Cost-Sensitive Decision Tree Pruning: Use of the ROC Curve

Andrew P. Bradley and Brian C. Lovell.

Cooperative Research Centre for
Sensor Signal and Information Processing,
Department of Electrical and Computer Engineering,
The University of Queensland, QLD 4072. Australia.

bradley@elec.uq.edu.au, lovell@elec.uq.edu.au

Abstract

This paper discusses a revised form of decision tree pruning that is sensitive to the relative costs of the misclassification of examples. A brief overview of existing decision tree pruning methods is given together with the rationale behind these techniques. Then, the two types of misclassifications, false negatives and false positives, are defined and related to three concepts from statistical pattern recognition: the receiver operating characteristic (ROC) curve; statistical hypothesis testing; and the Neyman-Pearson method. Details of the implementation of two cost-sensitive pruning algorithms, based on the well known Pessimistic and Minimum Error pruning techniques, are discussed. Results are then presented for both these techniques on two machine learning datasets and related to ROC curves and the Neyman-Pearson method. Thus we show that decision trees can be made to conform to specified operating criteria given in terms of the probabilities of false negatives and false positives. As a result of this analysis, it is noted that, on the data sets chosen, unequal misclassification costs actually increased the overall accuracy of the classification scheme. It is concluded that the application of the ROC curve, from statistical pattern recognition to machine learning, and to decision tree pruning in particular, can provide increased flexibility and accuracy.

Index Terms — Decision Trees, Pruning Methods, Cost-sensitive pruning, Receiver Operating Characteristic, Neyman-Pearson method.

1 Introduction

Decision tree pruning has been an active area of research for the past decade [3,6,9,11,12,13, 14], with many empirical comparisons being carried out in that time [5,10,13]. However, the idea of cost-sensitive pruning has received much less investigation [3,8,15] and, as this paper demonstrates, allows for additional flexibility and even increased performance to be obtained from a pruning strategy.

This work arises from the development a new machine learning technique, the Multiscale Classifier [2], and an investigation into pruning strategies to be used in conjunction with

it[1]. Our motivation has come from the differing misclassification costs associated with the automated diagnosis of cervical cancer. Here, the cost associated with wrongly classifying an abnormal slide as normal is high (the abnormality may go untreated), whilst, there is a lesser cost associated with wrongly predicting a normal slide as abnormal (the slide has to be screened again by a Cytologist). The need to control misclassifications of this type is, in general, important in machine learning and the use of cost-sensitive pruning is a logical way to do this in decision trees.

1.1 Background

Decision trees are constructive learning algorithms in that they generate new nodes that partition feature space to classify the training data. Tree growth terminates either when all of the training data is correctly classified or when it is decided that no extra information can be obtained by adding nodes to the tree. When the training data is noisy or uncertain the resulting decision trees tend to be very large, as nodes are created for very few training examples. This is known as "over-specialisation," since the decision tree usually has poor performance on new, previously unseen data [3].

There are two solutions to the problem of over-specialisation, either tree growth is terminated before it over-specialises, called (*pre-pruning*), or the over-specialised tree is pruned to remove error prone nodes, called (*post-pruning*). *Pre-pruning* will not add a branch when it adds no information, however, subsequently the branch may be found useful when combined with another. *Post-pruning*, does not suffer from this problem of localised information and has been shown, in general, to be superior [3]. Tree Pruning is of use in data domains that represent relatively simple underlying relationships [16] and are not deterministic [10]. Most "Real World" data sets are likely to be of this type [7] and therefore pruning of some kind will, in general, be required.

2 Decision Tree Pruning

The idea behind decision tree pruning (*post-pruning*) is to identify the least reliable nodes of the tree and remove them. This produces a simplified tree that reflects the underlying structure of the data. Pruning will increase the number of classification errors on the training set, but will in general, improve classification accuracy on new, unseen, data.

Pruning methodologies can be separated into two groups, those that estimate the probability of misclassification of a subtree, and therefore which leaves to prune, using an independent test set *e.g.,* Error-Complexity Pruning [3], Critical Value Pruning [9], Reduced Error Pruning [13], and Iterative Growing and Pruning [6], and those that prune only on the information gained when the tree was constructed, *i.e.,* from the training data *e.g.,* Pessimistic Pruning [12], and Minimum Error Pruning [11]. Quinlan [13] has also examined a method of simplifying decision trees by translating the tree structure into a single level rule set, or *decision list*, removing irrelevant input terms from these rules. These *production rules* are then in a form more amenable to human comprehension than standard decision trees.

