Thomas Rusch

Recursive Partitioning of Models of a Generalized Linear Model Type

Thesis

DOKTORAT DER SOZIAL- UND WIRTSCHAFTSWISSENSCHAFTEN

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED

1. Beurteilerin/1. Beurteiler: **ao. Univ. Prof. Dr. Reinhold Hatzinger**

2. Beurteilerin/2. Beurteiler: **Univ. Prof. Dr. Kurt Hornik**

Eingereicht am: _____

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED

Titel der Dissertation:

Recursive Partitioning of Models of a Generalized Linear Model Type

Dissertation zur Erlangung des akademischen Grades

einer Doktorin/eines Doktors

der Sozial- und Wirtschaftswissenschaften an der Wirtschaftsuniversität Wien

eingereicht bei

1. Beurteilerin/1. Beurteiler: **ao. Univ. Prof. Dr. Reinhold Hatzinger**

2. Beurteilerin/2. Beurteiler: **Univ. Prof. Dr. Kurt Hornik**

von **MMag. Thomas Rusch, Bakk.**

Fachgebiet: **Statistik**

Wien, im **Mai, 2012**

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

EFMD EQUIS ACCREDITED

Ich versichere:

1. dass ich die Dissertation selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

2. dass ich diese Dissertation bisher weder im In- noch im Ausland (einer Beurteilerin/ einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

3. dass dieses Exemplar mit der beurteilten Arbeit übereinstimmt.

Datum _____ Unterschrift _____

Recursive Partitioning of Models of a Generalized Linear Model Type

Thomas Rusch
Matr.Nr.:0005783
Institute for Statistics and Mathematics, WU
thomas.rusch@wu.ac.at

Supervisor 1: Reinhold Hatzinger
reinhold.hatzinger@wu.ac.at

Supervisor 2: Kurt Hornik
kurt.hornik@wu.ac.at

Für Christof, Claudia, Hubert und Monika

Danksagung

Ich möchte mich zu Beginn bei Reinhold Hatzinger bedanken. Er bot mir die Möglichkeit im Umfeld des Instituts für Statistik und Mathematik der WU eine von ihm betreute Dissertation durchzuführen und darüber hinaus als sein Assistent wertvolle weitere Erfahrung zu sammeln. Sein stets offenes Ohr für meine Anliegen und seine Loyalität haben entscheidend dazu beigetragen dieses Dissertationsprojekt erfolgreich abschliessen zu können. Ganz besonders dankbar bin ich dafür, dass er ständig großes Vertrauen in mich legte, sich wann immer nötig für mich einsetzte und mir ermöglichte meinen eigenen Forschungsinteressen nachzugehen, selbst wenn sie abseits der Seinigen lagen.

Ebenfalls danken möchte ich Kurt Hornik, von dessen wissenschaftlicher Exzellenz zu lernen und zu profitieren, ein Privileg darstellt, das gar nicht überschätzt werden kann. Seine, selbst wenn kritisch, stets konstruktiven Anregungen und Kommentare zu meiner Arbeit führten überhaupt erst zu akzeptablen Endergebnissen und waren immer wieder Ansporn und Motivation. Dass er dabei gleichzeitig eine angenehme Art im persönlichen Umgang pflegt, möchte ich ihm hoch anrechnen.

Weiters bedanken möchte ich mich bei Alois Geyer, Bettina Grün, Paul Hofmarcher, Wolfgang Jank, Ilro Lee, Manfred Lueger, Patrick Mair und Marcus Wurzer sowie dem R Core Development Team, die alle direkt auf die eine oder andere Art das Gelingen dieser Dissertation ermöglichten. Auch allen ungenannten Menschen die indirekt Einfluß ausübten sei gedankt.

Besonderer Dank in dieser Arbeit und darüber hinaus jedoch gebührt Achim Zeileis. Seine Arbeit im Bereich modell-basierten rekursiven Partitionierens und seine R Implementation stellte überhaupt erst den Ausgangspunkt dieser Dissertation dar. Darüber hinaus war er stets unglaublich hilfsbereit und geduldig wenn ich mich mit Fragen oder Problemen an ihn wendete. Er fungierte öfter als nicht als der Hauptansprechpartner bei Problemen jeglicher statistischer Art und sein schier unerschöpflich erscheinendes Wissen aus allen möglichen Bereichen der Statistik, seine beeindruckende wissenschaftliche Kompetenz sowie seine Hilfsbereitschaft machten ihn zu einem Leuchtturm, wann immer ich im Dunkeln wandelte.

