Model-Based Recursive Partitioning

Achim Zeileis

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

- Motivation: Trees and leaves
- Methodology
  - Model estimation
  - Tests for parameter instability
  - Segmentation
  - Pruning
- Applications
  - Costly journals
  - Beautiful professors
  - Choosey students
- Software
Motivation: Trees


- **Data models**: Stochastic models, typically parametric.
- **Algorithmic models**: Flexible models, data-generating process unknown.

**Example**: Recursive partitioning models dependent variable $Y$ by “learning” a partition w.r.t explanatory variables $Z_1, \ldots, Z_l$.

**Key features**:

- Predictive power in nonlinear regression relationships.
- Interpretability (enhanced by visualization), i.e., no “black box” methods.
Motivation: Leaves

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

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

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.

Goal: Fitting local models by partitioning of the sample space.
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 sub-samples 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.
1. Model estimation

**Estimating function:** $\hat{\theta}$ can also be defined in terms of

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

where $\psi(Y, \theta) = \partial \Psi(Y, \theta)/\partial \theta$.

**Idea:** In many situations, a single global model $\mathcal{M}(Y, \theta)$ that fits all $n$ observations cannot be found. But it might be possible to find a partition w.r.t. the variables $Z = (Z_1, \ldots, Z_l)$ so that a well-fitting model can be found locally in each cell of the partition.

**Tool:** Assess parameter instability w.r.t. to partitioning variables $Z_j \in \mathcal{Z}_j (j = 1, \ldots, l)$. 
2. Tests for parameter instability

Generalized M-fluctuation tests capture instabilities in $\hat{\theta}$ for an ordering w.r.t $Z_j$.

**Basis:** Empirical fluctuation process of cumulative deviations w.r.t. to an ordering $\sigma(Z_{ij})$.

$$W_j(t, \hat{\theta}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{\lfloor nt \rfloor} \psi(Y_{\sigma(Z_{ij})}, \hat{\theta}) \quad (0 \leq t \leq 1)$$

**Functional central limit theorem:** Under parameter stability $W_j(\cdot) \xrightarrow{d} W^0(\cdot)$, where $W^0$ is a $k$-dimensional Brownian bridge.
2. Tests for parameter instability

Test statistics: Scalar functional $\lambda(W_j)$ that captures deviations from zero.

Null distribution: Asymptotic distribution of $\lambda(W^0)$.

Special cases: Class of test encompasses many well-known tests for different classes of models. Certain functionals $\lambda$ are particularly intuitive for numeric and categorical $Z_j$, respectively.

Advantage: Model $\mathcal{M}(Y, \hat{\theta})$ just has to be estimated once. Empirical estimating functions $\psi(Y_i, \hat{\theta})$ just have to be re-ordered and aggregated for each $Z_j$. 
2. Tests for parameter instability

Splitting numeric variables: Assess instability using sup$LM$ statistics.

$$\lambda_{\text{sup}LM}(W_j) = \max_{i=i_0,...,i_{\bar{i}}} \left( \frac{i}{n} \cdot \frac{n-i}{n} \right)^{-1} \left\| W_j \left( \frac{i}{n} \right) \right\|_2^2.$$  

**Interpretation:** Maximization of single shift $LM$ statistics for all conceivable breakpoints in $[i, \bar{i}]$.

**Limiting distribution:** Supremum of a squared, $k$-dimensional tied-down Bessel process.
2. Tests for parameter instability

Splitting categorical variables: Assess instability using $\chi^2$ statistics.

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

Feature: Invariant for re-ordering of the $C$ categories and the observations within each category.

Interpretation: Captures instability for split-up into $C$ categories.

Limiting distribution: $\chi^2$ with $k \cdot (C - 1)$ degrees of freedom.
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

$$\sum_b \sum_{i \in I_b} \psi(Y_i, \theta_b).$$

$B = 2$: Exhaustive search of order $O(n)$.

$B > 2$: Exhaustive search is of order $O(n^{B-1})$, but can be replaced by dynamic programming of order $O(n^2)$. Different methods (e.g., information criteria) can choose $B$ adaptively.

