# Recursive Partitioning

by

**Hugh A. Chipman, Acadia University**

## 1 Introduction

Recursive partition (RP) models are a flexible method for specifying the conditional distribution of a variable $y$, given a vector of predictor values $x$. Such models use a tree structure to recursively partition the predictor space into subsets where the distribution of $y$ is successively more homogeneous. The terminal nodes of the tree correspond to the distinct regions of the partition, and the partition is determined by splitting rules associated with each of the internal nodes. By moving from the root node through to the terminal node of the tree, each observation is then assigned to a unique terminal node where the conditional distribution of $y$ is determined. The two most common response types are continuous and categorical, with corresponding tasks often known as regression and classification.

Given a data set, a common strategy for finding a good tree is to use a greedy algorithm to grow a tree and then to prune it back to avoid overfitting. Such greedy algorithms typically grow a tree by sequentially choosing splitting rules for nodes on the basis of maximizing some fitting criterion. This generates a sequence of trees each of which is an extension of the previous tree. A single tree is then selected by pruning the largest tree according to a model choice criterion such as cost-complexity pruning, cross-validation, or hypothesis tests of whether two adjoining nodes should be collapsed into a single node.

Early work in RP models includes Morgan and Sonquist (1963), who developed a recursive partitioning strategy (AID - Automatic Interaction Detection) for a continuous response. There were many offshoots of this work, including Kass (1980) and Hawkins and Kass (1982). Recursive partitioning models were popularized in the statistical community by the book "Classification and Regression Trees" by Breiman, Friedman, Olshen and Stone (1984). RP models have also been developed in the machine learning community, with work by Quinlan on the ID3 (1986 and references therein) and C4.5 (1993) algorithms being among the most widely recognized.

## 2 Structure of a RP model

A RP model describes the conditional distribution of $y$ given a vector of predictors $x = (x_1, x_2, \ldots, x_p)$. This model has two main components: a tree $T$ with $b$ terminal nodes, and a parameter $\Theta = (\theta_1, \theta_2, \ldots, \theta_b)$ which associates the (possibly vector-valued) parameter $\theta_j$ with the $j^{th}$ terminal node. If $x$ lies in the region corresponding to the $j^{th}$ terminal node then $y \mid x$ has distribution $f(y \mid \theta_j)$, where we use $f$ to represent a parametric family indexed by $\theta_j$. The model is called a regression tree or a classification tree according to whether the response $y$ is quantitative or qualitative, respectively. An example of a RP model with binary splits is displayed in Figure 1, and data sampled from its induced partition is displayed in Figure 2.

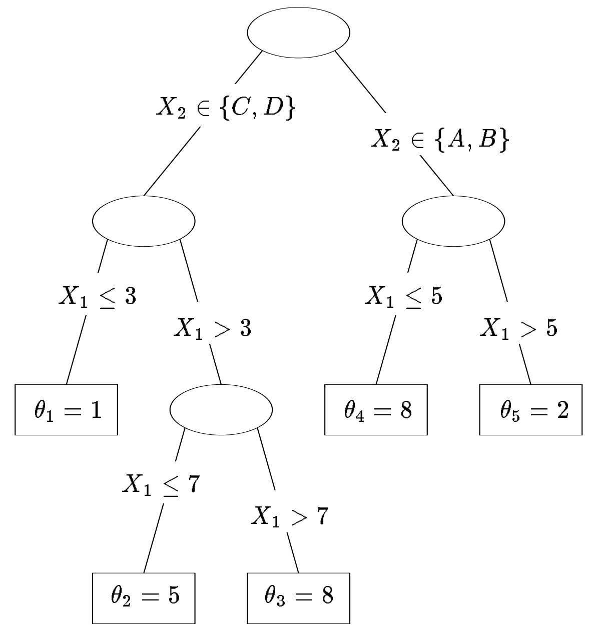

Figure 1: A regression tree where $y \sim N(\theta, 2^2)$ and $x = (x_1, x_2)$.

