International Statistical Review

International Statistical Review (2014), 82, 3, 329–348 doi:10.1111/insr.12016

Fifty Years of Classification and Regression Trees[1]

Wei-Yin Loh

Department of Statistics, University of Wisconsin, Madison, WI 53706, USA
E-mail: loh@stat.wisc.edu

Summary

Fifty years have passed since the publication of the first regression tree algorithm. New techniques have added capabilities that far surpass those of the early methods. Modern classification trees can partition the data with linear splits on subsets of variables and fit nearest neighbor, kernel density, and other models in the partitions. Regression trees can fit almost every kind of traditional statistical model, including least-squares, quantile, logistic, Poisson, and proportional hazards models, as well as models for longitudinal and multiresponse data. Greater availability and affordability of software (much of which is free) have played a significant role in helping the techniques gain acceptance and popularity in the broader scientific community. This article surveys the developments and briefly reviews the key ideas behind some of the major algorithms.

Key words: Classification trees; regression trees; machine learning; prediction.

1 Introduction

As we reach the 50th anniversary of the publication of the first regression tree algorithm (Morgan & Sonquist, 1963), it seems appropriate to survey the numerous developments in the field. There have been previous reviews, but some are dated (e.g., Murthy, 1998) and others were written as brief overviews (e.g., Loh, 2008a; 2011; Merkle & Shaffer, 2009; Strobl *et al.*, 2011) or simple introductions intended for non-statistics audiences (e.g., De'ath & Fabricius, 2000; Harper, 2003; Lemon *et al.*, 2005). Owing to the large and increasing amount of literature (in statistics, computer science, and other fields), it is impossible, of course, for any survey to be exhaustive. We have therefore chosen to focus more attention on the major algorithms that have stood the test of time and for which software is widely available. Although we aim to provide a balanced discussion, some of the comments inevitably reflect the opinions of the author.

We say that X is an *ordered* variable if it takes numerical values that have an intrinsic ordering. Otherwise, we call it a *categorical* variable. Automatic Interaction Detection (AID) (Morgan & Sonquist, 1963) is the first regression tree algorithm published in the literature. Starting at the root node, AID recursively splits the data in each node into two children nodes. A split on an ordered variable X takes the form "$X \leq c$". If X has n distinct observed values, there are $(n-1)$ such splits on X. On the other hand, if X is a categorical variable having m distinct observed values, there are $(2^{m-1}-1)$ splits of the form "$X \in A$", where A is a subset of the X values. At any node t, let $S(t)$ denote the set of training data in t and let

[1] This paper is followed by discussions and a rejoinder.

$\bar{y}_t$ be the sample mean of Y in t. Let $\phi(t)$ denote the node "impurity" of t. Using the sum of squared deviations $\phi(t) = \sum_{i \in S(t)} (y_i - \bar{y}_t)^2$, AID chooses the split that minimizes the sum of the impurities in the two children nodes. Splitting stops when the reduction in impurity is less than a preset fraction of the impurity at the root node. The predicted Y value in each terminal node is the node sample mean. The result is a piecewise constant estimate of the regression function.

THeta Automatic Interaction Detection (THAID) (Messenger & Mandell, 1972) extends these ideas to classification, in which Y is a categorical variable. THAID chooses splits to maximize the sum of the number of observations in each modal category (i.e., the category with the most observations). Alternative impurity functions are the entropy, $\phi(t) = -\sum_j p(j|t) \log p(j|t)$, and the Gini index, $\phi(t) = 1 - \sum_j p^2(j|t)$, where $p(j|t)$ is the proportion of class j observations in node t. Messenger & Mandell (1972) attributed the Gini index to Light & Margolin (1971).

Figure 1 shows a classification tree model for the iris data that Fisher (1936) used to introduce linear discriminant analysis (LDA). Four measurements (petal length and width, and sepal length and width) were recorded on 150 iris flowers, with 50 from each of the Setosa, Versicolour, and Virginica types. The tree splits only on petal length and width.

Despite their novelty, or perhaps owing to it, AID and THAID did not initially attract much interest in the statistics community. Einhorn (1972) showed by simulation that AID can severely overfit the data. Doyle (1973) pointed out that if two or more variables are highly correlated, at most one may appear in the tree structure. This problem of *masking* can lead to spurious conclusions about the relative importance of the variables. Bishop *et al.* (1975) criticized AID for ignoring the inherent sampling variability of the data. Around the same time though, the idea of recursive partitioning was gaining steam in the computer science and engineering communities as more efficient algorithms for carrying out the search for splits began to appear (Chou 1969; Henrichon & Fu, 1973; Meisel & Michalopoulos, 1977; Payne & Meisel, 1977; Sethi & Chatterjee, 1991).

2 Classification Trees

We begin with classification trees because many of the key ideas originate here.

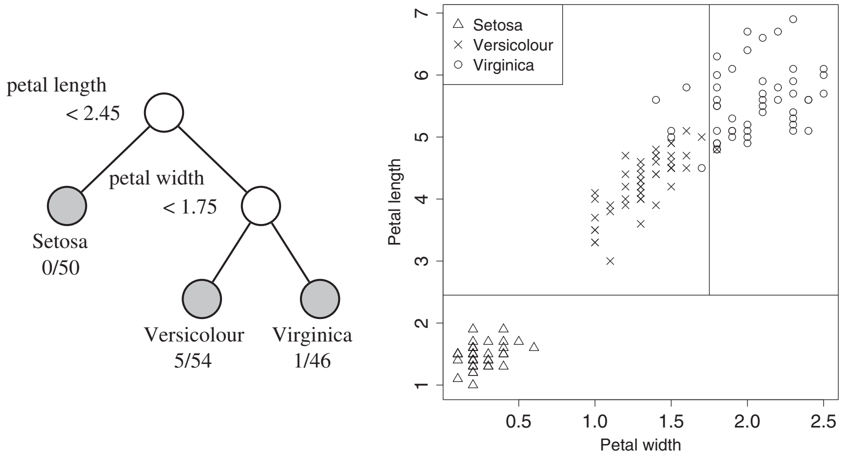


Figure 1. *Classification tree model for iris data. At each intermediate node, an observation goes to the left child node if and only if the stated condition is true. The pair of numbers beneath each terminal node gives the number misclassified and the node sample size.*

2.1 CART

Classification And Regression Trees (CART) (Breiman *et al.*, 1984) was instrumental in regenerating interest in the subject. It follows the same greedy search approach as AID and THAID, but adds several novel improvements. Instead of using stopping rules, it grows a large tree and then prunes the tree to a size that has the lowest cross-validation estimate of error. The pruning procedure itself is ingenious, being based on the idea of weakest-link cutting, with the links indexed by the values of a cost-complexity parameter. This solves the under-fitting and over-fitting problems of AID and THAID, although with increased computation cost. To deal with missing data values at a node, CART uses a series of "surrogate" splits, which are splits on alternate variables that substitute for the preferred split when the latter is inapplicable because of missing values. Surrogate splits are also used to provide an importance score for each X variable. These scores, which measure how well the surrogate splits predict the preferred splits, can help to detect masking. CART can also employ linear splits, that is, splits on linear combinations of variables, by stochastic search. Brown *et al.* (1996) proposed a linear programming solution as an alternative. Breiman *et al.* (1984) obtained conditions for all recursive partitioning techniques to be Bayes risk consistent. CART is available in commercial software. It is implemented as RPART (Therneau & Atkinson, 2011) in the R system (R Core Team 2014).

2.2 CHAID

CHi-squared Automatic Interaction Detector (CHAID) (Kass, 1980) employs an approach similar to stepwise regression for split selection. It was originally designed for classification and later extended to regression. To search for an X variable to split a node, the latter is initially split into two or more children nodes, with their number depending on the type of variable. CHAID recognizes three variable types: categorical, ordered without missing values (called *monotonic*), and ordered with missing values (called *floating*). A separate category is defined for missing values in a categorical variable. If X is categorical, a node t is split into one child node for each category of X. If X is monotonic, t is split into 10 children nodes, with each child node defined by an interval of X values. If X is floating, t is split into 10 children nodes plus one for missing values. Pairs of children nodes are then considered for merging by using Bonferroni-adjusted significance tests. The merged children nodes are then considered for division, again by means of Bonferroni-adjusted tests. Each X variable is assessed with a Bonferroni-adjusted p-value, and the one with the smallest p-value is selected to split the node. CHAID is currently available in commercial software only.

2.3 C4.5

C4.5 (Quinlan, 1993) is an extension of the ID3 (Quinlan, 1986) classification algorithm. If X has m distinct values in a node, C4.5 splits the latter into m children nodes, with one child node for each value. If X is ordered, the node is split into two children nodes in the usual form "$X < c$". C4.5 employs an entropy-based measure of node impurity called *gain ratio*. Suppose node t is split into children nodes $t_1, t_2, \ldots, t_r$. Let $n(t)$ denote the number of training samples in t, and define $\phi(t) = -\sum_j p(j|t) \log p(j|t)$, $f_k(t) = n(t_k)/n(t)$, $\phi_X(t) = \sum_{k=1}^{r} \phi(t_k) f_k(t)$, $g(X) = \phi(t) - \phi_X(t)$, and $h(X) = -\sum_k f_k(t) \log f_k(t)$. The gain ratio of X is $g(X)/h(X)$. Although C4.5 takes almost no time on categorical variable splits, the strategy has the drawback that if X has many categorical values, a split on X may produce children nodes with so few observations in each that no further splitting is possible—see Loh (2008a) for an example. C4.5 trees are pruned with a heuristic formula instead of cross-validation.

If there are missing values, the gain function is changed to $g(X) = F\{\phi(t) - \phi_X(t)\}$, where F is the fraction of observations in a node non-missing in X. The $h(X)$ function is extended by the addition of a "missing value" node t_{r+1} in its formula. If an observation is missing the value of a split variable, it is sent to every child node with weights proportional to the numbers of non-missing observations in those nodes. Empirical evidence shows that C4.5 possesses excellent speed and good prediction accuracy, but its trees are often substantially larger than those of other methods (Lim *et al.*, 2000; Loh, 2009).

Source code for C4.5 can be obtained from www.rulequest.com/Personal/c4.5r8.tar.gz. It is also implemented as J48 in the WEKA (Hall *et al.*, 2009) suite of programs.

2.4 FACT and QUEST

Fast and Accurate Classification Tree (FACT) (Loh & Vanichsetakul, 1988) is motivated by recursive LDA, which generates linear splits. As a result, it splits each node into as many children nodes as the number of classes. To obtain univariate splits, FACT uses analysis of variance (ANOVA) F-tests to rank the X variables and then applies LDA to the most significant variable to split the node. Categorical X variables are transformed first to dummy 0–1 vectors and then converted to ordered variables by projecting the dummies onto the largest discriminant coordinate. Splits on the latter are expressed back in the form $X \in A$. Missing X values are estimated at each node by the sample means and modes of the non-missing ordered and categorical variables, respectively, in the node. Stopping rules based on the ANOVA tests are used to determine the tree size.

One weakness of the greedy search approach of AID, CART, and C4.5 is that it induces biases in variable selection. Recall that an ordered X variable taking n distinct values generates $(n-1)$ splits. Suppose X_1 and X_2 are two such variables with n_1 and n_2 distinct values, respectively. If $n_1 < n_2$ and both variables are *independent* of Y, then X_2 has a larger chance to be selected than X_1. The situation is worse if X_2 is a categorical variable, because the number of splits grows exponentially with n_2. Breiman *et al.* (1984, p. 42) noted this weakness in the CART algorithm, and White & Liu (1994) and Kononenko (1995) demonstrated its severity in C4.5. We will say that an algorithm is *unbiased* if it does not have such biases. Specifically, if all X variables are independent of Y, an unbiased algorithm gives each X the same chance of being selected to split a node.

FACT is unbiased if all the X variables are ordered, because it uses F-tests for variable selection. But it is biased toward categorical variables, because it employs LDA to convert them to ordered variables before application of the F-tests. Quick, Unbiased and Efficient Statistical Tree (QUEST) (Loh & Shih, 1997) removes the bias by using F-tests on ordered variables and contingency table chi-squared tests on categorical variables. To produce binary splits when the number of classes is greater than 2, QUEST merges the classes into two superclasses in each node before carrying out the significance tests. If the selected X variable is ordered, the split point is obtained by either exhaustive search or quadratic discriminant analysis. Otherwise, if the variable is categorical, its values are transformed first to the largest linear discriminant coordinate. Thus, QUEST has a substantial computational advantage over CART when there are categorical variables with many values. Linear combination splits are obtained by applying LDA to the two superclasses. The trees are pruned as in CART.

2.5 CRUISE

Whereas CART always yields binary trees, CHAID and C4.5 can split a node into more than two children nodes, their number depending on the characteristics of the X variable. Classification Rule with Unbiased Interaction Selection and Estimation (CRUISE) (Kim & Loh, 2001) is a descendent of QUEST. It splits each node into multiple children nodes, with their number depending on the number of distinct Y values. Unlike QUEST, CRUISE uses contingency table chi-squared tests for variable selection throughout, with the values of Y forming the rows and the (grouped, if X is ordered) values of X forming the columns of each table. We call these "main effect" tests, to distinguish them from "pairwise interaction" tests that CRUISE also performs, which are chi-squared tests cross-tabulating Y against Cartesian products of the (grouped) values of pairs of X variables. If an interaction test between X_i and X_j, say, is most significant, CRUISE selects X_i if its main effect is more significant than that of X_j, and vice versa. Split points are found by LDA, after a Box–Cox transformation on the selected X variable. Categorical X variables are first converted to dummy vectors and then to their largest discriminant coordinate, following FACT and QUEST. CRUISE also allows linear splits using all the variables, and it can fit a linear discriminant model in each terminal node (Kim & Loh, 2003).

Kim & Loh (2001) showed that CART is biased toward selecting split variables with *more* missing values and biased toward selecting surrogate variables with *fewer* missing values. The cause is due to the Gini index being a function of the class proportions and not the class sample sizes. CRUISE and QUEST are unbiased in this respect. CRUISE has several missing value imputation methods, the default being imputation by predicted class mean or mode, with class prediction based on a non-missing X variable.

2.6 GUIDE

Generalized, Unbiased, Interaction Detection and Estimation (GUIDE) (Loh, 2009) improves upon QUEST and CRUISE by adopting their strengths and correcting their weaknesses. One weakness of CRUISE is that there are many more interaction tests than main effect tests. As a result, CRUISE has a greater tendency to split on variables identified through interaction tests. GUIDE restricts their frequency by using the tests only if no main effect test is significant at a Bonferroni-corrected level. This reduces the amount of computation as well. Further, GUIDE uses a two-level search for splits when it detects an interaction between X_i and X_j, say, at a node t. First, it finds the split of t on X_i and the splits of its two children nodes on X_j that yield the most reduction in impurity. Then it finds the corresponding splits with the roles of X_i and X_j reversed. The one yielding the greater reduction in impurity is used to split t.

Besides univariate splits, GUIDE can employ bivariate linear splits of two X variables at a time. The bivariate linear splits can be given higher or lower priority over univariate splits. In the latter case, linear splits are considered only if no interaction tests are significant after another Bonferroni correction. Although bivariate linear splits may be less powerful than linear splits on all X variables together, the former are still applicable if the number of X variables exceeds the number of observations in the node.

Other improvements in GUIDE include (i) assigning missing categorical values to a "missing" category, (ii) fitting bivariate kernel or nearest-neighbor node models, and (iii) using the node chi-squared test statistics to form an importance score for each variable (Loh, 2012). Smyth *et al*. (1995) and Buttrey & Karo (2002) proposed fitting kernel density estimation and nearest-neighbor models, respectively, in the terminal nodes of a CART tree or a C4.5 tree. Executable codes for CRUISE, GUIDE, and QUEST are distributed free from http://www.stat.wisc.edu/~loh/.

2.7 CTREE and Other Unbiased Approaches

Conditional Inference Trees (CTREE) (Hothorn *et al*., 2006b) is another algorithm with unbiased variable selection. It uses p-values from permutation distributions of influence function-based statistics to select split variables. Monte Carlo or asymptotic approximations to the p-values are employed if they cannot be computed exactly. CTREE does not use pruning; it uses stopping rules based on Bonferroni-adjusted p-values to determine tree size. The algorithm is implemented in the R package PARTY.

Shih (2004), Shih & Tsai (2004), and Strobl *et al.* (2007a) proposed to correct the selection bias of CART by choosing splits based on p-values of the maximal Gini statistics. The solutions are limited, however, to ordered X variables and to classification and piecewise constant regression trees, and they increase computation cost.

2.8 Ensemble, Bayesian, and Other Methods

There is much interest recently on the use of ensembles of classifiers for predictions. In this approach, the predicted value of an observation is based on the majority "vote" from the predicted values of the classifiers in the ensemble. *Bagging* (Breiman, 1996) uses an ensemble of unpruned CART trees constructed from bootstrap samples of the data. *Random forest* (Breiman, 2001) weakens the dependence among the CART trees by using a random subset of X variables for split selection at each node of a tree. Hothorn & Lausen (2005) applied bagging to the original variables as well as the predicted values of other classifiers, such as LDA, nearest neighbor, and logistic regression.

Boosting (Freund & Schapire, 1997) sequentially constructs the classifiers in the ensemble by putting more weight on the observations misclassified in the previous step. Hamza & Larocque (2005) found random forest to be better than boosting CART, but Gashler *et al.* (2008) showed that random forest can perform poorly if there are irrelevant variables in the data. Dietterich (2000) reviewed ensemble methods in the computer science literature.

