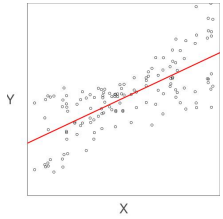
universität
innsbruck

The Power of Unbiased Recursive Partitioning: A Unifying View of CTree, MOB, and GUIDE

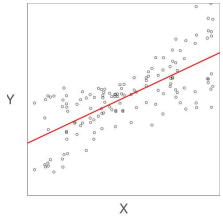
Lisa Schlosser, Torsten Hothorn, Achim Zeileis

http://www.partykit.org/partykit

Motivation

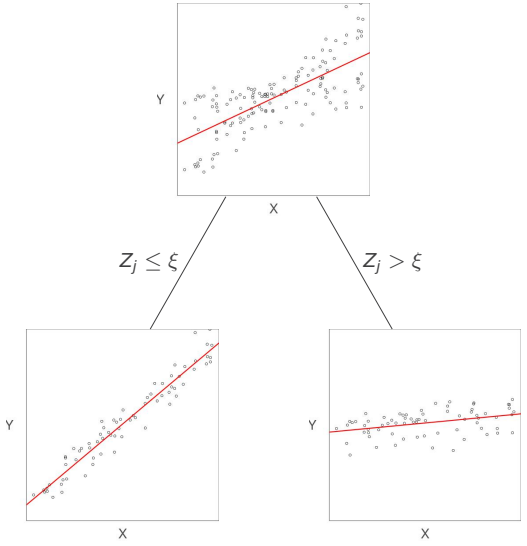


Motivation

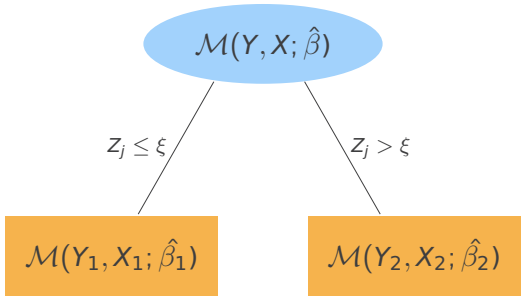


Other covariates $Z_1, \ldots, Z_p$?

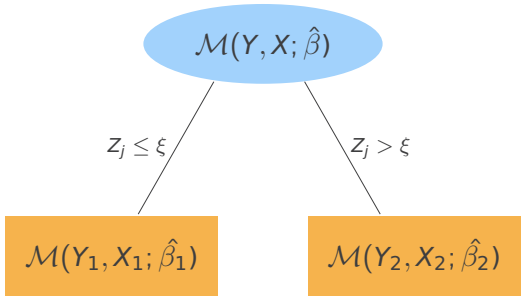
Motivation



Motivation


$\mathcal{M}(Y, X; \hat{\beta})$
$z_j \leq \xi$
$z_j > \xi$
$\mathcal{M}(Y_1, X_1; \hat{\beta}_1)$
$\mathcal{M}(Y_2, X_2; \hat{\beta}_2)$

Motivation



$\mathcal{M}$ can also be a more general model (possibly without X).

Unbiased recursive partitioning

GUIDE: Loh (2002, *Statistica Sinica*).

- First *unbiased* algorithm for recursive partitioning of linear models.
- Separation of split variable and split point selection.
- Based on χ^2 tests.

CTree: Hothorn, Hornik, Zeileis (2006, *JCGS*).

- Proposed as unbiased recursive partitioning for nonparametric modeling.
- Based on conditional inference (or permutation tests).
- Can be model-based via model scores as the response transformation.

MOB: Zeileis, Hothorn, Hornik (2008, *JCGS*).

- Model-based recursive partitioning using M-estimation (ML, OLS, CRPS, . . .).
- Based on parameter instability tests.
- Adapted to various psychometric models: Rasch, PCM, Bradley-Terry, MPT, SEM, networks, ...

Unbiased recursive partitioning

Basic tree algorithm:

1. Fit a model $\mathcal{M}(Y, X; \hat{\beta})$ to the response Y and possible covariates X.
2. Assess association of $\mathcal{M}(Y, X; \hat{\beta})$ and each possible split variable Z_j and select the split variable Z_{j^*} showing the strongest association.
3. Choose the corresponding split point leading to the highest improvement of model fit and split the data.
4. Repeat steps 1–3 recursively in each of the resulting subgroups until some stopping criterion is met.

Here: Focus on split variable selection (step 2).

