boosting" a decision tree classifier.

Cost is an important consideration in operational applications of remote sensing, and training a classifier often represents a significant proportion of these costs. The training time for the ANN classifier was about 58 CPU minutes on a Sun dual-processor workstation, compared to 0.7 CPU seconds using a personal computer with a Pentium II processor for the unboosted DT. Even with the use of boosting, the decision tree classifier required about 7.1 CPU seconds on a slow Pentium II machine to perform 14 boosting iterations, which is still far less than the time taken to train the ANN classifier. In terms of design effort, training time requirements, and classification accuracy, the ML method offers advantages over the ANN and DT procedures when the data are measured on an interval or ratio scale.

6. Conclusions

The main aim of this study is to assess the utility of DT classifiers for land cover classification using multispectral and hyperspectral data, and to compare the performance of the DT classifier with that of the ANN and ML classifiers. The specific objectives are to study the behaviour of decision tree classifiers with changes in training data size, choice of attribute selection measures, pruning methods, and boosting. The results presented above suggest several conclusions. First, the performance of both univariate and multivariate DT is always affected by the size of the training data set. This is a predictable outcome, but the use of common training and test data sets shows that the behaviour of the univariate DT is more systematic than that of the multivariate DT and that, in the case of the univariate DT, at least 300 pixels per training class were needed to provide the most suitable combination of classification accuracy and sample size. The study also concludes that DT classifiers are not recommended for high-dimensional data sets. Other results indicate that the choice of an appropriate pruning method has a positive effect in improving classification accuracy, while the use of attribute selection measures was

Table 4
Classification accuracies achieved using 2000 training and 3800 test data with all 65 features of the hyperspectral data set (test data set 2, La Mancha)

	Classifier used			
	ML	NN	Univariate decision tree	Boosted univariate decision tree
Accuracy (%)	94.7	92.5	86.0	90.5

found not to be important. The use of boosting is recommended; in this study, it resulted in an improvement in classification accuracy of about 3–4%, at little cost in computer time or complexity of use.

Studies carried out using ML and ANN classifiers for the same data sets indicate that the DT performs slightly better than the ML classifier, using ETM+ data. The performance of the ANN is also slightly better than that of the unboosted univariate decision tree classifier for ETM+ data, but the difference is not statistically significant, as shown in Table 4. A number of studies (Foody & Arora, 1997; Kavzoglu, 2001) suggest that the performance of an ANN classifier depends on the values of a number of parameters that the user must define in advance, while the performance of a univariate DT classifier depends on the pruning method used in the design of the tree. The training time for a neural classifier is large when compared to that of the univariate decision tree classifier. Even with the use of boosting, the training time for the decision tree is still short compared to the demands of the ANN classifier, but the performance of the boosted decision tree is better than that of the ANN classifiers.

When hyperspectral data are used, the performance of both univariate and multivariate DT classifiers declines as the number of features increases, while both the ML and ANN classifiers produce overall classification accuracy values that are higher than those produced by the DT classifier (both boosted and unboosted). This may be due to the combination of the requirement of a large training data set size and the use of a single feature to split the training data. As a result of correlations among the features forming a hyperspectral data set, it may be the case that a combination of features is needed for an informed decision to be made. The probable reason for the poor performance of the multivariate DT classifier on high-dimensional data could be a result of local feature selection, suggesting that the use of DT classifiers with high-dimensional data is limited. Finally, results also show that boosting of a univariate DT classifier did not work well for high-dimensional data.

In summary, the ML procedure performs adequately or better in the experiments we have performed, using both multispectral ETM+ data and hyperspectral DAIS data. The DT method may produce a slightly higher level of classification accuracy than ML for multispectral data, but for data dimensionalities greater than 20–25, the unboosted DT gives a much lower accuracy (95% for ML and 86% for the unboosted DT, using 55 or more features). These results were confirmed by a second experiment. It follows that the ML algorithm is preferred unless there are particular reasons for believing that data do not follow a Gaussian (or, at least, a unimodal) distribution. As noted in Section 4.1, the adequacy of the training and test data sets in characterising the variability present in each of the classes is possibly a more important factor in determining classification accuracy than the nature of the classification algorithm that is used, especially for cleanly structured data.

Acknowledgements

Dr. Pal thanks the Association of Commonwealth Universities (ACU), London, for providing a research scholarship to fund his postgraduate research. The DAIS data used in this study were collected and processed by the DLR, Germany, and were kindly made available by Prof. J. Gumuzzio of the Autonomous University of Madrid, Spain. The School of Geography, University of Nottingham, provided research facilities. We are grateful for the suggestions of three anonymous referees, and for the comments of the editor, Dr. M.E. Bauer, all of which have led to improvements in the presentation of this paper.

References

Bennett, K. P., & Blue, J. A. (1998). A support vector machine approach to decision trees. *Proceedings of the IEEE international joint conference on neural networks, Anchorage, Alaska* (pp. 2396–2401).

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Borak, J. S., & Strahler, A. H. (1999). Feature selection and land cover classification of a MODIS-like data set for semi-arid environment. *International Journal of Remote Sensing*, *20*, 919–938.

Borgelt, C., Gebhardt, J., & Kruse, R. (1996). Concepts for probabilistic and possibilistic induction of decision trees on real world data. *Proceedings of 4th European congress on intelligent techniques and soft computing, Aachen, Germany, vol. 3* (pp. 1556–1560).

