Model-Based Recursive Partitioning: Ideas, Theory, and Implementation

Achim Zeileis, Florian Wickelmaier, Julia Kopf, Carolin Strobl

http://statmath.wu.ac.at/~zeileis/
Overview

- Motivation: Trees and leaves
- Theory
  - Model estimation
  - Tests for parameter instability
  - Segmentation
  - Pruning
- Applications
  - Linear regression: Costly journals
  - Bradley-Terry model: A model model
  - Rasch model: Differential item functioning
- Implementation
  - Generic object-oriented infrastructure
  - Model plugins
  - Methods
Motivation: Trees

Breiman (2001, *Statistical Science*) distinguishes two cultures of statistical modeling:

- **Data models**
  - Stochastic models, typically parametric.
  - Predominant modeling strategy in the social sciences.
  - Regression models are workhorse for empirical analyses.

- **Algorithmic models**
  - Flexible models, data-generating process unknown.
  - Example: Regression trees model dependent variable $Y$ by “learning” a partition w.r.t explanatory variables $Z_1, \ldots, Z_l$.
  - Few applications in the social sciences.
Motivation: Leaves

**Examples for trees**: CART and C4.5 in statistical and machine learning, respectively.

**Key features**: Predictive power in nonlinear regression relationships, and interpretability (enhanced by visualization), i.e., no “black box”.

**Typically**: Simple models for univariate $Y$, e.g., mean or proportion.

**Idea**: More complex models for multivariate $Y$, e.g., multivariate normal model, regression models, etc.

**Here**: Synthesis of parametric data models and algorithmic tree models.
Recursive partitioning

**Base algorithm:**

1. Fit model for $Y$.
2. Assess association of $Y$ and each $Z_j$.
3. Split sample along the $Z_j^*$ with strongest association: Choose breakpoint with highest improvement of the model fit.
4. Repeat steps 1–3 recursively in the subsamples until some stopping criterion is met.

**Here:** Segmentation (3) of parametric models (1) with additive objective function using parameter instability tests (2) and associated statistical significance (4).
1. Model estimation

Models: $\mathcal{M}(Y, \theta)$ with (potentially) multivariate observations $Y \in \mathcal{Y}$ and $k$-dimensional parameter vector $\theta \in \Theta$.

Parameter estimation: $\hat{\theta}$ by optimization of objective function $\Psi(Y, \theta)$ for $n$ observations $Y_i$ ($i = 1, \ldots, n$):

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

Special cases: Maximum likelihood (ML), weighted and ordinary least squares (OLS and WLS), quasi-ML, and other M-estimators.

Central limit theorem: If there is a true parameter $\theta_0$ and given certain weak regularity conditions, $\hat{\theta}$ is asymptotically normal with mean $\theta_0$ and sandwich-type covariance.
2. Tests for parameter instability

**Estimating function:** Model deviations can be captured by

\[
\psi(y_i, \hat{\theta}) = \frac{\partial \psi(Y, \theta)}{\partial \theta} \bigg|_{y_i, \hat{\theta}}
\]

also known as *score function* or contributions to the *gradient*.

**Fluctuation processes:** Systematic changes in parameters over the variables \( Z = (Z_1, \ldots, Z_l) \) can be assessed by cumulative sums of the empirical estimating functions.

**Fluctuation tests:** Aggregate process to test statistics.

- Andrews’ supLM test for numerical \( Z_j \),
- \( \chi^2 \)-type test for categorical \( Z_j \).
2. Tests for parameter instability
3. Segmentation

**Goal:** Split model into \( b = 1, \ldots, B \) segments along the partitioning variable \( Z_j \) associated with the highest parameter instability. Local optimization of

\[
\sum_{b} \sum_{i \in l_b} \psi(Y_i, \theta_b).
\]

**Here:** \( B = 2 \), binary partitioning.
4. Pruning

**Pruning:** Avoid overfitting.

**Pre-pruning:** Internal stopping criterion. Stop splitting when there is no significant parameter instability.

**Post-pruning:** Grow large tree and prune splits that do not improve the model fit (e.g., via crossvalidation or information criteria).

**Here:** Pre-pruning based on Bonferroni-corrected $p$ values of the fluctuation tests.
Costly journals

**Task:** Price elasticity of demand for economics journals.


**Model:** Linear regression via OLS.

- Demand: Number of US library subscriptions.
- Price: Average price per citation.
- Log-log-specification: Demand explained by price.
- Further variables without obvious relationship: Age (in years), number of characters per page, society (factor).
Costly journals

Node 2 (n = 53)

Node 3 (n = 127)

log(subscriptions) vs log(price/citation) for node 1 with age as the predictor, p < 0.001.

- Node 2 for age ≤ 18 with 18 data points.
- Node 3 for age > 18 with 127 data points.
### Costly journals

#### Recursive partitioning:

<table>
<thead>
<tr>
<th>Regressors</th>
<th>Partitioning variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Const.)</td>
<td>Price</td>
</tr>
<tr>
<td>log(Pr./Cit.)</td>
<td>3.280</td>
</tr>
<tr>
<td></td>
<td>0.660</td>
</tr>
<tr>
<td></td>
<td>0.998</td>
</tr>
<tr>
<td></td>
<td>0.999</td>
</tr>
</tbody>
</table>

(Wald tests for regressors, parameter instability tests for partitioning variables.)
Implementation

Software provided in R system for statistical computing and graphics. Available under General Public License from Comprehensive R Archive Network at http://CRAN.R-project.org/:

- Trees/recursive partitioning: party.
- Structural change inference: strucchange.
- Bradley-Terry/Rasch trees: psychotree.
Implementation

Generic infrastructure:

- Model-based recursive partitioning in function `mob()` from package `party`.
- Takes care of all steps except model fitting (i.e., estimation of $\theta$ by minimization $\Psi$).
- Leverages inference methods from package `strucchange`.
- Models can be plugged in using object-oriented approach: Model objects and methods.
- Visualization can be customized by panel functions (in `grid`).
Implementation

Model plugins:

- Model fitting function (à la `lm()` or – even better – `lm.fit()`) that returns a classed object (like “lm”).
- Required methods: `estfun()`, `weights()`, `reweight()` (at least for 0/1 weights), and extractor for objective function (e.g., `deviance()` or `logLik()`).
- Optional methods (reused if available): `print()`, `predict()`, `coef()`, `summary()`, `residuals()`.
- Additional glue: S4 “StatModel” objects (modeltools package). Separate data handling (in particular, formula processing) from model fitting. (Hopefully facilitated in future versions.)
Implementation

Examples:

- `bttree()`: Interface for `mob()` with `btReg.fit()`.
- `raschtree()`: Interface for `mob()` with `RaschModel.fit()`.
Summary

Model-based recursive partitioning:

- Synthesis of classical parametric data models and algorithmic tree models.
- Based on modern class of parameter instability tests.
- Aims to minimize clearly defined objective function by greedy forward search.
- Can be applied general class of parametric models.
- Alternative to traditional means of model specification, especially for variables with unknown association.
- Object-oriented implementation freely available: Extension for new models requires some coding but not too extensive if interfaced model is well designed.