Split variable selection

General testing strategy:

1. Evaluate a discrepancy measure capturing the observation-wise goodness of fit of $\mathcal{M}(Y, X; \hat{\beta})$.
2. Apply a statistical test assessing dependency of the discrepancy measure to each possible split variable Z_j.
3. Select the split variable Z_j^* showing the smallest p-value.

Discrepancy measures: (Model-based) transformations of Y (and X, if any), possibly for each model parameter.

- (Ranks of) Y.
- (Absolute) deviations $Y - \bar{Y}$.
- Residuals of $\mathcal{M}(Y, X; \hat{\beta})$.
- Score matrix of $\mathcal{M}(Y, X; \hat{\beta})$.
- ...

Discrepancy measures

Example: Simple linear regression $\mathcal{M}(Y, X; \beta_0, \beta_1)$, fitted via ordinary least squares (OLS).

Residuals:

$$r(Y, X, \hat{\beta}_0, \hat{\beta}_1) = Y - \hat{\beta}_0 - \hat{\beta}_1 \cdot X$$

Discrepancy measures

Example: Simple linear regression $\mathcal{M}(Y, X; \beta_0, \beta_1)$, fitted via ordinary least squares (OLS).

Residuals:

$$r(Y, X, \hat{\beta}_0, \hat{\beta}_1) = Y - \hat{\beta}_0 - \hat{\beta}_1 \cdot X$$

Model scores: Based on log-likelihood or residual sum of squares.

$$s(Y, X, \hat{\beta}_0, \hat{\beta}_1) = \left(\frac{\partial r^2(Y, X, \hat{\beta}_0, \hat{\beta}_1)}{\partial \beta_0} \quad , \quad \frac{\partial r^2(Y, X, \hat{\beta}_0, \hat{\beta}_1)}{\partial \beta_1} \right)$$

Discrepancy measures

Example: Simple linear regression $\mathcal{M}(Y, X; \beta_0, \beta_1)$, fitted via ordinary least squares (OLS).

Residuals:

$$r(Y, X, \hat{\beta}_0, \hat{\beta}_1) = Y - \hat{\beta}_0 - \hat{\beta}_1 \cdot X$$

Model scores: Based on log-likelihood or residual sum of squares.

$$s(Y, X, \hat{\beta}_0, \hat{\beta}_1) = \left(\frac{\partial r^2(Y, X, \hat{\beta}_0, \hat{\beta}_1)}{\partial \beta_0} \quad , \quad \frac{\partial r^2(Y, X, \hat{\beta}_0, \hat{\beta}_1)}{\partial \beta_1} \right)$$

$$\Downarrow \qquad\qquad \Downarrow$$

$$-2 \cdot r(Y, X, \hat{\beta}_0, \hat{\beta}_1) \qquad -2 \cdot r(Y, X, \hat{\beta}_0, \hat{\beta}_1) \cdot X$$

A unifying view

Algorithms: CTree, MOB, GUIDE are all ‘flavors’ of the general framework.

Building blocks: For standard setup.

	Scores	Binarization	Categorization	Statistic
CTree	Model scores	–	–	Sum of squares
MOB	Model scores	–	–	Maximally selected
GUIDE	Residuals	✓	✓	Sum of squares

Remarks:

- All three algorithms allow for certain modifications of standard setup.
- Further differences, e.g., null distribution, pruning strategy, etc.

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

$$s(Y, X, \hat{\beta}_0, \hat{\beta}_1) = -2 \cdot \begin{pmatrix} r(Y_1, X_1, \hat{\beta}_0, \hat{\beta}_1) & r(Y_1, X_1, \hat{\beta}_0, \hat{\beta}_1) \cdot X_1 \\ r(Y_2, X_2, \hat{\beta}_0, \hat{\beta}_1) & r(Y_2, X_2, \hat{\beta}_0, \hat{\beta}_1) \cdot X_2 \\ \vdots & \vdots \\ r(Y_n, X_n, \hat{\beta}_0, \hat{\beta}_1) & r(Y_n, X_n, \hat{\beta}_0, \hat{\beta}_1) \cdot X_n \end{pmatrix}$$

