Appears in ECML-98 as a research note

Pruning Decision Trees with Misclassification Costs

Jeffrey P. Bradford[1] Clayton Kunz[2]
Ron Kohavi[2] Cliff Brunk[2] Carla E. Brodley[1]

[1] School of Electrical Engineering
Purdue University
West Lafayette, IN 47907
{jbradfor,brodley}@ecn.purdue.edu

[2] Data Mining and Visualization
Silicon Graphics, Inc.
2011 N. Shoreline Blvd.
Mountain View, CA 94043
{clayk,ronnyk,brunk}@engr.sgi.com

Abstract. We describe an experimental study of pruning methods for decision tree classifiers when the goal is minimizing *loss* rather than *error*. In addition to two common methods for error minimization, CART's cost-complexity pruning and C4.5's error-based pruning, we study the extension of cost-complexity pruning to loss and one pruning variant based on the Laplace correction. We perform an empirical comparison of these methods and evaluate them with respect to loss. We found that applying the Laplace correction to estimate the probability distributions at the leaves was beneficial to all pruning methods. Unlike in error minimization, and somewhat surprisingly, performing no pruning led to results that were on par with other methods in terms of the evaluation criteria. The main advantage of pruning was in the reduction of the decision tree size, sometimes by a factor of ten. While no method dominated others on all datasets, even for the same domain different pruning mechanisms are better for different loss matrices.

1 Pruning Decision Trees

Decision trees are a widely used symbolic modeling technique for classification tasks in machine learning. The most common approach to constructing decision tree classifiers is to grow a full tree and prune it back. Pruning is desirable because the tree that is grown may overfit the data by inferring more structure than is justified by the training set. Specifically, if there are no conflicting instances, the training set error of a fully built tree is zero, while the true error is likely to be larger. To combat this overfitting problem, the tree is pruned back with the goal of identifying the tree with the lowest error rate on previously unobserved instances, breaking ties in favor of smaller trees (Breiman, Friedman, Olshen & Stone 1984, Quinlan 1993). Several pruning methods have been introduced in the literature, including cost-complexity pruning, reduced error pruning, pessimistic pruning, error-based pruning, penalty pruning, and MDL pruning. Historically, most pruning algorithms have been developed to minimize the expected *error* *rate* of the decision tree, assuming that classification errors have the same unit cost.

Our objective in this paper is different than the above-mentioned studies. Instead of pruning to minimize *error*, we aim to study pruning algorithms with the goal of minimizing *loss*. In many practical applications one has a *loss matrix* associated with classification errors (Turney 1997), and pruning should be performed with respect to the loss matrix. Pruning for loss minimization can lead to different pruning behavior than does pruning for error minimization. In this paper, we investigate the behavior of several pruning algorithms. In addition to the two most common methods for error minimization, cost-complexity pruning (Breiman et al. 1984) and error-based pruning (Quinlan 1993), we study the extension of cost-complexity pruning to loss and a pruning variant based on the Laplace correction (Good 1965, Cestnik 1990). We perform an empirical comparison of these methods and evaluate them with respect to loss under two different matrices. We found that even for the same domain, different pruning mechanisms are better for different loss matrices. In addition, we found that adjusting the probability distributions at the leaves using the Laplace correction was beneficial to all methods.

2 The Pruning Algorithms and Evaluation Criteria

Most pruning algorithms perform a post-order traversal of the tree, replacing a subtree by a single leaf node when the estimated error of the leaf replacing the subtree is lower than that of the subtree. The crux of the problem is to find an *honest* estimate of error (Breiman et al. 1984), which is defined as one that is not overly optimistic for a tree that was built to minimize errors in the first place. The resubstitution error (error rate on the training set) does not provide a suitable estimate because a leaf-node replacing a subtree will never have fewer errors on the training set than the subtree. Two commonly used pruning algorithms for error minimization are C4.5's error-based pruning (Quinlan 1993) and CART's cost-complexity pruning (Breiman et al. 1984).

We attempted to extend several error-based pruning to loss-based pruning. In some cases the extensions are obvious, but C4.5's error-based pruning based on confidence intervals does not extend easily. The naive idea of computing a confidence interval for each probability and computing the losses based on the upper bound of the interval for each class yields a distribution that does not add to one. Experimental results we made on some variants (e.g. normalizing the probabilities) did not perform well. Instead, we decided to use a Laplace-based pruning method.

The Laplace-based pruning method we introduce here has a similar motivation to C4.5's error-based pruning. The Laplace correction method biases the probability towards a uniform distribution. Specifically, if a node has m instances, c of which are from a given class, in a k-class problem, the probability assigned to the class is $(c+1)/(m+k)$ (Good 1965, Cestnik 1990). The Laplace correction makes the distribution at the leaves more uniform and less extreme.

Given a node, we can compute the expected loss using the loss matrix. The expected loss of a subtree is the sum of expected loss of the leaves.