Abstract

This thesis is concerned with recursive partitioning of models of a generalized linear model type (GLM-type), i.e., maximum likelihood models with a linear predictor for the linked mean, a topic that has received constant interest over the last twenty years. The resulting tree (a "model tree") can be seen as an extension of classic trees, to allow for a GLM-type model in the partitions. In this work, the focus lies on applied and computational aspects of model trees with GLM-type node models to work out different areas where application of the combination of parametric models and trees will be beneficial and to build a computational scaffold for future application of model trees. In the first part, model trees are defined and some algorithms for fitting model trees with GLM-type node model are reviewed and compared in terms of their properties of tree induction and node model fitting. Additionally, the design of a particularly versatile algorithm, the MOB algorithm [Zeileis *et al.*, 2008] in R is described and an in-depth discussion of how the functionality offered can be extended to various GLM-type models is provided. This is highlighted by an example of using partitioned negative binomial models for investigating the effect of health care incentives. Part II consists of three research articles where model trees are applied to different problems that frequently occur in the social sciences. The first uses trees with GLM-type node models and applies it to a data set of voters, who show a non-monotone relationship between the frequency of attending past elections and the turnout in 2004. Three different type of model tree algorithms are used to investigate this phenomenon and for two the resulting trees can explain the counter-intuitive finding. Here model tress are used to learn a nonlinear relationship between a target model and a big number of candidate variables to provide more insight into a data set. A second application area is also discussed, namely using model trees to detect ill-fitting subsets in the data. The second article uses model trees to model the number of fatalities in Afghanistan war, based on the WikiLeaks Afghanistan war diary. Data pre-processing with a topic model generates predictors that are used as explanatory variables in a model tree for overdispersed count data. Here the combination of model trees and topic models allows to flexibly analyse database data, frequently encountered in data journalism, and provides a coherent description of fatalities in the Afghanistan war. The third paper uses a new framework built around model trees to approach the classic problem of segmentation, frequently encountered in marketing and management science. Here, the framework is used for segmentation of a sample of the US electorate for identifying likely and unlikely voters. It is shown that the framework's model trees enable accurate identification which in turn allows efficient targeted mobilisation of eligible voters.

Zusammenfassung

Diese Arbeit beschäftigt sich mit rekursivem Partitionieren von verallgemeinerten linearen modell-artigen Modellen (GLM-artige), d.h. Maximum-Likelihood-Modelle mit linearem Prediktor für eine Funktion des Erwartungswertes. Der so resultierende "Modellbaum" kann als Erweiterung klassischer Baumverfahren verstanden werden, die es erlaubt GLM-artige Modelle in den Partitionen des Baumes anzupassen. In dieser Arbeit liegt der Fokus auf angewandten und computationalen Aspekten von Modellbäumen mit GLM-artigen Knotenmodellen um sowohl Bereiche abzustecken in denen die Anwendung vorteilhaft sein kann, als auch ein Gerüst zur Verfügung zu stellen, das die Erweiterung von Modellbäumen für zukünftige Anwendungen erleichtern soll. Im ersten Teil werden Modellbäume definiert und verschiedene Algorithmen zum Lernen von Modellbäumen diskutiert sowie einige Eigenschaften der Baumstruktur und der Knotenmodelle verglichen. Zusätzlich wird der besonders vielseitige MOB Algorithmus [Zeileis *et al.*, 2008], der in R implementiert ist, genauer beleuchtet und eine Diskussion der Erweiterung der Funktionalität auf noch unimplementierte GLM-artige Modelle unternommen. Das wird mit einem Beispiel eines partitionierten Modells zur Abschätzung von Effekten einer Zusatzversicherung auf Spitalbesuche illustriert. Teil II besteht aus drei Fachartikeln in denen Modellbäume für verschiedene Anwendungsprobleme aus den Sozialwissenschaften verwendet werden. Der erste Artikel führt Modellbäume mit GLM-artigen Knotenmodellen ein und wendet diese auf einen WählerInnendatensatz an. In diesem zeigt sich ein nicht-monotoner Zusammenhang zwischen der Häufigkeit der Teilnahme an bisherigen Wahlen und der Wahrscheinlichkeit 2004 wählen zu gehen. Drei unterschiedlich geschätzte Modellbäume werden verwendet um diesen Zusammenhang zu untersuchen, von denen zwei eine befriedigende Erklärung liefern können. Entsprechend werden Modellbäume hierbei dazu verwendet einen komplexen, nichtlinearen Zusammenhang eines Modells mit einer großen Zahl an weiteren Variablen zu lernen, um so mehr Einsicht in die Daten zu bekommen. Ein zweiter Anwendungsbereich wird ebenfalls besprochen, die Verwendung von Modellbäumen zur Identifikation von Subgruppen mit hoher Anpassungsgüte eines GLM-artigen Modells. Der zweite Artikel verwendet Modellbäume um basierend auf den WikiLeaks Afghanistan Daten Todesfälle im Afghanistankrieg zu modellieren. Aus Berichten zu Vorfällen werden mit Themenmodellen neue Variablen generiert, die in weiterer Folge als erklärende Variablen in einem Modellbaum für Zähldaten mit großem Streubereich verwendet werden. Hier ermöglicht die Kombination von Modellbäumen und Themenmodellen eine flexible Analyse von Daten aus Datenbanken, wie sie oft im Bereich des Datenjournalismus vorkommen und erlaubt eine koherente Charakterisierung der Umstände von Todesfällen im Afghanistankrieg. Der dritte Artikel schlägt einen auf Modellbäumen basierenden methodischen Rahmen vor um das klassische Problem der Segmentierung, wie es häufig im Marketing vorkommt, zu lösen. Konkret wird hier eine Stichprobe US-amerikanischer Wahlberechtigter segmentiert um Personen zu identifizieren, die eher schon oder eher nicht wählen gehen. Es wird gezeigt, dass der methodische Rahmen relativ akkurate Identifikation erlaubt, was in weiterer Folge effiziente Wahlberechtigtenmobilisierung nach sich ziehen kann.