Another class of ensemble methods is Bayesian model averaging, where prior distributions are placed on the set of tree models and stochastic search is used to find the good ones. Chipman *et al.* (1998) used a prior distribution that can depend on tree size and shape, and Denison *et al.* (1998) used a truncated Poisson prior that puts equal weight on equal-sized trees. For split point selection on an ordered X variable, Chipman *et al.* (1998) used a discrete uniform prior on the observed values of X, and Denison *et al.* (1998) used a continuous uniform distribution on the range of X.

2.9 Importance Scores

Many tree algorithms produce importance scores of the X variables. CART bases the scores on the surrogate splits, but because the latter are subject to selection bias, the scores are similarly biased. Sandri & Zuccolotto (2008) proposed a method to correct the bias. GUIDE uses as importance score a sum of weighted chi-squared statistics over the intermediate nodes, with node sample sizes as weights. A chi-squared approximation to the null distribution of the importance scores is used to provide a threshold for identifying the noise variables. Random forest derives its importance scores from changes in prediction error after random permutation of the X variable values. Strobl *et al.* (2007b) showed that the scores are biased toward correlated variables, and Strobl *et al.* (2008) proposed an alternative permutation scheme as a solution.

2.10 Comparisons

Figures 2 and 3 show the tree models and their partitions given by C4.5, CHAID, CRUISE, CTREE, GUIDE, QUEST, and Recursive PARTitioning (RPART) for the iris data. The X

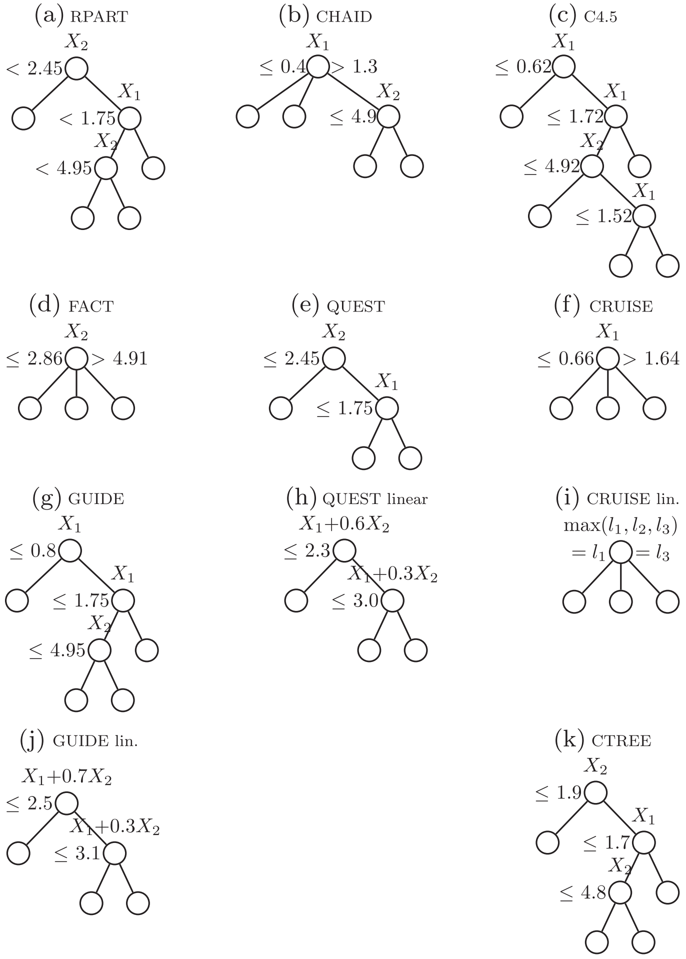


Figure 2. *Classification trees for iris data.* X_1, X_2, X_3, *and* X_4 *denote petal width and length, and sepal width and length, respectively. Functions* $l_1 = -7 - 3X_1 + 9X_2$, $l_2 = -52 + 11X_1 + 21X_2$, $l_3 = -93 + 23X_1 + 25X_2$.

variables are restricted to petal length and width for CHAID and for the CRUISE and QUEST linear split models to allow their partitions to be plotted in the space of these two variables. No such restriction is necessary for the other methods because they only split on these two variables. Although the tree structures may appear different, the methods give the same predictions for a large majority of the observations. The plots show that the CHAID split points are rather poor and that those of C4.5 and CTREE are at observed data values.

Lim *et al.* (2000) compared the prediction accuracy and computation speed of 33 classification algorithms on a large number of data sets without missing values. Twenty-two algorithms were classification trees; two were neural networks; and the others included LDA, nearest neighbor, logistic regression, and POLYchotomous regression and multiple CLASSification (POLYCLASS) (Kooperberg *et al.*, 1997), a logistic regression model based on linear splines and their tensor products. POLYCLASS and logistic regression were found to have the lowest and second lowest, respectively, mean error rates. QUEST with linear splits ranked fourth best overall. POLYCLASS was, however, among the slowest. C4.5 trees had on average about twice as many terminal nodes as QUEST.

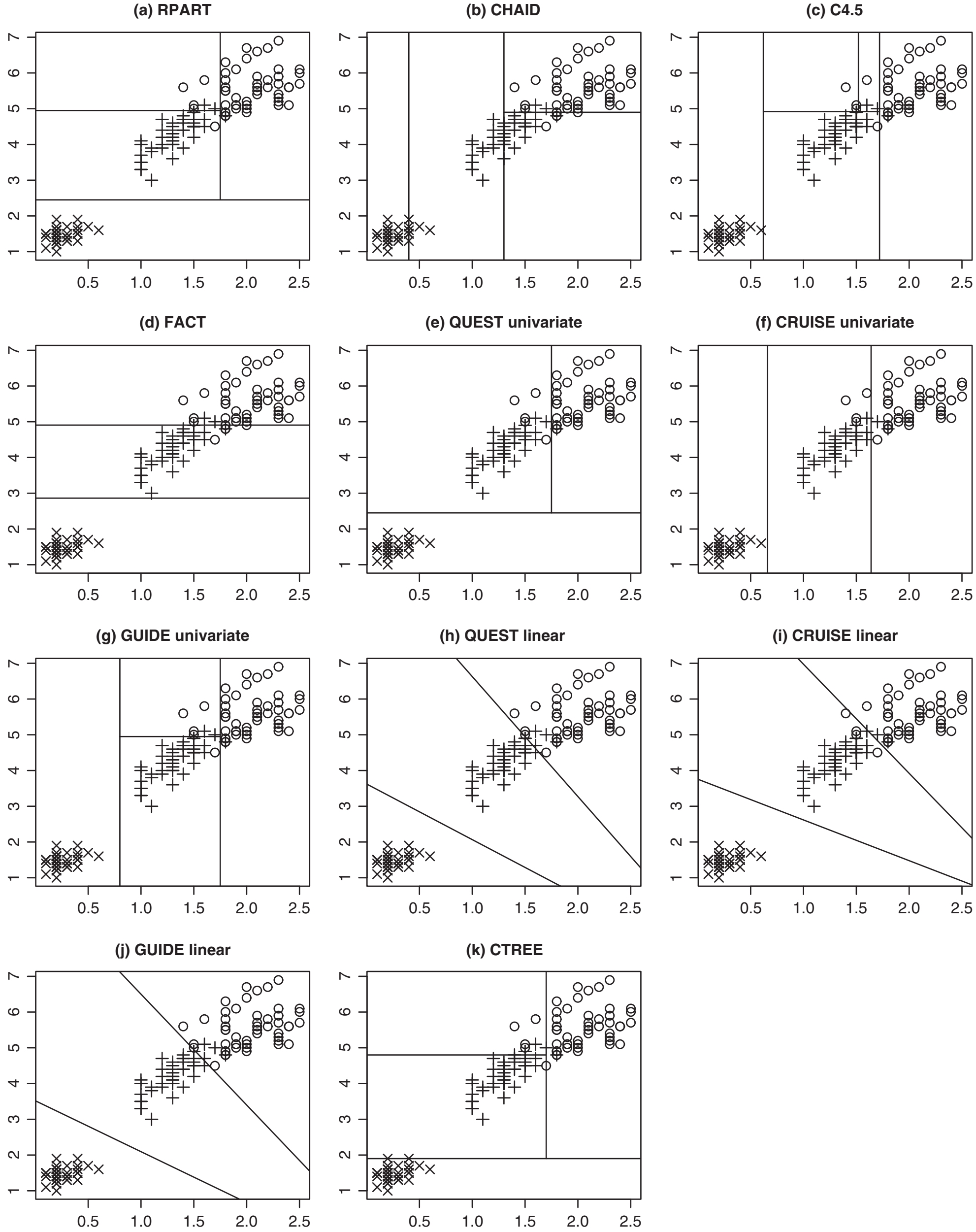


Figure 3. *Plots of petal length versus petal width with classification tree partitions; Setosa, Versicolour, and Virginica are marked by triangles, circles, and crosses, respectively.*

Perlich *et al*. (2004) compared logistic regression with C4.5 and found the former to be more accurate for small sample sizes and the latter better for large sample sizes. Loh (2009) found that among the newer classification tree algorithms, GUIDE had the best combination of accuracy and speed, followed by CRUISE and QUEST.

Ding & Simonoff (2010) studied the effectiveness of various missing value methods for classification trees with binary Y variables. After comparing several types of missing value mechanisms, they concluded that the use of a missing category to handle missing values (as used in CHAID and GUIDE) is best if the test sample has missing values and if missingness is not independent of Y.

3 Regression

AID and CART construct piecewise constant regression trees by using the node mean of Y as predicted value and the sum of squared deviations as node impurity function. Subsequent developments fall under one of two directions: (i) piecewise linear or higher order least-squares models and (ii) piecewise constant or linear models with other loss functions.

3.1 Least Squares

Although it is straightforward conceptually to extend the CART algorithm to piecewise linear models, this can be too time consuming in practice because it fits a linear model in each child node for *every* potential split of a node. To reduce the amount of computation, Alexander & Grimshaw (1996) proposed fitting a simple linear regression model in each node, with the linear predictor being the X variable yielding the smallest sum of squared residuals. M5 (Quinlan, 1992) and its implementation M5′ (Wang & Witten, 1996) fit a piecewise multiple linear tree model by using a less exhaustive but much faster approach. M5 first grows a piecewise constant tree and then fits a stepwise multiple linear model in each node t, using as linear predictors only those variables that are used to split the nodes below t. Thus, M5 avoids having to fit two linear models for every potential split. But because they are originally piecewise constant models, the M5 trees tend to be quite large. Torgo (1997) took a similar approach, but allowed kernel regression and nearest-neighbor models in addition to linear models in the terminal nodes.

Smoothed and Unsmoothed Piecewise POlynomial Regression Trees (SUPPORT) (Chaudhuri *et al.*, 1994) uses a different approach that applies classification tree techniques to the residuals. At each node, it first fits a linear model to the data and classifies the observations into two classes according to the signs of their residuals. Then, as in FACT, it performs two-sample tests of differences between the class means and the class variances for each X variable. The most significant X is selected to split the node with the split point being the average of the two class means. As a result, only one linear model needs to be fitted at each node. Conditions for asymptotic consistency of the function estimate and its derivatives from recursive partitioning methods are given in Chaudhuri *et al.* (1994).

It is harder to achieve unbiased variable selection in piecewise multiple linear regression trees because an X variable can be used in one or both of two roles: (a) as a candidate for split selection (called a “split” variable) and (b) as a linear predictor in the linear model (called a “fit” variable). Because the residuals are uncorrelated with split-and-fit variables, but are not necessarily uncorrelated with split-only variables, the p-values of the former tend to be stochastically larger than those of the latter. As a result, SUPPORT is biased toward selecting split-only variables. One way to correct the bias is to scale down the p-values (or scale up the test statistic values) of the split-and-fit variables. GUIDE (Loh, 2002) uses bootstrap calibration to find the scale factor.

There are also extensions in other directions. CTREE (Hothorn *et al.*, 2006b) uses permutation tests to construct unbiased piecewise constant regression trees for univariate, multivariate, ordinal, or censored Y variables. Regression Trunk Approach (RTA) (Dusseldorp & Meulman, 2004) combines the regression tree approach with linear regression to detect interactions between a treatment variable and ordered X variables. RTA first fits a linear main effects model to all the data. Then it uses the residuals to construct a piecewise constant regression tree model for each treatment group. Simultaneous Threshold Interaction Modeling Algorithm (STIMA) (Dusseldorp *et al.*, 2010) improves upon RTA by estimating the linear regression and tree models simultaneously.

Ciampi *et al.* (2002) proposed the use of soft thresholds (sigmoidal functions) instead of hard thresholds (indicator functions) for splits on ordered variables. Chipman *et al.* (2002) extended the Bayesian approach of Chipman *et al.* (1998) to piecewise linear regression trees. Fan & Gray (2005) and Gray & Fan (2008) used a genetic algorithm for tree construction. Guerts *et al.* (2006) proposed selecting splits from randomly picked subsets of split variables and split points. Su *et al.* (2004) extended CART by using maximum likelihood to choose the splits in a piecewise constant regression model, but instead of the usual negative log-likelihood, they used Akaike information criterion (AIC) and an independent test sample to prune the tree.

Yildiz & Alpaydin (2001, 2005a, 2005b) and Gama (2004) compared linear splitting with linear node modeling. Their results suggest that linear splitting and linear fitting yield similar gains in prediction accuracy, and both are superior to univariate splits and constant node models. Loh *et al.* (2007) showed that the prediction accuracy of piecewise linear regression trees can be improved by truncating or Winsorizing the fitted values. Kim *et al.* (2007) and Loh (2008b) showed that using only one or two regressor variables in the node models can be useful for data visualization.

3.1.1 Baseball example

To compare the methods, we use them to predict the 1987 salaries (in thousands of dollars) of 263 professional baseball players. The data, from Statlib (http://lib.stat.cmu.edu), contain 22 predictor variables, of which six are categorical, as shown in Table 1. They were used for a poster session contest at an American Statistical Association meeting. After reviewing the submitted solutions and performing their own analysis, Hoaglin & Velleman (1995) chose the following model fitted to log-salary:

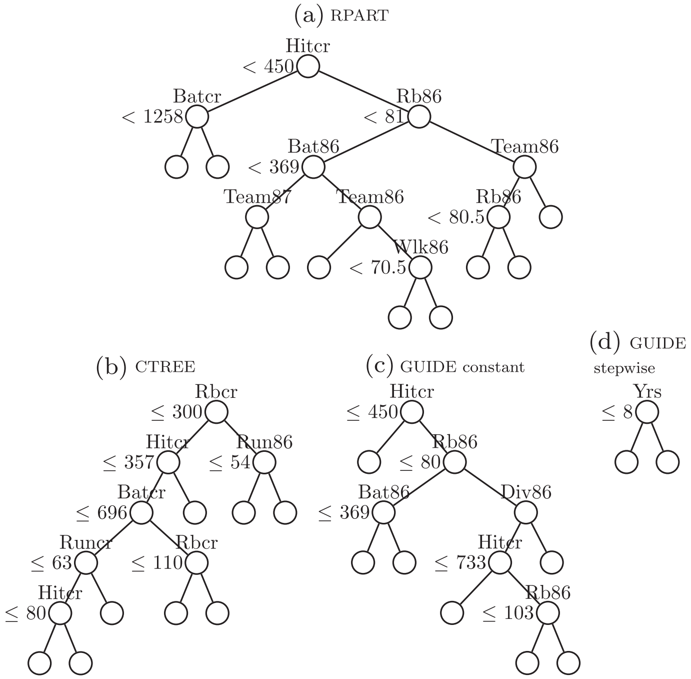
$$\log(\texttt{Salary}) = \beta_0 + \beta_1 \texttt{Runcr}/\texttt{Yrs} + \beta_2 \sqrt{\texttt{Run86}} + \beta_3 \min[(\texttt{Yrs}-2)_+, 5] + \beta_4 (\texttt{Yrs}-7)_+.$$

Reasons for the data transformations include dealing with the range restriction on salary, collinearity, variance heterogeneity, and other difficulties typically encountered in linear regression.

As tree models are not limited by these difficulties, we fit salary without transformations to any variables. Figure 4 shows the RPART, CTREE, and two GUIDE tree models (one with

Table 1. *Predictor variables for baseball data.*

Bat86	# times at bat in 1986	Batcr	# times at bat during career
Hit86	# hits in 1986	Hitcr	# hits during career
Hr86	# home runs in 1986	Hrcr	# home runs during career
Run86	# runs in 1986	Runcr	# runs during career
Rb86	# runs batted in 1986	Rbcr	# runs batted in during career
Wlk86	# walks in 1986	Wlkcr	# walks during career
Leag86	league at end of 1986 (2 cat.)	Leag87	league at start of 1987 (2 cat.)
Team86	team at end of 1986 (24 cat.)	Team87	team at start of 1987 (24 cat.)
Div86	division at end of 1986 (2 cat.)	Yrs	# years in the major leagues
Pos86	position in 1986 (23 cat.)	Puto86	# put outs in 1986
Asst86	# assists in 1986	Err86	# errors in 1986


(a) RPART

(b) CTREE

(c) GUIDE constant

(d) GUIDE stepwise

Figure 4. *Regression tree models for baseball data.*

a constant fitted in each node and the other with a stepwise linear model in each node). The initial splits of the RPART and GUIDE piecewise constant trees are identical except for one split point. RPART, however, shows a preference for splits on the two `Team` variables, which have more than eight million ($2^{23} - 1$) splits each. None of the piecewise constant models select `Yrs`, which features prominently in the Hoaglin–Velleman model. The piecewise linear GUIDE model, in contrast, splits just once, and on `Yrs`. The tree is short because each node is fitted with a multiple linear model.

