Robust Linear Discriminant Trees

Extended Abstract submitted to Artificial Intelligence & Statistics

George H. John
Computer Science Department
Stanford University
Stanford, CA 94305
gjohn@cs.Stanford.EDU

June 30, 1994

Category: Learning From Data
Keywords: Decision Trees, Learning Algorithms, Regularization

1 Introduction

We present a new method for the induction of classification trees with linear discriminants as the partitioning function at each internal node. This paper presents two main contributions: 1) a novel objective function called *soft entropy*, which is used to identify optimal coefficients for the linear discriminants , and 2) a novel, simple method for removing outliers called *iterative re-filtering*, which boosts performance on many datasets. These two ideas are presented in the context of a single learning system called DT-SEPIR. In comparisons with similar decision tree algorithms and a neural net algorithm, DT-SEPIR exhibits superior performance.

2 Building Trees

The tree-building algorithm is the common recursive partitioning algorithm – take the whole set of data, partition it somehow into two subsets, and continue partitioning recursively until all subsets are *pure*, that is, all samples in the subset come from the same class. Each node in our classification tree is a linear discriminant, parametrized by a vector of coefficients $\vec{\beta}$. We note that the problem of finding the "best" discriminant is simply a function optimization problem: $\vec{\beta}^* = \text{argmin}_{\vec{\beta}} f(\vec{\beta})$, where $\vec{\beta} \in \mathbb{R}^n, f : \mathbb{R}^n \mapsto \mathbb{R}$.

Optimization problems may be described by the domain, the objective function which evaluates elements of the domain, and a method of using the objective function to find the optimal element within the domain. The domain $\mathbb{R}^n$ is given above and the function optimization method we use is just a version of steepest descent without line search.

2.1 Motivation for Soft Entropy

Our objective function is the soft entropy function, a new continuous approximation to the entropy criterion used previously in univariate decision tree induction. Previous work in multivariate decision trees has used off-the-shelf training algorithms for "perceptrons" (linear discriminants), thus minimizing a mean squared error criterion(Brodley & Utgoff 1992). Soft entropy is to be preferred over MSE for several reasons. Figure 1 shows a dataset and the linear discriminants minimizing MSE and soft entropy. Clearly the objective of partitioning in a decision tree is not to minimize error immediately (which is what the MSE solution gives) but rather to divide the dataset in such a way that a small tree is found which minimizes errors. We would expect entropy to perform better in this role than MSE.

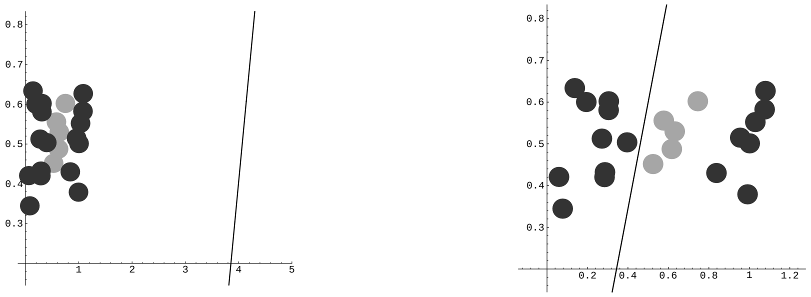

Figure 1: Left: The linear discriminant that minimizes MSE. Right: The linear discriminant that minimizes soft entropy.

Figure 2: All four splits shown on the left have equivalent entropy. On the right, the split that minimizes soft entropy is shown.

The entropy criterion is described in Morgan & Messenger (1973). Soft entropy is quite similar, but is to be preferred in continuous spaces. Figure 2 shows several candidate splits of a simple dataset. All of the splits shown have equivalent entropy, but clearly the hyperplane that divides the space down the middle is preferred. Thus we believe that in continuous spaces a good splitting criterion requires more granularity than entropy provides. The GINI criterion used in linear splits in CART (Breiman, Friedman, Olshen & Stone 1984) also suffers from this problem. A further advantage is that soft entropy is differentiable and thus amenable to the efficient use of a wider variety of function optimizers.