Contents

Part I.

Introduction and Computational Aspects

1. Introduction

1.1. Introduction to the Thesis

Tree models date back to "Automatic Interaction Detection" [AID; Morgan and Sonquist, 1968] and have been popularised by Breiman *et al.* [1984] and Quinlan [1993] with their famous CART and C4.5 algorithms for regression and classification problems. See Zhang and Singer [2010], Clarke *et al.* [2009] or Hastie *et al.* [2009] for a discussion of the history and construction principles of tree based models. They were invented as a flexible, data-driven, nonlinear and nonparametric alternative to conventional regression or classification models. Their classic idea is to partition the data set based on the predictor variables into maximally homogeneous subsets and to assign the *same single value* to all observations in the partition. Algorithmically, this is usually achieved with a greedy, forward selection procedure, mostly by recursive partitioning ("divide-and-conquer"). With trees the focus has been shifted towards interactions of predictors as the driving force in explaining and predicting responses.

The increased availability of computational power, the flexibility and the simplicity of interpretation of regression and classification trees has arguably lead to growing interest in tree models in recent years. This applies especially to areas where large or unstructured data sets are common, where there is only limited knowledge about the underlying data generating processes, where exploration of the data is equally or more important than classic inference, where robust and flexible methods are sought and where predictive accuracy is particularly important. In short, areas for which the term "data mining" has been established. Hastie *et al.* [2009, p. 352] even go as far to claim that "[...] trees come closest to meeting the requirements for serving as an off-the-shelf procedure for data mining. They are relatively fast to construct and they produce interpretable models (if the trees are small). [...] They naturally incorporate mixtures of numeric and categorical predictor variables and missing values. They are invariant under (strictly monotone) transformations of the individual predictors. As a result, scaling and/or more general transformations are not an issue, and they are immune to the effects of predictor outliers. They perform internal feature selection as an integral part of the procedure. They are thereby resistant, if not completely immune, to the inclusion of many irrelevant predictor variables. These properties of decision trees are largely the reason that they have emerged as the most popular learning method for data mining."

Over roughly the last 20 years there has been a growing body of literature on extending tree algorithms to allow for *more than just a constant single value* in the partitions [e.g., Chaudhuri *et al.*, 1995, Gama, 2004, Quinlan, 1993, Chaudhuri *et al.*, 1994, Loh, 2002, Landwehr *et al.*, 2005, Chan and Loh, 2005, Zeileis *et al.*, 2008, Strobl *et al.*, 2010, 2011, Sela and Simonoff, 2012, Potts and Sammut, 2005]. Along these lines, this thesis considers merging trees with models of a generalized linear model type (GLM-type), i.e., of maximum or quasi-likelihood models with linear predictors for the mean [for a comprehensive treatment of the group of generalized linear models see McCullagh and Nelder, 1989]. Building upon the work done up to this point in this area, most notably Zeileis *et al.* [2008], this dissertation approaches the topic by focusing on the applied side of model trees with GLM-type node models and by shedding more light on

computational aspects. It contains a chapter in which computational aspects of different algorithms (with a focus on the MOB algorithm [Zeileis *et al.*, 2008]) are compared, described and discussed. For the latter, it provides an in-depth description of the MOB implementation in R and on how to extend the current functionality of the `mob` function from **party** [Hothorn *et al.*, 2012a] in R [R Development Core Team, 2012] to allow for arbitrary GLM-type models in the nodes. This can be found in Chapter 2 in Part I.

