Model-based Recursive Partitioning

Achim Zeileis
Wirtschaftsuniversität Wien

Torsten Hothorn
Friedrich-Alexander-Universität Erlangen-Nürnberg

Kurt Hornik
Wirtschaftsuniversität Wien

Abstract

Recursive partitioning is embedded into the general and well-established class of parametric models that can be fitted using M-type estimators (including maximum likelihood). An algorithm for model-based recursive partitioning is suggested for which the basic steps are: (1) fit a parametric model to a data set, (2) test for parameter instability over a set of partitioning variables, (3) if there is some overall parameter instability, split the model with respect to the variable associated with the highest instability, (4) repeat the procedure in each of the daughter nodes. The algorithm yields a partitioned (or segmented) parametric model that can be effectively visualized and that subject-matter scientists are used to analyze and interpret.

Keywords: change points, maximum likelihood, parameter instability, recursive partitioning.

1. Introduction

Since the appearance of the first tree-structured regression analysis (Automated Interaction Detection, **?**), virtually every publication in this field highlights two features of trees: 1. interpretability—enhanced by visualizations of the fitted decision trees—and 2. predictive power in non-linear regression relationships. The latter is of diminishing importance because modern approaches to predictive modeling such as boosting (e.g., simple L_2 boosting by **?**), random forests (**?**) or support vector machines (**?**) are often found to be superior to trees in purely predictive settings (e.g., **?**). However, a simple graphical representation of a complex regression problem is still very valuable, probably increasingly so.

In the last decade, the incorporation of (simple) parametric models into trees has been receiving increased interest. Research in this direction was mainly motivated by the fact that constant fits in each node tend to produce large and thus hard to interpret trees (see e.g., **?**). Several algorithms have been suggested both in the statistical and machine learning communities that attach parametric models to terminal nodes or employ linear combinations to obtain splits in inner nodes. In machine learning, such approaches are known as *hybrid*, *model* or *functional trees* (**?**) with M5 (**?**) being the most prominent representative. The key developments in statistics are due to Wei-Yin Loh and his coworkers. GUIDE (**?**), CRUISE (**?**) and LOTUS (**?**) attach parametric models to terminal nodes, and **?** suggest an extension to count data. Some of these algorithms (in particular CRUISE or QUEST, **?**) additionally allow to employ parametric models to obtain splits in inner nodes. Furthermore, maximum likelihood trees (**?**) embed regression trees with a constant fit in each terminal node into maximum likelihood estimation.

Building on these ideas, we carry the integration of parametric models into trees one step further and provide a rigorous theoretical foundation by introducing a new unified framework that embeds recursive partitioning into statistical model estimation and variable selection. Within this framework, a segmented parametric model is fitted by computing a tree in which every leaf is associated with a fitted model such as, e.g., a maximum likelihood model or a linear regression. The model's objective function is used for estimating the parameters and the split points; the corresponding model scores are tested for parameter instability in each node to assess which variable should be

used for partitioning. The benefits of employing this approach are: The objective function used for parameter estimation is also used for partitioning (testing *and* split point estimation). The recursive partitioning allows for modeling of non-linear relationships and automated detection of interactions among the explanatory variables. The statistical formulation of the algorithm ensures the validity of interpretations drawn from the depicted model. Moreover, the use of well-known parametric models provides subject-matter scientists with a segmented model that they are used to analyze and interpret.

The remainder of the paper is organized as follows: Section **??** establishes the class of models the framework is based on before Section **??** describes the suggested model-based recursive partitioning algorithm in detail. Section **??** provides a set of illustrations and applications along with benchmark comparisons with other tree-based algorithms. Section **??** provides a short discussion of some further details of the algorithm before Section **??** concludes the paper with a summary.

2. Segmented models

Consider a parametric model $\mathcal{M}(Y, \theta)$ with (possibly vector-valued) observations $Y \in \mathcal{Y}$ and a k-dimensional vector of parameters $\theta \in \Theta$. Given n observations Y_i $(i = 1, \ldots, n)$ the model can be fitted by minimizing some objective function $\Psi(Y, \theta)$ yielding the parameter estimate $\hat{\theta}$

$$\hat{\theta} \quad = \quad \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi(Y_i, \theta). \tag{1}$$

Estimators of this type include various well-known estimation techniques, the most popular being ordinary least squares (OLS) or maximum likelihood (ML) among other M-type estimators. In the case of OLS, Ψ is typically the error sum of squares and, in the case of ML, it is the negative log-likelihood. In the latter case, it could be the full likelihood of the variable Y or the conditional likelihood if Y can be split into dependent and explanatory variables $Y = (y, x)^\top$.

