Paper presented at the 22nd Asian Conference on Remote Sensing, 5 - 9 November 2001, Singapore.

DECISION TREE BASED CLASSIFICATION OF REMOTELY SENSED DATA

Mahesh Pal and Paul M. Mather
School of Geography
University of Nottingham
Nottingham
U.K. NG7 2RD
pal@geography.nottingham.ac.uk
paul.mather@nottingham.ac.uk

KEY WORDS: Decision tree classifier, Boosting, Classification accuracy.

ABSTRACT Over the last decade, many applications of neural network classifiers for land use classification have been reported in the literature, but few studies have assessed the use of decision tree classifiers. These latter techniques have substantial advantages for remote sensing classification problems due to their nonparametric nature, simplicity, robustness with respect to non-linear and noisy relations among input features and class labels, and their computational efficiency. A decision tree classifier has a simple form which can be compactly stored and that efficiently classifies new data. Decision tree classifiers can perform automatic feature selection and complexity reduction, while the tree structure gives easily understandable and interpretable information regarding the predictive or generalisation ability of the data. A decision tree recursively partitions a data set into smaller subdivisions on the basis of tests applied to one or more features at each node of the tree. In this study, a decision tree classifier is used for land use classification using Landsat-7 ETM+ data for an agricultural area near Littleport (Cambridgeshire), UK, for the year 2000. A field study was carried out to collect ground truth information about the various land use classes in the study area. Six land use classes (wheat, sugar beet, potatoes, peas, onions and lettuce) are selected for classification, and a univariate decision tree classifier is used for the labelling of the image pixels. The results of this study suggest that the decision tree classifier performs well, producing an overall accuracy of about 84.5%. The boosting technique, which improves the classification accuracy of a base classifier, was applied and the classification accuracy was increased by about 2 percent to 86.5%.

1. INTRODUCTION

Classification is a method by which labels or class identifiers are attached to the pixels making up a remotely sensed image on the basis of their characteristics. These characteristics are generally measurements of their spectral response in different wavebands. They may also include other attributes (e.g., texture) or temporal signatures. This labelling process is implemented through pattern recognition procedures, the patterns being vectors of pixel characteristics. The most commonly used classification methodologies used in remote sensing are unsupervised procedures such as ISODATA and supervised methods, the most popular of which is the maximum likelihood (ML) algorithm. The ML classifier is based on a probabilistic classification procedure, which assumes that each spectral class can be adequately described or modelled by a multivariate normal probability distribution in feature space. The performance of this type of classifier thus depends on how well the data match the pre-defined model. If the data are complex in structure then to model the data in an appropriate way can become a real problem. In order to overcome this problem, which is inherent in statistical approaches, non-parametric classification techniques such as artificial neural networks (ANN) and rule-based classifiers are increasingly being used. Decision tree classifiers have, however, not been used as widely by the remote sensing community for land use classification despite their non-parametric nature and their attractive properties of simplicity, flexibility, and computational efficiency (Friedl et. al., 1997). The non-parametric property means that non-normal, non-homogenous and noisy data sets can be handled, as well as non-linear relations between features and classes, missing values, and both numeric and categorical inputs (Quinlan, 1993).

2. DECISION TREE CLASSIFIERS

In the usual approach to classification, a common set of features is used jointly in a single decision step. An alternative approach is to use a multistage or sequential hierarchical decision scheme. The basic idea involved in any multistage approach is to break up a complex decision into a union of several simpler decisions, hoping the final solution obtained in this way would resemble the intended desired solution. Hierarchical classifiers are a special type of multistage classifier that allows rejection of class labels at intermediate stages.

Classification trees offer an effective implementation of such hierarchical classifiers. Indeed, classification trees have become increasingly important due to their conceptual simplicity and computational efficiency. A decision tree classifier has a simple form which can be compactly stored and that efficiently classifies new data. Decision tree classifiers can perform automatic feature selection and complexity reduction, and their tree structure provides easily understandable and interpretable information regarding the predictive or generalisation ability of the classification.
To construct a classification tree by heuristic approach, it is assumed that a data set consisting of feature vectors and their corresponding class labels are available. The features are identified based on problem specific knowledge. The decision tree is then constructed by recursively partitioning a data set into purer, more homogenous subsets on the basis of a set of tests applied to one or more attribute values at each branch or node in the tree. This procedure involves three steps: splitting nodes, determining which nodes are terminal nodes, and assigning class label to terminal nodes. The assignment of class labels to terminal nodes is straightforward: labels are assigned based on a majority vote or a weighted vote when it is assumed that certain classes are more likely than others.

