Unbiased Recursive Partitioning II: A Parametric Framework Based on Parameter Instability Tests

Achim Zeileis Torsten Hothorn Kurt Hornik

http://www.ci.tuwien.ac.at/~zeileis/
Overview

- Model-based recursive partitioning
  - Parametric models
  - Parameter estimation
  - Segmented models
- The recursive partitioning algorithm
  - Tests for parameter instability
  - Assessing numerical/categorical variables
- Illustrations
  - Artificial data
  - Boston housing data
- Summary
Model-based recursive partitioning

Starting point: Recursive partitioning algorithms (including conditional inference trees) learn a partition/segmentation from data and then fit a naive model in each terminal node, e.g., a mean, relative frequencies or a Kaplan-Meier curve.

Idea: Employ parametric models in each node.

Goal: Algorithm for constructing segmented parametric models by recursive partitioning.
Parametric models

Consider models $M(Y, \theta)$ 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 fit by minimizing some objective function $\Psi(Y, \theta)$ yielding the parameter estimate $\hat{\theta}$

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n} \Psi(Y_i, \theta).$$
Parameter estimation

Under mild regularity conditions it can be shown that the estimate $\hat{\theta}$ can also be computed by solving the first order conditions

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

where

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

is the score function or estimating function corresponding to $\Psi(Y, \theta)$. 
Parameter estimation

This type of estimators includes maximum likelihood (ML), ordinary least squares (OLS), Quasi-ML and further M-type estimators.

Example: \( M(Y, \theta) \) could be a multivariate normal model for \( Y \sim \mathcal{N}(\mu, \Sigma) \) such that \( \theta = (\mu, \Sigma) \).

Example: \( M(Y, \theta) \) could be a generalized linear model for \( Y = (y, x)^\top \) such that

\[
g(\mathbb{E}(y)) = x^\top \theta.
\]
**Segmented models**

**Idea:** In many situations, it is unreasonable to assume that a single global model $\mathcal{M}(Y, \theta)$ can be fit to all $n$ observations. But it might be possible to partition the observations with respect to covariates $Z = (Z_1, \ldots, Z_l)$ such that a fitting model can be found in each cell of the partition.

**Goal:** Learn partition via recursive partitioning with respect to $Z_j \in Z_j$ ($j = 1, \ldots, l$).
Segmented models

**Example:** Regression trees.
The parameter $\theta$ describes the mean of the univariate observations $Y_i$ and is estimated by OLS or equivalently ML in a normal model. The variables $Z_j$ are the regressors considered for partitioning.

**Example:** Changepoint or structural change analysis.
A (generalized) linear regression model with $Y_i = (y_i, x_i)^\top$ and regression coefficients $\theta$ is segmented with respect to a single variable $Z_1$ (i.e., $l = 1$), typically time.
Segmented models

Given a partition, the estimation of the parameters $\theta$ that minimize the corresponding global objective function $\sum_{b=1}^{B} \sum_{i \in I_b} \psi(Y_i, \theta^{(b)})$ can be easily achieved by computing the locally optimal parameter estimates $\hat{\theta}^{(b)}$ in each segment $b$ (with corresponding indices $I_b$).

If it is unknown, minimization of $\psi$ is more complicated (if trivial partitions are excluded). But it is easily possible to optimally split the observations with respect to only a single variable $Z_1$ into $B$ segments. Typically $B = 2$ is chosen.
A single optimal split into $B = 2$ partitions can easily be computed in $O(n)$ by exhaustive search.

For $B > 2$, when an exhaustive search would be of order $O(n^{B-1})$, the optimal partition can be found using a dynamic programming approach of order $O(n^2)$ (Hawkins, 2001; Bai & Perron, 2003) or via iterative algorithms (Muggeo, 2003).

Various algorithms for adaptively choosing the number of segments $B$ are available, e.g., via information criteria.
The recursive partitioning algorithm

The generic recursive partitioning algorithm presented in Part I can be used almost directly.

The only difference is that now each node is associated with a parametric model.

**Question**: How should we assess the association of a fitted model with a covariate $Z_j$?

**Answer**: Test for instability of the parameters of the model with respect to this variable $Z_j$. 
The recursive partitioning algorithm

1. Fit the model once to all observations in the current node by estimating \( \hat{\theta} \) via minimization of \( \Psi \).
2. Assess whether the parameter estimates are stable with respect to every ordering \( Z_1, \ldots, Z_l \). 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 \( \Psi \) (either for a fixed number of splits, or choose the number of splits adaptively).
4. Split this node into daughter nodes and repeat the procedure.
Generalized M-fluctuation tests (Zeileis & Hornik, 2003) can be used for assessing whether the parameter estimates $\hat{\theta}$ are stable over a certain variable or not.

The basic idea is to use an empirical fluctuation process of cumulative scores for a particular ordering of the observations

$$W(t, \hat{\theta}) = \hat{J}^{-1/2}n^{-1/2} \sum_{i=1}^{\lfloor nt \rfloor} \psi(Y_i, \hat{\theta}) \quad (0 \leq t \leq 1)$$

which is governed by a functional central limit theorem (FCLT). It converges to a Brownian bridge $W^0$. 

Tests for parameter instability
Tests for parameter instability

A test statistic can be derived by applying a scalar functional \( \lambda(\cdot) \) to the fluctuation process, the limiting distribution is just the same functional (or its asymptotical counterpart) applied to the limiting process \( \lambda(W^0(\cdot)) \).