**Here:** 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 cross-validation or information criteria).

**Here**: Pre-pruning based on Bonferroni-corrected $p$ values of the fluctuation tests.
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-logSpecification: Demand explained by price.
- Further variables without obvious relationship: Age (in years), number of characters per page, society (factor).
Costly journals

\[ \text{age} \quad p < 0.001 \]

\[ \leq 18 \quad > 18 \]

Node 2 (n = 53)

Node 3 (n = 127)
Costly journals

Recursive partitioning:

<table>
<thead>
<tr>
<th></th>
<th>Regressors</th>
<th>Partitioning variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Const.) log(Pr./Cit.)</td>
<td>Price</td>
</tr>
<tr>
<td>1</td>
<td>4.766 -0.533</td>
<td>3.280</td>
</tr>
<tr>
<td></td>
<td>&lt; 0.001 &lt; 0.001</td>
<td>0.660</td>
</tr>
<tr>
<td>2</td>
<td>4.353 -0.605</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>&lt; 0.001 &lt; 0.001</td>
<td>0.998</td>
</tr>
<tr>
<td>3</td>
<td>5.011 -0.403</td>
<td>0.608</td>
</tr>
<tr>
<td></td>
<td>&lt; 0.001 &lt; 0.001</td>
<td>0.999</td>
</tr>
</tbody>
</table>

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

**Task:** Correlation of beauty and teaching evaluations for professors.


**Model:** Linear regression via WLS.

- Response: Average teaching evaluation per course (on scale 1–5).
- Explanatory variables: Standardized measure of beauty and factors gender, minority, tenure, etc.
- Weights: Number of students per course.
Beautiful professors

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>Men</th>
<th>Women</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Constant)</td>
<td>4.216</td>
<td>4.101</td>
<td>4.027</td>
</tr>
<tr>
<td>Beauty</td>
<td>0.283</td>
<td>0.383</td>
<td>0.133</td>
</tr>
<tr>
<td>Gender (= w)</td>
<td>−0.213</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minority</td>
<td>−0.327</td>
<td>−0.014</td>
<td>−0.279</td>
</tr>
<tr>
<td>Native speaker</td>
<td>−0.217</td>
<td>−0.388</td>
<td>−0.288</td>
</tr>
<tr>
<td>Tenure track</td>
<td>−0.132</td>
<td>−0.053</td>
<td>−0.064</td>
</tr>
<tr>
<td>Lower division</td>
<td>−0.050</td>
<td>0.004</td>
<td>−0.244</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.271</td>
<td></td>
<td>0.316</td>
</tr>
</tbody>
</table>

(Remark: Only courses with more than a single credit point.)
Beautiful professors

Hamermesh & Parker:

- Model with all factors (main effects).
- Improvement for separate models by gender.
- No association with age (linear or quadratic).

Here:

- Model for evaluation explained by beauty.
- Other variables as partitioning variables.
- Adaptive incorporation of correlations and interactions.
Beautiful professors

- **Node 1**: Gender
  - Male $p < 0.001$
  - Female

- **Node 2**: Age
  - $p = 0.008$
  - $\leq 50$
  - $> 50$

- **Node 3 (n = 113)**

- **Node 4 (n = 137)**

- **Node 5**: Age
  - $p = 0.014$
  - $\leq 40$
  - $> 40$

- **Node 6 (n = 69)**

- **Node 7**: Division
  - $p = 0.019$
  - Upper
  - Lower

- **Node 8 (n = 81)**

- **Node 9 (n = 36)**
Beautiful professors

Recursive partitioning:

<table>
<thead>
<tr>
<th></th>
<th>(Const.)</th>
<th>Beauty</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3.997</td>
<td>0.129</td>
</tr>
<tr>
<td>4</td>
<td>4.086</td>
<td>0.503</td>
</tr>
<tr>
<td>6</td>
<td>4.014</td>
<td>0.122</td>
</tr>
<tr>
<td>8</td>
<td>3.775</td>
<td>−0.198</td>
</tr>
<tr>
<td>9</td>
<td>3.590</td>
<td>0.403</td>
</tr>
</tbody>
</table>

Model comparison:

<table>
<thead>
<tr>
<th>Model</th>
<th>$R^2$</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>full sample</td>
<td>0.271</td>
<td>7</td>
</tr>
<tr>
<td>nested by gender</td>
<td>0.316</td>
<td>12</td>
</tr>
<tr>
<td>recursively partitioned</td>
<td>0.382</td>
<td>10 + 4</td>
</tr>
</tbody>
</table>
Beautiful professors

Single credit courses:

- Different type of courses: Yoga, aerobic, etc.
- Associated with second strongest instability (after gender).
- Sub-samples too small for separated models: 18 (m), 9 (f).
Choosy students

Task: Choice of university in student exchange programmes.


