Cline: New Multivariate Decision Tree Construction Heuristics

M.Fatih Amasyalı[1], Okan Ersoy[2]

[1] Department of Computer Engineering, Yıldız Technical University, Istanbul, TURKEY mfatih@ce.yildiz.edu.tr
http://www.ce.yildiz.edu.tr/myindex.php?id=14

[2] School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47907, USA,
ersoy@purdue.edu

***Abstract*-Decision trees are often used in pattern recognition and regression problems. They are attractive due to high performance and easy-to-understand rules. Many different decision tree construction algorithms have been developed because of their popularity. In this work, we describe some new heuristic tree construction algorithms and test with 8 benchmark datasets. We compare the new method with other 21 tree induction algorithms. The results show that cline heuristics can be used in all types of classification problems because of its simplicity and acceptable performance.**

1 Introduction

Decision tree algorithms are one of the most used tools for pattern recognition, regression, machine learning and data mining. A typical decision tree algorithm begins with all data, splits data into two subsets based on the vales of one or more attributes and then recursively splits each subset into finer ones until each subset contains single-class data. A decision tree contains hierarchic structure of internal nodes and leaves. Instances are classified by navigating them from the root of tree down to a leaf. Each leaf is assigned to one class. Each node of a decision tree has a criterion. Decision trees are named according to how many attributes are used in the nodes. Univariate decision trees use only one attribute whereas multivariate trees use more than one attribute. Univariate and multivariate decision trees which are constructed for the same dataset are illustrated in Figure 1. Many studies show that the multivariate trees are generally smaller but less understandable [1].

In this work, new heuristic tree construction algorithms(Cline) were developed and compared with other 21 decision tree construction methods. A detailed explanation of the Cline algorithm is given in the second section. In the third section, the comparison of Cline and other methods on 8 different benchmark datasets is discussed. The conclusions and future works are given in the last section.

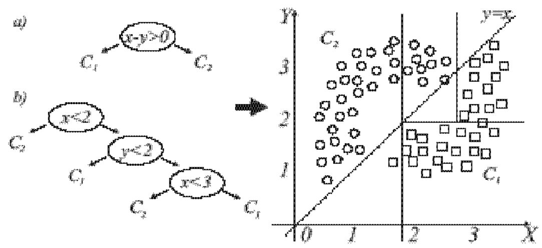

Fig. 1. (a) Multivariate (b) Univariate decision trees[2].

2 Cline Decision Tree

The Cline decision tree is a multivariate tree. At each node, all attributes of instances are used for decision making. The Cline tree is also a binary tree because two branches are generated from each internal node. It can be used for all classification problems based on numerical values.

The main idea of the Cline is the division of the feature space into two regions with a hyperplane. The determinations of the hyperplanes are discussed in the next section. The division process is repeated until each region contains single-class data. In the two-dimensional space, the boundary which separates classes is a line, in three dimensional space, it is a plane, and in higher dimensional spaces, it is a hyperplane. At each node, there is a hyperplane and the test instances are directed within the tree according to whether the test point is on one side or the other of the hyperplane.

2.1 Determination of the Boundary HyperPlanes

We developed five heuristic methods (CL2, CL4, CLM, CLLDA, CLLVQ) developed to determine the boundary hyperplanes. The methods will be explained in two dimensional (2-D) space for the purpose of easy visualization. In 2-D space, classes are separated with a line. For example, Figure 2 illustrates 10 instances from two classes in a sample training set.

CL2: In Figure 2a, the nearest two points which are from different classes are ‘A’ and ‘B’. There is only one line which passes through the midpoint of this line and perpendicular to it.

1-4244-0020-1/05/$20.00 ã2005 IEEE

This line is referred to as classification line and used in CL2 algorithm.

CL4: 'D' point is the second nearest point (which is from other class) to 'A' point after the 'B' point. Similarly, 'C' point is the second nearest point to 'B' point after the 'A' point. There is only one line which passes through the midpoint of the A-C line and the midpoint of the B-D line, say, Line 2. Then, the line perpendicular to Line 2 and passing through its midpoint as shown in Figure 2b will be called classification line and used in CL4 algorithm.