**Advantage:** The model just has to be estimated once. For testing, the scores of the fitted model \( \hat{\psi} \) just have to be re-ordered for each variable.

Let \( W_j(t) \) be the fluctuation process for the observations ordered by \( Z_j \).
Assessing numerical variables

The most intuitive functional for assessing the stability with respect to a numerical partitioning variable $Z_j$ is the supLM statistic of Andrews (1993).

$$\lambda_{\text{supLM}}(W_j) = \max_{i=\bar{i},...,\bar{i}} \left( \frac{i}{n} \cdot \frac{n-i}{n} \right)^{-1} \left\| W_j \left( \frac{i}{n} \right) \right\|_2^2.$$  

This gives the maximum of the single changepoint LM statistics over all possible changepoints in $[\bar{i}, \bar{i}]$.

The limiting distribution is given by the supremum of a squared, $k$-dimensional tied-down Bessel process.
Assessing categorical variables

To assess the stability of a categorical variable with $C$ levels, a $\chi^2$ statistics is most intuitive

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

because it is insensitive to re-ordering of the levels and the observations within the levels.

It essentially captures the instability when splitting the model into $C$ groups.

The limiting distribution is $\chi^2$ with $k \cdot (C - 1)$ degrees of freedom.
Pruning

The algorithm described so far employs a **pre-pruning** strategy, i.e., uses an internal stopping criterion: if no variable exhibits significant association, i.e., significant parameter instability, the algorithm stops.

Alternatively/additionally, a **post-pruning** strategy can be used. This seems particularly attractive if ML is used for parameter estimation. Then a ML tree can be grown which is consequently associated with a segmented ML model. This can be pruned afterwards using information criteria for example.
Example: Artificial data

Artificial data from a segmented univariate linear regression. The segmentation is explained by 2 numerical partitioning variables. Furthermore, 2 numerical and 2 categorical variables with additional “noise” are in the data set.

The data-generating mechanism is:

\[
\begin{align*}
    a \leq 1 : & \quad y = 1 + x + \varepsilon, \\
    a > 1, b \leq 1 : & \quad y = 2 + x + \varepsilon, \\
    a > 1, b > 1 : & \quad y = 2 + \varepsilon,
\end{align*}
\]

where \( x \sim \mathcal{U}(0, 2) \) and \( \varepsilon \sim \mathcal{N}(0, 1) \).
Example: Artificial data
Example: Artificial data
Example: Artificial data
Example: Artificial data

R> fm <- mob(y ~ x | a + b + e + f + g + h, data = dat1)

Fluctuation tests of splitting variables:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>stat.</td>
<td>2.310366e+01</td>
<td>10.0350125</td>
<td>7.8502106</td>
<td>1.609714</td>
<td>3.8000510</td>
<td>2.7036527</td>
</tr>
<tr>
<td>p value</td>
<td>3.576589e-04</td>
<td>0.1142662</td>
<td>0.2584384</td>
<td>1.000000</td>
<td>0.4337418</td>
<td>0.6085756</td>
</tr>
</tbody>
</table>

Best splitting variable: a
Perform split? yes

Node properties:
) a <= 1.106652; criterion = 1, statistic = 23.104
) a > 1.106652

R> plot(fm)
Example: Artificial data

Node 2 (n = 112)

\[ a \]

\[ p < 0.001 \]

\[ \leq 1.107 \]
\[ > 1.107 \]

Node 4 (n = 47)

\[ b \]

\[ p = 0.003 \]

\[ \leq 0.999 \]
\[ > 0.999 \]

Node 5 (n = 41)
Example: Artificial data

Artificial data from a segmented quadratic regression. The segmentation is explained by 2 categorical and 1 numerical variables, plus 4 additional “noise” variables.

The data-generating mechanism is:

\[
\begin{align*}
& a = a_1, b = b_2 \quad : \quad y = 0 + 4 \cdot x + 0 \cdot x^2 + \varepsilon, \\
& a = a_1, b \neq b_2 \quad : \quad y = 2 + 1 \cdot x + 1 \cdot x^2 + \varepsilon, \\
& a \neq a_1, d \leq 1 \quad : \quad y = 1 + 3 \cdot x + 0 \cdot x^2 + \varepsilon, \\
& a \neq a_1, d > 1 \quad : \quad y = 1.5 + 0 \cdot x + 1.5 \cdot x^2 + \varepsilon,
\end{align*}
\]

where \( x \sim \mathcal{U}(0, 2) \) and \( \varepsilon \sim \mathcal{N}(0, 0.5) \).
Example: Artificial data
Example: Artificial data
Example: Artificial data
Example: Artificial data

Node 3 (n = 21)

Node 4 (n = 41)

Node 6 (n = 72)

Node 7 (n = 66)
Example: Boston housing data

**Goal:** Explain median value of houses in suburbs of Boston by various numerical covariates.

**Here:** Segment a linear regression with explanatory variables \( \log(\text{average number of rooms}) \) and \( \log(\text{lower status percentage}) \). All remaining variables are used as partitioning variables.
Example: Boston housing data
Example: Boston housing data
Example: Boston housing data
Example: Boston housing data
Model-based recursive partitioning:

- based on well-established statistical models,
- aims at minimizing a clearly defined objective function (and not certain heuristics),
- unbiased due to separation of variable and cutpoint selection,
- statistically motivated stopping criterion,
- employs general class of tests for parameter instability.