Figure 5 plots the observed versus fitted values of the tree models and those from ordinary least squares, Hoaglin and Velleman, Random forest, and GUIDE forest. The two forest methods are similar, their only difference being that Random forest uses the CART algorithm for split selection and GUIDE forest uses its namesake algorithm. Although the ordinary least-squares model compares quite favorably with the others, it has a fair number of negative fitted values. The piecewise constant models are easily identified by the vertical stripes in the plots. As every model has trouble predicting the highest salaries, we conclude these salaries cannot be adequately explained by the variables in the data. We note that the GUIDE stepwise model (which uses a single tree) fits the data about as well as Random forest (which uses 500 trees) and that GUIDE forest fits the data visibly better than Random forest here.

3.2 Poisson, Logistic, and Quantile Regression

Efforts have been made to extend regression tree methods beyond squared error loss. Ciampi (1991) extended CART to fit a generalized linear regression model in each node, choosing the split that most reduces the sum of deviances in the children nodes. The trees are pruned by significance tests or the AIC.

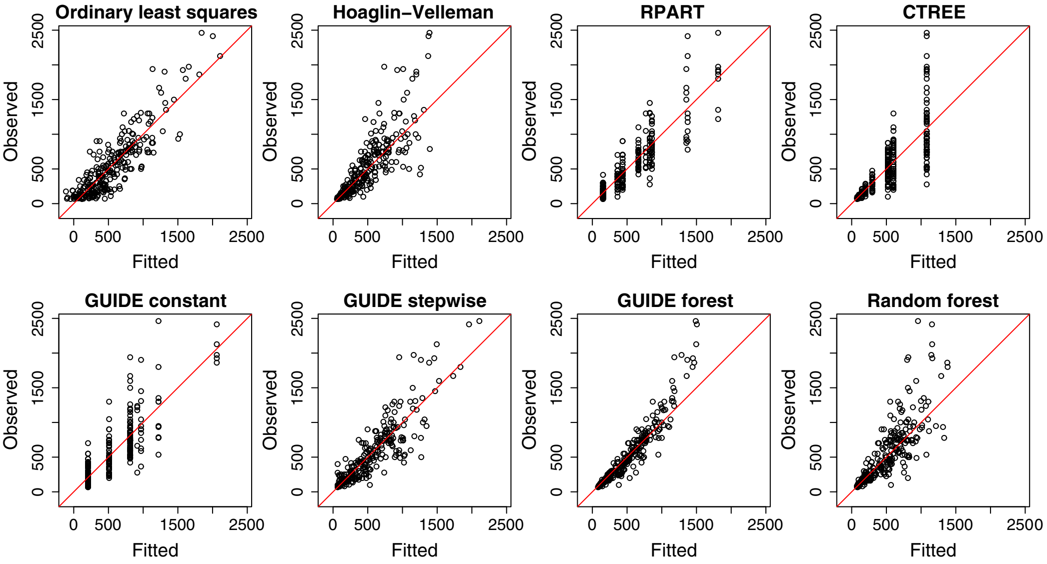


Figure 5. *Plots of observed versus fitted values for baseball data.*

Chaudhuri *et al.* (1995) extended SUPPORT to piecewise linear Poisson and logistic regression. For Poisson regression, they used adjusted Anscombe residuals. For logistic regression, they estimated the probability function at each node by using both logistic regression and a nearest-neighbor method and defined the "residual" as the difference between the two estimated values. Ahn & Chen (1997) used similar ideas to construct logistic regression trees for clustered binomial data.

Chaudhuri & Loh (2002) and Loh (2006b) extended GUIDE to quantile and Poisson regression, respectively. LOTUS (Chan & Loh, 2004; Loh, 2006a) uses the same ideas to fit a linear logistic regression model in each node. To attain unbiasedness, LOTUS uses a trend-adjusted chi-squared test for X variables that are used for splitting and fitting.

Landwehr *et al.* (2005) constructed logistic regression trees by using the LogitBoost (Friedman *et al.*, 2000) technique to fit a logistic regression model to each node and the C4.5 method to split the nodes. Choi *et al.* (2005) used the GUIDE split selection method to construct regression trees for overdispersed Poisson data. Lee & Yu (2010) employed the CART approach to model ranking response data.

MOdel-Based recursive partitioning (MOB) (Zeileis *et al.*, 2008) fits least-squares, logistic, and other models, using the score functions of M-estimators. Special cases include standard maximum and pseudo-likelihood models. It achieves unbiased variable selection by choosing split variables on the basis of structural break tests for the score function. The split point (for ordered X) or split set (for categorical X) is obtained by maximizing the change in an objective function. The tree is not pruned; instead, stopping rules based on Bonferroni-adjusted p-values are used to control tree growth. MOB has been extended to psychometric models such as the Bradley–Terry model (Strobl *et al.*, 2011) and the Rasch model (Strobl *et al.*, 2010) and to generalized linear models and maximum likelihood models with linear predictors (Rusch & Zeileis, 2013). The algorithm is not unbiased if some X variables are used for both fitting and splitting.

3.3 Censored Response Variables

Gordon & Olshen (1985) extended CART to censored response variables by fitting a Kaplan–Meier survival curve to the data in each node and using as node impurity the minimum Wasserstein distance between the fitted Kaplan–Meier curve and a point-mass function. Segal (1988) chose splits to maximize a measure of between-node difference instead of within-node homogeneity. The measures include two-sample rank statistics such as the logrank test (Peto & Peto, 1972). Davis & Anderson (1989) adapted CART to fit a constant hazard to each node, using exponential log-likelihood as impurity function.

Ciampi *et al.* (1986) compared stepwise Cox regression, correspondence analysis, and recursive partitioning models for censored response data. Stepwise Cox regression finds a prognostic index (a linear combination of the X variables) and then partitions the data at the quartiles of the index. Correspondence analysis converts each ordered X into a vector of indicator variables (one indicator for each observed value) and groups the Y values into a small number of categories. The first canonical variable is used as the prognostic index to partition the data. Recursive partitioning converts each categorical variable into a vector of indicators and partitions the data on the indicators. RECursive Partitioning and AMalgamation (RECPAM) (Ciampi *et al.*, 1988) extends these ideas to allow merging of terminal nodes. For regression with censored response data, the split criterion is a dissimilarity measure such as likelihood ratio or the logrank, Wilcoxon–Gehan, and Kolmogorov–Smirnov statistics. For classification, the split criterion is the multinomial likelihood. Splits may be univariate or Boolean intersections of univariate splits. Missing values may be given a separate category or be dealt with through surrogates splits as in CART. Importance scores are given by the sum of the dissimilarities of each variable over all the nodes. Tree size is determined by cross-validation or AIC.

LeBlanc & Crowley (1992) fitted a proportional hazards model with the hazard rate in node t being $\lambda_t(u) = \theta_t \lambda_0(u)$, where θ_t is a constant and $\lambda_0(u)$ is the baseline hazard function. For tree construction and pruning, the baseline cumulative hazard $\Lambda_0(u)$ is estimated by the Nelson–Aalen estimator (Aalen, 1978). The θ_t is a one-step estimate from the full maximum likelihood. Split selection and pruning are based on the one-step deviance. The rest of the algorithm follows CART. LeBlanc & Crowley (1993) used logrank test statistics to select splits and the sum of logrank test statistics over intermediate nodes as measure of goodness of split for pruning. Crowley *et al.* (1995) noted that, without a node impurity measure, cross-validation pruning cannot be employed with Segal's (1988) approach. They also showed that the split selection method of Gordon & Olshen (1985), based on L_p and Wasserstein metrics, can perform poorly even with mild censoring. Bacchetti & Segal (1995) considered left-truncated survival times and splits on time-dependent covariates, by letting each observation go into both child nodes at the same time. This approach precludes classifying each subject in exactly one terminal node. They noted that splits on time-dependent variables can yield unstable Kaplan–Meier estimates of the survival functions.

Jin *et al.* (2004) used between-node variance of restricted mean survival time as node impurity to construct survival trees. For clustered survival data, Gao *et al.* (2004) fitted a proportional hazards model with subject frailty to each node; see also Su & Fan (2004) and Fan *et al.* (2006). Hothorn *et al.* (2004) used bagging to obtain an ensemble of survival trees and obtained a Kaplan–Meier survival curve for each subject from the bootstrap observations belonging to the same terminal nodes as the subject. Molinaro *et al.* (2004) used inverse probabilities of censoring as weights to construct trees for censored data. Hothorn *et al.* (2006a) employed the idea to predict mean log survival time from random forests with case weights. Ishwaran *et al.* (2004) applied the random forest (Breiman, 2001) technique to construct relative risk forests using piecewise proportional hazards; see also Ishwaran *et al.* (2006) who obtained variable importance scores. Clarke & West (2008) fitted Bayesian Weibull tree models to uncensored survival data with split criteria on the basis of Bayes factors, and Garg *et al.* (2011) used a similar approach to fit exponential models. Cho & Hong (2008) constructed median regression trees by using the Buckley–James (1979) method to estimate the survival times of the censored observations and then fitting a piecewise constant quantile regression model to the completed data.

Loh (1991) and Ahn & Loh (1994) extended SUPPORT to piecewise proportional hazards models. Ahn (1994a, 1994b, 1996a, 1996b) did the same for piecewise parametric survival models.

3.4 Longitudinal and Multiresponse Variables

Segal (1992) was among the first to extend CART to longitudinal data by using as node impurity a function of the likelihood of an autoregressive or compound symmetry model. If there are missing response values, the expectation–maximization (EM) algorithm (Laird & Ware, 1982) is used to estimate the parameters. Abdolell *et al.* (2002) used the same approach, but with a likelihood-ratio test statistic as impurity function.

Zhang (1998) extended CART to multiple binary response variables, using as node impurity the log-likelihood of an exponential family distribution that depends only on the linear terms and the sum of second-order products of the responses. Zhang & Ye (2008) applied the technique to ordinal responses by first transforming them to binary-valued indicator functions; see also Zhang & Singer (2010). Their approach requires covariance matrices to be computed at every node.

For longitudinal data observed at very many times, Yu & Lambert (1999) treated each response vector as a random function and reduced the dimensionality of the data by fitting each trajectory with a spline curve. Then they used the estimated coefficients of the basis functions as multivariate responses to fit a regression tree model.

De'ath (2002) avoided the problem of covariance estimation by using as node impurity the total sum of squared deviations from the mean across the response variables. Larsen & Speckman (2004) used the Mahalanobis distance, but estimated the covariance matrix from the whole data set.

Hsiao & Shih (2007) showed that multivariate extensions of CART are biased toward selecting variables that allow more splits. They proposed using chi-squared tests of conditional independence (conditioning on the components of the response vector) of residual signs versus grouped X values to select the split variables. The method may lack power if the effects of the X variables are not in the same direction across all the Y variables.

Lee (2005) applied the GUIDE approach to multiple responses with ordered X variables by fitting a generalized estimating equation model to the data in each node and taking the average of the Pearson residuals over the responses variables, for each observation. The observations are classified into two groups according to the signs of the average residuals, and the X with the smallest p-value from two-sample t-tests is chosen to split the node. Although unbiased, the method is not sensitive to all response trajectory shapes. Loh & Zheng (2013) solved this problem by using the residual vector patterns, rather than their averages, to choose the split variables. The solution is applicable to data observed at random time points.

Sela & Simonoff (2012) proposed the RE-EM method (Sela & Simonoff, 2011), which fits a model consisting of the sum of a random effects term and a tree-structured term.

The procedure mimics the EM algorithm (Laird & Ware, 1982) by iterating between estimating the tree structure, assuming that the random effects are correct, and estimating the random effects, assuming that the tree structure is correct.

4 Conclusion

Research in classification and regression trees has seen rapid growth, and applications are increasing at an even greater rate. Interpretability of the tree structures is a strong reason for their popularity among practitioners, but so are reasonably good prediction accuracy, fast computation speed, and wide availability of software.

Despite 50 years of progress, however, many hard problems remain. One of them is how best to deal with missing values. Ding & Simonoff (2010) made a good start, but their results apply only to classification with binary responses. Much of the difficulty is due to missing value techniques interacting with other algorithm components and with the type of variables and the causes of the missingness. Another challenging problem is how to deal with time-varying covariates in regression trees for longitudinal and censored response data. This is not surprising given that traditional (non-tree) solutions require various model assumptions that are hard to justify in a tree-structured framework. Computationally efficient approaches to search for effective linear combination splits is yet another elusive problem, especially in the regression context. It is harder if the linear splits are coupled with linear model fits in the nodes, because the latter are already quite effective in reducing prediction error. Thus, the linear splits need to be so much more effective to justify the increase in computation and loss of interpretability. In this age of large data sets, there are also new problems, such as algorithms that scale well with sample size (Dobra & Gehrke, 2002; Gehrke, 2009) and incremental tree construction algorithms for streaming data (Alberg *et al.*, 2005; Potts & Sammut, 2011; Taddy *et al.*, 2012).

The rise of ensemble and other methods has made it difficult for single-tree methods to compete in terms of prediction accuracy alone. Comparisons based on real and simulated data sets suggest that the accuracy of the best single-tree algorithm is on average about 10% less than that of a tree ensemble, although it is certainly not true that an ensemble always beats a single tree (Loh, 2009). An ensemble of, say, 500 trees is, however, often practically impossible to understand. Importance scores can rank order the variables, but they do not explain how the variables influence the predictions. Thus, the biggest advantage of single-tree models remains their model interpretability, although interpretability rapidly diminishes with tree size. But because inferences from the tree structures can be compromised by selection bias, future algorithms will need to be unbiased to be useful in applications where interpretability is important.

Acknowledgements

The author is grateful to Probal Chaudhuri for suggesting and encouraging him to write this survey and to four anonymous reviewers for their helpful comments. Research for this article was partially supported by US Army Research Office grant W911NF-09-1-0205, National Institutes of Health grant P50CA143188, and Eli Lilly and Company. CART® is a registered trademark of California Statistical Software, Inc.

References

Aalen, O.O. (1978). Nonparametric inference for a family of counting processes. *Ann. Stat.*, **6**, 701–726.

Abdolell, M., LeBlanc, M., Stephens, D. & Harrison, R.V. (2002). Binary partitioning for continuous longitudinal data: categorizing a prognostic variable. *Stat. Med.*, **21**, 3395–3409.

Ahn, H. (1994a). Tree-structured exponential regression modeling. *Biometrical J.*, **36**, 43–61.

Ahn, H. (1994b). Tree-structured extreme value model regression. *Commun. Stat.-Theor. M.*, **23**, 153–174.

Ahn, H. (1996a). Log-gamma regression modeling through regression trees. *Commun. Stat.-Theor. M.*, **25**, 295–311.

Ahn, H. (1996b). Log-normal regression modeling through recursive partitioning. *Comput. Stat. Data Anal.*, **21**, 381–398.

Ahn, H. & Chen, J. (1997). Tree-structured logistic model for over-dispersed binomial data with application to modeling developmental effects. *Biometrics*, **53**, 435–455.

Ahn, H. & Loh, W.-Y. (1994). Tree-structured proportional hazards regression modeling. *Biometrics*, **50**, 471–485.

Alberg, D., Last, M. & Kandel, A. (2012). Knowledge discovery in data streams with regression tree methods. *Wil. Interdiscip. Rev.: Data Mining and Knowledge Disc.*, **2**, 69–78.

Alexander, W.P. & Grimshaw, S.D. (1996). Treed regression. *J. Comput. Graph. Stat.*, **5**, 156–175.

Bacchetti, P. & Segal, M.R. (1995). Survival trees with time-dependent covariates: application to estimating changes in the incubation period of AIDS. *Lifetime Data Anal.*, **1**, 35–47.

Bishop, Y.M.M., Fienberg, S.E. & Holland, P.W. (1975). *Discrete Multivariate Analysis.* Cambridge, MA: MIT Press.

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

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, **24**, 123–140.

Breiman, L. (2001). Random forests. *Mach. Learn.*, **45**, 5–32.

Brown, D.E., Pittard, C.L. & Park, H. (1996). Classification trees with optimal multivariate decision nodes. *Pattern. Recogn. Lett.*, **17**, 699–703.

Buckley, J.J. & James, I.R. (1979). Linear regression with censored data. *Biometrika*, **66**, 429–436.

Buttrey, S.E. & Karo, C. (2002). Using k-nearest-neighbor classification in the leaves of a tree. *Comput. Stat. Data Anal.*, **40**, 27–37.

Chan, K.-Y. & Loh, W.-Y. (2004). LOTUS: an algorithm for building accurate and comprehensible logistic regression trees. *J. Comput. Graph. Stat.*, **13**, 826–852.

Chaudhuri, P., Huang, M.-C., Loh, W.-Y. & Yao, R. (1994). Piecewise-polynomial regression trees. *Stat. Sinica*, **4**, 143–167.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y. & Yang, C.-C. (1995). Generalized regression trees. *Stat. Sinica*, **5**, 641–666.

Chaudhuri, P. & Loh, W.-Y. (2002). Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli*, **8**, 561–576.

Chipman, H.A., George, E.I. & McCulloch, R.E. (1998). Bayesian CART model search. *J. Amer. Statist. Assoc.*, **93**, 935–948.