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

$$s(Y, X, \hat{\beta}_0, \hat{\beta}_1) = -2 \cdot \begin{pmatrix} r(Y_1, X_1, \hat{\beta}_0, \hat{\beta}_1) & r(Y_1, X_1, \hat{\beta}_0, \hat{\beta}_1) \cdot X_1 \\ r(Y_2, X_2, \hat{\beta}_0, \hat{\beta}_1) & r(Y_2, X_2, \hat{\beta}_0, \hat{\beta}_1) \cdot X_2 \\ \vdots & \vdots \\ r(Y_n, X_n, \hat{\beta}_0, \hat{\beta}_1) & r(Y_n, X_n, \hat{\beta}_0, \hat{\beta}_1) \cdot X_n \end{pmatrix}$$

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

$$r(Y, X, \hat{\beta}_0, \hat{\beta}_1) = \begin{pmatrix} r(Y_1, X_1, \hat{\beta}_0, \hat{\beta}_1) \\ r(Y_2, X_2, \hat{\beta}_0, \hat{\beta}_1) \\ \vdots \\ r(Y_n, X_n, \hat{\beta}_0, \hat{\beta}_1) \end{pmatrix}$$

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

$$r(Y, X, \hat{\beta}_0, \hat{\beta}_1) = \begin{pmatrix} r(Y_1, X_1, \hat{\beta}_0, \hat{\beta}_1) \\ r(Y_2, X_2, \hat{\beta}_0, \hat{\beta}_1) \\ \vdots \\ r(Y_n, X_n, \hat{\beta}_0, \hat{\beta}_1) \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} > 0 \\ \leq 0 \\ \vdots \\ > 0 \end{pmatrix}$$

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

$$Z_j = \begin{pmatrix} Z_{j1} \\ Z_{j2} \\ \vdots \\ Z_{jn} \end{pmatrix}$$

General framework

Building blocks:

- Residuals vs. full model scores.
- Binarization of residuals/scores.
- Categorization of possible split variables.

$$Z_j = \begin{pmatrix} Z_{j1} \\ Z_{j2} \\ \vdots \\ Z_{jn} \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \text{Q3} \\ \text{Q1} \\ \vdots \\ \text{Q2} \end{pmatrix}$$

Pruning

Goal: Avoid overfitting.

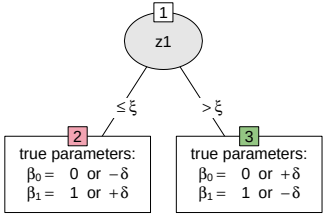
Two strategies:

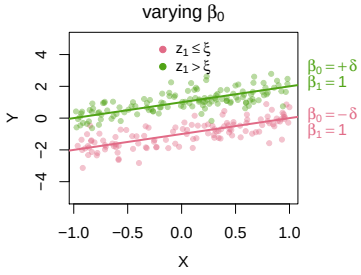
- *Pre-pruning:* Internal stopping criterion based on Bonferroni-corrected *p*-values of the underlying tests. Stop splitting when there is no significant association.
- *Post-pruning:* First grow a very large tree and afterwards prune splits that do not improve the model fit, either via cross-validation (e.g., cost-complexity pruning as in CART) or based on information criteria (e.g., AIC or BIC).

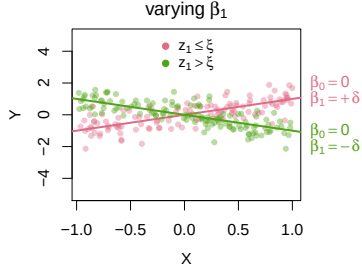
Simulation

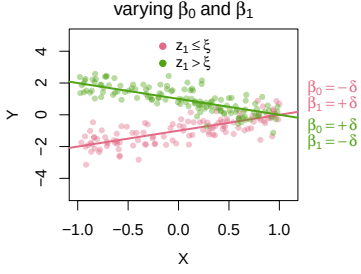
Name	Notation	Specification
Variables:		
Response	Y	$= \beta_0(Z_1) + \beta_1(Z_1) \cdot X + \epsilon$
Regressor	X	$\mathcal{U}([-1, 1])$
Error	ϵ	$\mathcal{N}(0, 1)$
True split variable	Z_1	$\mathcal{U}([-1, 1])$ or $\mathcal{N}(0, 1)$
Noise split variables	$Z_2, Z_3, \ldots, Z_{10}$	$\mathcal{U}([-1, 1])$ or $\mathcal{N}(0, 1)$
Parameters/functions:		
Intercept	β_0	0 or $\pm\delta$
Slope	β_1	1 or $\pm\delta$
True split point	ξ	$\in \{0, 0.2, 0.5, 0.8\}$
Effect size	δ	$\in \{0, 0.1, 0.2, \ldots, 1\}$