CLM: In Figure 2c, 'A' point is the mean point of one class. 'B' point is the mean point of the other class. There is only one line which passes through the midpoint of A-B line and perpendicular to it. This line is referred to as classification line and used in CLM algorithm.

CLLDA: Linear Discriminant Analysis (LDA) [3] transforms the multidimensional data into one dimension such that the distance between class centroids is maximized. In Figure 2d, the nearest two points which are from different classes are 'A' and 'B'. There is only one line which passes through the midpoint of A-B line and perpendicular to LDA's transformation hyperplane. This line is referred to as classification line and used in CLLDA algorithm.

CLLVQ : Using Linear Vector Quantization (LVQ) [4], two class centroids (A, B) are found. There is only one line which passes through the midpoint of A-B line and perpendicular to it. This line is referred to as classification line and used in CLLVQ algorithm.

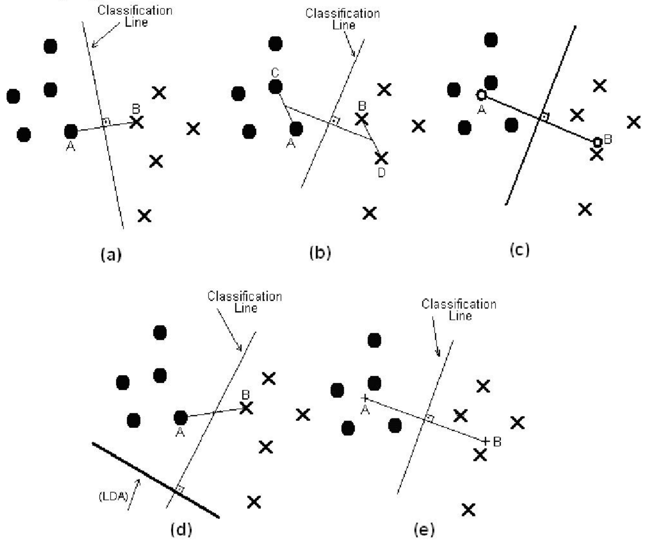

Fig. 2. Finding Hyperplane a) CL2 b) CL4 c) CLM d) CLLDA e) CLLVQ.

2.2 Pruning Process

In most cases, growing a decision tree until all leaves contain data for a single class causes overfitting. To avoid overfitting, most of the decision tree induction algorithms (C4.5 [5], OC1 [6] etc.) use pruning. Data is divided into three subsets (train, prune, and test) for pruning. Several pruning methods are developed in the literature[7]. In reduced error pruning, the procedure checks whether replacing a node with the most frequent class does not reduce the tree's classification accuracy with the prune set. If so, the node is pruned. In cline heuristics, two reduced error pruning methods are used.

1) Top-down pruning: start pruning from the node at the top of the tree.

2) Down-up pruning: start pruning from the node at the lower levels of the tree and having a leaf.

2.3 Types of Tree Classification

During decision making, the starting point is the top node of the tree. The test point moves around branches according to whether the test point is on one side or the other of a test node. The stopping point is the leaf node which has a class label. Determining the class of test sample is done by two cline heuristics:

1) Using Leaves: Test sample's class is leave's class.

2) Using Branches: During the tree construction process each node's class distribution is recorded. During testing, the class's distributions which are at the test sample's path on the tree are summed. The test sample's class is the class having highest probability.

2.4 Conversion of a Binary Classifier to a N-Class Classifier

The original Cline algorithm is developed for the two-class problem. There are two known methods to convert this type of classifier (Cline, SVM etc.) to a N-class classifier[8].

1- One vs All: For N-category classification, construct N one-versus-all binary classifiers, each to distinguish one class from all other classes. For each classifier, all the training set is used. The final class is decided by consensus.

2- One vs One: For N-category classification, construct N*(N-1)/2 binary classifiers, each to distinguish one class from another class. For each classifier, a subset of the training set including only the instances from the two classes used in the binary classifier is used.

The second method is commonly used in literature. In this work, One vs One is used.

3 Experimental Results

As our first and simple application, the boundary lines are seen in Figure 3 for two artificial datasets (XOR and Spiral). Bold lines are the boundaries. The datasets are observed to be separated successfully with the Cline algorithm.