Chipman, H.A., George, E.I. & McCulloch, R.E. (2002). Bayesian treed models. *Mach. Learn.*, **48**, 299–320.

Cho, H.J. & Hong, S.-M. (2008). Median regression tree for analysis of censored survival data. *IEEE T. Syst. Man Cy. A.*, **38**, 715–726.

Choi, Y., Ahn, H. & Chen, J. (2005). Regression trees for analysis of count data with extra Poisson variation. *Comput. Stat. Data Anal.*, **49**, 893–915.

Chou, P.A. (1991). Optimal partitioning for classification and regression trees. *IEEE T. Pattern. Anal.*, **13**, 340–354.

Ciampi, A. (1991). Generalized regression trees. *Comput. Stat. Data. Anal.*, **12**, 57–78.

Ciampi, A., Couturier, A. & Li, S.L. (2002). Prediction trees with soft nodes for binary outcomes. *Stat. Med.*, **21**, 1145–1165.

Ciampi, A., Hogg, S.A., McKinney, S. & Thiffault, J. (1988). RECPAM: A computer program for recursive partition and amalgamation for censored survival data and other situations frequently occurring in biostatistics. *Comput. Meth. Prog. Bio.*, **26**, 239–256.

Ciampi, A., Thiffault, J., Nakache, J-P. & Asselain, B. (1986). Stratification by stepwise regression, correspondence analysis and recursive partition: A comparison of three methods of analysis for survival data with covariates. *Comput. Stat. Data Anal.*, **4**, 185–204.

Clarke, J. & West, M. (2008). Bayesian Weibull tree models for survival analysis of clinico-genomic data. *Stat. Meth.*, **5**, 238–262.

Crowley, J., LeBlanc, M., Gentleman, R. & Salmon, S. (1995). Exploratory methods in survival analysis. In *Analysis of Censored Data*, vol. 27, Eds. H.L. Koul & J.V. Deshpande, pp. 55–77. Institute of Mathematical Statistics, IMS Lecture Notes-Monograph Series: Hayward, CA.

Davis, R.B. & Anderson, J.R. (1989). Exponential survival trees. *Stat. Med.*, **8**, 947–961.

De'ath, G. (2002). Multivariate regression trees: a new technique for modeling species-environment relationships. *Ecology*, **83**, 1105–1117.

De'ath, G. & Fabricius, K.E. (2000). Classification and regression trees: A powerful yet simple technique for ecological data analysis. *Ecology*, **81**, 3178–3192.

Denison, D.G.T., Mallick, B.K. & Smith, A.F.M. (1998). A Bayesian CART algorithm. *Biometrika*, **85**, 363–377.

Dietterich, T.G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems, LBCS-1857,* pp. 1–15. New York: Springer.

Ding, Y. & Simonoff, J.S. (2010). An investigation of missing data methods for classification trees applied to binary response data. *J. Mach. Learn. Res.*, **11**, 131–170.

Dobra, A. & Gehrke, J.E. (2002). SECRET: A scalable linear regression tree algorithm. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 481–487. Edmonton, Canada: ACM Press.

Doyle, P. (1973). The use of automatic interaction detector and similar search procedures. *Oper. Res. Quart.*, **24**, 465–467.

Dusseldorp, E., Conversano, C. & Van Os, B.J. (2010). Combining an additive and tree-based regression model simultaneously: STIMA. *J. Comput. Graph. Stat.*, **19**, 514–530.

Dusseldorp, E. & Meulman, J.J. (2004). The regression trunk approach to discover treatment covariate interaction. *Psychometrika*, **69**, 355–374.

Einhorn, H.J. (1972). Alchemy in the behavioural sciences. *Public Opin. Quart.*, **36**, 367–378.

Fan, G. & Gray, J.B. (2005). Regression tree analysis using TARGET. *J. Comput. Graph. Stat.*, **14**, 1–13.

Fan, J., Su, X., Levine, R.A., Nunn, M.E. & LeBlanc, M. (2006). Trees for correlated survival data by goodness of split, with applications to tooth prognosis. *J. Amer. Statist. Assoc.*, **101**, 959–967.

Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems. *Ann. Eugenic.*, **7**(2), 179–188.

Freund, Y. & Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, **55**(1), 119–139.

Friedman, J., Hastie, T. & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Ann. Stat.*, **38**(2), 337–374.

Gama, J. (2004). Functional trees. *Mach. Learn.*, **55**, 219–250.

Gao, F., Manatunga, A.K. & Chen, S. (2004). Identification of prognostic factors with multivariate survival data. *Comput. Stat. Data Anal.*, **45**, 813–824.

Garg, L., McClean, S., Meenan, B.J. & Millard, P. (2011). Phase-type survival trees and mixed distribution survival trees for clustering patients' hospital length of stay. *Informatica*, **22**, 57–72.

Gashler, M., Giraud-Carrier, C.G. & Martinez, T.R. (2008). Decision tree ensemble: small heterogeneous is better than large homogeneous. In *Seventh International Conference on Machine Learning and Applications*, pp. 900–905. Washington, DC: IEEE Computer Society.

Gehrke, J. (2009). Scalable decision tree construction. In *Encyclopedia of Database Systems*, Eds. L. Liu & T. Ozsu, pp. 2469–2474. New York: Springer.

Gordon, L. & Olshen, R.A. (1985). Tree-structured survival analysis. *Cancer Treat. Rep.*, **69**, 1065–1069.

Gray, J.B. & Fan, G. (2008). Classification tree analysis using TARGET. *Comput. Stat. Data Anal.*, **52**, 1362–1372.

Guerts, P., Ernst, D. & Wehenkel, L. (2006). Extremely randomized trees. *Mach. Learn.*, **63**, 3–42.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, **11**, 10–18.

Hamza, M. & Larocque, D. (2005). An empirical comparison of ensemble methods based on classification trees. *J. Stat. Comput. Sim.*, **75**, 629–643.

Harper, P.R. (2005). A review and comparison of classification algorithms for medical decision making. *Health Policy*, **71**, 315–331.

Henrichon, Jr., E.G. & Fu, K.-S. (1969). A nonparametric partitioning procedure for pattern classification. *IEEE T. Comput.*, **C-18**, 614–624.

Hoaglin, D.C. & Velleman, P.F. (1995). A critical look at some analyses of major league baseball salaries. *Am. Stat.*, **49**, 277–285.

Hothorn, T., Bühlmann, P., Dudoit, S., Molinaro, A. & van der Laan, M.J. (2006a). Survival ensembles. *Biostatistics*, **7**, 355–373.

Hothorn, T., Hornik, K. & Zeileis, A. (2006b). Unbiased recursive partitioning: a conditional inference framework. *J. Comput. Graph. Stat.*, **15**, 651–674.

Hothorn, T. & Lausen, B. (2005). Bundling classifiers by bagging trees. *Comput. Stat. Data Anal.*, **49**, 1068–1078.

Hothorn, T., Lausen, B., Benner, A. & Radespiel-Tröger, M. (2004). Bagging survival trees. *Stat. Med.*, **23**, 77–91.

Hsiao, W.-C. & Shih, Y.-S. (2007). Splitting variable selection for multivariate regression trees. *Stat. Probab. Lett.*, **77**, 265–271.

Ishwaran, H., Blackstone, E.H., Pothier, C.E. & Lauer, M. (2004). Relative risk forests for exercise heart rate recovery as a predictor of mortality. *J. Amer. Statist. Assoc.*, **99**, 591–600.

Ishwaran, H., Kogalur, U.B., Blackstone, E.H. & Lauer, M. (2006). Random survival forests. *Ann. Appl. Stat.*, **2**, 841–860.

Jin, H., Lu, Y., Stone, K. & Black, D.M. (2004). Survival analysis based on variance of survival time. *Med. Decis. Making*, **24**, 670–680.

Kass, G.V. (1980). An exploratory technique for investigating large quantities of categorical data. *Ann. Appl. Stat.*, **29**, 119–127.

Kim, H. & Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *J. Amer. Statist. Assoc.*, **96**, 589–604.

Kim, H. & Loh, W.-Y. (2003). Classification trees with bivariate linear discriminant node models. *J. Comput. Graph. Stat.*, **12**, 512–530.

Kim, H., Loh, W.-Y., Shih, Y.-S. & Chaudhuri, P. (2007). Visualizable and interpretable regression models with good prediction power. *IIE Transactions*, **39**, 565–579.

Kononenko, I. (1995). On biases in estimating multi-valued attributes. In *14th International Joint Conference on Artificial Intelligence*, pp. 1034–1040. Burlington, MA: Morgan Kaufmann.

Kooperberg, C., Bose, S. & Stone, C.J. (1997). Polychotomous regression. *J. Amer. Statist. Assoc.*, **92**, 117–127.

Laird, N.M. & Ware, J.H. (1982). Random-effects models for longitudinal data. *Biometrics*, **38**, 963–974.

Landwehr, N., Hall, M. & Frank, E. (2005). Logistic model trees. *Mach. Learn.*, **59**, 161–205.

Larsen, D.R. & Speckman, P.L. (2004). Multivariate regression trees for analysis of abundance data. *Biometrics*, **60**, 543–549.

LeBlanc, M. & Crowley, J. (1992). Relative risk trees for censored survival data. *Biometrics*, **48**, 411–425.

LeBlanc, M. & Crowley, J. (1993). Survival trees by goodness of split. *J. Amer. Statist. Assoc.*, **88**, 457–467.

Lee, P.H. & Yu, P.L.H. (2010). Distance-based tree models for ranking data. *Comput. Stat. Data Anal.*, **54**, 1672–1682.

Lee, S.K. (2005). On generalized multivariate decision tree by using GEE. *Comput. Stat. Data Anal.*, **49**, 1105–1119.

Lemon, S.C., Roy, J., Clark, M.A., Friedman, P.D. & Rakowski, W. (2003). Classification and regression tree analysis in public health: Methodological review and comparison with logistic regression. *Ann. Behav. Med.*, **26**, 172–181.

Light, R.J. & Margolin, B.H. (1971). An analysis of variance for categorical data. *J. Amer. Statist. Assoc.*, **66**, 534–544.

Lim, T.-S., Loh, W.-Y. & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Mach. Learn.*, **40**, 203–228.

Loh, W.-Y. (1991). Survival modeling through recursive stratification. *Comput. Stat. Data Anal.*, **12**, 295–313.

Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Stat. Sinica*, **12**, 361–386.

Loh, W.-Y. (2006a). Logistic regression tree analysis. In *Handbook of engineering statistics,* Ed. H. Pham, pp. 537–549. New York: Springer.

Loh, W.-Y. (2006b). Regression tree models for designed experiments. In *Second E. L. Lehmann Symposium*, Vol. 49, Ed. J. Rojo. IMS Lecture Notes-Monograph Series, pp. 210–228. Bethesda, MD: Institute of Mathematical Statistics.

Loh, W.-Y. (2008a). Classification and regression tree methods. In *Encyclopedia of Statistics in Quality and Reliability*, Eds. F. Ruggeri, R. Kenett & F.W. Faltin, pp. 315–323. Chichester, UK: Wiley.

Loh, W.-Y. (2008b). Regression by parts: fitting visually interpretable models with GUIDE. In *Handbook of Data Visualization,* Eds. C. Chen, W. Härdle & A. Unwin. pp. 447–469. New York: Springer.

Loh, W.-Y. (2009). Improving the precision of classification trees. *Ann. Appl. Stat.*, **3**, 1710–1737.

Loh, W.-Y. (2011). Classification and regression trees. *Wil. Interdiscip. Rev.: Data Mining and Knowledge Disc.*, **1**, 14–23.

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.

Loh, W.-Y., Chen, C.-W. & Zheng, W. (2007). Extrapolation errors in linear model trees. *ACM Trans. On Knowledge Disc. From Data*, **1**. DOI: 10.1145/1267066.1267067.

Loh, W.-Y. & Shih, Y.-S. (1997). Split selection methods for classification trees. *Stat. Sinica*, **7**, 815–840.

Loh, W.-Y. & Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion). *J. Amer. Statist. Assoc.*, **83**, 715–728.

Loh, W.-Y. & Zheng, W. (2013). Regression trees for longitudinal and multiresponse data. *Ann. Appl. Stat.*, **7**(1), 495–522.

Meisel, W.S. & Michalopoulos, D.A. (1973). A partitioning algorithm with application in pattern classification and the optimization of decision trees. *IEEE T. Comput.*, **C-22**, 93–103.

Merkle, E.C. & Shaffer, V.A. (2011). Binary recursive partitioning: Background, methods, and application to psychology. *Brit. J. Math. Stat. Psy.*, **64**, 161–181.

Messenger, R. & Mandell, L. (1972). A modal search technique for predictive nominal scale multivariate analysis. *J. Amer. Statist. Assoc.*, **67**, 768–772.

Molinaro, A.M., Dudoit, S. & van der Laan, M.J. (2004). Tree-based multivariate regression and density estimation with right-censored data. *J. Multivariate Anal.*, **90**, 154–177.

Morgan, J.N. & Sonquist, J.A. (1963). Problems in the analysis of survey data, and a proposal. *J. Amer. Statist. Assoc.*, **58**, 415–434.

Murthy, S.K. (1998). Automatic construction of decision trees from data: a multi-disciplinary survey. *Data Min. Knowl. Disc.*, **2**, 345–389.

Payne, H.J. & Meisel, W.S. (1977). An algorithm for constructing optimal binary decision trees. *IEEE T. Comput.*, **C-26**, 905–916.

Perlich, C., Provost, F. & Simonoff, J.S. (2004). Tree induction vs. logistic regression: A learning-curve analysis. *J. Mach. Learn. Res.*, **4**, 211–255.

Peto, R. & Peto, J. (1972). Asymptotically efficient rank invariant test procedures. *J. R. Stat. Soc. Ser.-A*, **135**, 185–207.

Potts, D. & Sammut, C. (2005). Incremental learning of linear model trees. *Mach. Learn.*, **61**, 5–48.

Quinlan, J.R. (1986). Induction of decision trees. *Mach. Learn.*, **1**, 81–106.

Quinlan, J.R. (1992). Learning with continuous classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence,* pp. 343–348. Singapore: World Scientific.

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

R Core Team. (2014). *R: A language and environment for statistical computing*, R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

Rusch, T. & Zeileis, A. (2013). Gaining insight with recursive partitioning of generalized linear models. *J. Stat. Comput. Sim.*, **83**(7), 1301–1315.

Sandri, M. & Zuccolotto, P. (2008). A bias correction algorithm for the Gini variable importance measure in classification trees. *J. Comput. Graph. Stat.*, **17**, 611–628.

Segal, M.R. (1988). Regression trees for censored data. *Biometrics*, **44**, 35–47.

Segal, M.R. (1992). Tree structured methods for longitudinal data. *J. Amer. Statist. Assoc.*, **87**, 407–418.

Sela, R.J. & Simonoff, J.S. (2011). *Reemtree: Regression trees with random effects*. R package version 0.90.3.

Sela, R.J. & Simonoff, J.S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Mach. Learn.*, **86**, 169–207.

Sethi, I.K. & Chatterjee, B. (1977). Efficient decision tree design for discrete variable pattern recognition problems. *Pattern Recogn.*, **9**, 197–206.

Shih, Y.S. (2004). A note on split selection bias in classification trees. *Comput. Stat. Data Anal.*, **45**, 457–466.

Shih, Y.S. & Tsai, H.W. (2004). Variable selection bias in regression trees with constant fits. *Comput. Stat. Data Anal.*, **45**, 595–607.

Smyth, P., Gray, A. & Fayyad, U.M. (1995). Retrofitting decision tree classifiers using kernel density estimation. In *Proceedings of the 12th International Conference on Machine Learning,* pp. 506–514. Burlington, MA: Morgan Kaufmann.

Strobl, C., Boulesteix, A. & Augustin, T. (2007a). Unbiased split selection for classification trees based on the Gini index. *Comput. Stat. Data Anal.*, **52**, 483–501.

Strobl, C., Boulesteix, A., Kneib, T., Augustin, T. & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, **9**, 307.

Strobl, C., Boulesteix, A., Zeileis, A. & Hothorn, T. (2007b). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, **8**, 25.

Strobl, C., Kopf, J. & Zeileis, A. (2010). A new method for detecting differential item functioning in the Rasch model. Tech. Rep. 92, Department of Statistics, Ludwig-Maximilians-Universitaet Muenchen.

Strobl, C., Malley, J. & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application and characteristics of classification and regression trees, bagging and random forests. *Psychol. Methods*, **14**, 323–348.

Strobl, C., Wickelmaier, F. & Zeileis, A. (2011). Accounting for individual differences in Bradley–Terry models by means of recursive partitioning. *J. Educ. Behav. Stat.*, **36**(2), 135–153.

Su, X. & Fan, J. (2004). Multivariate survival trees: A maximum likelihood approach based on frailty models. *Biometrics*, **60**, 93–99.

Su, X. G., Wang, M. & Fan, J.J. (2004). Maximum likelihood regression trees. *J. Comput. Graph. Stat.*, **13**, 586–598.

Taddy, M.A., Gramacy, R.B. & Polson, N.G. (2011). Dynamic trees for learning and design. *J. Amer. Statist. Assoc.*, **106**, 109–123.

Therneau, T.M. & Atkinson, B. (2011). *rpart: Recursive partitioning*, R port by Brian Ripley. R package version 3.1-50.

Torgo, L. (1997). Functional models for regression tree leaves. In *Proceedings of the Fourteenth International Conference on Machine Learning,* Ed. D.H. Fisher. pp. 385–393. Burlington, MA: Morgan Kaufmann.