By partitioning the data into a training set and pruning set, the accuracy of the classifier may be reduced because of the reduced amount of training data. Although, cross-validation [3] or iterative growing and pruning [6] can be used to alleviate this problem, it will slow

[1]The current Microsoft Windows executable version of The Multiscale Classifier is available via anonymous ftp (ftp.cssip.elec.uq.oz.au:pub/cssip/software/msc).

the whole training process down. The error estimates obtained using an independent test set are unbiased and, if the test set is large enough, reliable. However, the error estimates obtained using only the training data are biased[2]. Statistical correction techniques can, and are, used to remove the bias from this error estimate. A pruned tree will have decision nodes (leaves) that contain training examples of more than one class. So, instead of a class being associated with a leaf, the leaf now has a *class distribution* is associated with it. This means that classifications now have a level of certainty associated with them depending on this class distribution.

2.1 Pruning and the Multiscale Classifier

The Multiscale Classifier (MSC) [2] is an incremental decision tree construction algorithm. It works by successively splitting feature space in half, using finer levels of resolution as required to separate points in decision space. It is termed a *consistent* learning algorithm, in that, it constructs a tree to fit the training data exactly. Therefore, a pruning strategy is of vital importance if the MSC is going to be consistent with Bayes risk classification [20]. The MSC also has a non-binary tree structure, decisions being made on more than one attribute at each branch. Each branch may lead to, up to 2^N leaves, where N is the number of input attributes.

Despite these differences from conventional decision trees [3,14], Mingers's finding, that there is no significant relationship between tree creation method and pruning performance [10], means that the pruning techniques already described and empirically tested in the literature can be applied to the MSC. Conversely, the results reported here, will also apply to other decision tree construction algorithms. The methodology of cost-sensitive decision tree pruning being completely general. For this work, we chose to implement the revised forms of Pessimistic [14] and Minimum error [4] pruning, because they do not require a separate test set, are simple to implement, and are computationally light. Also, in comparative studies they were found to perform at least as well as the other techniques [5,10,13]. Cost-sensitive decision tree construction cannot easily be applied to the MSC because it uses a binary rather than a statistical splitting criterion at the tree construction stage [2].

3 Cost-Sensitive Pruning

When pruning a decision tree the expected error rate of a subtree is calculated with the assumption that all the classes are equally probable and equally important. Minimum Error pruning was extended to take into account different *a priori* class probabilities [4], estimated from the distribution of the training data. However, in most "real world" classification problems there is also a *cost* associated with misclassifying examples from each class, C_c.

For binary classification problems these errors are called *false positives* and *false negatives*. In the field of statistical hypothesis testing they are referred to as *Type I* and *Type II* errors [21]. A Type I error is the rejection of the null hypothesis when it is true and a Type II error is the acceptance of the null hypothesis when it is false. The analogous false positive is a classification of positive given to an example that is actually negative, and a false negative is the negative classification of an example that is actually positive. They are defined as follows:

$$P(\text{False Positive}) = P(\text{Classify Positive} \mid \text{Negative}), \tag{1}$$

$$P(\text{False Negative}) = P(\text{Classify Negative} \mid \text{Positive}). \tag{2}$$

[2] As pruning always reduces classification accuracy on the training data.

In general the probability of a false positive, (denoted α), is referred to as the *level of significance* of a test, and the probability of a false negative, (denoted β), the *power* of a test. Two related accuracy measures are then the *sensitivity*, $(1-\beta)$, and the *specificity*, $(1-\alpha)$ of a test. In signal detection theory the plot of α versus $1-\beta$ is known as the "Receiver Operating Characteristic" or ROC curve [17]. In this domain it is used to measure how well a receiver can detect signal from noise, but in general it measures the ability to classify positive from negative examples. In statistical pattern recognition, this method of treating the two types of errors separately is called the Neyman-Pearson method [20]. Here, we fix one of the class error probabilities, usually from a performance specification, and then minimize the other class error probability with this constraint. Because of the difficulties in solving this constrained minimization, a practical approach to the Neyman-Pearson criterion is to vary the decision threshold between two class distributions and plot the locus of the points obtained for α and $1-\beta$ *i.e.,* a ROC curve, choosing an operational point from that.