Simulation 1: True tree structure

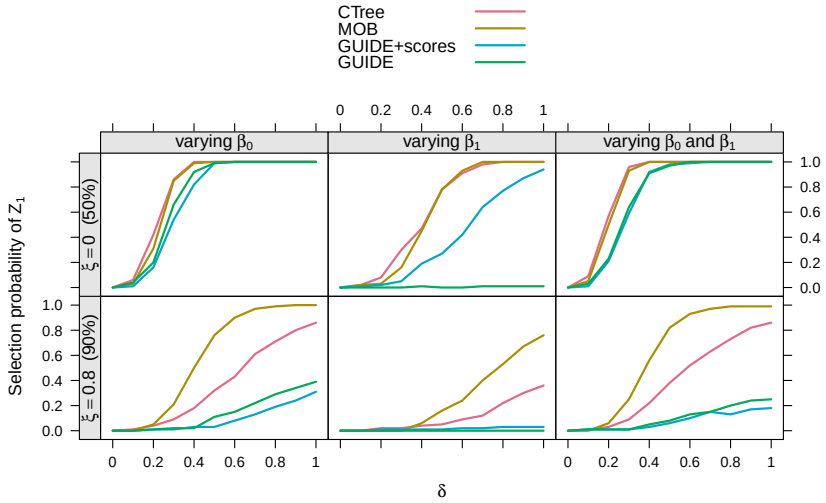




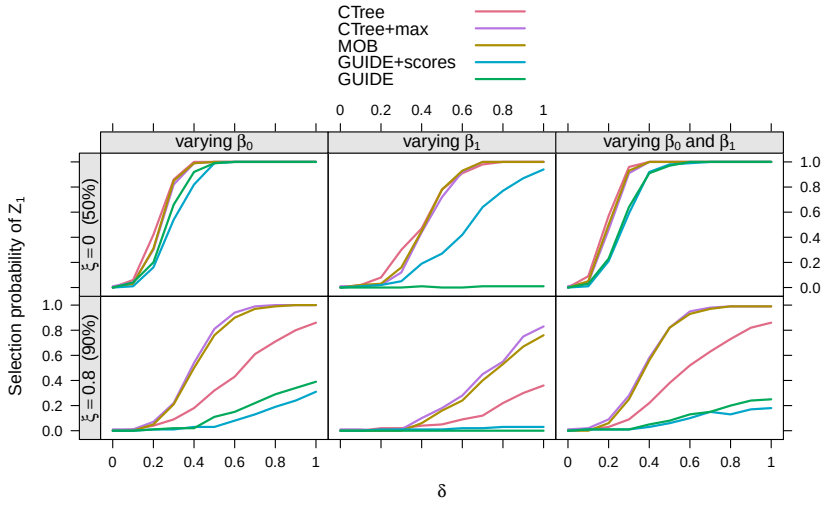




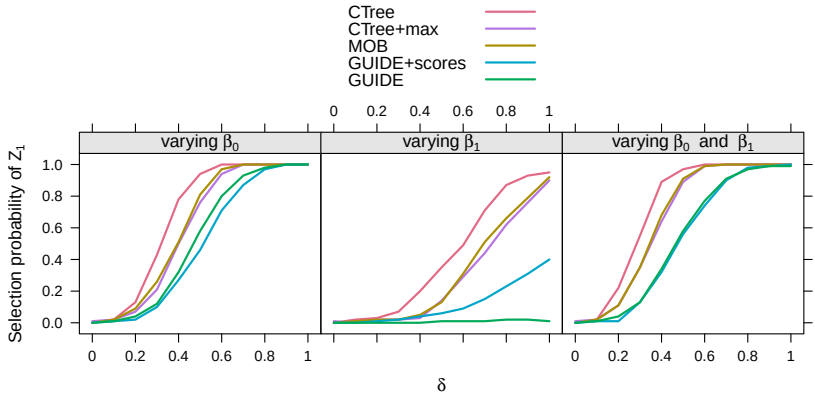
Simulation 1: Residuals vs. full model scores



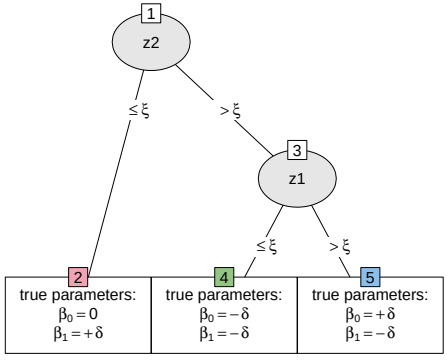
Simulation 1: Maximum vs. linear selection

