Bivariate Oblique Decision Tree Algorithms Based on Linear Discriminant Analysis

B Liu, J Song and S R Jiao

School of Statistics, Capital University of Economics and Business, Beijing, 100070, China
songjie@cueb.edu.cn   Email: xxliubiao@163.com

Abstract. To solve the problem of the low efficiency of the traditional decision tree algorithm in dividing decision boundaries that are not parallel to the coordinate axis as well as the problem of the difficulties and the limited interpretability of some oblique decision tree algorithms in solving high-dimensional data covariance matrix, the present study proposed a series of bivariate oblique decision tree algorithm based on LDA. In the algorithms, LDA was used to determine the division boundary, and impurity and variance analysis were used as a criterion for the selection of split variables, thus ensuring the interpretability of this method while improving the efficiency of defining the decision boundaries that were not parallel to the coordinate axis. Experiments conducted in multiple data sets show that algorithms has achieved satisfactory results.

1. Introduction
The decision tree algorithm has become one of the most widely used models in the field of data mining due to its simple generation rules and highly readable model structure. However, with the increasing complexity of the data structure, conventional decision tree algorithms select a single variable for division each time, ignoring the correlation between variables. The decision surface parallel to the coordinate axis sometimes builds a tree with an unusually large structure in order to obtain a higher accuracy, which violates the intuitive and straightforward original intention of the decision tree. In order to solve these problems, Breiman proposed a multivariate tree CART-LC that fine-tunes weights one by one to reduce impurity [1]. In CART-LC, all input dimensions can be used at each decision node to obtain a hyperplanar decision surface in any direction. After that, many scholars have made improvements based on CART-LC and proposed a series of oblique decision tree (ODT) algorithms that use multiple variables at nodes for the division. ODT can generate decision surfaces in any direction, considering the correlation between variables while significantly reducing the size of the tree, it has become an important optimization direction of the decision tree algorithm. The linear discriminant analysis (LDA)-based FACT algorithm [2] and the QUEST algorithm based on statistical testing [3] introduced statistical methods into the decision tree algorithm. They improved the decision tree by applying traditional statistical methods in the selection of dividing variables and splitting points. CRUISE algorithm [4] implemented the selection of dividing variables and construction of dividing criteria into two steps based on FACT and QUEST. It corrected the deviation caused by the traditional decision tree using a greedy search and directly used the two-variable linear discriminant model in the leaf nodes to simplify the tree structure. The statistical decision tree algorithm is essentially a single-variate decision tree, and the Cline algorithm uses LDA and KNN to construct a multivariate decision boundary, applying statistical methods to ODT [5]. GUIDE algorithm fine-tunes the test p-value...
standard based on CRUISE and applies a binary linear kernel and nearest-neighbor method to the segmentation criteria [6]. The purpose of this research is to develop a series of LDA-based binary ODT algorithms in order to solve the contradiction between the pursuit of purity and simplifying the tree in traditional algorithms. The test p-values of variance analysis and impurity are used as the dividing criteria at the nodes to solve the problems of difficulty in solving the covariance matrix of high-dimensional data and poor interpretability in the Cline algorithm and improve the instability of the GUIDE algorithm using the $\chi^2$ statistic to measure the correlation between numerical variables.

2. The proposed algorithm

The proposed algorithm adopts a bivariate linear discriminant function for the division at the nodes, preserving the structure of CART tree. It uses the test p-value of variance analysis and impurity as the dividing criteria. The linear combination of two variables is used for each division. Thus, each node of the generated ODT can be visualized, which not only retains the interpretability of the CART algorithm but also improves the division efficiency of the decision tree for the decision boundary not parallel to the coordinate axis, as shown in Figure 1.

![Decision boundary of CART and C Method](image)

Figure 1. Comparison of decision boundary between CART and C method

The proposed algorithm divides each node by a linear discriminant function fitted by two variables based on the Cline and GUIDE algorithms. The proposed algorithm effectively solves the covariance matrix of high-dimensional data and poor interpretability in the Cline algorithm and also improves the instability of GUIDE algorithm using the $\chi^2$ statistic to measure the correlation between numerical variables. Only two variables that can distinguish the best different types of data are selected for linear discrimination. The solution process can be formulated as the selection of variate pairs, where the parameters of the linear discriminant function are simultaneously estimated under the optimization goal of the greatest decrease in impurity.

The proposed algorithm includes a series of preprocessing steps. First, the discrete variables are converted into continuous variables. Specifically, the discrete variables are first converted into dummy variables, and then they are projected into numerical variables on the largest discriminant coordinates. Finally, the projected numerical variables are Box-Cox transformed; when processing is missing values, simple imputation is used to fill in the missing values. If the missing variables are continuous, the mean is used to fill in the missing values. In contrast, if the missing variables are discrete, the mode is used to fill in the missing values. The algorithm uses the pre-pruning method and performs the grid search on the parameters of the pre-pruning strategy.
In the proposed algorithm, the multi-classification problem is formulated as multiple binary classifications. For a multi-class problem \( J > 2 \), one class versus all other classes framework is employed, where one of the classes is selected for the binary classification while the remaining \( J-1 \) classes are considered as negative class. By repeating \( J \) times of binary classification, the proposed algorithm solves the \( J \)-classification problem.

