Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference, 1992, 97-101.

DECISION TREE CONSTRUCTION VIA LINEAR PROGRAMMING

KRISTIN P. BENNETT
COMPUTER SCIENCES DEPARTMENT, UNIVERSITY OF WISCONSIN
1210 WEST DAYTON STREET, MADISON, WISCONSIN 53706 *

Abstract. Linear-combination splits in decision trees allow multivariate relations to be expressed more accurately and succinctly than univariate splits alone. We propose the use of linear programming for determining linear-combination splits within two-class decision trees. The problem of determining an optimal linear-combination split to distinguish two sets can be formulated as a single linear program. Fast and powerful techniques exist for solving linear programs. The linear programming approach eliminates the problems of stopping criteria and local minima that plague gradient and perceptron approaches. Computational comparison of the proposed algorithm and classical univariate split algorithms indicates that the linear programming approach quickly produces smaller trees that generalize well.

1 Introduction Typically tree-structured classification algorithms such as CART [3] and ID3 [12] use univariate splits, i.e. splits based on a single variable. While univariate trees are easy to interpret logically, complex trees may be required to express multivariate relations. Linear-combination (LC) splits allow multivariate splits to be expressed more succinctly potential resulting in simpler trees with less nodes. The CART package, perceptron trees [20], and neural tree networks [16] all utilize LC splits. The potential difficulties with these splitting algorithms are discussed in Section 2. Finding the best LC split can be posed as a linear program (LP) that minimizes a weighted sum of the misclassification errors. The LP can be solved efficiently using fast algorithms that avoid local minima.

The paper is organized as follows. Section 2 discusses the LP formulation. Comparisons of the LP approach with other LC splitting methods are made. Section 3 describes the LP decision tree approach. Section 4 contains results of experiments comparing the LP approach with CART and C4.5 [12, 14] decision-tree approaches. Section 5 concludes with a summary.

2 LP Approach The optimal LC split consists of a separating plane that minimizes some measure of misclassification error. In this section, we propose an LP [2] which finds such a plane, and compare the LP with other LC splitting methods. We first describe our notation. For a vector x in the n-dimensional real space R^n, x_+ will denote the vector in R^n with components $(x_+)_i := \max\{x_i, 0\}$, $i = 1, \ldots, n$ (the *plus* function). The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. A_i will denote its ith row. The 1-norm of x, $\sum_{i=1}^{n} |x_i|$, will be denoted by $\|x\|_1$. A vector of ones in a space of arbitrary dimension will be denoted by e.

2.1 LP Formulation Let the two classes be represented by the two point-sets $\mathcal{A}$ and $\mathcal{B}$ in the n-dimensional real space R^n. Each training example in $\mathcal{A}$ and $\mathcal{B}$ is represented by a row of the $m \times n$ matrix A and the $k \times n$ matrix B respectively. When the sets $\mathcal{A}$ and $\mathcal{B}$ are linearly separable the goal is to find a "strictly separating plane" by solving the inequalities: $Aw > e\gamma$, $e\gamma > Bw$, where w is an n-dimensional "weight" vector representing the normal to an optimal "separating" plane, and γ is a real number which is a "threshold" that locates the strictly separating plane $wx = \gamma$. If a point A_i in $\mathcal{A}$ is correctly classified, then $-A_i w + \gamma < 0$ and consequently $(-A_i w + \gamma)_+ = 0$. If A_i is incorrectly classified then

* This work has been partially supported by Air Force Laboratory Graduate Fellowship SSAN 531-56-2969, Air Force Office of Scientific Research Grant AFOSR-89-0410 and National Science Foundation Grant CCR-9101801. The author's email address is bennett@cs.wisc.edu.