Example: (Multivariate) normal distribution. The observations Y are normally distributed with mean μ and covariance matrix Σ: $Y \sim \mathcal{N}(\mu, \Sigma)$ with the combined parameter vector $\theta = (\mu, \Sigma)$.

Example: Generalized linear model (GLM). The observations can be split into a dependent variable y and covariates or regressors x, i.e., $Y = (y, x)^\top$. The model equation is $g(\mathsf{E}(y)) = x^\top \theta$ where y has a pre-specified exponential family distribution, $g(\cdot)$ is a known link function and θ are the regression coefficients.

In many situations, it is unreasonable to assume that a single global model $\mathcal{M}(Y, \theta)$ fits all n observations well. But it might be possible to partition the observations with respect to some covariates such that a well-fitting model can be found in each cell of the partition. In such a situation, we can use a recursive partitioning approach based on ℓ partitioning variables $Z_j \in \mathcal{Z}_j$ $(j = 1, \ldots, \ell)$ to adaptively find a good approximation of this partition.

More formally, we assume that a partition $\{\mathcal{B}_b\}_{b=1,\ldots,B}$ of the space $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_\ell$ exists with B cells (or segments) such that in each cell $\mathcal{B}_b$ a model $\mathcal{M}(Y, \theta_b)$ with a cell-specific parameter θ_b holds. We denote this segmented model by $\mathcal{M}_\mathcal{B}(Y, \theta)$ where θ is now the full combined parameter $\theta = (\theta_1, \ldots, \theta_B)^\top$.

Special cases of such segmented models are classification and regression trees where many partitioning variables Z_j but only very simple models $\mathcal{M}$ are used, and structural break models that find partitions with respect to time.

Example: For regression trees a simple model $\mathcal{M}$ is chosen: the parameter θ describes the mean of the univariate observations Y_i and is estimated by OLS (or equivalently ML in a normal model with the variance treated as a nuisance parameter). The variables Z_j are the regressors considered for partitioning.

Example: In change point or structural change analysis, typically a linear regression model with $Y_i = (y_i, x_i)^\top$ and regression coefficients θ is segmented with respect to only a single variable Z_1 (i.e., $\ell = 1$) which is usually time (**??**).

Given the correct partition $\{\mathcal{B}_b\}$ the estimation of the parameters θ that minimize the global objective function can be easily achieved by computing the locally optimal parameter estimates $\hat{\theta}_b$ in each segment $\mathcal{B}_b$. However, if $\{\mathcal{B}_b\}$ is unknown, minimization of the global objective function

$$\sum_{b=1}^{B} \sum_{i \in I_b} \Psi(Y_i, \theta_b) \quad \rightarrow \quad \min, \tag{2}$$

over all conceivable partitions $\{\mathcal{B}_b\}$ (with corresponding indexes I_b, $b = 1, \ldots, B$) is more complicated, even if the number of segments B is fixed: If there is more than one partitioning variable ($\ell > 1$), the number of potential partitions quickly becomes too large for an exhaustive search. If, in addition, the number of segments B is unknown, the problem becomes even more severe—at least, if trivial partitions, such as the partition, where each observation is its own segment, are excluded, e.g., by requiring some minimum segment size. Furthermore, in this case, some means should be taken to avoid overfitting by increasing B.

In short, determining the optimal partition (with respect to Ψ) is difficult, even for fixed B. However, if there is only $\ell = 1$ partitioning variable, the optimal split(s) can be found easily: both the statistics and econometrics literature on change point and structural change analysis discuss various algorithms for segmenting models over a single variable, typically time. To exploit this methodology for finding a partition close to the optimal one in $\ell > 1$ dimensions, we suggest a greedy forward search where the objective function Ψ can at least be optimized locally in each step. A detailed description of this algorithm is given in the next section.