A node is not partitioned if one or more of the following conditions are met: There are very few cases in the node (default is 5); At most one class has more than \( m \) cases, where \( m \) is user-specified (default is \( m = \max\{2, N/200\} \), where \( N \) is the number of cases in the learning sample); All the cases go down the same branch if the node is split.

### 2.1. Methods based on an impurity

When the algorithm uses separate entropy gain (C method) and misclassification rate (E method) as the dividing criteria, for the dataset \( D_t \) at any node \( t \), the algorithm first traverses all variate pairs, fits \( C^K_2 \) binary linear discriminant functions. Then, it selects the variable pairs with the best division effects and corresponding linear discriminant functions to divide the dataset. At any node, the algorithm flow can be expressed as:

#### Algorithm 1 C method and E method

**Input:** Dataset \( D_t = \{(x_1, x_2, ..., x_K, y)_{n \times (k+1)}\} \)

**Output:** Node \( t \)

1. The class with the highest proportion will be returned, if \( D_t \) satisfies the stop conditions;

2. (C) Entropy(t) = \(-\sum_{i=1}^{l} p(y_i) \log_2 p(y_i)\), for \( a, b \) in 1 to \( k \), \( a \neq b \) do
   a) Fit a linear discriminator classifier \( M \) to \( D_t' = \{(x_a, x_b, y)_{n \times 3}\} \), and get the decision boundary \( x_b - u x_a = v \). Specially, fit the linear discriminator classifier \( M \) with each single variable when matrix \( D_M = \{(x_a, x_b)_{n \times 2}\} \) is a singular matrix;
   b) Divide \( D_t \) with the decision boundary of classifier \( M \);
   c) \( H(D_t|M) = -\sum_{i=1}^{l} p(m_j) \sum_{j=1}^{l} p(y_i|m_j) \log_2 p(y_i|m_j)\);
   d) Gain_{Gain}(t) = Entropy(t) - H(D_t|M);

\((a_0, b_0, u_0, v_0) = \text{argmax} \text{ Gain}_{Gain}(t)\);

2. (E) for \( a, b \) in 1 to \( k \), \( a \neq b \) do
   a) Fit a linear discriminator classifier \( M \) to \( D_t' = \{(x_a, x_b)_{n \times 3}\} \), and get the decision boundary \( x_b - u x_a = v \). Specially, fit the linear discriminator classifier \( M \) with each single variable when matrix \( D_M = \{(x_a, x_b)_{n \times 2}\} \) is a singular matrix;
   b) Classification error(t) = 1 - max[p(i|t)]
   \((a_0, b_0, u_0, v_0) = \text{argmin} \text{ Classification error}(t)\);

3: Divide \( D_t \) to \( S_1 = \{(x_1, x_2, ..., x_k, y), x_{b_0} - u_0 x_{a_0} \leq v_0\} \) and \( S_2 = \{(x_1, x_2, ..., x_k, y), x_{b_0} - u_0 x_{a_0} > v_0\} \) with \( x_{b_0} - u_0 x_{a_0} = v_0 \);

4: Execute the algorithm recursively with \( S_1 \) and \( S_2 \) as inputs.

### 2.2. Methods based on variance analysis

The algorithm uses the p-value of the ANOVA test (M1 method) and the p-value of the MANOVA test (M2 method) as the dividing criteria. For the dataset \( D_t \) at any node \( t \), the M1 method first traverses all the variate pairs and fits \( C^K_2 \) binary linear discriminant functions. Variable pairs are combined into a new variable according to the discriminant function. The discriminant function with variable pairs having the most significant p-value of ANOVA is used to divide the dataset. Similarly, in the M2 method, variable pairs that have the most significant p-value of MANOVA are selected to fit the binary linear discriminant function and divide the dataset. At any node, the algorithm flow can be expressed as:

#### Algorithm 2 M1 method and M2 method

**Input:** Dataset \( D_t = \{(x_1, x_2, ..., x_K, y)_{n \times (k+1)}\} \)

**Output:** Node \( t \)
1: The class with the highest proportion will be returned, if $D_t$ satisfies the stop conditions;
2: (M1) for $a, b$ in 1 to k, $a \neq b$
do
a) Fit a linear discriminator classifier $M$ to $D_t' = \{(x_a, x_b, y)_{n \times 3}\}$, and get the decision boundary: $x_a - u x_b = v$. Specially, fit the linear discriminator classifier $M$ with each single variable when matrix $D_M = \{(x_a, x_b)_{n \times 2}\}$ is a singular matrix;
b) $x' = x_a - u x_b - v$;
c) Compute $p_{value}$ of ANOVA for $D_{anova} = \{(x', y)_{n \times 2}\}$; 
$(a_0, b_0, u_0, v_0) = \text{argmin}_{a,b} p_{value}$;
2: (M2) for $a, b$ in 1 to k, $a \neq b$
do
a) Compute $p_{value}$ of MANOVA for $D_{manova} = \{(x_a, x_b, y)_{n \times 3}\}$ 
$(a_0, b_0) = \text{argmin}_{a,b} p_{value}$ . Fit a linear discriminator classifier $M$ to $D_t' = \{(x_a, x_b, y)_{n \times 3}\}$, and get the decision boundary: $x_{b_0} - u x_{a_0} = v$. Specially, fit the linear discriminator classifier $M$ with each single variable when matrix $D_M = \{(x_a, x_b)_{n \times 2}\}$ is a singular matrix;
3: Divide $D_t$ to $S_1 = \{(x_1, x_2, ..., x_k, y), x_{b_0} - u x_{a_0} \leq v_0\}$ and $S_2 = \{(x_1, x_2, ..., x_k, y), x_{b_0} - u x_{a_0} > v_0\}$ with $x_{b_0} - u x_{a_0} = v_0$;
4: Execute the algorithm recursively with $S_1$ and $S_2$ as inputs.