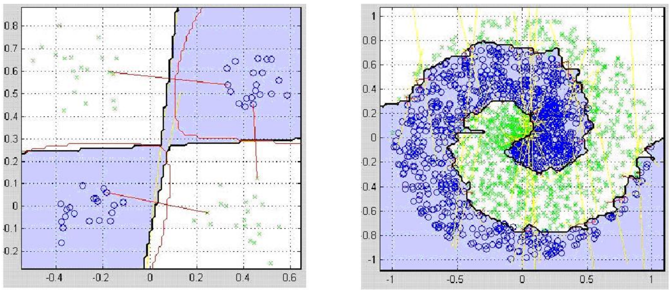

Fig. 3. The boundary lines which are created by CL2 for two bivariate two class problems.

T.S. Lim et al.' (2000) compared 32 new and old classification algorithms on 8 benchmark dataset with 10 fold cross validation [9]. 21 algorithms out of 32 are tree construction algorithms. We used Lim's classification accuracy results to compare them with our developed methods. The detailed descriptions of 21 tree construction algorithms are given in Lim's paper [9]. The 8 UCI datasets [10] used are shown in Table 2 with their names and characteristics.

Table 1. Summary of datasets

Dataset Name	Dataset Code	Attributes	Classes	Cases
Breast cancer Wisconsin	Bcw	9	2	683
Boston housing	Bos	12	3	506
Congressional voting	Vot	16	2	435
Bupa liver disorders	Bld	6	2	345
StatLog heart disease	Hea	7	2	270
Pima Indians diabetes	Pid	7	2	532
StatLog image	Seg	19	7	2310
StatLog vehicle silhouette	Veh	18	4	3772

In this work five different methods were used for determining hyperplanes, two methods were used for types of tree classification, three methods were used for tree pruning. In other words, 30(2*3*5) different cline versions are run on the 8 datasets. The average accuracies of cline methods are given at Table 3. The average of 10 cline versions that use top-down pruning is given at fifth line. Among pruning methods, no pruning has highest accuracy. Among determining hyperplane methods, CLLDA has highest accuracy. Among types of tree classification, using branch has highest accuracy.

Table 2. Average accuracy rates of cline methods.

Methods	Success Ratio
Using leaves	0.7888
Using branch	0.7914
No pruning	0.7920
Top-down	0.7878
Down-up	0.7905
CL2	0,7720
CL4	0,7728
CLLDA	0,8074
CLLVQ	0,7978
CLM	0,7988

Five different hyperplane methods are compared in Figure 4.a. In this figure 'd n' corresponds to using branch and no pruning, 'd td' corresponds to using branch and top-down pruning, 'd du' corresponds to using branch and down-up pruning, 'y n' corresponds to using leaves and no pruning, 'y td' corresponds to using leaves and top-down pruning, 'y du' corresponds to using leaves and down-up pruning.

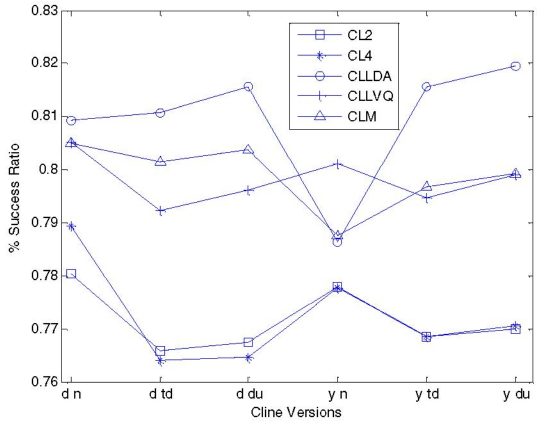

Figure 4a. Comparison of determining hyperplane methods.