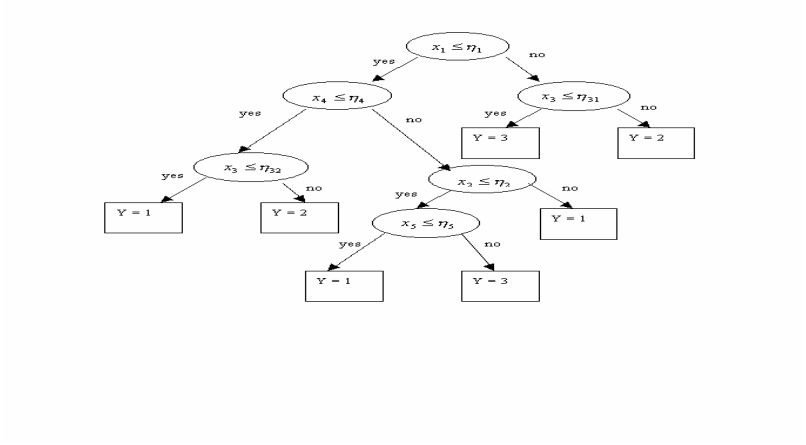


Figure 1. A classification tree for a five dimensional feature space and three classes. The x_i `s are the feature values, the $\eta_i$ `s are the thresholds, and y is the class label.

A tree is composed of a root node (containing all the data), a set of internal nodes (splits), and a set of terminal nodes (leaves). Each node in a decision tree has only one parent node and two or more descendent node (figure 1). A data set is classified by moving down the tree and sequentially subdividing it according to the decision framework defined by the tree until leaf is reached.

2.1. Pruning Decision Tree

Decision tree classifiers divide the available training data into subsets, each representing a single class. The result of this procedure is often a very large and complex tree. In most cases, fitting a decision tree until all leaves contain data for a single class may overfit to the noise in the training data, as the training samples may not be representative of the population. If the training data contain errors, then overfitting the tree to the data in this manner can lead to poor performance on unseen cases. To minimise this problem, the original tree must be pruned to reduce classification errors when data outside of the training set are to be classified.
A decision tree is not usually simplified by deleting the whole tree in favour of a leaf. Instead, parts of the tree that do not contribute to classification accuracy on unseen cases, thus producing less complex and more comprehensible tree, are removed. For details, the reader is referred to Quinlan (1993), Mingers (1989), and Brieman et al. (1984).

2.2. Boosting

In recent years, a number of works proposing the use of combining the predictions of multiple classifiers to produce a single classifier have been reported. The resulting classifier, referred to as an ensemble, is generally found to be more accurate than any of the individual classifiers making up the ensemble. For example, ensembles of neural network (Giacinto et. al., 1997) and integration of the results of different type of classifiers (Wilkinson et al., 1995) are found to be effective in improving classification accuracy. Much of this research is focused on improving the classification accuracy, as accuracy is the primary concern in all applications of learning. Only a few authors have reported the use of boosting, another technique to improve the performance of any learning algorithm (e.g., Fridl, et

al., 1999). The basic difference between the use of ensembles of classifiers and boosting is that boosting uses the same learning algorithm that consistently generates multiple classifiers in an iterative manner.
Boosting is a general method for improving the performance of any learning algorithm. Boosting can be used to reduce the error of any weak learning algorithm that consistently generate classifiers on various distributions over the training data, and then combining the classifications produced by the weak learner into a single composite classifier.

3. STUDY AREA AND DATA

The study area selected for this experiment is part of an agricultural region located near the town of Littleport in Cambridgeshire, in eastern England. Landsat-7 ETM+ data acquired on 19th June, 2000, is used. The classification problem involved the identification of six land cover types; namely, wheat, potatoes, sugar beet, onions, peas, and lettuce that cover the bulk of the area of interest.
The ERDAS Imagine image processing software (version 8.4) was used to register the images to the Ordnance Survey of Great Britain's National Grid by applying a linear transformation. The RMSE (Root Mean Square Error) values estimated for image transformations were less than one pixel. A sub-image of size 286-pixels (columns) by 386-pixels (rows) covering the area of interest was extracted for further analysis.
Field data providing information on the location of particular crop types were collected from farmers and their representative agencies supplemented by field observation. Parcel boundaries were digitised using Arc Info software and each polygon was given a label corresponding to the crop it contained. As boundary regions are often found to have the mixed pixels, a buffer zone was defined around the field boundaries and masked out of the study. A random sampling is carried out to select the pixels for training and testing the classifier.