Regarding applications, this thesis digs deeper into the practical application of this type of extended trees and highlights their versatility for a broad array of statistical problems, such as mixture modelling and segmentation, prediction, goodness-of-fit assessment, and flexible and nonlinear modelling in data sets with many predictors and/or observations. Since this is a cumulative thesis, the applications correspond to three stand-alone original research articles that have recursively partitioned model trees with GLM-type node models as their common backdrop. All applications come from the social sciences and touch on one or more of the following areas: political science, marketing, journalism, armed conflict research and debt management. Each paper uses model trees with GLM-type node models at one point but they all differ in to what end the model trees are used for and which node model they employ. The papers can be found in Part II of the thesis and are briefly summarised below.

Gaining Insight with Recursive Partitioning of Generalized Linear Models In this paper [Rusch and Zeileis, 2013, see Chapter 3] the MOB algorithm of Zeileis *et al.* [2008] for generalized linear models (GLM) and related models (GLM-type) is described against the backdrop of classic GLM theory. Two examples are used to illustrate the applicability of the technique: For the first, model trees with logistic regression models are used as an exploratory technique to find an explanation for a counter-intuitive or surprising result of a global model, namely to identify why in a sample of 19634 people from Ohio, the individual likelihood of turnout in the US 2004 presidential is not monotonically increasing with the percentage of attended elections in the future, but actually bell-shaped. The model tree approach detects manifest subgroups that indeed display a monotonically increasing relationship while other groups show a monotonically decreasing relationship. This leads to the surprising functional form of the global model for the whole data set. Additionally, the model tree approach delivers an interpretable characterisation of these groups. Trees for these data are grown with MOB, LOTUS [Chan and Loh, 2005] and LMT [Landwehr *et al.*, 2005] and the algorithms are compared in terms of accuracy, parsimony and provided explanation. For the second example, the MOB algorithm with log linear Weibull node models is used to detect subsets of the original data that cause ill model fit. The data set itself is artificial but mimics the data used in Schober and Rusch [2010]. In this example an *a priori* assumed model, that has been used before in this context, fits badly to the data. The model tree approach can identify that this happens because the data set is actually merged from two different sources. It detects subsets for which the model fits well and poorly, corresponding to the two sources. Additionally, the paper compares various model tree algorithms in terms of node model versatility and properties of the tree induction.

Model Trees with Topic Model Pre-processing: An Approach for Data Journalism Illustrated with the WikiLeaks Afghanistan War Logs This (currently unpublished) paper [Rusch *et al.*, 2013, see Chapter 4] is a shortened version of Rusch *et al.* [2011] and uses a model tree with negative binomial node models with unknown shape parameter to mine fatality numbers in the Wikileaks Afghanistan war log. The WikiLeaks Afghanistan data consist of 76911 reports that cover the time period between January 2004 and December 2009 of the war in Afghanistan. They are low-level data with each entry describing a single incident in the war. The data set consists of 32 columns with numerical and factorial variables, of which nine are counts of killed/wounded/detained people of four groups. Only six variables are actually useful explanatory variables. But each incident also has a text summary which gives detailed information about what has been happening in that particular incident. To process the text summaries and extract information, Latent Dirichlet Allocation [Blei *et al.*, 2003], a topic model technique for clustering words into topics and documents into mixtures of topics, is used to generate new predictors. These predictors are more or less artificial tags of which topic a report belongs to. The predictors are then used to model the number of fatalities reported in each incident. In this data set, the fatality numbers are highly overdispersed count data for which the negative binomial distribution (in the gamma-Poisson mixture formulation) is an appropriate and popular parametric statistical model. In lack of a substantive theory to build the model upon, learning the model with recursive partitioning allows the predictors to have a complex influence in the model and to structure the response based on an appropriate node model. In this application, model trees are therefore basically used to estimate a restricted class of mixtures of negative binomial distributions.

Influencing Elections with Statistics: Targeting Voters with Logistic Regression Trees The third paper [Rusch *et al.*, 2012a, see Chapter 5] is concerned with the problem of voter targeting, i.e., with identifying those individuals that should be targeted in a mobilisation campaign. This is a special instance of the classic marketing problem of customer segmentation, but in this context the problem has been rarely investigated scientifically. In this paper a new framework of using logistic regression trees for prediction and identification of likely voters/non-voters that may allow more efficient campaign resource allocation is proposed. The framework (coined LORET) subsumes a number of techniques that are or can be used for this problem, namely logistic regression, classification trees and model trees with logistic regression models in the node. A sample of eligible and registered voters for the 2004 US presidential election from Ohio is used to illustrate targeting with LORET and assess performance of the different LORET types. The sample consists of 19634 people for whom there are voting records from 1990 to 2004 and roughly 80 demographic, behavioural and institutional covariates. The target variable is individual turnout in the 2004 presidential general election. Regarding the explanatory variables, the most important variables for targeting are individual historic voting records and age [Malchow, 2008], but as in this data set usually there are more variables available. A further aim of this investigation is therefore not only to compare targeting with the different LORET types but also to compare the impact of using only standard