Treating the two types of errors separately means that we can associate a *cost* with each type of error. Normally, a classification scheme will be specified in terms of an operational point that defines limits on both types of errors, and therefore minimizes the overall cost [22]. This operational point can then be extended to an operational characteristic that gives a system a number of different operational modes. Again, in the case of screening for cervical cancer, an automated screener can be used in one of two roles: as a primary screener, which initially screens *all* of the incoming slides, with $\alpha \approx 50\%$, $\beta \ll 1\%$; and as a quality control checker which "double checks" only those slides conventionally screened and passed as normal, with $\alpha < 10\%$, $\beta \approx 5\%$ [1].

3.1 Implementation

We have already shown why it is important to take relative misclassification costs into consideration when deciding which leaves of a decision tree should be removed. So, the expected error rate, E_c, as determined using the Minimum error [4] or Pessimistic [14] pruning, should be weighted by a measure of the misclassification cost for that class, C_c.

$$EK_c = E_c \cdot C_c \tag{3}$$

This concept is directly related related to Error-Complexity Pruning [3] where the actual *size* of the decision tree is used as an additional cost in the pruning strategy. It should be noted that the Pessimistic pruning algorithm has to be slightly modified, this is because the class of a replacement node for a subtree is no longer always the most frequently occurring class, but is the class with the smallest weighted error estimate (EK_c). This condition is already the case for the Minimum Error algorithm, where the *a priori* class probabilities have a similar effect on the error estimates.

The error estimates of a subtree for each of the pruning strategies are now weighted using the misclassification cost associated with each class. So, for Pessimistic pruning, $U_{CF}(E,N)$ is upper limit on the probability of error of a subtree, with confidence level CF[3]. For a subtree classifying N cases correctly and E wrongly, the cost-sensitive error is given by:

$$EK_c = U_{CF}(E,N) \cdot C_c, \tag{4}$$

For Minimum Error pruning, for a subtree covering N examples, of which n_c are in class c,

[3] Based on the Binomial distribution.

and class c having an *a priori* probability of p_{ac}, the cost-sensitive subtree error is given by:

$$EK_c = \frac{N - n_c + (1 - p_{ac})m}{N + m} \cdot C_c. \tag{5}$$

Here, the parameter m controls the amount of pruning that takes place and is related to the amount of noise in the data domain. Similar cost-sensitive pruning equations for the original Minimum Error, and Pessimistic algorithms, plus a number of other pruning strategies can be found in Knoll *et al* [8]. In addition Breiman, *et al* show the use of misclassification costs both when constructing and pruning CART decision trees [3].

4 Experimental Work

In this paper we wish to demonstrate that cost-sensitive pruning of decision trees can be used to control the number of false positive and false negative classifications. Thus providing the required classification modes and additional flexibility. It should be noted that we are assuming an operational point will be specified prior to construction of the decision tree. The ROC curve being used to find suitable values of relative misclassification cost to position the pruned decision tree at, or near to, the specified operational point. The two data domains chosen for this study were:

1. BREAST: Diagnosing breast cancer [23]. There are 9 integer inputs, each with a value between 1 and 10. The two output classes, benign and malignant, are non-uniformly distributed (65.5% and 34.5% respectively).
2. PIMA: Diagnosing diabetes among female Pima Indians [19]. There are 8 continuously valued inputs, 2 non-uniformly distributed output classes (65.1% and 34.9%) and a total of 768 data points.

The actual datasets and further information regarding them can be obtained from a machine-readable data repository at the University of California, Department of Information and Computer Science (ftp:ics.uci.edu:pub/machine-learning-databases).

4.1 Experimental Procedure

The experimental procedure used consisted of the following steps:

1. Randomly partition the data into two sets: a training set, comprising 70% of the data; and a test set containing the remaining 30%. This partitioning is done so that the class distributions in each set remained approximately equal [22].

   Note: For the PIMA domain the input data contained continuous attributes that, for MSC, required scaling to the range $[0, 1)$. This was done by a linear transformation using the maximum and minimum values of each attribute as follows:

   $$y = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{6}$$

2. Train the MSC on each training set in turn until 100% accuracy is achieved on the training set, then store the decision tree.

3. Use Minimum error pruning on the decision tree with the following, approximately logarithmic scale, relative misclassification costs (shown as false negative:false positive):
   (1:8,1:4,1:2,1:1.5,1:1.25,1:1,1.25:1,1.5:1,2:1,4:1,8:1,16:1,32:1).
   After pruning with each cost, note the number of false positive and false negative classifications.
4. Repeat steps 1 to 3, 9 more times and average the results for both datasets. Repeat this process for Pessimistic pruning.