Additionally, we prefer the function optimization approach to less direct methods relying on normality and homoscedasticity, such as the methods in Loh & Vanichsetakul (1988). The liberation from such assumptions seems worth the computational price of the function optimization approach.

2.2 Definition of Soft Entropy

Soft entropy is a significant but simple modification of entropy, so we begin by discussing entropy and then show how it is modified to give the soft entropy criterion. We are concerned with $E(\vec{\beta})$, the expected mutual information of the class label and the partition described by the linear discriminant wiht coefficients $\vec{\beta}$.

$$E(\vec{\beta}) = \sum_{i \in \{l,r\}} \frac{p_i + n_i}{p + n} I(p_i, n_i) \tag{1}$$

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \tag{2}$$

p_i and n_i refer to the number of positive and negative instances in the left ($i = l$) or right ($i = r$) subsets[1]. p and n are similarly the number of positive and negative instances in the entire dataset to be split.

We *soften* entropy by allowing an instance to be partially a member of both the left and right subset of a split.

$$discr(\vec{x}) = \frac{1}{1 + e^{-\vec{\beta}.\vec{x}}} \tag{3}$$

[1]To simplify the presentation we will only discuss binary classification tasks, but extension to multiple classes is straightforward.

Table 1: Test set accuracy and concept size (number of nodes) results on several datasets.

	Breast	Vote	Monks1	Monks2	Monks3
DT-SEPIR	**99.3 (3)**	98.5 (3)	**100 (7)**	**100 (5)**	91.7 (5)
DT-SEP	95.6 (7)	**99.2 (5)**	**100 (7)**	**100 (5)**	91.7 (5)
DT	95.6 (13)	97.8 (13)	**100 (7)**	**100 (5)**	93.8 (11)
OC1-P	95.6 (7)	95.6 (3)	82.9 (5)	96.3 (5)	**97.7 (3)**
OC1	98.5 (19)	97.0 (17)	91.2 (9)	94.2 (5)	90.3 (5)
Cascor	97.4 (5.9)	97.9 (1.2)	96.1 (1)	98.2 (1)	91.4 (1)

$$p = \sum_{\vec{x} \in \mathrm{dataset}} 1_{\mathrm{class}(\vec{x})=\mathrm{p}} \tag{4}$$

$$p_l = \sum_{\vec{x} \in \mathrm{dataset}} discr(\vec{x}) \times 1_{\mathrm{class}(\vec{x})=\mathrm{p}} \tag{5}$$

$$p_r = \sum_{\vec{x} \in \mathrm{dataset}} (1 - discr(\vec{x})) \times 1_{\mathrm{class}(\vec{x})=\mathrm{p}} \tag{6}$$

and similarly for n, n_l, n_r.

Ordinarily we think of a discriminant as a function mapping an instance $\vec{x}$ into one of two discrete values, say $\{0, 1\}$. We might let $discr(x) = 1$ if $\vec{\beta}.\vec{x} > 0$ and $discr(x) = 0$ otherwise. This would be a *hard* discriminant. Allowing instances to be partially assigned by the discriminant to both the left *and* right subsets amounts to simply allowing $discr(x) \in [0, 1]$ as above.

The soft entropy of $\vec{\beta}$, $SE(\vec{\beta})$, is simply $E(\vec{\beta})$ when $discr$ is defined softly as above. We only approximate the hard discriminant with the soft discriminant during optimization of $\vec{\beta}$. When an optimal $\vec{\beta}$ is found and the data is partitioned, instances are sent wholly either left or right.

Note that the approach here is more general than only defining a soft version of entropy – any splitting criterion which depends only on the statistics p, p_i, n, n_i (e.g., GINI (Breiman et al. 1984), C-SEP (Fayyad & Irani 1992)) can be similarly "softened" by taking $discr(x)$ to be continuous in $[0, 1]$ and using the above equations for these statistics.

3 Regularization: Pruning and Filtering

While studying the trees induced by the algorithm, we found that typical leaf subsets contained around 50 to 100 instances, while other leaves contained only a few. Clearly some sort of pruning is called for. In our learning system, pruning simply replaces a decision node with a leaf whenever the number of positive or negative instances matching that node drops below a fixed parameter. (The value 5 was used in all experiments[2].)