3. The recursive partitioning algorithm

The basic idea is that each node is associated with a single model. To assess whether splitting of the node is necessary a fluctuation test for parameter instability is performed. If there is significant instability with respect to any of the partitioning variables Z_j, split the node into B locally optimal segments and repeat the procedure. If no more significant instabilities can be found, the recursion stops and returns a tree where each terminal node (or leaf) is associated with a model of type $\mathcal{M}(Y, \theta)$. More precisely, the steps of the algorithm are

1. Fit the model once to all observations in the current node by estimating $\hat{\theta}$ via minimization of the objective function Ψ.
2. Assess whether the parameter estimates are stable with respect to every ordering $Z_1, \ldots, Z_\ell$. If there is some overall instability, select the variable Z_j associated with the highest parameter instability, otherwise stop.
3. Compute the split point(s) that locally optimize Ψ, either for a fixed or adaptively chosen number of splits.
4. Split the node into daughter nodes and repeat the procedure.

The details for steps 1–3 are specified in the following. To keep the notation simple, the dependence on the current segment is suppressed and the symbols established for the global model are used, i.e., n for the number of observations in the current node, $\hat{\theta}$ for the associated parameter estimate and B for the number of daughter nodes chosen.

3.1. Parameter estimation

This step of the algorithm is common practice: Under mild regularity conditions (see e.g., **?**), it can be shown that the estimate $\hat{\theta}$ defined by Equation **??** can also be computed by solving the first order conditions

$$\sum_{i=1}^{n} \psi(Y_i, \hat{\theta}) \quad = \quad 0, \tag{3}$$

where

$$\psi(Y, \theta) \quad = \quad \frac{\partial \Psi(Y, \theta)}{\partial \theta} \tag{4}$$

is the score function or estimating function corresponding to $\Psi(Y, \theta)$. Analytical closed form solutions for $\hat{\theta}$ are available only in certain special cases, but for many models of interest well-established fitting algorithms for computing $\hat{\theta}$ are available (e.g., OLS estimation via QR decomposition for linear regression or ML via iterative weighted least squares for GLMs). The score function evaluated at the estimated parameters $\hat{\psi}_i = \psi(Y_i, \hat{\theta})$ is then inspected for systematic deviations from its mean 0 in the next section.

3.2. Testing for parameter instability

The task in this step of the algorithm is to find out whether the parameters of the fitted model are stable over each particular ordering implied by the partitioning variables Z_j or whether splitting the sample with respect to one of the Z_j might capture instabilities in the parameters and thus improve the fit. To assess the parameter instability, a natural idea is to check whether the scores $\hat{\psi}_i$ fluctuate randomly around their mean 0 or exhibit systematic deviations from 0 over Z_j. These deviations can be captured by the empirical fluctuation process

$$W_j(t) \quad = \quad \hat{J}^{-1/2} n^{-1/2} \sum_{i=1}^{\lfloor nt \rfloor} \hat{\psi}_{\sigma(Z_{ij})} \qquad (0 \leq t \leq 1) \tag{5}$$

where $\sigma(Z_{ij})$ is the ordering permutation which gives the antirank of the observation Z_{ij} in the vector $Z_j = (Z_{1j}, \ldots, Z_{nj})^\top$. Thus, $W_j(t)$ is simply the partial sum process of the scores ordered by the variable Z_j, scaled by the number of observations n and a suitable estimate $\hat{J}$ of the covariance matrix $\mathsf{COV}(\psi(Y, \hat{\theta}))$, e.g., $\hat{J} = n^{-1} \sum_{i=1}^{n} \psi(Y_i, \hat{\theta}) \psi(Y_i, \hat{\theta})^\top$, but other robust estimators such as HC (heteroskedasticity consistent) and HAC (heteroskedasticity and autocorrelation consistent) estimators are also applicable. This empirical fluctuation process is governed by a functional central limit theorem (**?**) under the null hypothesis of parameter stability: it converges to a Brownian bridge W^0. A test statistic can be derived by applying a scalar functional $\lambda(\cdot)$ capturing the fluctuation in the empirical process to the fluctuation process $\lambda(W_j(\cdot))$ and the corresponding limiting distribution is just the same functional (or its asymptotic counterpart) applied to the limiting process $\lambda(W^0(\cdot))$.