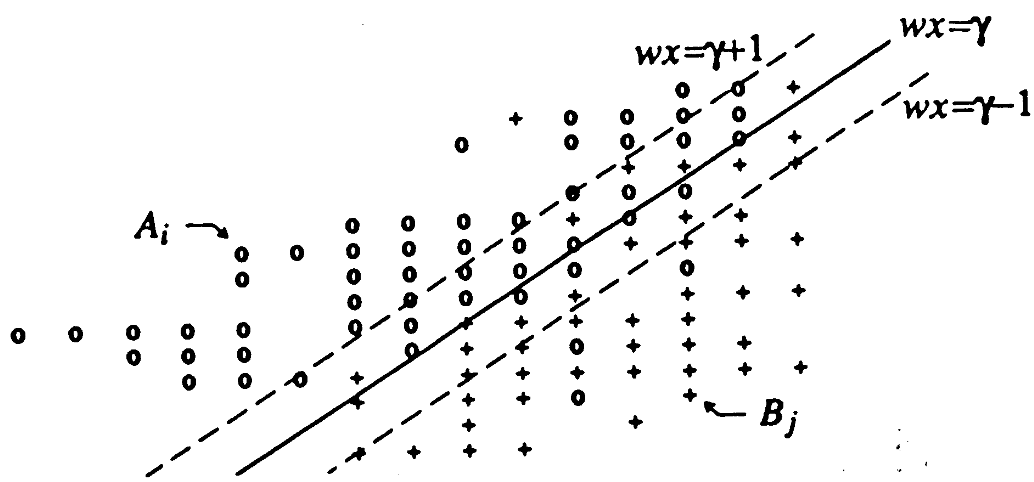

Figure 1: Linear-Combination Split by Linear Programming

$(-A_iw+\gamma)_+ \geq 0$. Similarly, $(B_iw-\gamma)_+$ provides a measure of misclassification for a point B_i in $\mathcal{B}$.

When the sets are linearly inseparable, an optimal separating plane is defined as a plane that minimizes a weighted sum of the misclassifications. Such an optimal plane can by obtained by solving the minimization problem:

$$\text{(2.1.1)} \qquad \min_{w\neq 0,\gamma} \ \frac{1}{m}\|(-Aw+e\gamma)_+\|_1 + \frac{1}{k}\|(Bw-e\gamma)_+\|_1$$

The constraint $w \neq 0$ is essential. Without it the point, $w = 0$, $\gamma = 0$, is an optimal solution and no separating plane is obtained. Problem (2.1.1) can be modified to remove the nonlinear constraint $w \neq 0$ as follows [2]:

$$\text{(2.1.2)} \qquad \min_{w,\gamma} \frac{1}{m}\|(-Aw+e\gamma+e)_+\|_1 + \frac{1}{k}\|(Bw-e\gamma+e)_+\|_1$$

Problem (2.1.2) always generates a *strictly* separating plane $wx = \gamma$ for linearly separable sets $\mathcal{A}$ and $\mathcal{B}$. The added term e ensures that no points of either class will be directly on the separating plane for the linearly separable case. For linearly inseparable sets $\mathcal{A}$ and $\mathcal{B}$, (2.1.2) will generate an optimal separating plane $wx = \gamma$, with $w \neq 0$, that minimizes the average violations

$$\frac{1}{m}\sum_{i=1}^{m}(-A_iw+\gamma+1)_+ + \frac{1}{k}\sum_{i=1}^{k}(B_iw-\gamma+1)_+.$$

Points of $\mathcal{A}$ which lie on the wrong side of the plane $wx = \gamma+1$, i.e. $\{x|wx < \gamma+1\}$, and points of $\mathcal{B}$ which lie on the wrong side of the plane $wx = \gamma-1$, i.e. $\{x|wx > \gamma-1\}$, are the only points that contribute to the violations. Figure 1 depicts an actual error-minimizing plane $wx = \gamma$ obtained by minimizing (2.1.2). Problem (2.1.2) can be transformed [2] to the equivalent LP:

$$\text{(2.1.3)} \qquad \min_{w,\gamma,y,z} \{\frac{1}{m}ey + \frac{1}{k}ez | y \geq -Aw+e\gamma+e, \ \ z \geq Bw-e\gamma+e, \ y \geq 0, \ z \geq 0\}$$

We briefly mention the additional desirable properties of LP (2.1.3) and recommend that the reader consult [2] for a complete discussion and proofs. The constant 1 locating the planes $wx = \gamma + 1$ and $wx = \gamma - 1$ can be considered a positive scale factor, and can be replaced by any $\zeta > 0$ as follows: $wx = \gamma + \zeta$ and $wx = \gamma - \zeta$. The linear program (2.1.3) will generate the same error-minimizing solution $wx = \gamma$ for any $\zeta > 0$. The weights of $\frac{1}{m}$ and $\frac{1}{k}$ on the sums ensure that a nontrivial w is always generated without imposing any extraneous constraints. The LP can be solved in polynomial time in theory and very quickly in practice. [8]. Computational results on real-world problems show that the LP (2.1.3) is preferable to other [18, 9, 7] LP-based approaches for linearly inseparable sets.