variables vs. standard and additional variables. A bootstrap validation approach is used to generate training and test samples to gauge model performance. The combinations of LORET methods and variables are then compared in terms of predictive accuracy and efficiency (assessed by a linear cost-benefit function) for each out-of-bag sample. In line with the thesis objective, the focus of this paper lies in investigating the predictive capabilities of model trees by themselves and in comparison to non-parametric trees and non-partitioned rivals.

The remainder of this dissertation is as follows: In the next section I will continue with a more formal definition of the idea of model trees with GLM-type models. The next chapter, Chapter 2, is concerned with algorithmic approaches to model trees, all of which allow for one or more GLM-type models in the nodes. In Section 2.1.1, I review algorithms from the machine learning and statistics literature and briefly describe the conceptual ideas behind the various algorithms. In Section 2.1.2, I propose a generic algorithm for unbiased model trees along the lines of Gama [2004]. In Section 2.2 I will elaborate on the specific implementation of MOB for GLM-type models in R (the `mob` function from **party** [Hothorn *et al.*, 2012a] and how the provided functionality can be extended to allow for new GLM-type node models not yet implemented. This is illustrated with the extension to negative binomial models with unknown shape parameter in the nodes [as used in Rusch *et al.*, 2013] and can serve as a manual for enhancing `mob` functionality as well as writing objects of class StatModel. To illustrate the usage, an example of the effect of add-on health care insurance on the number of hospital visits is used. This is then followed by the second part, the collection of the three papers.

1.2. Model Trees with Generalized Linear Model Type Node Models

Model trees are results of specific algorithms that partition a predictor space in a certain way, and fit a statistical model in the resulting partitions that may be more sophisticated than fitting a constant. They can therefore be seen as statistical models that are a collection of simpler statistical models fitted to subsets or segments of the data. The segmentation itself is learned from the data. Generally, there are a number of ways how a segmentation can be learned. The way chosen for trees is to use a greedy, forward selection (usually by recursive partitioning) and hence restricting the possible segmentations to hierarchically nested, disjoint, tree-like partitions [see, e.g., Hastie *et al.*, 2009, for a discussion of the type of segmentation achieved].

1.2.1. Model Trees

Model trees generally look for a segmented (or piece-wise) model $\mathcal{M}_{\mathcal{B}}(\mathbf{Y})$, which is a collection of models $\mathcal{M}_{\mathcal{B}}(\mathbf{Y}) := \{\mathcal{M}_1(\mathbf{Y}), \ldots, \mathcal{M}_B(\mathbf{Y})\}$. Here $\mathbf{Y}$ are (possibly multivariate) observations from a space $\mathcal{Y}$. In principle the segmented model $\mathcal{M}_{\mathcal{B}}(\mathbf{Y})$ can employ

any type of statistical model for $\mathbf{Y}$ in the segments, i.e., for $\mathcal{M}_b(\mathbf{Y}), b = 1, \ldots, B$. It may even be the case that the models in the segments are different.

In case of GLM-type models as considered in this thesis, the segmented model is a collection of parametric node models and hence following Zeileis *et al.* [2008] or Rusch and Zeileis [2013] will be written as $\mathcal{M}_{\mathcal{B}}(\mathbf{Y}, \{\boldsymbol{\vartheta}_b\}_{b=1,\ldots,B})$. The modelled response in each segment follows a parametric distribution $P_{\boldsymbol{\vartheta}}$ from $\mathcal{P} = \{P_{\boldsymbol{\vartheta}} | \boldsymbol{\vartheta} \in \boldsymbol{\Theta}\}$ with $\boldsymbol{\vartheta}$ being of finite dimensionality. The existence of the real p-dimensional parameter vector in each segment $\boldsymbol{\vartheta}_b \in \boldsymbol{\Theta}_b$ has to be assumed and their collection over all segments is denoted by $\{\boldsymbol{\vartheta}_b\}_{b=1,\ldots,B}$. The set of oblique or perpendicular partitions $\{\mathcal{B}_b\}_{b=1,\ldots,B}$ of the space $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_l$ spanned by l covariates $Z_j, j = 1, \ldots, l$ gives rise to B segments within the data for which local parametric models $\mathcal{M}_b(\mathbf{Y}, \boldsymbol{\vartheta}_b), b = 1, \ldots, B$ may fit better than a global model $\mathcal{M}(\mathbf{Y}, \vartheta)$. All these local models may or may not have the same structural form, but they always differ in terms of $\boldsymbol{\vartheta}_b$. There are a number of ways how the segmentation can be learned and how model fitting can be achieved. For example, model tree algorithms may minimise the sum of some objective function Ψ, e.g., $\sum_{b=1}^{B} \sum_{i \in I_b} \Psi(\mathbf{Y}_i, \boldsymbol{\vartheta}_b)$ (with the corresponding indices $I_b, b = 1, \ldots, B$) over all conceivable partitions $\{\mathcal{B}_b\}_{b=1,\ldots,B}$ that will result in a set of vectors of parameter estimates $\{\hat{\boldsymbol{\vartheta}}_b\}$ (see Section 2.1.2). To find the optimal partition however is technically very difficult to achieve or may even be infeasible, which is why a greedy, hierarchical forward search of selecting only one covariate in each step is often suggested to approximate the optimal partition, also known as Hunt's algorithm [Hunt *et al.*, 1966].