Wang, Y. & Witten, I.H. (1996). *Induction of model trees for predicting continuous classes*, Working paper series, Department of Computer Science, University of Waikato.

White, A.P. & Liu, W.Z. (1994). Technical note: bias in information-based measures in decision tree induction. *Mach. Learn.*, **15**, 321–329.

Yildiz, O.T. & Alpaydin, E. (2001). Omnivariate decision trees. *IEEE T. Neural. Networ.*, **12**, 1539–1546.

Yildiz, O.T. & Alpaydin, E. (2005a). Linear discriminant trees. *Int. J. Pattern Recogn.*, **19**, 323–353.

Yildiz, O.T. & Alpaydin, E. (2005b). Model selection in omnivariate decision trees. In *Machine learning: ECML 2005, Proceedings*, Vol. 3720, Eds. J. Gama, R. Camacho, P. Brazdil, A. Jorge & L. Torgo. Lecture Notes in Artificial Intelligence, pp. 473–484. Berlin: Springer-Verlag.

Yu, Y. & Lambert, D. (1999). Fitting trees to functional data, with an application to time-of-day patterns. *J. Comput. Graph. Stat.*, **8**, 749–762.

Zeileis, A., Hothorn, T. & Hornik, K. (2008). Model-based recursive partitioning. *J. Comput. Graph. Stat.*, **17**, 492–514.

Zhang, H. (1998). Classification trees for multiple binary responses. *J. Amer. Statist. Assoc.*, **93**, 180–193.

Zhang, H. & Singer, B.H. (2010). *Recursive Partitioning and Applications*, 2nd ed. New York: Springer.

Zhang, H. & Ye, Y. (2008). A tree-based method for modeling a multivariate ordinal response. *Stat. and Its Interface*, **1**, 169–178.

[*Received February 2012, accepted February 2013*]

International Statistical Review

International Statistical Review (2014), 82, 3, 349–352 doi:10.1111/insr.12059

Discussions

Carolin Strobl

Universität Zürich, Zurich, Switzerland
E-mail: carolin.strobl@psychologie.uzh.ch

With 'Fifty Years of Classification and Regression Trees', Wei-Yin Loh has given a concise historical overview of the central developments in recursive partitioning. It is interesting to read how the successive improvements were triggered by their predecessor algorithms—especially as the author has (co-)authored many of the milestones in our field. Moreover, I appreciate his emphasis on open-source implementations, which make the methodology available to scientists from all disciplines and all around the world.

I have two questions to the author and would like to add a short comment about variable importance measures.

1 Missing Value Handling

As has been pointed out for several algorithms in the paper, the treatment of missing values is an interesting aspect that distinguishes recursive partitioning techniques from other statistical methods. The two approaches specific for recursive partitioning are (i) surrogate variables, which are correlated with the primary splitting variable and can thus be used to replace it for processing observations with missing values in the primary variable, and (ii) creation of separate nodes for missing values.

Both approaches are distinct from ad hoc approaches classically—while often unreflectedly—used in statistical analyses, such as case-wise deletion, but also from more advanced approaches like (multiple) imputation techniques. While the advantage of preserving observations with missing values and thus avoiding data loss is straightforward, I wonder how these approaches relate to the concepts of missing completely at random (MCAR), missing at random (MAR) and missing not at random (cf., e.g. Little & Rubin, 1986).

Hapfelmeier *et al.* (2014) systematically investigate the effects of MCAR, MAR and missing not at random on a random forest variable importance measure modified to be able to deal with missing values (including surrogate variables), but I am not aware of any such studies for the approach of creating separate nodes for missing values.

Do you know of any or can you infer how this approach performs under the different missingness mechanisms? In particular, is creating a separate node for missing values informative for predicting the response variable under the MCAR and MAR schemes, and/or could it be informative for narrowing down the missingness mechanism itself?

2 Ignorance of Variable Selection Bias

Because both Wei-Yin Loh and I have worked in this area, I assume we both find the development of unbiased split selection criteria to be one of the most important improvements over the early recursive partitioning algorithms. As pointed out in the paper, variable selection

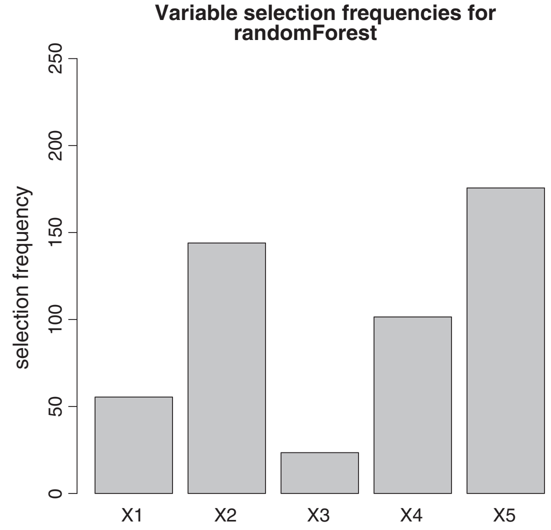


Figure 1. *Variable selection frequencies for a random forest algorithm with biased split selection (*`randomForest,` *Liaw & Wiener, 2002). In the underlying simulation design, only the second predictor variable X_2 is informative, but its selection frequency is outperformed by the irrelevant variable X_5, which is preferred only because it has more categories.*

bias is defined as an artificial preference for variables offering more cutpoints—even if all variables are noise variables containing no information.

What we should probably point out more clearly though is that this also means that when a predictor variable offering few cutpoints is in fact associated with the response—and thus should be found relevant by any reasonable statistical learning technique—it may still be outperformed by a less informative or even irrelevant competitor, just because the latter offers more cutpoints. This is illustrated in Figure 1 from Strobl *et al.* (2007) (for the selection frequencies of a random forest, but of course, the same can be observed for the selection frequencies of single trees).

In the simulation design underlying Figure 1, the predictor variables systematically differ in the number of cutpoints they offer: X_1 was generated from a normal distribution (thus offering a high number of different cutpoints), X_2 from a binomial distribution (offering only one cutpoint) and X_3 to X_5 from a multinomial distribution with 4, 10 and 20 categories, respectively (offering again an increasing number of cutpoints). Only X_2 was simulated to have a strong effect on the response class, whereas X_1, X_3, X_4 and X_5 are entirely uninformative noise variables.

Yet, we can clearly see in Figure 1 that the relevant variable X_2 is outperformed by the irrelevant noise variable X_5, which is preferred solely because it has more categories. (If the effect size of X_2 is modelled to be more moderate, it is also outperformed by noise variables with less categories.)

One should think that the results shown here, and in many previous studies that Wei-Yin Loh has summarized in his paper, are so clear that any statistically educated person should never want to use a biased recursive partitioning algorithm again. Yet I encounter so many cases where biased recursive partitioning algorithms are still employed in both applied and methodological publications—including some of those cited in ‘Fifty Years of Classification and Regression Trees’.

I really wonder why this is the case. Does it mean that the authors of those publications do not consider variable selection bias an issue of concern or willingly ignore decades of research? Or rather that we have not managed to bring our results to the attention of a broader scientific audience? Or maybe even that—at first sight—recursive partitioning looks so easy that anyone can do it without bothering to read up on it? I would be very interested to hear your opinion.

3 Variable Importance Measures

When giving up the interpretability of single trees for the stability of ensemble methods, variable importance measures are the only means to tease out at least some information from the otherwise black box. As Wei-Yin Loh has pointed out, these variable importance measures only provide a summarized impression and cannot be interpreted with respect to the direction or actual form of the relationship between predictor variables and response. Still, they can serve as a valuable tool in application areas where exploratory screening (cf., e.g. Lunetta *et al.*, 2004; Bureau *et al.*, 2005) is the only way to narrow down the number of candidate variables that need to be considered in more detail.

Even though many disciplines with a strong tradition in hypothesis-driven research, such as psychology, are still somewhat shy about these types of procedures, they have their right to exist as one legitimate means of generating hypotheses when no other means is available, as pointed out by Strobl (2013). What is crucial to note, however, is that when machine learning or other statistical techniques are used for screening or automated variable selection (cf., e.g. Diaz-Uriarte & de Andrés, 2006; Rodenburg *et al.*, 2008, for random-forest-based approaches), a newly drawn sample must be used to later conduct statistical significance tests on the selected variables. In some cases, it might even be possible to experimentally test their effects (e.g. by 'knocking out' a previously identified candidate gene).

To conclude with the issue of variable importance measures, let me add a short specification of the works of Strobl *et al.* (2007, 2008).

In Strobl *et al.* (2007), it is shown that—unsurprisingly—random forests built from trees with biased split selection criteria also show variable selection bias (as was illustrated in Figure 1) and that this bias also transfers to the Gini and permutation variable importance measures. However, what was very surprising to us was that even when random forests are built from trees with unbiased split selection criteria, like in the `cforest` function available in the R-package `party`, the widely used bootstrap sampling induces another source of bias, which again affects the variable selection frequencies but also results in an increased variance for the permutation variable importance. This is the reason that we discourage the use of bootstrap sampling and employ subsampling as the default in `cforest`.

Strobl *et al.* (2008), on the other hand, consider the consequences of correlations between predictor variables, which had previously been noted by Archer & Kimes (2008) and Nicodemus & Shugart (2007). In this situation, it is not *per se* clear how a good variable importance measure should behave, and even for parametric models like multiple linear regression, a variety of variable importance measures have been suggested (cf., e.g. Azen & Budescu, 2003), which vary in their particular treatment of correlated variables.

It is a matter of taste or philosophy—rather than an objectively defined bias in the statistical sense—how a variable importance measure should behave in the presence of correlated variables. My impression from speaking to applied researchers was, however, that they were interpreting the random forest permutation importance similar to the coefficients of a multiple regression model, which reflect the impact of a variable given all other variables in the model—which is not how Breiman's original permutation variable importance works. Therefore, we developed a conditional permutation scheme available for `cforest`, which more closely mimics the behaviour of multiple regression coefficients (that is, however, computationally only feasible if the number of correlated variables is not too high).

References

Archer, K.J. & Kimes, R.V. (2008). Empirical characterization of random forest variable importance measures. *Comput. Stat. Data Anal.*, **52**(4), 2249–2260.

Azen, R. & Budescu, D.V. (2003). The dominance analysis approach for comparing predictors in multiple regression. *Psychol. Methods*, **8**(2), 129–48.

Bureau, A., Dupuis, J., Falls, K., Lunetta, K.L., Hayward, B., Keith, T.P. & Eerdewegh, P.V. (2005). Identifying SNPs predictive of phenotype using random forests. *Genet. Epidemiol.*, **28**(2), 171–182.

Diaz-Uriarte, R. & de Andrés, S.A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, **7**(3).

Hapfelmeier, A., Hothorn, T., Ulm, K. & Strobl, C. (2014). A new variable importance measure for random forests with missing data. *Stat. Comput.*, **24**(1), 21–34.

Liaw, A. & Wiener, M. (2002). Classification and regression by `randomForest`. *R News*, **2**(3), 18–22.

Little, R. & Rubin, D. (1986). *Statistical Analysis with Missing Data.* New York: John Wiley & Sons, Inc.

Lunetta, K.L., Hayward, L.B., Segal, J. & Eerdewegh, P.V. (2004). Screening large-scale association study data: Exploiting interactions using random forests. *BMC Genetics*, **5**(32).

Nicodemus, K. & Shugart, Y.Y. (2007). Impact of linkage disequilibrium and effect size on the ability of machine learning methods to detect epistasis in case–control studies. In *Proceedings of the Sixteenth Annual Meeting of the International Genetic Epidemiology Society*, Vol. 31, North Yorkshire, UK, pp. 611.

Rodenburg, W., Heidema, A.G., Boer, J.M., Bovee-Oudenhoven, I.M., Feskens, E.J., Mariman, E.C. & Keijer, J. (2008). A framework to identify physiological responses in microarray based gene expression studies: selection and interpretation of biologically relevant genes. *Genet. Epidemiol.*, **33**(1), 78–90.

Strobl, C. (2013). Data mining. In *The Oxford Handbook on Quantitative Methods*, Ed. T. Little, pp. 678–700. USA, Chapter 29: Oxford University Press.

[*Received March 2014, accepted March 2014*]

International Statistical Review (2014), 82, 3, 352–357 doi:10.1111/insr.12065

Antonio Ciampi

Department of Epidemiology and Biostatistics, McGill University,1020 Pine Ave. West, Montreal H3A 1A2, Quebec, Canada
E-mail: antonio.ciampi@mcgill.ca

Fifty years ago, Morgan & Sonquist (1963) introduced the now famous AID algorithm. AID uses a sample to construct a tree-structured predictor for a specified continuous variable y given a specified vector of covariates x. The result is therefore, in contemporary language, a *regression* tree. The first paper to introduce *classification* trees in the modern sense appeared 9 years later, in 1972: it presented the algorithm THAID, as mentioned in the review (REF). We have to wait 12 more years to see both kinds of trees reunited in the very influential book by Breiman *et al.* (1984), ‘Classification and Regression Trees’. This title is often abbreviated as CART, which is somewhat confusing, because the acronym CART™ also denotes the proprietary software associated to the book. Notwithstanding the great merits of the CART book, it is more than fair to consider the 1963 paper as the seminal work for the *research area* known as ‘Classification and Regression Trees’, the object of Prof. Loh’s excellent review. One could argue that the 1963 paper, together with the CART book, is also at the origin of flexible statistical modelling (beyond variable selection algorithms in regression) such as MARS (Friedman, 1991) and PIMPLE (Breiman, 1991), and, indeed, of *statistical* machine learning (Hastie *et al.*, 2009) . Yet one cannot disagree with the author’s decision of restricting the review to classification and regression trees, and to draw the line at 'forests' and similar ensemble learning. As the old saying goes, 'Grab all, loose all'.

But what is exactly 'Classification and Regression Trees' as a research area? The answer is open to debate. One might say that it includes research on algorithms that construct a tree from data through some kind of recursive partitioning coupled with a rule or set of rules to determine tree size. It should be noted that partitioning also includes some choices as regards the handling of missing data. Choosing how to partition and how to determine tree size is not trivial, and there is a bewildering array of perfectly respectable approaches to these tasks, often leading to minimally divergent results. One might add that the terminal nodes (leaves) of the tree represent a simple prediction or classification rule, depending on whether we are concerned, respectively, with regression or classification trees. How simple? In a strict sense, the rule should be constant on a leaf, i.e. the same prediction/classification is attached to all observational units belonging to the same leaf.

Prof. Loh's review goes a little beyond the strict definition. Firstly, it does include a number of regression tree algorithms, including his own, that fit linear models at each node, thus relaxing the requirement of a constant predictor at each leaf. A minor criticism is that it would have been useful to explicitly note that the idea of fitting linear models in terminal nodes is not exclusive to *regression* trees but can also be implemented for *classification* trees; however, doing this would somewhat disrupt the scheme of the review, which is based on keeping regression and classification separate.

Secondly, the author also considers tree construction algorithms for predicting count data, censored data, multivariate binary data and longitudinal continuous data (functional data). He may have included, but this is by no means an important flaw, trees for predicting multivariate continuous data (beyond longitudinal data) (Gillo & Shelly, 1974): indeed, in one of the RECPAM articles (Ciampi *et al.*, 1991) that he reviews, there are examples of such trees. Minor details aside, the author commendably transcends the old identification of regression trees with trees to predict a continuous, scalar variable.

Thirdly, this review also mentions, although not in great depth, that 'hard' partitioning may be replaced by 'soft' partitioning, i.e. at a given node, an observational unit may be assigned to the issuing branches probabilistically rather than sharply.

Finally, the author discusses some global tree-construction algorithms, which look for the optimal tree within (a large subset of) all possible trees: an application of the genetic algorithm (REF), as well as two Bayesian 'model averaging' approaches are mentioned (REF).

In developing the review, again the author shows wisdom in concentrating on algorithms that have been extensively applied and validated. The variations in partitioning rules (including the treatment of missing data) and size determining rules are clearly if succinctly outlined. Comparative work is cited, and a simple and enlightening original comparison of the performance of several algorithms on a 'classic' data set is presented. The author cites in detail his own work, and I find this totally acceptable: indeed, one of Prof. Loh's major accomplishments is that of having blazed a trail within all the possible variants of tree-growing algorithms obtaining superior accuracy of prediction, high computational efficiency and major reduction of the inherent biases in the original CART (1984) approach.

One feature of this review I have particularly appreciated is the stress on tree algorithms that were developed within the machine-learning tradition at a time when the compartmentalisation separating statisticians from computer scientist was watertight. Indeed, Quinlan's work was just as influential among computer scientists as the CART book was among statisticians: the review clearly re-establishes the balance. In my opinion, if tree research will continue to advance in the near future, it will be because disciplinary boundaries are falling. Nowadays it is not rare to find, especially in the new generations, accomplished researchers who are excellent in both statistics and computing, regardless of their disciplinary background.

And now the hard question would be, is there a future for tree research? Prof. Loh points out that there are hard problems left to solve within the strict definition of regression and classification trees. He cites, in particular, the handling of missing data, the inclusion in tree-growing algorithms of longitudinal *predictor variables* and the introduction of splits based not only on unique variables (monothetic splits or nodes) but also on linear combinations of the original variables. Moreover, he hints at new developments that include incremental tree construction algorithms for streaming data: again, such algorithms go a little beyond the strict definition of classification and regression trees, which were originally conceived in a static setting, with a well-defined 'learning' data set. However, the main message of his concluding remarks is that tree-growing researchers are faced with a real dilemma: *either* one sticks to the classic definition of trees, and in so doing, accepts intrinsic limitations in predictive accuracy; *or* one pursues predictive accuracy by extending the definition of tree-growing algorithms towards ensemble learning, and sacrifices, in exchange, the advantage of highly interpretable predictions. It is difficult not to agree with this view, which I would define as 'realistically pessimist'.