This very general framework for testing parameter stability is called generalized M-fluctuation test and has been established by **?**. It has been shown to encompass a large number of structural change tests suggested both in the econometrics and statistics literature, including OLS-based CUSUM and MOSUM tests (**??**), score-based tests (**??**) and statistics based on Lagrange multiplier statistics (**??**)—an overview is given in **?**. In principle, any of the tests from this framework could be used in the recursive partitioning algorithm, but two different test statistics seem to be particularly attractive for assessing numerical and categorical partitioning variables Z_j respectively.

Assessing numerical variables: To capture the instabilities over a numerical variable Z_j, the following functional is most intuitive:

$$\lambda_{\sup LM}(W_j) \quad = \quad \max_{i = \underline{i}, \ldots, \overline{i}} \left(\frac{i}{n} \cdot \frac{n-i}{n} \right)^{-1} \left\| W_j \left(\frac{i}{n} \right) \right\|_2^2, \tag{6}$$

which is the maximum of the squared L_2 norm of the empirical fluctuation process scaled by its variance function. This is the supLM statistic of **?** which can be interpreted as the supremum of LM statistics against a single change point alternative where the potential change point is shifted over the interval $[\underline{i}, \overline{i}]$ that is typically defined by requiring some minimal segment size $\underline{i}$ and then $\overline{i} = n - \underline{i}$.[1] This test statistic is asymptotically equivalent to the supremum of likelihood-ratio (or

[1] To avoid that the test becomes liberal, it can additionally be imposed that in each node at least a certain fraction of the current observations are trimmed on each end. Typically, 10% are used for this (**?**). The trimming affects only the tests not the subsequent splitting.

Chow) statistics (**?**) but has the advantage that the model has to be fitted only once under the null hypothesis of parameter stability and not under the alternative for each conceivable change point. The limiting distribution of the supLM statistic is given by the supremum of a squared, k-dimensional tied-down Bessel process $\sup_t (t(1-t))^{-1} ||W^0(t)||_2^2$ from which the corresponding p value p_j can be computed (**?**).

Recently, **?** have also pointed out that parameter instability tests can be employed for growing linear regression trees. However, our approach makes the evaluation of all conceivable change points feasible (and not only a fixed number, such as 5), uses the correct limiting distribution (rather than a crude Bonferroni approximation) and can be applied to a more general class of models.

Assessing categorical variables: To capture the instability with respect to a categorical variable Z_j with C different levels or categories, a different statistic is required because, by definition, Z_j has ties and hence a total ordering of the observations is not possible. The most natural statistic, which is insensitive to the ordering of the C levels and the ordering of observations within each level, is given by

$$\lambda_{\chi^2}(W_j) \quad = \quad \sum_{c=1}^{C} \frac{|I_c|}{n}^{-1} \left\| \Delta_{I_c} W_j \left(\frac{i}{n} \right) \right\|_2^2 \tag{7}$$

where $\Delta_{I_c} W_j$ is the increment of the empirical fluctuation process over the observations in category $c = 1, \ldots, C$ (with associated indexes I_c), i.e., essentially the sum of the scores in category c. The test statistic is then the weighted sum of the squared L_2 norm of the increments which has an asymptotic χ^2 distribution with $k \cdot (C-1)$ degrees of freedom from which the corresponding p value p_j can be computed (**?**).

The advantage of using this approach, based on the empirical fluctuation processes from Equation **??** with the functionals from Equations **??** and **??**, is that the parameter estimates and corresponding score functions just have to be computed once in a node. For performing the parameter instability tests, the scores just have to be reordered and aggregated to a scalar test statistic each time.

To test whether there is some overall instability in the current node, it just has to be checked whether the minimal p value $\min_{j=1,\ldots,\ell} p_j$ falls below a pre-specified significance level α, that is typically corrected for multiple testing. If this is the case, the variable Z_{j^*} associated with the minimal p value is chosen for splitting the model in the next step of the algorithm.

3.3. Splitting

In this step of the algorithm the fitted model has to be split with respect to the variable Z_{j^*} into a segmented model with B segments where B can either be fixed or determined adaptively. For a fixed number of splits, two rival segmentations can be compared easily by comparing the segmented objective function $\sum_{b=1}^{B} \sum_{i \in I_b} \Psi(Y_i, \theta_b)$. Performing an exhaustive search over all conceivable partitions with B segments is guaranteed to find the optimal partition but might be burdensome, so several search methods are briefly discussed for numerical and categorical partitioning variables respectively.