2.2 Other Linear-Splitting Methods Other decision-tree algorithms have used variants of back propagation [15], variants of the perceptron algorithm, and heuristic searches. CART uses a heuristic search algorithm which is computationally costly and is prone to local minima [3]. Utgoff [20] employs a perceptron algorithm which addresses the cycling problem [11, 6]. Since the perceptron algorithm fails to converge for the linearly *inseparable* case, stopping conditions are more difficult to determine and there is no guarantee that an optimal solution will be found. Sankar and Mamonne's neural tree network [16] uses back propagation [15] modified to use the sum of the absolute value of the errors to train each unit. It suffers from the usual difficulties of back propagation: choice of parameters, local minima, and stopping conditions. The advantage of the LP approach used with the simplex method [4] is that there are no parameters, no problems with local minima or convergence, and it has well-defined, easy-to-check stopping conditions.

3 LP Tree Algorithm We call the LP-based tree algorithm multisurface method - tree (MSMT) because it is an extension of the multisurface method of pattern recognition [9, 10] to decision trees. For each node in the tree, the best split of the points reaching that node is found by solving LP (2.1.3) using the simplex method [4]. The node is split into two branches, and the same procedure is applied until there are mostly points of one class at the node or there are too few points at the node. In practice, we split the most impure nodes first, as measured by the information function popularized by ID3, and limit the tree to at most 10 splits. The leaf nodes are assigned the class of the majority of points at that node. We adopted the pessimistic pruning strategy proposed used in C4.5 [13, 14].

4 Computational Results In this section we give computational comparisons on several real-world databases: the Wisconsin Breast Cancer Database [10, 21], the Cleveland Heart Disease Database [5], and the Bank Failure Database [1]. We use MSMT, CART, and C4.5 (the new and improved ID3). Our original experimental design called for the linear-combination feature of CART. Unfortunately, our commercial CART package crashes after extensive computational time whenever the linear-combination feature is invoked. Thus CART used only univariate splits in conjunction with a cost-complexity pruning procedure. C4.5 used univariate splits with pessimistic pruning. The windowing feature of C4.5 was disabled because windowing did not seem to improve the C4.5 results significantly. Also, windowing could be used with any of the three algorithms if desired.

Table 1 summarizes the results on three databases. The Wisconsin Breast Cancer Database consists of 681 points of which 442 are benign and 239 are malignant, all in a 9-dimensional real space. The Cleveland Heart Disease Database [1] consists of 197 points in a 13-dimensional real space, of which 137 are negative and 60 are positive. Categorical

[1] Available via anonymous ftp from ics.uci.edu courtesy of the University of California-Irvine.

WISCONSIN BREAST CANCER

Method	Train Error	CV Error	Leaf Nodes	Time (secs)
MSMT	2.4%	3.0%	2	6.8
C4.5	2.8%	3.8%	11	3.7
CART	5.3%	5.3%	3	-

CLEVELAND HEART DISEASE

Method	Train Error	CV Error	Leaf Nodes	Time (secs)
MSMT	15.5%	18.2%	2	9.2
C4.5	9.4%	25.9%	28	1.0
CART	16.8%	20.5%	6	-

BANK FAILURE

Method	Train Error	CV Error	Leaf Nodes	Time (secs)
MSMT	6.4%	6.5%	3	156.3
C4.5	5.0%	7.2%	67	261.0

Table 1: Comparison of MSMT, C4.5, and CART on Three Databases
Train Error := % error on entire data set, CV Error := % cross-validation error (10-fold)

features within this database were converted to ordered integers for MSMT but not for C4.5 and CART. The Bank Failure Database consists of 4751 points in a 9-dimensional real-space with 4311 successful banks and 441 failed banks. This previously unpublished data set, collected by Richard S. Barr of Southern Methodist University and Thomas F. Siems of the Federal Reserve Bank of Dallas, has 9 numeric features which range from 0 to 1. The Bank Failure Database exceeded the space limitations for the CART program so there are no results for CART.

Ten-fold cross validation was used to measure generalization. The data was partitioned into 10 roughly-equal parts. For each part, a decision tree was created using the remaining nine parts and tested on the part. The cross-validation error is the total number of points misclassified on all 10 parts divided by the total number of points in the database. The times reported are the CPU time on a DECStation 5000/125 required to construct and prune one tree averaged over the 10 folds. The CART program performs additional computations and was executed on a different machine. Thus no times are reported for the CART algorithm. The percent training set error and the number of leaf nodes reported are the results from using the entire database one time.

MSMT quickly produced trees with fewer nodes and better generalization than the other two methods. The cross-validation error for MSMT was less than that for C4.5 and CART on all three databases. MSMT produced smaller trees in terms of leaf nodes than did C4.5 and CART. Dramatic reduction in tree size makes the tree easier to interpret and thus compensates for the slightly more complex LC splits. CART also had smaller trees than C4.5 probably because of its better but more expensive pruning algorithm. MSMT and C4.5 were very fast on the Breast Cancer data and the Heart Disease data. C4.5 is slightly faster especially on the Heart Disease Database which has categorical variables. C4.5 handles categorical variables very efficiently. MSMT, like other LC methods, requires that the attributes be either linearized or encoded as binary attributes in a higher dimensional space. Thus MSMT works best on numerical attributes. On the Bank Failure Database, MSMT was much faster than C4.5 indicating MSMT works well on larger data sets.