However, when faced with a dilemma of this importance, it may be useful to step back and take a fresh look at the premises that have lead us to the dilemma. Here is a short list of questions that occurred or re-occurred to me while reading Prof. Loh's paper. I use these questions as headings for grouping some considerations related to them.

1 What is the Root of this Dilemma?

The essential feature of tree growing, which is also the reason for its popularity, is the reduction of one *global* optimisation problem–given a learning set, which is considered as a sample from a target population, find the *best* predictor–to a sequence of *local* problems of decreasing sizes according to the general cognitive strategy of 'divide and conquer'. However, no matter how large our learning set is, one is led very quickly to work locally on fairly small data sets: and this is perhaps the root of most problems. The smaller the subsamples, the more variable and unstable are the choices of splits, and the less generalisable are the results to other future samples from the same population. There is no way out of this, *unless* we broaden somewhat the definition of tree growing, to make it *a little less local*. This leads to the next question.

2 Should We Redefine Tree Growing and How?

The author and I agree on the fact that ensemble learning, at least as realised in available algorithms, is only distantly related to tree growing. From the point of view of 'tree growers', the loss in interpretability of such algorithms is too large. To take the example of random forest, the trees of the forest are far too numerous to help developing an interpretation; moreover, there is too much randomness in the generation of each tree.

On the other hand, some promising novel ideas have been put forward leading to algorithms that deserve to be considered as part of the tree-growing family. The Bayesian tree approach, although similar to ensemble learning, does recover some interpretability. Indeed, the analyst has the choice of using model averaging for prediction, while basing interpretation on a (usually) small number of trees: the one(s) with largest posterior probability.

TARGET, a tree-growing approach based on the genetic algorithm, also yields a 'best' tree, although it does *not* proceed by recursive partitioning. However, it is still not known whether in practice the theoretical superiority of the global search does translate into substantially superior predictive accuracy.

Trees with soft nodes were proposed in an attempt to gain predictive accuracy while mitigating the inevitable loss of interpretability. Predictive accuracy is increased by using at each node, *all* data–but with observational units weighted according to the probability of belonging to the node. Loss of interpretability is mitigated by retaining the monothetic feature of classical trees (one split, one variable). However, trees with soft nodes do not seem very useful when there are many categorical predictors. Also, empirical results obtained so far seem to indicate that the gain in predictive accuracy of soft trees with respect to hard trees is real but unimpressive.

All in all, it seems that the most promising ways to extend tree growing beyond its strict definition is to look for an interesting compromise between interpretability and predictive accuracy. But. . .

3 What is Interpretability? or, Better, What Kind of Interpretability do We Really Need?

At first sight, interpretability of a tree seems to hinge on the monothetic nature of the nodes. However, a closer look suggests that this point of view may be misleading: monothetic splits are *not* always what we need. In fact, as already noted, Prof. Loh identifies the introduction of splits based on linear combination of variables as one of the most important open problems in tree-growing research. When do we need to depart from monothetic splits? For instance, consider medical data: typically, predictors are categorical variables based on qualitative observations of symptoms and signs, and/or imprecise measurements of indices that are known to be, at best, proxy of some underlying construct, e.g. ‘cardiovascular health’. Is it really useful to choose, say, ‘elevated total blood cholesterol’ to create a node? Looking for splits based on a linear combination of variables is a natural alternative to looking for a monothetic split; however, how to do this remains problematic, and in fact, we risk to loose interpretability *without* improving predictive accuracy. So, it may be that new ideas are needed to look for polythetic splits: such ideas may arise from a creative interplay of clinical expertise and statistical modelling. Using techniques such as PLS regression at each node (Eriksson *et al.*, 2009), one may extract from the data a split defining statement of the kind clinicians are used to while making diagnosis and prognosis, e.g. ‘*if* the subject has one or more characteristics of the following list. . . ., *then go left*’.

The other essential pillar of interpretability for a tree-based predictor is its simple architecture: a hierarchy of (hard) nodes. It is possible, in my opinion, to make this framework more flexible without completely loosing interpretability. Perhaps we may consider interpretable an architecture consisting of hierarchically structured ‘black boxes’, each being based on a limited number of variables that in some intuitive sense ‘go together’. Moreover, ‘soft nodes’ rather than ‘hard nodes’ could link these black boxes. For example, suppose we want to predict cardiovascular mortality: we may aim to construct a predictor by stringing together a black box based on measurements of blood lipid levels, another black box based on family history data and yet another black box based on demographics. If such a predictor works well, it is possible that a knowledgeable user may find it interpretable. Arguably, he or she may *prefer* this alternative view, as it recognises some of the complexities of the specific predictive task. Now, such system of black boxes linked by soft nodes already exists in machine learning, and is known as ‘hierarchy of experts’ (Jacobs *et al.*, 1991). However, to the best of my knowledge, there is no popular algorithm for *constructing* hierarchy of experts from data, including, as possible, domain specific knowledge. In other words, the concept of hierarchy of experts does provide an excellent framework to imagine algorithms, but is not (yet) a ready-to-use discovery tool. There is here a great opportunity for tree growers: they could use their unique expertise to build problem-specific hierarchy of experts using both data and knowledge bases.

The last question concerns the role of tree-growing research in the context of new challenges arising from the increasing complexity of available data. Volume can be seen as an aspect of complexity; in this respect, I will not add to what Prof. Loh has already mentioned in this review, citing, among others, recent papers on tree-growing algorithm for streaming data. Instead, I wish to briefly discuss another type of complexity, which cannot be dealt with without rethinking prediction and prediction accuracy.

4 What Kind of Prediction Tools do We Really Need to Explore New Types of Data?

Again, I will discuss an example from clinical biostatistics. It becomes increasingly common to collect longitudinal data not only for a particular outcome variable but also for several clinical indices and for several categorical outcome variables. In other words, data become available that summarise the history of a disease as observed on a population of patients over a time window of considerable width. Clearly, a first task for the analyst is to develop the appropriate statistical models for the stochastic process underlying life history data: this task that has been successfully accomplished for a broad variety of situations (REF) (Skrondal & Rabe-Hesketh, 2004; Tenenhaus *et al.*, 2005; Vermunt, 1997). But then, typically, the analyst is also asked to assess the impact of covariates, e.g. patient and treatment characteristics, on the type of disease history that a patient is likely to experience. This is a prediction task, in a very true sense, but is not a standard one: we are very far from the classical problem of predicting a continuous or categorical variable. The statistician's automatic reflex would be to develop some (generalised) linear regression model that should describe the dependence of *some features* of the disease history process on the covariate of interest. However, these features will be represented by a high-dimensional parameter, so that a hypothetic regression model would be extremely hard to interpret. In contrast, a tree-growing approach, if it could be developed, would lead to a fairly straightforward interpretation. *If it could be developed...* Developing this approach is a serious but not impossible task. The tree-growing approach has been formulated and reformulated in abstract terms by several authors, leading to some of the extensions reviewed by Prof. Loh. Further and bolder developments are possible. Conceptually, all we need to do is to define a reasonable measure of 'goodness of split' for the appropriate stochastic process underlying the available data. *If this can be accomplished*, then virtually any tree-growing algorithm can be adapted to the new situation. The adaptation will be, in general, far from trivial and will require new statistical and computational developments: in other words, a great amount of original tree research may be produced, well beyond the present perspective.

To conclude, I wish to thank Prof. Loh for an excellent review of the status of tree research as it celebrates its 50th anniversary. Because I recognise that it would be very hard to do better, I have focussed on potential for future development. I am cautiously optimistic about the *next* 50 years of tree research. The reason for my optimism is the increasing cooperation of researchers from several disciplines that have in the past ignored each other, often 'rediscovering the wheel'. The reason for my caution is that the task of forming the next generation of tree researchers is fraught with many obstacles, but a discussion of this is well beyond the scope of my contribution.

References

Breiman, L. (1991). The Π method for estimating multivariate functions from noisy data. *Technometrics*, **33**, 125–143.

Ciampi, A., du Berger, R., Taylor, G. & Thiffault, J. (1991). RECPAM: a computer program for recursive partition and amalgamation for survival data and other situations frequently occurring in biostatistics. III. Classification according to a multivariate construct. Application to data on Haemophilus influenzae type B meningitis. *Comput. Methods and Prog. in Biomed.*, **36**, 51–61.

Eriksson, L., Trygg, J. & Wold, S. (2009). PLS-trees®, a top-down clustering approach. *J. Chemom.*, **23**, 569–580.

Friedman, J.H. (1991). Multivariate adaptive regression splines. *The Ann. of Stat.*, **19**, 1–67.

Gillo, M.W. & Shelly, M.W. (1974). Predictive modeling of multi-variable and multivariate data. *J. Amer. Stat. Assoc.*, **69**, 646–653.

Hastie, T., Tibshirani, R. & Friedman, J.H. (2009). *The Elements of Statistical Learning*, 2nd ed. New York: Springer.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. & Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Comput.*, **3**, 79–87.

Skrondal, A. & Rabe-Hesketh, S. (2004). *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models.* Boca Raton: Chapman and Hall/CRC.

Tenenhaus, M., Esposito Vinzi, V., Chatelinc, Y.M. & Lauro, C. (2005). PLS Path Modeling. *Comput. Stat. & Data Anal.*, **48**, 159–205.

Vermunt, J.K. (1997). *Log-linear Models for Event Histories*, Advanced Quantitative Techniques in the Social Sciences Series, vol. 8. Thousand Oakes: Sage Publications.

[*Received April 2014, accepted April 2014*]

International Statistical Review (2014), 82, 3, 357–359 doi:10.1111/insr.12061

Hongshik Ahn

State University of New York, Stony Brook, NY, USA
E-mail: hongshik.ahn@stonybrook.edu

The author presented a nice review of classification and regression trees by providing a discussion of major developments of the methods in the last 50 years. The author has made a great contribution to this field through developing fast and unbiased algorithms and applying the methods to various application areas. There has been a remarkable improvement in tree-structured methods. Due to the rapid advancement of computing capacity, even more computer intensive methods such as ensemble approach have been introduced.

Here, we will focus on discussing the properties of ensemble methods. There is a trade-off between a single tree and an ensemble method. Ensemble methods give higher prediction accuracy than a single tree in general. However, the ensemble method cannot compete with a single tree in interpretability as the author pointed out.

Three ensemble voting approaches, bagging, boosting and random subspace (Ho, 1998), have received attention. Because bagging and boosting were discussed in the paper, I will briefly discuss random subspace. Random subspace method combines multiple classification trees constructed in randomly selected subspaces of the variables. The final classification is obtained by an equal weight voting of the base trees. Ahn *et al.* (2007) proposed classification by ensembles from random partitions (CERP). CERP is similar to random subspace, but the difference is that base classifiers in an ensemble are obtained from mutually exclusive sets of predictors in CERP to increase diversity, whilst they are obtained by a random selection with overlap in random subspace.

The improvement in prediction accuracy in an ensemble from a single tree can be illustrated using a binomial model. If we assume independence amongst the n classifiers and equal prediction accuracy p of each classifier, where n is odd, the prediction accuracy of an ensemble classifier with majority voting is strictly increasing when $p > 0.5$ and strictly decreasing when $p < 0.5$ (Lam & Suen, 1997). The improvement of the prediction accuracy can be calculated using the beta-binomial model (Williams, 1975) when the the accuracies of the classifiers are positively correlated and using the extended beta-binomial model (Prentice, 1986) when they are negatively correlated.

The improvement of the ensemble accuracy illustrated earlier is valid under the assumption of equal accuracy of the base classifiers and equal correlation amongst the classifiers. Without these constraints, Breiman (2001) obtained the upper bound for the generalisation error. Convergence of the generalisation error rate depends on the average correlation, and it converges to zero when the classifiers are independent.

Logistic Regression Ensembles (LORENS: Lim *et al.*, 2010) is a logistic regression ensemble. LORENS uses the CERP algorithm to classify binary responses using the logistic regression model as a base classifier. This method enables class prediction by an ensemble of logistic regression models for a high-dimensional data set, which is impossible by a single logistic regression model due to the restriction that the sample size needs to be larger than the number of predictors. It is not as computer intensive as tree-based ensemble methods, whilst it does not lose the ensemble accuracy for high-dimensional data.

Recently, Kim *et al.* (2011) proposed weight-adjusted voting for ensemble (WAVE of classifiers). This method assigns unique voting weights to each classifier in the ensemble. Using an iterative process, a weight vector for the classifiers and another weight vector for the instances are obtained in the learning phase of model formation. They then proved the convergence of these vectors. After the final iteration, hard-to-classify instances get higher weights and subsequently, better performing classifiers on the hard-to-classify instances are assigned larger weights. Because a closed-form solution of the weight vectors can be obtained, WAVE does not need the iteration process.

In the evaluation of the performance of the classification methods, the sensitivity and specificity, positive predictive value, negative predictive value and receiver operating characteristic (ROC) curve also need to be considered. Most of the widely used classification methods have difficulties with unbalanced class sizes and almost always favour the majority class in order to increase the prediction accuracy.

Classification by ensembles from random partitions uses a different threshold from 0.5 in classification by logistic regression tree ensemble for unbalanced data. In a two-way classification, when r is the proportion of the positive responses in a data set, a threshold of r tends to give a better balance and a threshold of $1-r$ results in the highest accuracy (Chen *et al.*, 2006). Whilst a threshold of $1-r$ tends to yield the highest prediction accuracy, it worsens the balance by predicting more samples to the majority class. Pazzani *et al.* (1994) and Domingos (1999) assign a high cost to the misclassification of the minority class in order to improve a balance between sensitivity and specificity.

References

Ahn, H., Moon, H., Fazzari, M.J., Lim, N., Chen, J.J. & Kodell, R.L. (2007). Classification by ensembles from random partitions of high-dimensional data. *Comput. Stat. Data Anal.*, **51**, 6166–6179.

Chen, J.J., Tsai, C.A., Moon, H., Ahn, H. & Chen, C.H. (2006). The use of decision threshold adjustment in class prediction. *SAR & QSAR Environ. Res.*, **17**, 337–351.

Domingos, P. (1999). Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155–164. San Diego, California: ACM Press.

Ho, T.K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**, 832–844.

Kim, H., Kim, H., Moon, H. & Ahn, H. (2011). A weight-adjusted voting algorithm for ensembles of classifiers. *J. Korean Stat. Soc.*, **40**, 437–449.

Lam, L. & Suen, C.Y. (1997). Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Trans. Syst. Man Cybern.*, **27**, 553–568.

Lim, N., Ahn, H., Moon, H. & Chen, J.J. (2010). Classification of high-dimensional data with ensemble of logistic regression models. *J. Biopharm. Stat.*, **20**, 160–171.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T. & Brunk, C. (1994). Reducing misclassification costs: Knowledge-intensive approaches to learning from noisy data. In *Proceedings of the 11th International Conference on Machine Learning,* ML-94, pp. 217–225. New Brunswick:New Jersey.

Prentice, R.L. (1986). Binary regression using an extended beta-binomial distribution, with discussion of correlation induced by covariate measurement errors. *J. Amer. Statist. Assoc.*, **81**, 321–327.

Williams, D.A. (1975). The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity. *Biometrics*, **31**, 949–952.

[*Received March 2014, accepted March 2014*]

International Statistical Review (2014), 82, 3, 359–361 doi:10.1111/insr.12060

Chi Song and Heping Zhang

Department of Biostatistics, Yale University School of Public Health
E-mail: heping.zhang@yale.edu

We wish to congratulate the author for a nice overview of tree-based methods, and the author clearly highlighted the recursive partitioning technique (Friedman, 1977; Breiman *et al.*, 1984; Zhang & Singer, 2010) behind the tree-based methods. As the author summarized, there are two major types of tree methods: classification trees and regression trees, as precisely reflected in the title of the classical book by Breiman *et al.* (1984). In our own experience, for regression problems, other nonparametric methods, including adaptive splines (Friedman, 1991) that are based on a similar partitioning technique, appear more desirable than regression trees, with the exception of survival analysis (Zhang, 1997; 2004; Zhang & Singer, 2010).

With the advent of high-throughput genomic technologies, classification trees have become one of the most common and convenient bioinformatic tools. In what follows, we would like to share some of the recent developments in this area.

Genome-wide association studies (GWASs) collect data for hundreds of thousands or millions of single-nucleotide polymorphisms (SNPs) to study diseases of complex inheritance patterns, which can be recorded qualitatively (e.g. breast cancer) or in a quantitative scale (e.g. blood pressure). GWASs typically employ the case–control design, and the logistic regression model is generally applied to assess the association between each of the SNPs and the disease response, although more advanced techniques, especially nonparametric regression, have been proposed to incorporate multiple SNPs and interactions.