Model: Paired comparison via Bradley-Terry(-Luce).

- Ranking of six European management schools: London (LSE), Paris (HEC), Milano (Luigi Bocconi), St. Gallen (HSG), Barcelona (ESADE), Stockholm (HHS).
- Interviews with about 300 students from WU Wien.
- Additional information: Gender, studies, foreign language skills.
Choosy students

1. Italian
   - Good: p < 0.001
   - Poor

2. Spanish
   - Good: p = 0.011
   - Poor

5. French
   - Good: p < 0.001
   - Poor

6. Study
   - Good
   - Other: p = 0.01

Node 3 (n = 8)

Node 4 (n = 40)

Node 7 (n = 60)

Node 8 (n = 104)

Node 9 (n = 89)
## Choosy students

### Recursive partitioning:

<table>
<thead>
<tr>
<th></th>
<th>London</th>
<th>Paris</th>
<th>Milano</th>
<th>St. Gallen</th>
<th>Barcelona</th>
<th>Stockholm</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.22</td>
<td>0.13</td>
<td>0.16</td>
<td>0.07</td>
<td>0.41</td>
<td>0.01</td>
</tr>
<tr>
<td>4</td>
<td>0.42</td>
<td>0.09</td>
<td>0.35</td>
<td>0.05</td>
<td>0.06</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>0.33</td>
<td>0.42</td>
<td>0.06</td>
<td>0.07</td>
<td>0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>8</td>
<td>0.39</td>
<td>0.23</td>
<td>0.09</td>
<td>0.14</td>
<td>0.09</td>
<td>0.06</td>
</tr>
<tr>
<td>9</td>
<td>0.39</td>
<td>0.10</td>
<td>0.08</td>
<td>0.16</td>
<td>0.17</td>
<td>0.10</td>
</tr>
</tbody>
</table>

(Standardized ranking from Bradley-Terry model.)
**Software**

**Implementation:** In R system for statistical computing.

- Object-oriented implementation of model-based recursive partitioning in function `mob()` from package `party`.
- Underlying inference methods in package `strucchange`.
- Convenient interfaces for linear regression (`lm.fit()`), generalized linear models (`glm.fit()`), and survival regression (`survreg()`) are readily available.
- Currently: Hand-crafted code for Bradley-Terry model (interfacing `glm.fit()`), not in package.
Software

Extension requirements:

- S4 “StatModel” objects (*modeltools* package): Separate data handling (in particular, formula processing) from model fitting.
- Fitted models must provide methods: `estfun()`, `weights()`, `reweight()` (at least for 0/1 weights), and extractor for objective function (default: `deviance()`).
- Further methods are re-used (if available): `print()`, `predict()`, `coef()`, `summary()`, `residuals()`, `logLik()`.

**Easy if:** Model already available in R with

- Fitted model class with all the usual extractor functions.
- Access to empirical estimating functions (`estfun()` method).
- In addition to formula interface (à la `lm()`): Fitting function (à la `lm.fit()`) that returns sufficiently post-processed output.
Software

Caveats:

- For visualization: Panel-generating function for grid graphics.
- \texttt{mob()} interprets \texttt{weights} as case weights (and expects the “StatModel” objects to do the same).
- Non-standard formula processing for multivariate responses.
- Hopefully: New model/formula interface soon on R-Forge.

Example: Simple implementation of basic Bradley-Terry model.