3. Experiment results and analysis
In order to verify the applicability of the proposed algorithm, comparative experiments with mainstream machine learning methods were conducted on three classification datasets. The average prediction accuracy for the test set with 100 times k-fold cross-validation is used as the evaluation metric. The value of k was determined by the distribution of the variable to be predicted.

3.1. Dataset
The datasets used in the experiment were all from the UCI machine learning database [7]. Table 1 lists the information of the dataset, including the sample size, the number of independent variables, and the attributes of dependent variables.

| Code | N | Variables | J |
|------|---|-----------|---|
| Iris | 150 | 5 | 3 |
| BC   | 683 | 10 | 2 |
| Fish | 158 | 7 | 7 |

3.2. Experimental results
The proposed algorithm was implemented using R language programming. Eight mainstream machine learning models were adopted as compared methods, including CART, SVM_Linear, SVM_Radial, Logistic Regression (LOG), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K Nearest Neighbor (KNN), and Naive Bayes (NB). The average prediction accuracy for the test set with 100 times k-fold cross-validation is used as the evaluation metric. The compared results are shown in Table 2.

| Method | Train | Test |
|--------|-------|------|
|        | Iris(k=5) | BC(k=10) | Fish(k=2) |
|        | Iris(k=5) | BC(k=10) | Fish(k=2) |
| Iris   | 97.70 | 97.72 | 82.00 | 93.43 | 95.71 | 78.00 |
| Iris   | 96.79 | 96.96 | 98.29 | 93.07 | 95.54 | 95.61 |
| Iris   | 97.59 | 97.13 | 98.67 | 95.04 | 95.96 | 97.03 |
| Iris   | 97.60 | 97.13 | 98.67 | 95.04 | 95.97 | 97.03 |
As shown in Table 2, the classification accuracies of the two proposed algorithms are comparable to the mainstream machine learning models. When dealing with problems with a large number of dependent variables, the classification accuracies of the Method, M1 method, and M2 method are better than that of CART algorithm and SVM. As shown in Figure 2, the prediction accuracy of M1 method (right tree) was 96.63%, while that of CART (left tree) was 96.78%. It indicates that the proposed algorithm can generate a smaller tree than the CART structure while achieving the same level of prediction accuracy. At the same time, as shown in Figure 3, the proposed algorithm can visualize a binary linear discriminant function at each node.
4. Conclusions
A series of bivariate ODT algorithms based on LDA are proposed. In the proposed algorithms, entropy gain, misclassification gain, and test p-value of variance analysis are used as the dividing criteria. The experimental results demonstrate that the proposed algorithm can achieve higher prediction accuracy but a simpler structure than CART when LDA performance is secure. This can be attributed to the fact that the proposed algorithm can retain the main effects of the target variable of the two variable pairs in original variables, while CART ignores this main effect.

However, two proposed methods can only handle numerical data like LDA. Future work will consider how to add categorical data into the discriminant function. Also, future work will include a discussion of the problem of post-reduction.

Acknowledgements
The authors would like to thank The Capital University of Economics and Business Graduate Science and Technology Innovation project to support this work.

References
[1] Breiman L, Friedman J, Olshen R, et al. Classification and Regression Trees[M]. New York: Chapman & Hall, 1984.
[2] Loh, W.-Y. & Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion). J. Amer. Statist. Assoc., 83, 715–728.
[3] Loh, W.-Y. & Shih, Y.-S. (1997). Split selection methods for classification trees. Stat. Sinica, 7, 815–840.
[4] Kim H, Loh W Y. Classification Trees With Unbiased Multiway Splits[J]. Journal of the American Statistical Association, 2001, 96(454).
[5] Amasyali M F , Ersoy O. Cline: new multivariate decision tree construction heuristics[C]. Istanbul, Turkey: Icsc Congress on Computational Intelligence Methods & Applications, 2005.
[6] Loh, W.-Y. (2012). Variable selection for classification and regression in large p, small n problems. In Probability approximations and beyond, Vol. 205, Eds. A. Barbour, H.P. Chan & D. Siegmund. Lecture Notes in Statistics— Proceedings. pp. 133–157. New York: Springer.
[7] https://archive.ics.uci.edu/ml/index.php