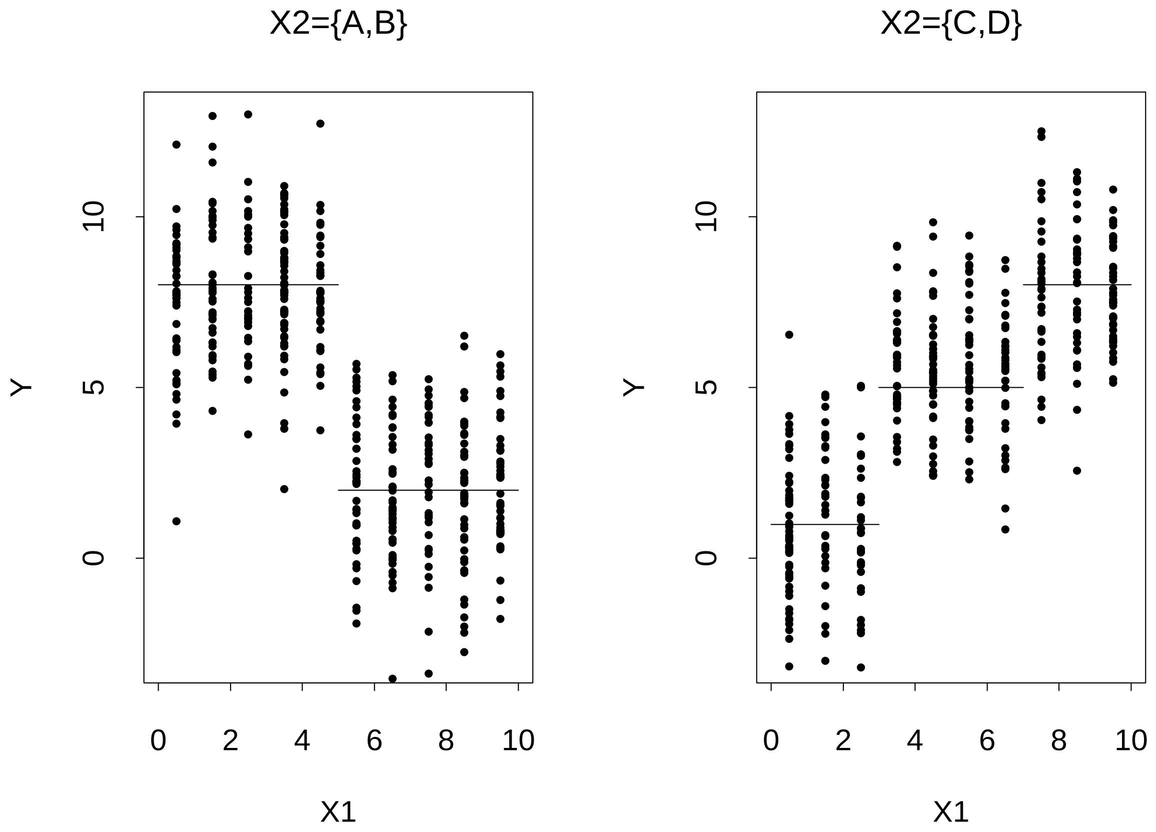

Figure 2: A realization of 800 observations sampled from the tree model depicted in Figure 1.

Before describing the example tree, we discuss the general structure of a RP model for the case of a binary tree. A binary tree $T$ subdivides the predictor space as follows: Each internal node has an associated splitting rule which uses a predictor to assign observations to either its left or right child node. The terminal nodes thus identify a partition of the predictor space according to the subdivision defined by the splitting rules. For quantitative predictors, the splitting rule is based on a split value $s$, and assigns observations for which $\{x_i \leq s\}$ or $\{x_i > s\}$ to the left or right child node respectively. For qualitative predictors, the splitting rule is based on a category subset $C$, and assigns observations for which $\{x_i \in C\}$ or $\{x_i \notin C\}$ to the left or right child node respectively.

Several assumptions have been made to simplify exposition. First, splitting rules are assumed to subdivide a region into two sub-regions, giving a binary tree. Second, only one predictor variable is assumed to be used for each splitting rule. Both these restrictions can be relaxed.

For illustration, Figure 1 depicts a regression tree model where $y \sim N(\theta, 2^2)$ and $x = (x_1, x_2)$. $x_1$ is a quantitative predictor taking values in [0,10], and $x_2$ is a qualitative predictor with categories $(A, B, C, D)$. The binary tree has 9 nodes of which $b = 5$ are terminal nodes. The terminal nodes subdivide the $x$ space into 5 nonoverlapping regions. The splitting variable and rule are displayed at each internal node. For example, the leftmost terminal node corresponds to $x_1 \leq 3.0$ and $x_2 \in \{C, D\}$. The $\theta_i$ value which identifies the mean of $y$ given $x$ is displayed at each terminal node. Note that $\theta_i$ decreases in $x_1$ when $x_2 \in \{A, B\}$, but increases in $x_1$ when $x_2 \in \{C, D\}$. A realization of 800 observations sampled from this model is displayed in Figure 2.