Pruning is a method of local pattern selection, since instances are effectively removed from nodes deep in the tree. The pruned tree misclassifies these instances, which we presume to be *outliers*. A novelty in our iterative re-filtering approach is to *rebuild* the tree using the reduced training set, the training set with outliers removed. The hypothesis is thus that data which is *locally* uninformative is also *globally* uninformative. We thus hope to build more *robust* trees (Hubel 1977).

4 Experiments

We ran DT-SE, DT-SEP (pruned), DT-SEPIR (pruned + iterative refiltering), OC1 (Murthy, Kasif, Salzberg & Beigel 1993*b*, Murthy, Salzberg & Kasif 1993*a*), and Cascade Correlation (Fahlman & Lebiere 1990) on several datasets (Thrun et al. 1991, Murphy & Aha 1994), and results appear in Table 1. DT-SEPIR is the best or second best algorithm on both real datasets. OC1 is similar to CART with a slight enhancement to the optimization method. Like CART, OC1 suffers from the use of a *hard* objective function (see Figure

[2] A more sensible strategy will be implemented shortly.

2). Cascade correlation results are averaged over 10 runs. Performance was generally better than OC1 but worse than DT.

5 Related Work

John (1994) gives an annotated bibliography of 30 papers on linear discriminant decision trees in machine learning, pattern recognition, and neural networks.

Although this work was developed independently, Brent (1991) discusses essentially the same technique for softening a related splitting criterion, using conjugate gradients for the optimization. Brent (1991) did not discuss pruning, nor did it present comparisons with similar work.

Acknowledgements

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

References

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), *Classification and Regression Trees*, Chapman & Hall, New York.

Brent, R. P. (1991), "Fast training algorithms for neural networks", *IEEE Transactions on Neural Networks* **2**(3), 346–354.

Brodley, C. E. & Utgoff, P. E. (1992), Multivariate versus univariate decision trees, Technical Report COINS TR 92-8, Department of Computer Science, University of Massachusetts, Amherst, MA, 01003.

Fahlman, S. E. & Lebiere, C. (1990), The cascade-correlation learning architecture, *in* D. Touretzky, ed., "Advances in Neural Information Processing Systems", Vol. 2, Morgan Kaufmann.

Fayyad, U. M. & Irani, K. B. (1992), The attribute selection problem in decision tree generation, *in* "AAAI-92: Proceedings of the Tenth National Conference on Artificial Intelligence", American Association for Artificial Intelligence, AAAI Press / The MIT Press, pp. 104–110.

Hubel, P. J. (1977), *Robust Statistical Procedures*, Society for Industrial and Applied Mathematics, Pittsburgh, PA.

John, G. (1994), "Annotated bibliography of linear discriminant decision trees", Posted to the `comp.ai.neural-nets` Newsgroup. Author's electronic mail: gjohn@cs.stanford.edu.

Loh, W.-Y. & Vanichsetakul, N. (1988), "Tree-structured classification via generalized discriminant analysis", *Journal of the American Statistical Association* **83**(403), 715–725.

Morgan, J. N. & Messenger, R. C. (1973), *THAID: a sequential analysis program for the analysis of nominal scale dependent variables*, University of Michigan.

Murphy, P. M. & Aha, D. W. (1994), "UCI repository of machine learning databases", Available by anonymous ftp to `ics.uci.edu` in the `pub/machine-learning-databases` directory.

Murthy, S. K., Salzberg, S. & Kasif, S. (1993*a*), "OC1", Software available by ftp to `blaze.cs.jhu.edu`.

Murthy, S., Kasif, S., Salzberg, S. & Beigel, R. (1993*b*), OC1: Randomized induction of oblique decision trees, *in* "AAAI-93: Proceedings of the Eleventh National Conference on Artificial Intelligence", American Association for Artificial Intelligence, AAAI Press / The MIT Press, pp. 322–327.

Thrun, S. B. et al. (1991), The monk's problems – a performance comparison of different learning algorithms, Technical Report CMU-CS-91-197, CMU School of Computer Science.