5 Conclusions We have presented an LP method for constructing two-class decision trees. Unlike previous LC splitting methods, the LP approach has no problems with local minima, choice of parameters, and convergence criteria. The MSMT algorithm compares favorably with classical decision tree methods in terms of accuracy, training time, and size of trees. The LP described is limited to two-class problems. Work is in progress to generalize the LP to handle multi-category splits similar to those proposed for linear machine decision trees [19] and neural tree networks [16], and to do variable elimation by using the optimality conditions of the LP. We have demonstrated that LP-based decision tree algorithms compare very favorably with other approaches and warrant further investigation and application.

References

[1] Barr, R. S. (Southern Methodist University), & Siems, T. F. (Federal Reserve Bank of Dallas) (1990). Private Communication to O. L. Mangasarian, July 20, 1990.

[2] Bennett, K. P, & Mangasarian, 0. L. (1992). Robust Linear Programming Discrimination of Two Linearly Inseparable Sets. to appear in *Optimization Methods and Software.*

[3] Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* CA: Wadsworth International.

[4] Dantzig, G. B. (1963) *Linear Programming and Extensions*, Princeton, New Jersey: Princeton University Press.

[5] Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J., Sandhu, S., Guppy, K., Lee, S., & Froelicher, V., (1989). International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease. *American Journal of Cardiology*, 64, 304-310.

[6] Gallant, S. (1986). Optimal Linear Discriminants. *Proceedings of the International Conference on Pattern Recognition.* IEEE Computer Society Press, 849-852.

[7] Glover, F. (1990). Improved Linear Programming Models for Discriminant Analysis. *Decision Sciences*, 21,4, 771-785.

[8] Karmarkar, N. (1984). A New Polynomial Time Algorithm for Linear Programming, *Combinatorica*, 4, 373-395.

[9] Mangasarian, O. L. (1968). Multisurface Method of Pattern Separation. *IEEE Transactions on Information Theory*, IT-14(6), 801-807.

[10] Mangasarian, O. L., Setiono, R., & Wolberg, W.H. (1990). Pattern Recognition Via Linear Programming: Theory and Application to Medical Diagnosis. In Coleman T. F., & Li, Y. (Eds.). *Large-Scale Numerical Optimization* Philadelphia: SIAM, 22-30.

[11] Minsky, M., & Papert, S. (1972). *Perceptrons: An Introduction to Computational Geometry* (expanded edition). Cambridge, MA: MIT Press.

[12] Quinlan, J. R. (1984) Induction of Decision Trees. *Machine Learning*, 1, 81-106.

[13] Quinlan, J. R. (1987). Decision Trees as Probabilistic Classifiers. In Langley (Ed), *Proceedings of Fourth International Workshop on Machine Learning.* Los Altos, CA: Morgan Kaufmann.

[14] Quinlan, J. R. (1987). Simplifying Decision Trees, *International Journal of Man-Machine Studies*, 27, 1987.

[15] Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). Learning Internal Representations. In Rumelhart, D. E., & McClelland J. L. (Eds.) *Parallel Distributed Processing* (Vol. I), Cambridge, Ma: M.I.T. Press, 318-362.

[16] Sankar, A., & Mammone R. J. (1991). Speaker Independent Vowel Recognition using Neural Tree Networks. *Proceedings of IJCNN*, Seattle.

[17] Sankar, A., & Mammone, R. J. (1991). Optimal Pruning of Neural Tree Networks for Improved Generalization. *Proceedings of IJCNN*, Seattle.

[18] Smith, F. W. (1968). Pattern Classifier Design by Linear Programming. *IEEE Transactions on Computers C-17*, 4, 367-372.

[19] Utgoff, P. E., & Brodley, C. E., (1991). Linear Machine Decision Trees. COINS Technical Report 91-10, University of Massachusetts - Amherst.

[20] Utgoff, P. E. (1989). Perceptron Trees: A Case Study in Hybrid Concept Representations. *Connection Science*, 1, 4, 377-391.

[21] Wolberg, W. H., & O.L. Mangasarian, O. L. (1990). Multisurface Method of Pattern Separation Applied to Breast Cytology Diagnosis. *Proceedings of National Academy of Sciences, USA*, 87, 9193-9196.