1.2.2. Models of a Generalized Linear Model Type

Let $\mathbf{Y} = (y, \mathbf{x})$ denote a set of a response y and p-dimensional covariate vector $\mathbf{x} = (x_1, \ldots, x_p)$. For $i = 1, \ldots, n$ independent observations, the distribution of y_i can be any (regular) parametric distribution $P_{\boldsymbol{\vartheta}}$ from $\mathcal{P} = \{P_{\boldsymbol{\vartheta}} | \boldsymbol{\vartheta} \in \boldsymbol{\Theta}\}$ with $\boldsymbol{\vartheta}$ being of finite dimensionality and element of $\boldsymbol{\vartheta}$ will be (a known function of) the expected value $E(y_i) = \mu_i$. All y_i come from the same type of distribution. The n-dimensional vectors of fixed input values for the p explanatory variables are denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_p$. Following Aitkin *et al.* [2009], in these models it is assumed that the input vectors influence the conditional mean of y_i, $E(\mu_i | \mathbf{x}_i)$ only via a linear function, the linear predictor, $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$. More specifically, the conditional mean μ_i is seen as an invertible and smooth function of the linear predictor, i.e.

$$g(\mu_i) = \eta_i \text{ or } \mu_i = g^{-1}(\eta_i) \tag{1.1}$$

The function $g(\cdot)$ is called the link function. Please note that in these models it is assumed that the explanatory variables do not affect other parameters than the mean, i.e, parameters from the set $\{\boldsymbol{\vartheta} \setminus \mu\}$ ("nuisance parameters"), and therefore they are not modelled [Aitkin *et al.*, 2009]. Usually maximum likelihood estimation (MLE) is used to estimate the parameters.

Two groups of models will be distinguished in the following: generalized linear models [GLM; McCullagh and Nelder, 1989] and the somewhat more general GLM-type models.

The key difference lies in the fact that the responses y_i in GLM stem from an exponential family distribution with density [Aitkin *et al.*, 2009]

$$f(y_i; \mu_i, \phi) = \exp\{[y_i(\gamma')^{-1}(\mu_i) - \gamma(\gamma')^{-1}(\mu_i))]/\phi + \tau(y_i, \phi)\} \tag{1.2}$$

Here, the parameter of interest (natural or canonical parameter) is $(\gamma')^{-1}(\mu_i)$, ϕ is a scale parameter (known or seen as a nuisance) and γ and τ are known functions. If the link function has the form $g = (\gamma')^{-1}$, the link is called canonical. An important property of GLM and the reason for distinguishing them from GLM-type models is that the parameter vector $\boldsymbol{\beta}$ and the scale parameter ϕ can be obtained independently of each other with MLE [McCullagh and Nelder, 1989]. In GLM, estimates of parameters of the linear predictor $\hat{\boldsymbol{\beta}}$ are therefore (almost) independent of estimates of $\hat{\phi}$ under suitable limiting conditions [Aitkin *et al.*, 2009]. Accordingly, GLM-type models are models that integrate linear predictors for the mean as in (1.1), but for whom the mentioned properties of parameter orthogonality does not hold. However, most of the methodology presented throughout this thesis is valid beyond the standard GLM and also applies to models for distributions with non-orthogonal parameters. Among those are distributions to model survival such as the exponential distribution, the Weibull distribution or the extreme value distribution, or mixtures of exponential families such as the negative binomial distribution with unknown shape parameter. Here, therefore, the types of models are mostly treated interchangeably, unless the fact that for GLM-type models estimation of the linear predictor is confounded with nuisance parameter estimation is of special relevance.