If $y$ were a qualitative variable, a classification tree model would be obtained by using an appropriate categorical distribution at each terminal node. For example, if $y$ was binary with categories $C_1$ or $C_2$, one might consider the Bernoulli model $P(y \in C_1) = \theta = 1 - P(y \in C_2)$ with a possibly different value of $\theta$ at each terminal node. A standard classification rule for this model would then classify $y$ into the category yielding the smallest expected misclassification cost. When all misclassification costs are equal, this would be the category with largest probability.

## 3 Learning the RP model

To learn or estimate a RP model, we assume that a training sample consisting of tuples $(x_i, y_i), i = 1, \ldots, n$ is available. Both the tree $T$ and the terminal node parameters $\Theta$ must be estimated using the training data.

For a fixed $T$, a common assumption is that the response values are i.i.d. within each terminal node. The data in each terminal node can be considered a separate sample, and conventional estimation techniques (e.g. maximum likelihood) yield familiar node parameter estimates $\hat{\theta}_j$ such as the sample mean for a continuous normal response and sample proportions for a categorical multinomial response.

Armed with a recipe for estimating $\Theta$ given $T$, we can now consider estimation of $T$. First, an objective function must be specified, providing a mechanism to assess the quality of a particular tree $T$. The log-likelihood of the training data is one such criterion. For a normal response model, the corresponding criterion would be the minimization of a residual sum of squares. For a multinomial response, the multinomial log-likelihood would be used. Ciampi (1991) was one of the first to develop a likelihood-based approach to RP models. Other criteria have been proposed for specific response classes, such as the Gini index (Breiman et. al. 1984) for a categorical response.

With an objective function quantifying the quality of a tree, the estimation problem becomes a search over all possible trees to optimize the objective. Although splitting rules for continuous $x$ are real-valued, the objective function will only change when training points are moved among terminal nodes of the tree. Thus it is common to consider only splitting rules defined at data points, and require that each terminal node contain at least one training point. The search over the set of trees is thus a combinatorial search over a finite but very large discrete space.

The most common search algorithm is a greedy forward search, in which all training observations are initially grouped into a single node. The algorithm considers splitting into two child nodes, examining all possible splits on all possible variables. The splitting rule yielding the best value of the objective function (e.g. the smallest residual sum of squares when summed over the two child nodes) is selected. The procedure is repeated in each child node recursively until a large tree is grown.

Several strategies can be employed to decide how large a tree to grow. In the CHAID algorithm of Kass (1980), hypothesis tests were used to decide when to stop subdividing, yielding a final tree. Breiman et. al. (1984) suggest growing a maximal tree, and then pruning away sibling nodes that do not significantly improve the objective function over the value assigned to their parent node. Their reasoning was that the forward greedy search might sometimes stop early, missing significant effects. For example, in the tree displayed earlier, no initial split leads to a large reduction in residual sum of squares because of the interaction pattern. Their backward pruning was facilitated by the idea of cost-complexity pruning, in which a modified objective function was minimized:

$$\mathrm{Loss}(T;\alpha) = \mathrm{RSS}(T) + \alpha|T|, \tag{1}$$

where $|T|$ represents the number of terminal nodes of the tree. Penalty parameter $\alpha \geq 0$ controls the trade-off between tree size and accuracy. Breiman et. al. showed that (1) can be minimized as $\alpha$ increases from 0 to $\infty$ by considering a nested sequence of pruned trees, starting with the largest tree identified. The optimal $\alpha$ and a corresponding tree are selected so as to minimize a cross-validated estimate of the objective function.

While other methods for identifying the best tree have been proposed, the greedy forward search is quick and can be quite effective.

## 4 Strengths and weaknesses of RP models

The structure of RP models enables them to identify **interactions**. For instance, in Figures 1 and 2, we see an interaction effect between $X_1$ and $X_2$: If $X_2 = \{A, B\}$ then response $y$ decreases with increasing $X_1$. If $X_2 = \{C, D\}$ then response $y$ increases with increasing $X_1$. This is perhaps the greatest strength of RP models, and one of the reasons they are used for exploratory data analysis.