- Interfaces: \texttt{btl()} and \texttt{btl.fit()} plus methods.
- Workhorse: Set up design matrix, call \texttt{glm.fit()} with \texttt{family = binomial()}, suitably aggregate results.
- Glue code: S4 “BTL” object with few additional methods.
Implementation of simple Bradley-Terry models

Artificial data:

R> pc <- rbind(
+     c(1, 1, 1), # a > b > c
+     c(1, 1, 0), # a > c > b
+     c(1, 0, 0), # c > a > b
+     c(1, 1, 1)  # a > b > c
+   )
R> colnames(pc) <- c("ab", "ac", "bc")

Question: Proper data structures for paired comparison data?

Ideally: pc should be treated like a vector of length 4 (# subjects) with suitable meta-data that reflects # objects, labels, printing, etc.
Implementation of simple Bradley-Terry models

**Interfaces:** Formula interface and workhorse fitting function.

```r
R> btl(pc ~ 1)
Bradley-Terry-Luce model

Coefficients:
   a   b
1.7542 -0.4158

Standardized latent ranking:
   a   b   c
0.7769 0.0887 0.1344

R> pc_btl <- btl.fit(pc)
R> class(pc_btl)
[1] "btl"
```
Implementation of simple Bradley-Terry models

R> coef(pc_btl)
  a   b
1.7541765 -0.4158042

R> coef(pc_btl, log = FALSE)
  a   b   c
0.77686224 0.08870199 0.13443577

R> estfun(pc_btl)
    a   b
[1,] 0.2500030 0.4999987
[2,] 0.2500030 -0.5000014
[3,] -0.7500083 -0.5000014
[4,] 0.2500030 0.4999987

R> deviance(pc_btl)
[1] 11.36700

R> logLik(pc_btl)
'log Lik.' -5.683498 (df=2)
Implementation of simple Bradley-Terry models

R> btl.fit(y = pc, weights = c(1, 1, 1, 0))

Bradley-Terry-Luce model

Coefficients:
   a    b
  1.145 -1.145

Standardized latent ranking:
   a    b    c
0.70450 0.07133 0.22417
Implementation of simple Bradley-Terry models

**Interface:** “StatModel” glue code.

```r
R> class(BTL)
[1] "StatModel"
attr(.("package"))
[1] "modeltools"

R> mf <- ModelEnvFormula(ab + ac + bc ~ 1,
+    data = as.data.frame(pc))
R> pc_BTL <- fit(BTL, mf)
R> class(pc_BTL)
[1] "BTL" "btl"

R> pc_BTL

BTL coefficients:
   a      b
1.7542 -0.4158
```
Implementation of simple Bradley-Terry models

R> pc_BTL2 <- reweight(pc_BTL, weights = c(1, 1, 1, 0))
R> weights(pc_BTL2)
[1] 1 1 1 0
R> coef(pc_BTL2)

          a          b
1.145071 -1.145071

R> estfun(pc_BTL2)

          a          b
[1,] 0.3333340  0.6666672
[2,] 0.3333340 -0.3333340
[3,] -0.6666672 -0.3333340
[4,] 0.0000000  0.0000000
Implementation of simple Bradley-Terry models

```r
R> load("cems.rda")
R> cems <- cems[!apply(sapply(cems[,1:15], is.na), 1, all), ]
R> cems_mob <- mob(ab + ac + ad + ae + af + bc + bd + be +
+   bf + cd + ce + cf + de + df + ef ~ 1 | study + english +
+   french + spanish + italian + work + gender + intdegree,
+   data = cems, model = BTL, na.action = na.pass,
+   control = mob_control(minsplit = 5))
R> plot(cems_mob, terminal_panel = node_btlplot,
+   tnex = 2, tp_args = list(yscale = c(0, 0.5),
+   names = c("Lo", "Pa", "Mi", "SG", "Ba", "St")))
R> coef(cems_mob)
```

```
```
```r
  a   b   c   d   e
3 3 2.915557 2.37530103 2.6132469 1.7116796 3.5496433
4 4 2.657933 1.06913836 2.4611060 0.6068164 0.7413380
7 7 2.174962 2.42810393 0.4400174 0.5704167 0.9507593
8 8 1.794987 1.25646206 0.3024828 0.7576933 0.3729114
9 9 1.394938 0.03678015 -0.2427147 0.5071161 0.5702408
```
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 is limited if interfaced model is well designed.