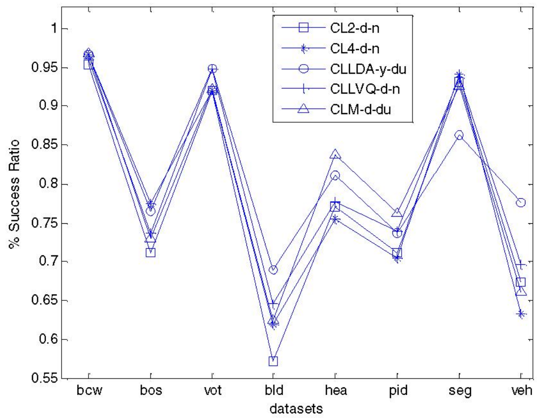

Figure 4b. Comparison of cline versions.

The classification accuracies of five cline versions out of 30 are given in Figure 4b. In this figure, the x axis corresponds to dataset codes. There is no single version which performs better than all of the others on all datasets. Therefore, it is possible to combine different successful cline versions into one tree.

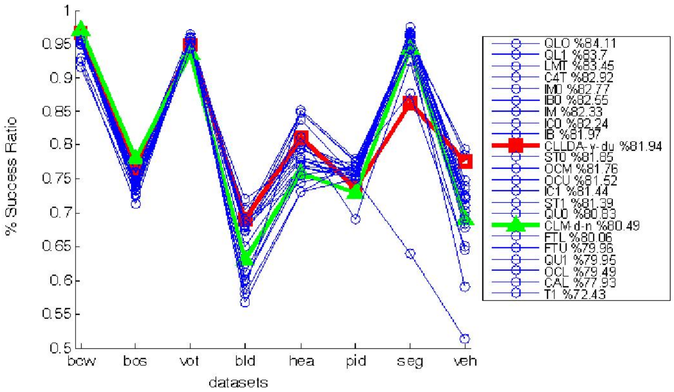

Figure 5. Accuracy comparisons of tree construction methods and two cline versions.

In Figure 5, 21 tree construction algorithms are compared with two cline version on 8 datasets. Cline versions' accuracies are competitive with other algorithms.

4 Conclusion and Future Work

In this paper, new heuristic tree construction algorithms - Cline- is proposed. The Cline algorithms construct binary, multivariate trees that can be used with numerical attributes. At each node of the tree, there is a hyperplane which separates the classes.

In this work, five different methods were developed for determining hyperplanes, two methods were used for types of tree classification, three methods were used for tree pruning. Consequently, 30(2*3*5) different cline versions were run on the 8 datasets using 10 fold cross validation. The best cline version out of 30 is CLLDA-y-du (using LDA in determining hyperplane, using leaves and down-up pruning).

For future work, we plan to improve performance in a number of ways, for example, by trying other pruning methods, and combining several cline versions into one tree, by construction of univariate cline versions and generalizing the method to the N-class problem by other ways. In addition, algorithm comparison can be done with other criteria such as tree size, learning time. Using branch method can be used with well-known tree classification algorithms.

In conclusion, the Cline algorithms can be used in classification applications because of their simplicity and acceptable performance.

References

[1] Xiao-Bai Li, James R. Sweigart, James T.C. Teng, Joan M.Donohue, Lori A. Thombs and S. Michael Wang, (2003), "Multivariate Decision Trees Using Linear Discriminants and Tabu Search", IEEE Transactions on Systems, Man, and Cybernetics,vol.33, No.2

[2] http://aragorn.pb.bialystok.pl/~mkret/docs/pkdd2000.pdf

[3] Fisher, R.A. (1936) The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7: 179-188.

[4] Kohonen T., "Self Organizing Maps", Springer Series in Information Sciences, p 245.

[5] Ross Quinlan (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.

[6] S. K. Murthy, S.Kasif, S. Salzberg, R. Beigel, (1994), "OC1: A randomized algorithm for building oblique decision trees", Journal of Artificial Intelligence Research, vol.2, p 1-32.

[7] F.Esposito, D. Malerba, G. Semeraro (1997), "A Comparative Analysis of Methods for Pruning Decision Trees", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No.5.

[8] Alpaydın E. (2004) "Introduction to Machine Learning", The MIT Press, p 135-140.

[9] Tjen-Sien Lim, Wei-Yin Loh,Yu-Shan Shih (2000), "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms", Machine Learning, vol.40, p 203-229.

[10] Blake, C., Merz, C. (1998) UCI Repository of machine learning databases, available at: http://www.ics.uci.edu/ mlearn/MLRepository.htm