A clear advantage of classification trees is that they make no model assumption and that they can select important variables (or features) and detect interactions among the variables. Zhang & Bonney (2000) was among the early applications of tree-based methods to genetic association analysis. Since then, interests in tree-based genetic analyses have grown substantially. For example, Chen *et al.* (2007) developed a forest-based method on haplotypes instead of SNPs to detect gene–gene interactions, and importantly, they detected both a known variant and an unreported haplotype that were associated with age-related macular degeneration. Wang *et al.* (2009) further demonstrated the utility of this forest-based approach. Yao *et al.* (2009) applied GUIDE to the Framingham Heart Study and detected combinations of SNPs that affect the disease risk. García-Magariños *et al.* (2009) demonstrated that the tree-based methods were effective in detecting interactions with pre-selected variables that were marginally associated with the disease outcome but were susceptible to the local maximum problem when many noise variables were present. Chen *et al.* (2011) combined the classification tree and Bayesian search strategy, which improved the power to detect high-order gene–gene interactions at the cost of high computation demand.

Tree-based methods are extensively used in gene expression analysis to classify tissue types. Here, the setting is very different from the GWAS applications. In GWAS applications, we deal with a very large number of discrete risk factors (e.g. the number of copies of a particular allele). In expression analysis, the number of variables is large but not so large, usually in the order of tens of thousands, and the variables tend to be continuous. For example, Zhang *et al.* (2001) demonstrated that classification trees can discriminate distinct colon cancers more accurately than other methods. Huang *et al.* (2003) found that aggregated gene expression patterns can predict the breast cancer outcomes with about 90% accuracy using tree models. Zhang *et al.* (2003) introduced deterministic forests for gene expression data in cancer diagnosis, which have a similar power to random forests but are easier in scientific interpretation. Pang *et al.* (2006) developed a random forest method incorporating pathway information and demonstrated that it has low prediction error in gene expression analysis. Furthermore, Díaz-Uriarte & De Andres (2006) demonstrated that random forest can be useful in variable selection by using a smaller set of genes and maintaining a comparable prediction accuracy. Of a related note, Wang & Zhang (2009) attempted to address the following basic question: how many trees are really needed in a random forest? They provided empirical evidence that a random forest can be reduced in size so much to allow scientific interpretation.

As more and more data are generated from new technologies such as the next-generation sequencing, tree-based methods will be very useful for analysing such large and complex data after necessary extensions. Closely related to genomic data analysis is the personalized medicine. Zhang *et al.* (2010) presented a proof of concept that tree-based methods have some unique advantages over parametric methods to identify patient characteristics that may affect their treatment responses. In summary, tree-based methods have thrived in the past several decades, and they will become more useful, and the methodological developments will be more challenging than ever, as more information increases in both size and complexity.

Acknowledgements

This research is supported in part by grant R01 DA016750 from the National Institute on Drug Abuse.

References

Chen, M., Cho, J. & Zhao, H. (2011). Detecting epistatic SNPs associated with complex diseases via a Bayesian classification tree search method. *Ann. of Human Gen.*, **75**, 112–121.

Chen, X., Liu, C., Zhang, M. & Zhang, H. (2007). A forest-based approach to identifying gene and gene–gene interactions. *Proceedings of the National Academy of Sciences*, **104**, 19199–19203.

Díaz-Uriarte, R. & De Andres, S.A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, **7**, 3.

Friedman, J.H. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Trans. Comput.*, **C-26**, 404–407.

Friedman, J.H. (1991). Multivariate adaptive regression splines. *IEEE Trans. Comput.*, **19**, 1–141.

García-Magariños, M., López-de Ullibarri, I., Cao, R. & Salas, A. (2009). Evaluating the ability of tree-based methods and logistic regression for the detection of SNP–SNP interaction. *Ann. of Human Gen.*, **73**, 360–369.

Huang, E., Cheng, S., Dressman, H., Pittman, J., Tsou, M., Horng, C.F., Bild, A., Iversen, E.S., Liao, M., Chen C.M., West, M. & Nevins, J.R. (2003). Gene expression predictors of breast cancer outcomes. *The Lancet*, **361**, 1590–1596.

Pang, H., Lin, A., Holford, M., Enerson, B.E., Lu, B., Lawton, M.P., Floyd, E. & Zhao, H. (2006). Pathway analysis using random forests classification and regression. *Bioinformatics*, **22**, 2028–2036.

Wang, M. & Zhang, H. (2009). Search for the smallest random forest. *Stat. Interface*, **2**, 381–388.

Wang, M., Zhang, M., Chen, X. & Zhang, H. (2009). Detecting genes and gene–gene interactions for age-related macular degeneration with a forest-based approach. *Stat. Biopharm. Res.*, **1**, 424–430.

Yao, L., Zhong, W., Zhang, Z., Maenner, M. & Engelman, C. (2009). Classification tree for detection of single-nucleotide polymorphism (SNP)-by-SNP interactions related to heart disease: Framingham Heart Study. *BMC Proceedings*, **3**, S83.

Zhang, H. (1997). Multivariate adaptive splines for analysis of longitudinal data. *J. Comput. Graph. Statist.*, **6**, 74–91.

Zhang, H. (2004). Mixed effects multivariate adaptive splines model for the analysis of longitudinal and growth curve data. *Stat. Methods Med. Res.*, **13**, 63–82.

Zhang, H. & Bonney, G. (2000). Use of classification trees for association studies. *Genet. Epidemiol.*, **19**, 323–332.

Zhang, H., Legro, R.S., Zhang, J., Zhang, L., Chen, X., Huang, H., Casson, P.R., Schlaff, W.D., Diamond, M.P., Krawetz, S.A., Coutifaris, C., Brzyski, R.G., Christman, G.M., Santoro, N. & Eisenberg, E. (2010). Decision trees for identifying predictors of treatment effectiveness in clinical trials and its application to ovulation in a study of women with polycystic ovary syndrome. *Human Reproduction*, **25**, 2612–2621.

Zhang, H., Yu, C. & Singer, B. (2003). Cell and tumor classification using gene expression data: Construction of forests. *Proceedings of the National Academy of Sciences*, **100**, 4168–4172.

Zhang, H., Yu, C., Singer, B. & Xiong, M. (2001). Recursive partitioning for tumor classification with gene expression microarray data. *Proceedings of the National Academy of Sciences*, **98**, 6730–6735.

[*Received March 2014, accepted March 2014*]

International Statistical Review (2014), 82, 3, 361–367 doi:10.1111/insr.12062

Thomas Rusch[1] and Achim Zeileis[2]

[1]*WU Vienna Vienna Austria*
E-mail: thomas.rusch@wu.ac.at
[2]*Universität Innsbruck, Innsbruck, Austria*
E-mail: Achim.Zeileis@R-Project.org

1 Introduction

We thank Wei-Yin Loh for this review paper. He provides a much-needed guide to tree methods currently available as well as the main ideas behind them, indicative of his experience with and knowledge of this topic. His contribution proves to be very valuable in bringing structure into the vast interdisciplinary field of tree algorithms: We found 83 different tree induction algorithms for different response types listed in his paper, and, along the lines of Loh's disclaimer, this is not even an exhaustive list.

The availability of so many different algorithms for fitting tree-structured models directly relates to the main point of our discussion: The tree literature is highly fragmented. Loh hints at that issue already on the first page, and we gladly take it up for discussion: There are so many recursive partitioning algorithms in the literature that it is nowadays very hard to see the wood for the trees.

In the remainder of our discussion paper, we identify causes for and consequences of this fragmentation, discuss what we perceive to be advantages and disadvantages of the current state of the tree algorithm literature and offer suggestions that might improve the situation in the years ahead by retaining advantages and overcoming disadvantages.

2 The Fragmentation of Tree Algorithms

Currently, there is an abundance of different tree algorithms coming from different communities including statistics, machine learning and other fields. We believe that this fragmentation emerged from various causes and has a number of implications for the development and application of tree models. Some of them are in our opinion good, some are not so good and some are rather unfortunate.

2.1 The Good

The area of tree algorithms is a popular and fruitful field of research in statistics, computer science and beyond. This leads to many people with different backgrounds contributing to the application and development of tree algorithms for various tasks. May this be to derive a set of if-then rules to make decisions, to analyse a large number of data relatively fast, to segment data, to detect or select important variables and interactions or to simply have an interpretable, visualisable, data-driven prediction machine with good performance, which is flexible and can be adapted easily to the problem at hand. An important contributing factor to their popularity is that recursive partitioning algorithms are easily adapted to different situations, as their core principles are easy to understand and intuitive. Most tree algorithms comprise a couple of similar steps, the difference between them entering at some point in the induction stage, where during development a concrete choice must be made – usually related to the loss function or measure of node impurity, split variable selection, split point selection or pruning. Thus by, for example, changing the loss function used to measure node impurity, a new tailored algorithm can be easily invented for a given problem. Owing to this, we now have tree algorithms for many types of problems and variables that we might encounter, say, for online or dynamic data, longitudinal data, big sample sizes, substantive data models, various error structures and so forth. Looking at it this way, the fragmentation reflects in part the diversity of the problems encountered by the community of scholars working in this field, as well as the many ideas they have and different applications they face. Given that there is no free lunch (Wolpert, 1996), a rich diversity in algorithmic solutions is to be welcomed as no single solution will always lead to the best results. Thus, tree models were and are an active field of research and hopefully will remain so in the future.

2.2 The Bad

This abundance of tree algorithms also has its dark side. For one, not only is it hard to keep up to date with various developments, but it is even harder to *choose* the 'right' algorithm for a given problem. Take the case of regression trees for explaining and predicting a metric outcome and assume that there is additive Gaussian error. Which algorithm to take? There are, among others, AID, CART, CTree, C4.5, GUIDE and M5 – all having different properties in different settings. There is a lack of guidance as to which algorithm to select. One might narrow the possibilities down by looking at certain additional, desirable properties like unbiasedness in split variable selection but that still leaves one with a number of possibilities to consider. People looking to solve their problems with tree algorithms might easily be intimidated by the large number of possibilities and if different tree algorithms give different answers. Perhaps this contributed to many people inventing a new algorithm for their specific problem rather than work through different properties of existing algorithms and benchmarking them against each other on their data, which in turn perpetuates the fragmentation problem.

The fact that tree methods can be easily adapted to new situations can backfire. First, this sometimes leads to new tree algorithms or changes to old ones that appear *ad hoc*, which was already noted by Murthy (1998). While experimentation is a necessary part of algorithm development, we should nevertheless take care to propose and use well-motivated, well-founded, methodologically sound procedures in the end. Second, modifications or improvements of old algorithms are often considered to be entirely new algorithms. They may be named differently

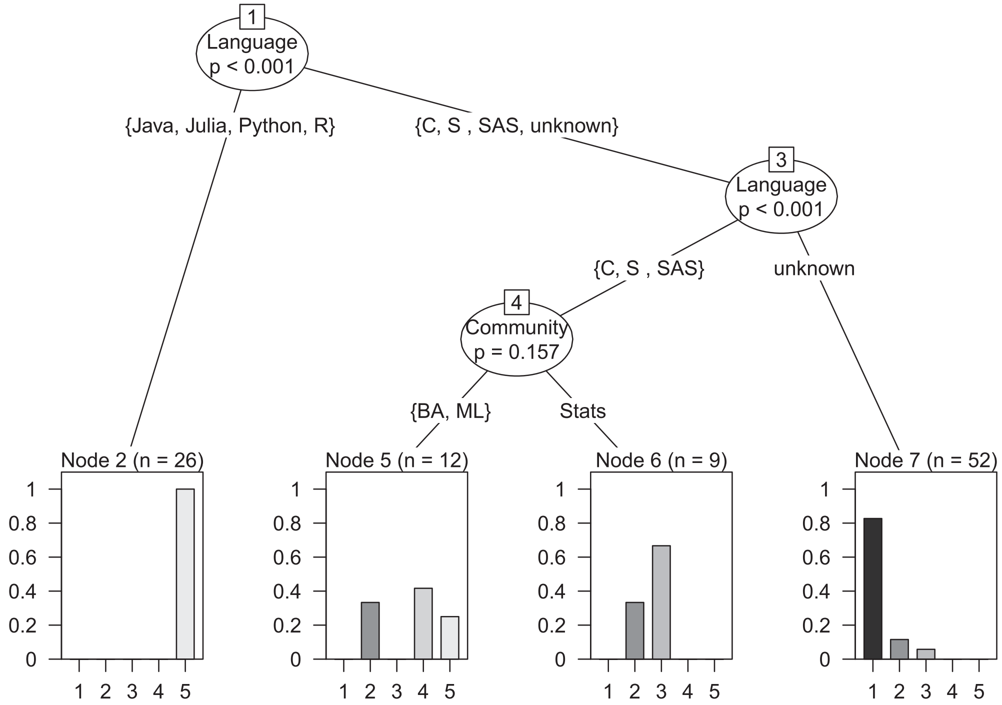


Figure 1. *A classification tree of tree algorithms fitted with CTree (Hothorn, Hornik, Zeileis, 2006). The target variable is the group of implementation as defined in Section 2.3. Predictor variables were the publication 'Year', the programming 'Language', whether it allows fitting of a classification ('CT'), regression ('RT'), or model tree ('MT'), author names ('FirstAuthor', 'SecondAuthor', 'LastAuthor'), 'Community' the algorithm is aimed at (machine learning vs. business analytics vs. statistics) and 'Journal' venue.*

and are often only viewed 'as a whole' rather than emphasising which steps in the tree induction are similar and which are different (e.g. the loss function might change, but the split variable selection is the same or vice versa). Hence, fragmentation is increased more than necessary, and common properties are obscured. This seems to tie in with a third bad effect: Many authors who propose or apply tree algorithms either are not aware of – or choose to ignore – similar work in that area. It happens that even recent papers do not refer to work carried out from 2000 onwards, therefore ignoring more than a decade of active development that may be highly relevant. On the one hand, this leads to reinventing the wheel, loss of time and resources and again more fragmentation of the literature. On the other hand, this also makes it hard for new algorithms to be noticed in an evergrowing, dense wood – an unfortunate situation for both developers and users alike. Perhaps all these points explain why, while there has been numerous new developments and improvements in tree modelling since the seminal work on C4.5 and CART in the 1980s, both remain the most popular tree algorithms for classification and regression (Wu *et al.*, 2008).

2.3 The Ugly

As algorithmic models, classification and regression trees are very closely tied to their respective implementations. With few exceptions, tree models usually *are* their specific implementation. Hence, they can be characterised by the specific combination of the generic computational steps that are adopted and by the way they are turned into a specific implementation – with both aspects being highly intertwined. While this algorithm–implementation dualism is rather natural for algorithmic models, there is also an ugly side to it: the potential lack of free (or any) implementations of new algorithms. This potential lack of access to the core of the actual tree model makes understanding, using, assessing and extending it much more difficult.

To illustrate this empirically, we consider 99 algorithms – including the 83 mentioned by Loh plus 16 very recent or less known ones. For 43 of these, we were not able to find an implementation[1] (see also Figure 2). More specifically, we placed all 99 algorithms into one of five broad classes pertaining to the availability of (free) software:

1. *Algorithms without an existing implementation:* These are algorithms for which a theoretical description was published, but no implementation seems to exist. Often, these are old algorithms or algorithms that were developed for a specific problem. Examples are AID or SUPPORT.
2. *Algorithms with a closed-source, for-profit implementation:* These are implementations of particular algorithms that are sold by a company. The code and specific implementation are kept a proprietary secret. Typical examples are M5, CHAID or CART.
3. *Algorithms with a closed-source, free-of-charge implementation:* These are implementations of particular algorithms that can be obtained free of charge, usually in an executable binary format. The code and specific implementation, however, is still kept a proprietary secret. Examples include GUIDE, CTMBR and HTL.
4. *Algorithms with an open-source, free-of-charge implementation:* These are implementations of particular algorithms that can be obtained free of charge and whose source code is open. However, these implementations either restrict or do not explicitly allow copying, adaptation and distribution of the source code. Examples currently include SECRET and C4.5.
5. *Algorithms with a free and open-source implementation:* These algorithms have implementations that follow the ideas of free software (FLOSS; free, libre open-source software, see Free Software Foundation, Inc., 2013). They are open source and give the user extensive rights with respect to copying, modification and distribution. Examples are most algorithms developed for FLOSS software packages like `Weka` or R, including re-implementations of closed-source algorithms, e.g. RPart, M5′, LMT and CTree and also C5.

We find that most algorithms belong to group 1 (43), followed by group 5 (29). The group of open-source algorithms/implementations (classes 4 and 5) only comprises 34% of all algorithms. To take a closer look at how this availability of (free) software depends on other characteristics of the algorithms; we naturally employ a classification tree (Figure 2, built by CTree). We see that an implementation in R, Python, Java (primarily in the packages `Weka`, `KNIME`, `RapidMiner`) and Julia is predictive of belonging to group 5, whereas other languages are either predictive for group 1 (if we do not know the language), of groups 2 and 3 respectively if suggested to the statistics community, and for groups 4 and 5 for implementations directed at the machine learning and business analytics communities. At any rate, it shows that unfortunately FLOSS is far from being the standard for tree modelling software.

As trees are inherently algorithmic, we view the implementation as an integral part of each algorithm. In our opinion, restrictions to viewing, modifying and sharing tree implementations are one of the main reasons for the bad sides of fragmentation discussed earlier. Depending on which group an algorithm belongs to this has different implications. For example, for algorithms belonging to groups 1 through 4, this leads to the need for authors proposing improvements to existing algorithms to implement the improved algorithm from scratch. Often, this adapted implementation is then again not FLOSS, and the problem perpetuates. Not being able to rely on existing code also applies to using an algorithm on different platforms. Another example is that the restriction of distribution and modification effectively prohibits to change the specific implementation (say, with regard to adapt it to parallel computing) or to improve it (say, with regard to speed). This also restricts the possibilities of combining the algorithms with other methods to form a pipeline of methods and distribute the bundle. For algorithms from groups 1 through 3, a further consequence is that specific steps that may not be well documented can be hard to reproduce [as was the case with M5 (Quinlan, 1993), which prompted M5′ (Wang & Witten, 1997) as a 'a rational reconstruction'].