This strength is also a weakness. If the relation between predictors and response is **additive**, very large trees will be needed to capture this relationship. For instance, if

$$y = x_1 + x_2 + x_3 + x_4 + x_5 + \text{error},$$

then a tree with 32 terminal nodes will be required to even approximate this function with a single step along each of the five predictor axes.

Trees are popular among practitioners because of their **interpretability**. It is natural to interpret the sequence of conditions leading to a terminal node of a tree. Care must be taken with such interpretations, especially if dependencies exist among predictors. In such cases, multiple trees with different splits on different variables may fit the data equally well.

In addition to dealing with mixed predictor types, RP models can handle missing values of predictors via several strategies. For missing predictor values in the training data, one could (i) treat "missing" as a new category for a categorical predictor, or (ii) identify surrogate splitting variables that produce splits similar to a missing predictor. If predictor values are missing when making predictions for new observations, either of these strategies may be employed, or one may terminate the branching process when a missing value is needed in a branch, and base predictions on the interior node.

The most common form of RP models utilize a single variable for each splitting rule. This **axis alignment** aids in interpretability, but can be a weakness if variation in the response occurs along a linear combination of predictors, rather than along the axes. The additive function of five variables mentioned above is an example of this.

By virtue of subdividing the data into smaller subgroups, an RP model can suffer from **sparsity**, especially if more complex statistical models are utilized in the terminal nodes. For instance, a significant challenge in modifying RP models for survival data with censoring (Leblanc and Crowley 1993) is the pooled nature of Kaplan-Meier estimates of the survival curve. This data sparsity is one of the primary reasons for the use of simple models in terminal nodes.

A weakness of RP models is **sensitivity** of results to small data perturbations. Breiman (1996) demonstrated that when RP models were fit to bootstrap samples of the data, there could be substantial variation in tree structure. While this would seem to be a weakness, Breiman leveraged this idea to produce Ensemble methods discussed below in Section 6.

Because of the greedy nature of the search over the space of trees, inference for the resultant model is difficult. Although confidence intervals and hypothesis tests can easily be constructed conditional on a specific tree $T$, the adaptive nature of the learning algorithm means that the statistical properties of estimators, intervals and tests will be seriously undermined. Methods that take account of the search include adjustments for multiple testing (Hawkins and Kass 1982) and Bayesian approaches (Chipman, George and McCulloch 1998; Denison Mallick and Smith 1998).

## 5 Extensions

The popularity of RP models has lead to a number of extensions and the development of related methods.

A variety of search strategies have been proposed as alternatives to the greedy forward stepwise approach. These include the use of stochastic search optimizers such as genetic algorithms (Fan and Gray 2005) and simulated annealing (Sutton 1991; Lutsko and Kuijpers 1994) and MCMC (Chipman et. al. 1998, Denison et. al. 1998). Tibshirani and Knight (1999) used the bootstrap to perturb data before executing a greedy search.

Variations on the tree structure have also been considered, including splitting rules based on linear combinations of real-valued predictors (Loh and Vanichsetakul, 1988). Some RP algorithms (e.g., AID) allow nodes to have more than two child nodes, complicating the search but sometimes making interpretation clearer. Quinlan's C4.5 splits categorical predictors by generating a different child node for each categorical level of the corresponding predictor.

The statistical model in terminal nodes has also been extended to richer models, such as linear regression (Alexander and Grimshaw, 1996; Chipman, George and McCulloch, 2002), generalized linear models (Chipman, George and McCulloch, 2003), and Gaussian process models (Gramacy and Lee, 2008).

# 6 Ensembles of trees

RP models have been used as a "base learner" in a number of algorithms that seek to achieve greater predictive accuracy by combining together multiple instances of a model.

In noticing the sensitivity of trees to small perturbations, Breiman (1996) developed a strategy known as bootstrap aggregation or "Bagging" for generating multiple trees and combining them to achieve greater prediction accuracy. For instance, with a continuous response, each bootstrap tree would be used to generate predictions at a particular test point, and these predictions would be averaged to form an ensemble prediction.

A further enhancement led to Random Forests (Breiman, 2001). Additional variation in the search algorithm was introduced by randomizing the choice of predictor in splitting rules. This led to a richer set of trees, and could further improve predictive accuracy.