Both Minimum Error and Pessimistic pruning require the specification of a pruning parameter, m and the confidence level, cf, respectively. Default values of $m = 8$ and $cf = 1$ were used in all cases.

5 Results

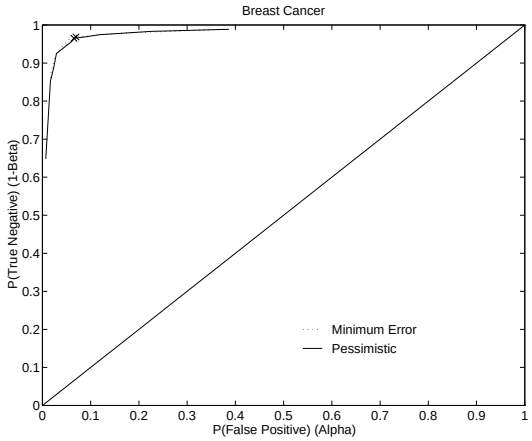

Figure 1: ROC curve for the breast cancer data.

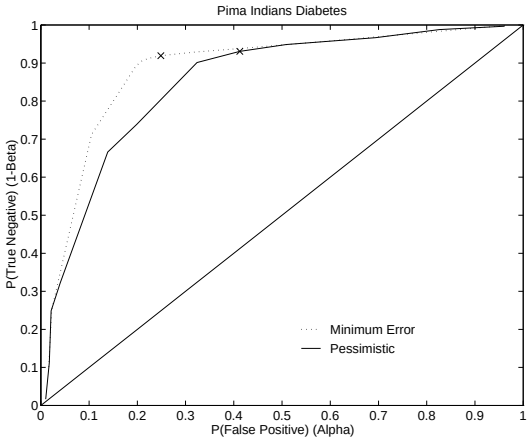

Figure 2: ROC curve for the Pima Indian's diabetes data.

Figures 1 and 2 show ROC curves for the breast cancer and Pima diabetes data sets respectively. The diagonal line from (0,0) to (1,1) shows the point of no discrimination between the two classes, *i.e.,* where the probability of a true positive $(1 - \beta)$ equals the probability of a false positive (α). The location of equal misclassification cost is shown on the curves with an 'x'.

6 Discussion

Figures 1 and 2 show that as the costs were varied the position of the classifier on the ROC curve varied. Though our method of constructing a ROC curve is based on predefined variations in misclassification cost, systematic approaches have been proposed [15] that reduce computational complexity. If one class has a high relative cost (greater than 16:1), the decision tree reduces to a single leaf of the class with the lowest cost. The resultant classifier is not of any practical use, but the reduction is sensible, since it indicates that it is too costly to predict any example as being from the high cost class.

It should be noted that the classifier can only be positioned at discrete points on the ROC curve. This is not due to the discrete steps in the costs used when pruning, but to the pruning methodology itself. When the decision tree is pruned, a subtree is reduced to a single leaf of the class with the minimum error. This single leaf can then subsequently lead to a number of false positive or negative classifications, so, the error rises in discrete steps. In the case of the non-binary MSC tree this effect could be reduced by further revising the pruning strategies to allow for partial removal of subtrees, *i.e.,* by allowing some of the leaves on a subtree to be merged. In this way a subtree would be replaced by a smaller subtree rather than by a single leaf. This strategy may also increase the overall accuracy of pruning, as erroneous single leaves can now be removed.

The point of perfect classification on the ROC curve is at $\alpha = 0, 1-\beta = 1$, the lowest overall error rate being obtained when the classifier is positioned as near to this corner as possible. Figures 1 and 2 show that equal costs, as indicated by the 'x', do *not* necessarily provide the lowest overall probability of error. In fact, for both pruning methods on both datasets, a slightly higher ($\approx$ 1:1.5) false positive cost actually increased the overall accuracy. However, as we have already discussed, overall accuracy is not, on its own, a good measure of performance. Measures such as sensitivity, specificity, and overall misclassification cost are more preferable [3,22]. This paper recommends that for the full evaluation of a machine learning technique, a ROC curve should always be plotted. Then, if a "single number" evaluation is required, a measure such as the area under the ROC curve [18], or the χ^2 value from the confusion matrix [21] should be used.

7 Conclusions

We have expanded Minimum Error and Pessimistic decision tree pruning so that they are sensitive to the cost of misclassifying examples from different classes. We have demonstrated the need for this type of pruning in order to produce different classification modes in decision trees. We have also shown how this information may be analysed using the ROC curve, a technique previously used in statistical pattern recognition. We recommended that ROC curves become more widely used in machine learning.