The cost-complexity-pruning (CCP) algorithm used in CART penalizes the estimated error based on the subtree size. Specifically, the error estimate assigned to a subtree is the resubstitution error plus a factor α times the subtree size. An efficient search algorithm can be used to compute all the distinct α values that change the tree size and the parameter is chosen to minimize the error on a holdout sample or using cross-validation. Once the optimal value of α is found, the entire training set is used to grow the tree and it is pruned using this optimal value. In our experiments, we have used the holdout method, holding back 20% of the training set to estimate the best α parameter.

Cost complexity pruning extends naturally to loss matrices. Instead of estimating the error of a subtree, we estimate its loss (or cost), using the resubstitution loss and penalizing by the size of the tree times the α factor as in error-based CCP.

3 A Comparison of Pruning Algorithms

Our goal in designing these experiments was to understand which pruning methods work well when the decision tree classifier is evaluated on loss given a loss matrix. The basic decision tree growing algorithm is implemented in $\mathcal{MLC}$++ (Kohavi, Sommerfield & Dougherty 1996) and called MC4 ($\mathcal{MLC}$++ C4.5). It is a Top-Down Decision Tree induction algorithm very similar to C4.5. The algorithm grows the decision tree following the standard methodology of choosing the best attribute according to the gain-ratio evaluation criterion and stopping when a node has two or fewer instances. The trees are pruned using the following pruning algorithms:

- **eb-fr** Error-based pruning (C4.5) with probabilities estimated using frequency counts.
- **eb-lc** Error-based pruning with probabilities estimated using the Laplace correction.
- **np-lc** No-pruning with probabilities estimated using the Laplace correction.
- **lp** Laplace-based pruning with probabilities estimated using the Laplace correction.
- **ccp-lc** Cost-complexity pruning based on loss with probabilities estimated using the Laplace correction.

The leaves of the trees are labeled with the class that minimizes expected loss based on the probability estimates at each leaf. In our initial experiments, the Laplace correction outperformed frequency counts in all variants. Therefore, excluding the basic method of error-based-pruning, all other pruning methods were run with the Laplace correction.

Ten datasets were chosen from the UCI repository (Merz & Murphy 1997): adult (salary classification based on census bureau data), breast cancer diagnosis, chess, crx (credit), german (credit), pima diabetes, road (dirt), satellite images, shuttle, and vehicle. In choosing the datasets, we decided on the following desiderata:

1. Datasets should be two-class to make the evaluation easier. This desideratum was hard to satisfy and we resorted to converting several multi-class problems into two-class problems by choosing the least prevalent class as the goal class.
2. Datasets should not have too many unknowns. To avoid another factor in this evaluation, we removed all instances with unknown values from the files.
3. The standard error of the estimated loss should be small. This was very important because with loss matrices the standard deviations of the estimates can be large. We therefore decided to require at least 500 instances and train on only 25% of the data, leaving the remaining instances for testing.

We wanted to test the following hypotheses:

1. The Laplace correction for estimating probabilities at the leaves leads to lower loss than frequency counts.
2. Considering the loss matrix during pruning leads to lower loss than pruning based on errors.

For all datasets we trained on 25% of the data and tested on 75% of the data, repeating the process 10 times. We compared performance of the pruning algorithms on two different loss matrices, which respectively set a loss of 10 and 100 for misclassifying the less frequent of the two classes. This was done to simulate real-world scenarios in which the less frequent class is the important class. Experiments were also done with the losses reversed, with similar conclusions to those shown below.

The results are displayed as graphs showing the average loss for the ten files as bars using the scale on the left, and the average relative loss as X-symbols with the scale on the right. The relative losses are computed as the ratio between the loss of the pruning method and eb-fr, our baseline method. These ratios are then averaged across the ten datasets to create summary graphs. In cases for which the losses are small, the ratio is a better indicator of performance. The average losses and average relative losses for the two loss matrices are shown in Figure 1. The following observations can be made:

1. Error-based pruning with frequency counts performs the worst.
2. The Laplace-based pruning (lp) performs the best on the 10 to 1 loss matrix and is comparable to the best on the 100 to 1 loss matrix.
3. No-pruning (np-lc) performs surprisingly well on both loss matrices!
4. Cost-complexity pruning (ccp-lc) is slightly inferior to no-pruning, but better than error-based pruning (eb) on the 100 to 1 loss matrix.
5. Tree sizes were radically different. The average tree sizes for the 10 to 1 loss matrix are: ccp(47), eb(118), lp(382), and np(670). Cost-complexity pruning was by far the smallest, confirming the observation by Oates & Jensen (1997) for error minimization.