Another form of ensemble model using RP models is boosting (Freund and Schapire, 1997). In this algorithm, a sequence of RP models are learned, each depending on those already identified via data weights that depend on predictive accuracy of earlier RP models. These weights encourage the next RP model to better fit those observations that have been incorrectly classified. At the end of the boosting sequence, an ensemble prediction is generated by a weighted combination of predictions from each learner in the ensemble.

Although neither boosting or random forests require that the base learner be a RP model, these have yielded the most popular and successful form of ensemble model.

# 7 Related work

A model closely related to RP models is the hierarchical mixture of experts model (Jordan and Jacobs, 1994). In this model, a different logistic function of the predictors is used in each interior node to probabilistically assign data points to the left and right children. In doing so, the hard boundaries associated with splitting rules are replaced with soft decisions indexed by continuous parameters. In terminal nodes, predictions are given by logistic regression. Tree size and topology is typically fixed in advance, and the tree learning algorithm becomes a continuous optimization problem.

## References

Alexander, W. P. and Grimshaw, S. D. (1996) "Treed Regression", *Journal of Computational and Graphical Statistics*, 5, 156–175.

Breiman, L., Friedman, J. Olshen, R. and Stone, C. (1984), *Classification and Regression Trees*, Wadsworth.

Breiman, L (1996), "Bagging Predictors", *Machine Learning*, 24, 123–140.

Breiman, L (2001) "Random Forests", *Machine Learning*, 45, 5–32.

Chipman, H. A., George, E.I. and McCulloch, R. E. (1998) "Bayesian CART model search", *Journal of the American Statistical Association*, 93, 935-948.

Chipman, H. A., George, E. I, and McCulloch, R. E. (2002) "Bayesian Treed Models", Machine Learning, 48, 299-320.

Chipman, H. A., George, E. I, and McCulloch, R. E. (2003) "Bayesian Treed Generalized Linear Models", in *Bayesian Statistics 7*, J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, Eds., Oxford University Press, Oxford, UK.

Ciampi, A. (1991) "Generalized Regression Trees", *Computational Statistics and Data Analysis*, 12, 57–78.

Denison, D., Mallick, B. and Smith, A.F.M. (1998) "A Bayesian CART Algorithm", *Biometrika* , 85, 363-377.

Fan, G., and Gray, J.B. (2005) "Regression analysis using TARGET", *Journal of Computational and Graphical Statistics*, 14, 206-218.

Gramacy, R. B. and Lee, H.K.H. (2008) "Bayesian treed Gaussian process models with an application to computer modeling", *Journal of the American Statistical Association*, 103, 1119-1130.

Hawkins, D. M. and Kass, G. V. (1982) "Automatic Interaction Detection", in *Topics in Applied Multivariate Analysis*, D. M. Hawkins, Ed., Cambridge University Press, Cambridge, UK.

Jordan, M.I. and Jacobs, R.A. (1994), " Mixtures of Experts and the EM Algorithm", *Neural Computation*, 6, 181-214.

Kass, G.V. (1980) "An exploratory technique for investigating large quantities of categorical data", *Applied Statistics*, 29, 119-127.

LeBlanc, M. and Crowley, J. (1993) “Survival Trees by Goodness of Split”, *Journal of the American Statistical Association*, 88, 457-467.

Loh, W.-Y. and Vanichsetakul, N. (1988) “Tree-structured Classification Via Generalized Discriminant Analysis”, *Journal of the American Statistical Association*, 83, 715-725.

Lutsko, J. F. and Kuijpers, B. (1994) “Simulated Annealing in the Construction of Near-Optimal Decision Trees”, in *Selecting Models from Data: AI and Statistics IV*, P. Cheeseman and R. W. Oldford, Eds., 453–462.

Morgan, J. A. and Sonquist, J.N. (1963) “Problems in the analysis of survey data: and a proposal” *Journal of the American Statistical Association*, 58, 415-34.

Quinlan, J. R. (1986) “Induction of Decision Trees”, *Machine Learning*, 1, 81–106.

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

Sutton, C. (1991), “Improving Classification Trees with Simulated Annealing”, *Proceedings of the 23rd Symposium on the Interface*, E. Keramidas, Ed., Interface Foundation of North America.

Tibshirani, R., and Knight, K. (1999), “Model Search by Bootstrap ‘Bumping’ ”, Journal of Computational and Graphical Statistics, 8, 671-686.