2. Computational Aspects

2.1. Algorithms for Model Trees

Tree algorithms that allow to fit more than a constant in each node have been around at least since Quinlan [1992], where he proposed the M5 algorithm, an algorithm that allowed for a linear model in the terminal nodes. Many of the later advances of classical model and tree hybrids is owed to Loh and his co-authors [Chaudhuri *et al.*, 1994, 1995, Loh and Shih, 1997, Loh, 2002, Kim and Loh, 2001, Chan and Loh, 2005] as well as Ahn and his colleagues [Ahn, 1994b,a, Ahn and Loh, 1994, Ahn, 1996b,a, Ahn and Chen, 1997, Choi *et al.*, 2005]. Additionally there has been notable work done by Landwehr *et al.* [2005], Gama [2004], Su *et al.* [2004], Potts and Sammut [2005], Zeileis *et al.* [2008] and Sela and Simonoff [2012].

2.1.1. Review of Model Tree Algorithms

In this section I briefly review some of the advances in the development of model tree algorithms over the last twenty years. This review only takes algorithms into account that allow node models that fit into the GLM-type framework as laid out in Section 1.2.2. Table 2.1 (an updated version of Table 3 in Rusch and Zeileis [2013]) lists properties of the tree built by the different algorithms described below, such as unbiasedness of the split variable selection as well as which node models can be used.

Functional Trees The ideas of Gama [2004] can be seen as a generic framework for building tree algorithms with univariate or multivariate splits and univariate or multivariate leave models. Since univariate splits are common in most tree algorithms, Gama suggests to include the predicted values in the node as an additional candidate for splitting and therefore achieving splits based on multiple variables. According to Gama tree models with multivariate splits and multivariate leave models have some advantage, mostly in large data sets. The Gama framework is rather unspecific about the crucial steps of model fitting, split variable selection and split point choice and should therefore not be seen as a specific model tree implementation. The provided example uses only Gaussian models in the nodes.

SUPPORT The SUPPORT algorithm [Chaudhuri *et al.*, 1994, 1995] was originally intended to fit piecewise-polynomial models with linear predictors and metric covariates. There the algorithm has been formulated for Gaussian models, binomial models with logit link and Poisson models. Later the specific approach of SUPPORT has been extended to quasi-binomial models [Ahn and Chen, 1997], Quasi-Poisson models [Choi *et al.*, 2005] and parametric survival regression models like proportional hazard models [Ahn and Loh, 1994], Weibull models [Ahn, 1994b], exponential models [Ahn, 1994a], log-normal [Ahn, 1996b] and log-gamma models [Ahn, 1996a]. This way, it became the first ample algorithm for model trees. SUPPORT works as follows: The model is fitted and then residuals are calculated. All negative and all positive residuals form a class respectively. After the observations have been grouped according to the sign of the residuals, test statistics for the difference in means or variances or both between the

observations belonging to the two classes are calculated for each explanatory variable and p-values are computed. The variable with the overall lowest p-value is selected for splitting and the split is carried out along the average of the two signed residual class means. Hence SUPPORT does not use an exhaustive search to find the split point, rather SUPPORT selects the split point by an analysis of the distribution of the residuals. For pruning, a cross-validatory look-ahead procedure is used to determine if the split should actually be carried out. This means the trees get pre-pruned while grown. SUPPORT also allows for weighted averaging to get a smooth functional form rather than a discontinuous one for the piecewise models. This algorithm requires the node models to be fitted once in every single node. Additionally, SUPPORT does not support the distinction between node model variables and partitioning variables. Moreover, the fact that the variable selection and the split point are not found independently of each other suggests that SUPPORT is not unbiased in variable selection.

MLRT This approach has been presented by Su *et al.* [2004]. The principal idea here is to embed trees in a maximum likelihood framework. For this, tree induction and node model estimation are carried out by using the same objective function (the likelihood). Tree induction happens by maximising the log-likelihood score for each possible split over all candidate variables. The best split is then chosen to be the one with the overall maximum likelihood over all permissible splits and therefore amounts to finding the change point for a parametric model. This approach makes split variable selection in MLRT biased. After a large tree is induced with MLRT, it is pruned back by a procedure based on information criteria. The best subtree is then selected with a cross-validation approach. In the paper by Su *et al.* [2004], the algorithm is only specified for a classic regression tree with a constant fitted in each node for a Gaussian response variable and hence basically reframes the CART algorithm for continuous responses [Breiman *et al.*, 1984]. Although this leads to a not very versatile algorithm, the principal idea can be extended quite easily to any GLM-type model.