Splitting numerical variables: Exhaustive search for a split into $B = 2$ segments is feasible in $O(n)$ operations. For $B > 2$, an exhaustive search would be of order $O(n^{B-1})$—however, the optimal partition can be found using a dynamic programming approach of order $O(n^2)$. This is an application of Bellman's principle and has been discussed in several places in the statistics and econometrics literature on change point and structural change analysis (see e.g., **???**, among others). Alternatively, iterative algorithms can be used that are known to converge to the optimal solution (e.g., **?**). If B is not fixed, but should be chosen adaptively, various methods are available (see e.g., **??**). In particular, information criteria can be used if the parameters are estimated by ML.

Splitting categorical variables: For categorical variables, the number of segments can not be larger than the number of categories $B \leq C$. Two simple approaches would be either to always split into all $B = C$ possible levels or alternatively to always split into the minimal number of $B = 2$ segments. In the latter case, the search for the optimal partition is of order $O(2^{C-1})$. For ordinal variables, it also makes sense to just split in the ordering of the levels, so that the search for a binary split is only of order $O(C)$. Again, information criteria could be an option to adaptively determine the number of splits, although this is less intuitive than for numerical variables.

In summary, two plausible strategies would be either to always use binary splits, i.e., use a fixed $B = 2$, or to determine B adaptively for numerical variables while always using $B = C$ for categorical variables. In Section **??** below, we adopt the former strategy of binary splits.

This concludes one iteration of the recursive partitioning algorithm and steps 1–3 are carried out again in each of the B daughter nodes until no significant instability is detected in step 2.

4. Illustrations and applications

To illustrate how model-based recursive partitioning can be used in practice, the general framework from the previous section is applied to various regression problems (linear regression, logistic regression, and survival regression). Four different data sets are analyzed in the following way: In a first step, the recursively partitioned model is fitted to the data and visualized for explanatory analysis, emphasizing that the algorithm can be used to build intelligible local models by automated interaction detection. In a second step, the performance of the algorithm is compared with other tree-based algorithms in two different respects: prediction and complexity. Comparing predictive performance of different learning algorithms is established practice—for (model-based) recursive partitioning comparing the model complexity (i.e., the number of splits and estimated coefficients) is equally important. As argued above, the strength of single tree-based classifiers is not so much predictive power alone, but that the algorithms are able to build interpretable models. Clearly, more parsimonious models are easier to interpret and hence are to be preferred (among those with comparable predictive performance).

For the linear regression applications, the model-based (MOB) recursive partitioning algorithm introduced here is compared to other algorithms previously suggested in the literature: GUIDE (**?**) and M5' (**?**), which is a rational reconstruction of M5 (**?**), as linear model trees; as well as CART (classification and regression trees, **?**) and conditional inference trees (CTree, **?**) as trees with constant models in the nodes. The logistic regression-based MOB trees are compared with logistic model trees (LMT, **?**) as well as various tree-based algorithms with constant models in the nodes: QUEST (**?**), the J4.8 implementation of C4.5 (**?**), CART and CTree.

All benchmark comparisons are carried out in the framework of **?** based on 250 bootstrap replications and employing the root mean squared error (RMSE) or misclassification rate on the out-of-bag (OOB) samples as predictive performance measure and the number of estimated parameters (splits and coefficients) as complexity measure. The median performances[2] on the bootstrap replications are reported in tabular form, simultaneous confidence intervals for performance differences (obtained by treatment contrasts with MOB as the reference category) are visualized. In addition, the tables contain the obvious complexity and prediction performance measures (RMSE or misclassification) on the original data set as an additional reference information (although the obvious prediction measures obviously represent no honest estimators).

Most computations have been carried out in the R system for statistical computing (**?**), in particular using the packages **party** (**?**), providing implementations of MOB and CTree, **rpart** (**?**), implementing CART, and **RWeka** (**?**), the R interface to **Weka** (**?**) containing implementations of M5', LMT and J4.8. For GUIDE and QUEST, the binaries distributed at `http://www.stat.wisc.edu/~loh/` were used.

[2]The median rather than the mean is used for two reasons: First, to account for the skewness of the performance distributions, and second due to the many ties in the complexity measure for the model-based tree algorithms (MOB, GUIDE, M5', LMT).