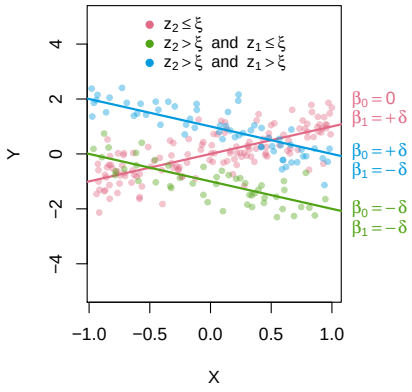


Simulation 1: Continuously changing parameters

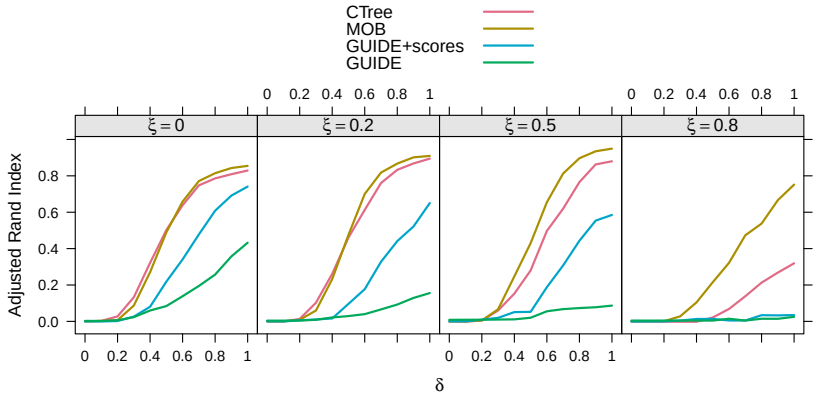


Simulation 2: True tree structure

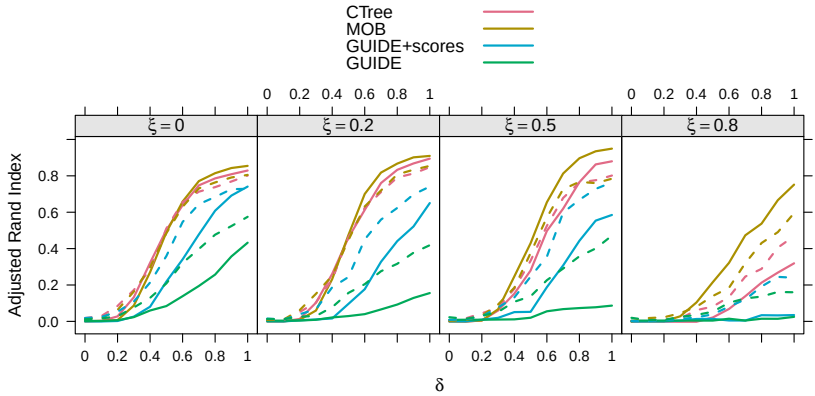




Simulation 2: Residuals vs. full model scores



Simulation 2: Pre-pruning vs. post-pruning



Recommendations

In this setting:

- Full model scores better than residuals only.
- Original values of scores/residuals better than binarized values.
- Categorization is simpler, but less powerful in margins.
- Maximally-selected statistics (as in MOB) more powerful for abrupt shifts.
- Linear statistics (default in CTree) more powerful for linear changes.
- If the significance tests perform well pre-pruning works well, otherwise post-pruning might be needed.

References

Schlosser L, Hothorn T, Zeileis A (2019). “The Power of Unbiased Recursive Partitioning: A Unifying View of CTree, MOB, and GUIDE.” arXiv:1906.10179, *arXiv.org E-Print Archive*. https://arxiv.org/abs/1906.10179.

Loh W-Y (2002). “Regression Trees with Unbiased Variable Selection and Interaction Detection.” *Statistica Sinica*, **12**(2), 361–386. http://www.jstor.org/stable/24306967

Hothorn T, Hornik K, Zeileis A (2006). “Unbiased Recursive Partitioning: A Conditional Inference Framework.” *Journal of Computational and Graphical Statistics*, **15**(3), 651–674. doi:10.1198/106186006X133933

Zeileis A, Hothorn T, Hornik K (2008). “Model-Based Recursive Partitioning.” *Journal of Computational and Graphical Statistics*, **17**(2), 492–514. doi:10.1198/106186008X319331