8 Acknowledgements

The authors are grateful to Alvin Mok for his work in taking initial research code and turning it into a user-friendly, well documented software package, and to Ross Burnett for his initial work in this area.

9 References

[1] T. L. Anderson, "Automatic Screening of Conventional Papanicolaou Smears," in *Compendium on the Computerized Cytology and histology laboratory*, G. L. Weid, P. H. Bartels, D. L. Rosenthal and U. Schenck, Eds. pp. 306–311, 1994c.

[2] A. P. Bradley and B. C. Lovell, "Inductive learning using multiscale classification," in *Proceedings of the Fifth Australian Conference on Neural Networks* . Brisbane, Australia: pp. 133–136, February 1994a.

[3] L. Breiman, J. Friedman, R. Olshen and C. Stone, in *Classification and Regression Trees*. Belmont: Wadsworth, 1984.

[4] B. Cestnik and I. Bratko, "On Estimating Probabilities in Tree Pruning," in *Machine Learning: EWSL-91: European Working Session on Learning*, Y. Kodratoff, Ed. Porto, Portugal: Lecture notes in Artificial Intelligence: 482, pp. 138–150, 1991.

[5] F. Esposito, D. Malerba and G. Semeraro, "Tree Pruning as a Search in the State Space," in *Machine Learning: Proceedings of ECML-93*. pp. 165–184, 1993.

[6] S. B. Gelfand, C. S. Ravishankar and E. J. Delp, "An Iterative Growing and Pruning Algorithm for Classification Tree Design," *PAMI*, 13, no. 2, pp. 163–174, 1991.

[7] R. C. Holte, "Very simple Classification Rules Perform Well on Most Commonly Used Datasets," *Machine Learning*, Vol. 11, pp. 63–91, 1993.

[8] U. Knoll, G. Nakhaeizadeh and B. Tausend, "Cost Sensitive Pruning of Decision Trees ," in *Machine Learning: Proceedings of ECML-94*. pp. 383–386, 1994.

[9] J. Mingers, "Expert Systems–rule induction with statistical data," *Journal of the operational research society*, Vol. 38, pp. 39–47, 1987.

[10] J. Mingers, "An empirical comparison of pruning methods for decision tree induction," *Machine Learning*, Vol. 4, pp. 227–243, 1989a.

[11] T. Niblett and I. Bratko, "Learning Decision Rules in Noisy Domains," in *Proceedings of Expert Systems 86 Conference*. Cambridge University Press, 1986.

[12] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, Vol. 1, pp. 81–106, 1986 .

[13] J. R. Quinlan, "Simplifying decision trees," *International journal of man-machine studies*, Vol 27, pp. 221–234, 1987.

[14] J. R. Quinlan, in *C4.5:Programs for Machine Learning*. San Mateo: Morgan Kaufmann, 1993.

[15] R. F. Raubertas, L. E. Rodewald, S. G. Humiston and P. G. Szilagyi, "ROC Curves for Classification Trees," *Medical Decision Making*, 14, no. 2, pp. 169–174, 1994.

[16] C. Schaffer, "Overfitting Avoidance as bias," *Machine Learning*, Vol. 10, pp. 153–178, 1993a.

[17] I. Selin, in *Detection Theory*. Princeton University Press, 1965.

[18] E. M. Sherwood, P. H. Bartels and G. L. Wied, "Feature selection in cell image analysis: use of the ROC curve," *Acta Cytol.*, Vol. 20, No 3, pp. 255–261, 1976.

[19] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler and R. S. Johannes, "Using the (ADAP) Learning Algorithm to Forecast the Onset of Diabetes Mellitus," in *Proceedings of the Symposium on Computer Applications and Medical Care*. IEEE Computer Society Press, pp. 261–265, 1988.

[20] C. W. Therrien, in *Decision Estimation and Classification: an introduction to pattern recognition and related topics*. John Wiley and Sons, 1989.

[21] R. E. Walpole and R. H. Myers, in *Probability and Statistics for Engineers and Scientists*. New York: Macmillan, 1990.

[22] S. M. Weiss and C. A. Kulikowski, in *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems*. San Mateo: Morgan Kaufmann, 1991.

[23] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," in *Proceedings of the National Academy of Sciences, U.S.A.*, vol. Vol. 87, no. 12. pp. 9193–9196, 1990.