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth.

Brodley, C. E., & Utgoff, P. E. (1992). *Multivariate versus univariate decision trees*. Technical report 92-8. Department of Computer Science, University of Massachusetts, Amherst, MA, USA.

Campbell, J. B. (1981). Spatial correlation effects upon accuracy of supervised classification of land cover. *Photogrammetric Engineering and Remote Sensing*, *47*, 355–363.

Esposito, F., Malerba, D., & Semeraro, G. (1997). A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 476–491.

Foody, G. M., & Arora, M. K. (1997). An evaluation of some factors affecting the accuracy of classification by an artificial neural network. *International Journal of Remote Sensing*, *18*, 799–810.

Foody, G. M., McCulloch, M. B., & Yates, W. B. (1995). The effects of training set size and composition on artificial neural network. *Photogrammetric Engineering and Remote Sensing*, *58*, 1459–1460.

Freund, Y., & Schapire, R. E. (1996). Experiments with new boosting algorithm. *Proceedings of the thirteenth international conference on*

machine learning (ICML '96) Bari, Italy, July 3–6, 1996 (pp. 148–156). San Francisco: Morgan Kaufmann.

Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, *61*, 399–409.

Gahegan, M., & West, G. (1998). The classification of complex data sets: An operational comparison of artificial neural networks and decision tree classifiers. *Proceedings of the 3rd international conference on geocomputation, University of Bristol, UK, 17–19 September 1998*, available at http://divcom.otago.ac.nz/SIRC/GeoComp/GeoComp98/61/gc_61.htm, accessed 10 April 2003.

Gelfand, S. B., Ravishankar, C. S., & Delp, E. J. (1991). An iterative growing and pruning algorithm for classification tree design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 163–174.

Hepner, G. F., Logan, T., Ritter, N., & Bryant, N. (1980). Artificial neural network classification using a minimal training set: Comparison to conventional supervised classification. *Photogrammetric Engineering and Remote Sensing*, *56*, 469–473.

Hsieh, P., & Landgrebe, D. (1998). *Classification of high dimensional data*. Technical report—ECE 98-4. School of Electrical and Computer Engineering Purdue University, West Lafayette, IN.

Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, *14*, 55–63.

Jackson, Q., & Landgrebe, S. (2001). An adaptive classifier design for high dimensional data analysis with a limited training data set. *IEEE Transactions on Geoscience and Remote Sensing*, *39*, 2664–2679.

Kavzoglu, T. (2001). *An investigation of the design and use of feed-forward artificial neural networks in the classification of remotely sensed images*. PhD thesis, University of Nottingham, Nottingham, UK.

Kononenko, I., & Hong, J. S. (1997). Attribute selection for modelling. *Future Generation Computer Systems*, *13*, 181–195.

Landgrebe, D. (2000). On the relationship between class definition precision and classification accuracy in hyperspectral analysis. *International geoscience and remote sensing symposium, Honolulu, Hawaii, 24–28 July 2000*. Available at: http://www.ece.purdue.edu/~landgreb/IGARSS.2000.pdf, accessed 15 August 2002.

Loh, W. -Y., & Shih, Y. -S. (1997). Split selection methods for classification trees. *Statistica Sinica*, *7*, 815–840.

Mather, P. M. (1999). *Computer processing of remotely-sensed images: An introduction* (2nd ed.). Chichester: Wiley.

Mingers, J. (1989a). An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, *4*, 227–243.

Mingers, J. (1989b). An empirical comparison of selection measures for decision tree induction. *Machine Learning*, *3*, 319–342.

Muchoney, D., Borak, J., Chi, H., Friedl, M., Gopal, S., Hodges, J., Morrow, N., & Strahler, A. (2000). Application of MODIS global supervised classification model to vegetation and land cover mapping of Central America. *International Journal of Remote Sensing*, *21*, 1115–1138.

Murthy, S. K., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, *2*, 1–32.

Oates, T., & Jenson, D. (1997). The effects of training set size on decision tree complexity. *Machine learning, Proceedings of the fourteenth international conference on machine learning* (pp. 254–262). San Francisco, CA: Morgan Kaufmann.

Paola, J. D., & Schowengerdt, R. A. (1995). A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification. *IEEE Transactions on Geoscience and Remote Sensing*, *33*, 981–996.

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man–Machine Studies*, *27*, 221–234.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufmann.

Quinlan, J. R. (1996). Bagging, boosting and C4.5. *Thirteenth national conference of artificial intelligence* (pp. 725–730). Portland, OR, USA: American Association for Artificial Intelligence.

Raudys, S., & Pikelis, V. (1980). On dimensionality, sample size, classification error, and complexity of classification algorithms in pattern recognition. *IEEE Transactions on Pattern Recorgnition and Machine Intelligence*, *3*, 242–252.

Schowengerdt, R. A. (1997). *Remote sensing: Models and methods for image processing*. San Diego: Academic Press.

Swain, P. H., & Hauska, H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, *3*, 142–147.

Utgoff, P. E., & Brodley, C. E. (1990). An incremental method of finding multivariate splits for decision trees. *Machine learning, Proceedings of the seventh international conference on machine learning* (pp. 58–65). Austin, TX: Morgan Kaufmann.

Wilkinson, G. G. (1997). Open questions in neurocomputing for earth observation. *Neuro-computational in remote sensing data analysis* (pp. 3–13). Berlin: Springer-Verlag.