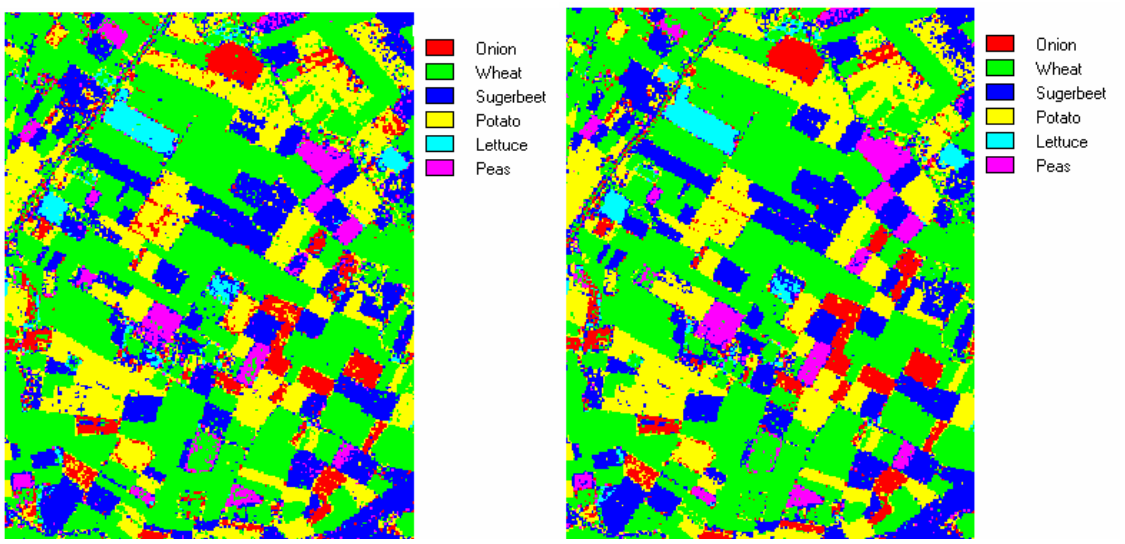
4. RESULTS AND CONCLUSIONS

A random sample of 5531 pixels was selected and divided in two parts. A total of 4181 pixels were used for training the classifier and remaining 1350 pixels, for testing the classifier. The results obtained by using decision tree classifier, without and with boosting are shown in table 1, and corresponding classified images are shown in Figure 2.

classifier	Accuracy	Kappa value
Decision tree (without boosting)	84.3	0.812
Decision tree (boosting with ten iteration)	86.7	0.840

Table 1

It is apparent that the classification accuracy increases by more than two percent following boosting. Although 2% may appear to be a small increase, it should be borne in mind that even small percentage increases are difficult to generate when the overall classification accuracy level exceeds 80%. We can, therefore, conclude that boosting is a useful technique for improving the performance of decision tree classifiers. As can be seen from the figure 2 (a), there are number of incorrectly classified pixels in several fields. After boosting the classifier, the number of incorrectly classified pixels reduces significantly, figure 2 (b), as this boosting algorithm assigns a weight to each training observation and those observations that were misclassified in the previous iteration are assigned a heavier weight in the next iteration. Thus, this algorithm forces the classification algorithm to concentrate on those observations that are more difficult to classify. This shows how boosting helps in reducing the number of misclassified pixels, thus increasing the classification accuracy. The decision tree classifier also has the advantage of ease of use. No lengthy training is required, as in the case of artificial neural networks, nor is any specific data model assumed, as in the case of statistical classifiers. However, as with all classification algorithms, the quality of the result (from a user perspective) depends crucially on the adequacy of the training data to represent the classes of interest, and on the number and nature of the chosen classes relative to the spatial scale of the imagery. The characteristics of the training data are particularly important. Sample size relative to data dimensionality and sampling design must be considered carefully, if acceptable performance is to be realised.
Same training and testing data set was also used with neural network classifier. An accuracy of 85.3% was achieved, which is slightly more than the decision tree classifier (without boosting). Further studies are being carried out to compare the usefulness of both these classifiers in land use classification, using ETM+ and other data sets.



(a) (b)

Figure 2. Classified images without boosting (a) and with boosting (b)

References

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., 1984. Classification and Regression Trees. Wadsworth , Monterey, CA.

Friedl, M.A., Brodley, C.E., and Strahler, A.H., 1999. Maximizing land cover classification accuracies produced by decision tree at continantal to global scales. IEEE Transactions on Geoscience and Remote Sensing. 37, 969-977.

Friedl, M.A., and Brodley, C.E., 1997. Decision tree classification of land cover from remotely sensed data. Remote Sensing of Environment. 61, 399-409.

Giacinto, G., and Roli, F., 1997. Ensembles of Neural Networks for soft classification of remote sensing images. Proceedings of the European Symposium on Intelligent Techniques, March 20-21, Bari, Italy, 166-170.

Wilkinson, G. G., Fierens, F., and Kanellopoulos, I., 1995. Integration of neural and statistical approaches in spatial data classification. Geographical Systems, 2, pp. 1-20.

Mingers, J., 1989. An empirical comparison of pruning methods for decision-tree induction. Machine Learning, 4(2), pp. 227-243.

Quinlan, J. R., 1993. *C4.5:* Programs for Machine Learning. Morgan Kaufmann, California.