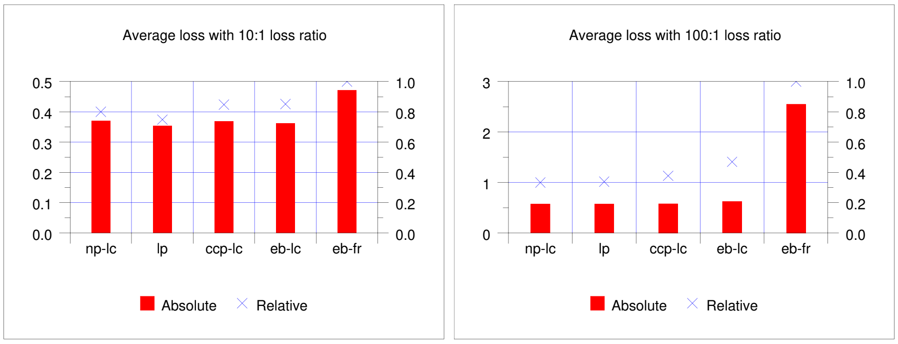

Fig. 1. Average absolute and relative losses for the different algorithms and for a 10 to 1 loss matrix (left) and a 100 to 1 matrix (right).

Our hypothesis that the Laplace correction for estimating probabilities at the leaves outperforms frequency counts was confirmed. It was also confirmed for the np and ccp pruning methods when they were run with frequency counts (results not shown). Interestingly, no-pruning performed *very* well, suggesting that when we have loss matrices and when tree size is not important, pruning need not be done if the Laplace correction is used. This result differs from error minimization, where pruning was consistently shown to help.

Pruning based on loss matrices performed better than pruning based on error for frequency counts for all methods. This result (for frequency counts) has been observed previously for reduced error/cost pruning (Draper, Brodley & Utgoff 1994). When the Laplace correction was used, pruning with loss matrices performed better than error-based pruning (eb-lc) for the 100:1 (ccp-lc, lp) but there was no significant difference for the 10:1 loss matrix.

For each pruning method, applying the Laplace correction improved performance on average. Only in a few cases did the Laplace correction lead to a higher distribution MSE (mean-squared-error) than frequency counts. The distribution MSE was similar for all the Laplace correction algorithms. The main difference between the pruning algorithms was in the tree size. The average tree sizes were ccp(27), eb(118), lp(280), and np(670).)

4 Conclusions

Of the two steps in inducing a decision tree—growing and pruning—we concentrated only on the latter stage. We view this as a good first step to study before studying different growing techniques as was done in Pazzani, Merz, Murphy, Ali, Hume & Brunk (1994).

We extended cost-complexity pruning to loss and introduced a new method that can be used with loss matrices, Laplace-pruning. Laplace-pruning was the best pruning method with the 10 to 1 loss matrix and tied for best pruning with no-pruning with the Laplace correction for the 100 to 1 loss matrix.

Our study revealed that using the Laplace correction at the leaves is extremely beneficial and aids all pruning methods used. We also found that for the datasets tested, pruning did not help much in reducing the loss, but did lead to smaller trees. Cost-complexity pruning was especially effective at reducing the tree size without significantly increasing the loss.

No single pruning algorithm dominated over all datasets in terms of loss, but more interestingly, even for a fixed domain, different pruning algorithms were better for different loss matrices. In the long version of this paper (Bradford, Kunz, Kohavi, Brunk & Brodley 1998) we showed ROC curves for different algorithms, including another pruning method. These differences, however, were not major. Given the fact that there was little difference in loss even for algorithms that did not use the loss matrix during tree pruning stage, we conclude that it will usually suffice to induce a single probability tree and use it with different loss matrices.

References

Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C. & Brodley, C. E. (1998), Pruning decision trees with misclassification costs (long).
`http://robotics.stanford.edu/~ronnyk/prune-long.ps.gz`.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Wadsworth International Group.

Cestnik, B. (1990), Estimating probabilities: A crucial task in machine learning, *in* L. C. Aiello, ed., 'Proceedings of the ninth European Conference on Artificial Intelligence', pp. 147–149.

Draper, B. A., Brodley, C. E. & Utgoff, P. E. (1994), 'Goal-directed classification using linear machine decision trees', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(9), 888–893.

Good, I. J. (1965), *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, M.I.T. Press.

Kohavi, R., Sommerfield, D. & Dougherty, J. (1996), Data mining using $\mathcal{MLC}$++: A machine learning library in C++, *in* 'Tools with Artificial Intelligence', IEEE Computer Society Press, pp. 234–245.
`http://www.sgi.com/Technology/mlc`.

Merz, C. J. & Murphy, P. M. (1997), UCI repository of machine learning databases.
`http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Oates, T. & Jensen, D. (1997), The effects of training set size on decision tree complexity, *in* D. Fisher, ed., 'Machine Learning: Proceedings of the Fourteenth International Conference', Morgan Kaufmann, pp. 254–262.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. & Brunk, C. (1994), Reducing misclassification costs, *in* 'Machine Learning: Proceedings of the Eleventh International Conference', Morgan Kaufmann.

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

Turney, P. (1997), Cost-sensitive learning.
`http://ai.iit.nrc.ca/bibliographies/cost-sensitive.html`.