We strongly believe that these and other implications of a lack of free implementations led to many synergy potentials having been lost over the years, partly because conceptual similarities and differences of various tree algorithms were not obvious enough and partly because the lack of reusable computational tools slowed down the pace development. Furthermore, there seems to be some confusion among practitioners as to which algorithms perform well (or even best) for their particular problems which often leads to suboptimal algorithms being used. This view appears to be shared increasingly by other researchers (e.g. Vukićević *et al.*, 2012).

3 A Possible Remedy

We have a suggestion as to what we think will improve or even solve the problems discussed in Sections 2.2 and 2.3 while retaining the advantages mentioned in Section 2.1: (Academic) publications of tree algorithms should be accompanied by free implementations (in the FLOSS sense). This means opening the source code of past and future implementations, giving users permission to modify, adapt and distribute it with an appropriate free software license and making the code/implementations easily publicly available, preferably with a low adoption threshold (e.g. in a popular language such as Python, C, Java or R).

Specifically, we think that the following six steps should be undertaken to reduce the bad and ugly aspects to a minimum while retaining the good:

- Every newly suggested algorithm or larger improvement should come with a FLOSS implementation.
- The source code of currently existing implementations should be opened, and they should be licensed with a FLOSS license.
- For algorithms for which there is no up-to-date or even existing software, any trademark should be relinquished, and the original source code should be made freely available or re-implemented as FLOSS (as has already happened with, e.g. J4.8 or RPart).
- All implementations should be made available on a public repository or archive, the author's homepages or as freely accessible supplementary material to articles.
- When a software is used, extended, modified and so on, the software (and not only the underlying algorithm) should be referred to and cited.
- When reviewing or editing papers or algorithms, we should point out the above and demand this as a new standard.

FLOSS software licenses should be used so that the copyright holders grant the rights to inspect, modify and (re)distribute the software. Most FLOSS licenses preserve the original copyright in such modified/extended versions and in an academic context citation of the software (and not only the underlying algorithm) is appropriate.

Then, improvements to algorithms do not need to be their own algorithm but can be suggested or submitted as patches or adaptations. Code building blocks (e.g. for split point selection, or predictions or tree visualisations) can be reused and recombined or be made computationally more efficient. Assessment and comparison of algorithms are facilitated, both for evaluating newly suggested methods and for choosing a particular model in practice. Hence, not only users of FLOSS implementations will profit but also the authors because their work is easier to understand, use and ultimately cite.

Finally, the aforementioned steps can also reduce the fragmentation and possibly achieve a certain degree of standardisation through developing and reusing computational ‘tree toolkits’. Some effort in this vein has been made already. For example, there are two R packages providing standardised frameworks: `partykit` (Hothorn & Zeileis, 2014) for representing, summarising and plotting of various tree models from different free software sources, and `caret` (Kuhn, 2008) for training, tuning and benchmarking of various tree algorithms (among many other methods). Other efforts of providing such standardisation exist as well (Vukićević *et al.*, 2012).

4 Conclusion

In this discussion, we follow up on Loh’s review paper and take a closer look at the fragmented field of tree models. While indicative of a creative, active, and diverse community of researchers, this fragmentation also leads to undesirable side effects making it hard to understand, assess, use and compare tree algorithms. Hence, a common language for describing tree models, both conceptually and (perhaps more importantly) computationally, is crucial to reduce the fragmentation. In particular, making FLOSS should be an integral part of the communication about classification and regression trees. We argue that this can alleviate many of the problems caused by or following from the fragmentation while retaining the good that comes from having a bright and vibrant community.

Especially in light of open research areas that Loh mentions in his conclusions, FLOSS implementations are an effective means for reducing fragmentation in the future and tackling open hard problems in tree algorithm research faster than it was possible before. The good news is that the tree community has already started to move into this direction, and an increase in FLOSS implementations can already be observed. With free platforms for statistical computing such as R or Python as well as initiatives like the ‘Foundation for Open Access Statistics’ (FOAS, http://www.foastat.org/), the conditions are now better than ever before. We should use this momentum. Rather than not seeing the wood for the trees, the whole community can grow a healthy, open and light forest of trees within which we can all walk with intimate familiarity.

Notes

[1] We searched with Google for all permutations of author names and algorithm names combined with the words “software” and “implementation”, as well as on the main author’s homepages.

Acknowledgements

We want to thank Wei-Yin Loh for his stimulating article and Probal Chaudhuri for inviting us to join this discussion.

References

Free Software Foundation, Inc. (2013). What is free software? Version 2013-06-18 05:16:52. http://www.fsf.org/about/what-is-free-software.

Hothorn, T. & Zeileis, A. (2014). `partykit`: A modular toolkit for recursive partytioning in R. *Working Paper 2014-10*, Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck. http://econpapers.repec.org/paper/innwpaper/2014-10.htm.

Kuhn, M. (2008). Building predictive models in R using the caret package. *J. Stat. Software*, **28**(5), 1–26.
Vukićević, M., Jovanović, M., Delibašić, B., Išljamović, S. & Suknović, M. (2012). Reusable component-based architecture for decision tree algorithm design. *Int. J. Artif. Intell. Tools*, **21**(05).
Wolpert, D.H. (1996). The lack of a priori distinctions between learning algorithms. *Neural Comput.*, **8**(7), 1341–1390.
Wu, X., Kumar, V., Quinlan, R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Philip, Y., Zhou, Z.-H., Steinbach, M., Hand, D. & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, **14**(1), 1–37.

[*Received April 2014, accepted April 2014*]

International Statistical Review (2014), 82, 3, 367–370 doi:10.1111/insr.12057

Rejoinder

Wei-Yin Loh

Department of Statistics, University of Wisconsin, Madison, WI 53706, USA
E-mail: loh@stat.wisc.edu

I thank the discussants for their thoughtful comments, which helped to fill in some gaps and expand the scope of the review. I will address each one below.

Carolin Strobl wonders when it is helpful to create separate nodes for missing values. CHAID seems to be the only algorithm to do this, but it has a procedure to merge some of the nodes before they are split further. Its effectiveness has not been studied. GUIDE treats missing values in a categorical variable as a separate category but does not assign them to a separate node. If there are missing values at a split on an ordered variable, GUIDE sends them to the same left or right child node, depending on which split yields the greater decrease in node impurity. Ding & Simonoff (2010) studied a simpler version of this technique, where missing ordered values are mapped to infinity and hence are always sent to the right child node. Using only binary-valued variables with training and test sets having missing values where missingness in a predictor variable depends on the values of the response variable (MAR), they found that this technique is better than case deletion, variable deletion, grand mean/mode imputation, surrogate splits (as used in RPART) and fractional weights (as used in C4.5). Case deletion and grand mean/mode imputation tend to be worst, a finding supported by Twala (2009), who considered missingness dependent on other predictor variables (MAR) and missingness due to truncation (MNAR) but not missingness dependent on the response variable. He found that the best method was an ensemble of classification trees constructed by multiple imputation of the missing values with the expectation–maximization algorithm (Dempster *et al.*, 1977; Rubin, 1987). It is not clear whether this is either due to multiple imputation or ensemble averaging. Note that because both studies employed C4.5 and RPART exclusively as the base classifiers, it is unknown if the conclusions extend to other methods. Further, some other missing value techniques, such as nodewise mean and mode imputation (FACT and QUEST) and alternative surrogate split methods (CRUISE), were not included.

Strobl wonders whether ignorance is the reason that biassed recursive partitioning methods continue to be used frequently. Many people still associate the term ‘classification and regression trees’ with CART and its software. Commercial software publishers perpetuate this misconception by largely basing their offerings on CART. The availability of RPART also encouraged the use and extension of CART (e.g. MVPART). On the other hand, selection bias may not cause serious harm if a tree model is used for prediction but not interpretation, in some situations. Selection bias can increase the likelihood of spurious splits on irrelevant variables, but if the sample size is large and there are not too many such variables, a correspondingly large tree may subsequently split on the important variables. If the spurious splits survive after pruning, they simply stratify the data into two or more subsets each having its own subtree, and overall prediction accuracy may be preserved; see Martin (1997) for a related discussion. If the sample size is small, however, then the spurious splits will increase the frequency of trivial pruned trees.

Strobl's comments on importance scores bring us back to the meaning of the 'importance' of a variable. Because it is not well defined, the concept has produced a plethora of importance measures. CART has a measure based on the efficiency of surrogates splits, and FACT has a measure based on F-statistics. At that time, both seemed reasonable as there was not much need for either, due to data sets being small and variables few. Now that data sets can contain thousands and even millions of variables, the situation is different. Further, there is evidence (e.g. Doksum *et al.*, 2008 and Loh, 2012) that when the number of variables is very large, some sort of preliminary variable selection can substantially improve the prediction accuracy of a model. Because importance measures are well suited (and are being used) for this task, it is time to examine the notion more carefully. As Strobl observes, some people consider the importance of a variable 'more or less on its own' [e.g. random forest (RF)], whereas others think of it as the residual effect after other variables are accounted for in a model (e.g. linear regression). Although the latter point of view is more specific, it lacks a sense of universality, because a variable can be important for one model but not for another. On the other hand, perhaps universality (i.e. being model free) is not attainable. Nonetheless, there is one property that every importance measure ought to have, namely, unbiasedness. Strobl *et al.* (2007) showed that RF is biased in the 'null' sense that, if all the variables are independent of the response, the frequencies with which they appear in the trees depend on their types.

A more general definition of unbiasedness, applicable to non-forest methods as well, is that under this null scenario, all variables are ranked equally on average. To see how RPART, RF, Cforest (CF, from the PARTY package) and GUIDE perform by this criterion, I simulated 5000 data sets with each set consisting of 100 observations on seven mutually independent predictor variables and one normally distributed response variable. Three predictor variables are continuous (normal, uniform and chi-squared with two degrees of freedom) and four are categorical with 2, 4, 10 and 20 equiprobable categories; all are independent of the response variable. Figure 3 shows the average rank of each variable for each method (rank 1 is most important and rank 7 the least). RPART and RF tend to find the variables with 10 and 20 categories the most important (although they differ on the *most* important) and the binary predictor the least. CF has a slight bias towards ranking the binary variable most important and the two variables with 10 and 20 categories least; GUIDE has a smaller bias towards ranking ordinal variables more important than categorical variables. The biases of CF and GUIDE are negligible, however, compared with that of the other two.

Antonio Ciampi touches on several philosophical issues. I will offer my take on some of his main points. The 'dilemma' between interpretability and accuracy is a result of the human mind's limitations in understanding complex structure. Fortunately, if the structure is comprehensible, the mind is exceptionally good at drawing insights that no machine can match. Therefore, rather than a dilemma, it is really a choice: construct simple models that humans can use to enhance their understanding of the problem or build complex models for automatic and accurate predictions. Both are laudable goals.

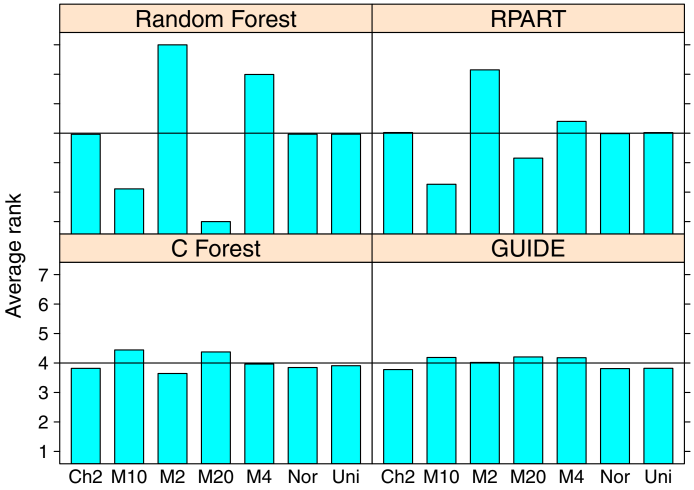


Figure 1. *Average ranks (with 1 being most important and 7 least) of variables for sample size 100 over 5000 simulation iterations. Simulation standard errors are less than 0.03. 'Ch2', 'Nor' and 'Uni' denote* χ^2_2*,* $N(0,1)$ *and* $U(0,1)$ *variables. 'Mk' denotes a multinomial (categorical) variable with k equiprobable levels. The response variable is* $N(0,1)$ *and all variables are mutually independent. A method is unbiased if each variable has average rank 4.0, which is marked by horizontal lines.*

Ciampi mentions several ways to improve the accuracy of single trees, such as using global model search (Bayesian and genetic algorithms) and probabilistic splits (soft nodes). The accuracy of Bayesian trees comes from model averaging; there is no evidence that the tree with the largest posterior probability has comparable accuracy. Global search techniques inevitably produce randomised solutions that may be undesirable in some applications. It is harder to follow the path of an observation in a model with probabilistic splits than it is in a model with conventional (hard) splits.

I agree that univariate (monothetic) splits are not the only ones that are interpretable. Ciampi's example of a (polythetic) split on the number of symptoms possessed by a patient is certainly interpretable and can be implemented as sums of indicator variables. But because there are numerous combinations of variables that can form the sums, this approach invites computational and selection bias problems. His idea of hierarchical tree structures is intriguing, particularly if the predictor variables are naturally clustered.

Ciampi notes that data have become more complex. In business, biology, medicine and other fields, predictor and response variables are increasingly observed as longitudinal series. Owing to difficulties caused by the number and location of the observation 'time' points varying between subjects, small number of subjects relative to number of time points, large number of baseline covariates and occurrence of missing values, the traditional approach of fitting parametric stochastic models to the processes is seldom feasible. A more practical solution may be a non-parametric approach that treats the longitudinal series as random functions (Loh & Zheng, 2013).

I thank Hongshik Ahn for his review of some of the more recent ensemble methods. The fact that the accuracy of an ensemble increases as the dependence among the component classifiers decreases, provided that the latter are equally accurate, motivates the construction of ensembles where each classifier is built from a mutually exclusive subset of predictors. But it is difficult to do this without destroying the requirement of equally accurate classifiers. This is obvious when there is exactly one informative predictor variable and many irrelevant ones. Then, all but one classifier (the one involving the informative predictor) do nothing except to dilute the accuracy of the ensemble. On the other hand, the classifier containing the informative variable should be more accurate than the one built with all the variables. This may explain the behaviour of CERP and LORENS. The WAVE method of adaptively assigning weights to classifiers seems to be a promising direction.

I thank Chi Song and Heping Zhang for the references to genetic applications. Subgroup identification, a key part of personalised medicine, is rapidly gaining attention. The goal is to find patient subgroups, defined by measurable patient characteristics (such as demographic, phenotype, genotype and protein biomarkers) prior to treatment, that respond differentially to treatment. Negassa *et al.* (2005), Su *et al.* (2009), Foster *et al.* (2011), Lipkovich *et al.* (2011) and Dusseldorp & Van Mechelen (2013) have proposed tree-based solutions. Alternatives that do not have selection bias have been implemented in the GUIDE software.

While I agree with Thomas Rusch and Achim Zeileis that in an ideal world, all published algorithms would be accompanied by free software, there are reasons why this does not always occur in practice. Quite often, the author of the software is a student who is not the architect of the algorithm. When the student graduates, there is no one to distribute and maintain the code. This was the case with the SUPPORT algorithm, although its best features have since been incorporated in GUIDE. Then, there is the author who plans only to publish a paper and move on to other problems, with no intention to distribute and maintain the software. As a result, the latter is developed only as far as it is needed for the examples and simulations in the paper.

Acknowledgements

This rejoinder was prepared with partial support from the National Science Foundation grant DMS-1305725.

References

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **39**, 1–38.

Doksum, K., Tang, S. & Tsui, K.-W. (2008). Nonparametric variable selection: The EARTH algorithm. *J. Amer. Statist. Assoc.*, **103**, 1609–1620.

Dusseldorp, E. & Van Mechelen, I. (2013). Qualitative interaction trees: A tool to identify qualitative treatment–subgroup interactions. *Stat. Med.*, **33**, 219–237, DOI 10.1002/sim.5933.

Foster, J.C., Taylor, J.M.G. & Ruberg, S.J. (2011). Subgroup identification from randomized clinical trial data. *Stat. Med.*, **30**, 2867–2880.

Lipkovich, I., Dmitrienko, A., Denne, J. & Enas, G. (2011). Subgroup identification based on differential effect search–a recursive partitioning method for establishing response to treatment in patient subpopulations. *Stat. Med.*, **30**, 2601–2621.

Martin, J.K. (1997). An exact probability metric for decision tree splitting and stopping. *Mach. Learn.*, **28**, 257–291.

Negassa, A., Ciampi, A., Abrahamowicz, M., Shapiro, S. & Boivin, J.R. (2005). Tree-structured subgroup analysis for censored survival data: Validation of computationally inexpensive model selection criteria. *Stat. Comput.*, **15**, 231–239.

Rubin, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys.* New York: Wiley.

Su, X., Tsai, C.L., Wang, H., Nickerson, D.M. & Bogong, L. (2009). Subgroup analysis via recursive partitioning. *J. Mach. Learn. Res.*, **10**, 141–158.

Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Appl. Artificial Intell.*, **23**, 373–405.

[*Received March 2014, accepted March 2014*]