LOTUS The LOTUS algorithm [Chan and Loh, 2005] is a specific model tree algorithm only intended for binary responses and with logistic regression in the nodes. The algorithm works like this: A logistic model is fitted in the current node. Depending on the node model, different χ^2 test statistics are then employed to select the splitting variables. For a multiple structural model, a linear probability model is used to approximate the logistic model in the χ^2 test statistic. Metric covariates get discretised to conduct the test. The candidate variable with the lowest p-value is then chosen for splitting. After the variable is selected, the split point is chosen by minimising the sum of the deviance of the logistic model in two candidate partitions defined by split point candidates that are heuristically found. This algorithm's usage of a χ^2 test for variable selection leads to a split variable selection that is unbiased. However using the χ^2 test also for metric variables does not take all the available information into account. Concerning the model in the nodes, LOTUS allows to either fit the "optimal" simple model, a prespecified model or a stepwise adapted multiple model in the terminal nodes. The logistic regression in

the nodes only allows to use metric predictors.

LMT This algorithm [Landwehr *et al.*, 2005] is specifically designed to build a model tree with a logistic model for binary or multinomial responses in the nodes. It aims at high predictive accuracy rather than interpretability. Different from many other algorithms, it allows for binary as well as multiway splits (for categorical variables). LMT employs boosting [LogitBoost; Friedman *et al.*, 2000] to fit the logistic model in the nodes and a tree induction very similar to C4.5 [Quinlan, 1993]. The latter makes LMT biased in variable selection. The pruning algorithm is borrowed from CART (cost-complexity pruning with a cross validation to select the optimal subtree, Breiman *et al.* [1984]). LMT always conducts adaptive variable selection with the LogitBoost algorithm in the nodes which reflects the intention to use it for prediction rather than explanation. LMT is a slow algorithm because of its combination of tree induction, boosting iterations and cross validation for pruning and finding the optimal number of boosting iterations but can be sped up by a number of heuristics [cf. Landwehr *et al.*, 2005].

GUIDE The GUIDE algorithm [Loh, 2002, 2009] is the first algorithm specifically designed to avoid the problem of split variable selection bias and has been invented as a classification, regression and model tree algorithm. As far as model trees are concerned, it has originally been proposed for node models for Gaussian responses (linear and polynomial models) and has later been extended to linear and polynomial quantile regression [Chaudhuri and Loh, 2002], Poisson models [Loh, 2008], Quasi-Poisson models [Choi *et al.*, 2005], proportional hazard models and longitudinal models [Loh and Zheng, 2012] as well as generalized estimating equations [Lee, 2005] and bears some similarities with SUPPORT or LOTUS. It selects the splitting variable with adjusted or classic χ^2-test statistics between residuals and predictors for main or pairwise interaction effects similar to LOTUS. To ensure unbiasedness, GUIDE further employs a built-in bias correction for the resulting p-values based on the bootstrap [Efron and Tibshirani, 1994]. The actual split is either chosen based on the error sum of squares (metric variables) or binomial variances (categorical variables) via an exhaustive search over all possible splits or based on the median of the splitting variable (similar to SUPPORT) or over a sample of order statistics. Pruning of a large tree is carried out with classic cost-complexity pruning and cross validation [Breiman *et al.*, 1984]. The GUIDE algorithm allows for specification of variables as node model variables or partitioning variables or both. Additionally, GUIDE allows stepwise variable selection in the node model.

M5 and M5' The algorithm M5 [Quinlan, 1992] can be seen as the father of all model-based trees. The inner workings of the proprietary M5 are not well documented which lead Wang and Witten [1997] to "rationally reconstruct" the algorithm. The result was coined M5', which will be described now. M5' (and M5) was developed for fitting linear regression models in tree nodes. Split variable selection is based on minimising intra-subset variation hence a node's standard deviation is the measure of node impurity. The expected reduction in standard deviation because of a split is then used to choose the

split point. One cannot specify a node model but all variables are partitioning and node model variables at the same time. Pruning is carried out more or less the same way as in C4.5. More specifically, a whole tree is grown and all subsequent variables that are used after a split in the whole tree are used in the regression function. Then greedy variable selection is used for each node. The tree is then pruned back as long as the error (the scaled up average absolute residual) decreases. Then the tree is smoothed for adjacent linear models. M5' has a correction for selection bias of categorical variables with many splits but it is not enough to make variable selection as unbiased as in GUIDE, LOTUS or MOB.

MOB The recently proposed MOB algorithm [Zeileis *et al.*, 2008] utilizes a rigorous framework of model parameter estimation with and parameter stability tests for M-estimators, which includes maximum likelihood or quasi likelihood models as a special case, and hence is in principle the most versatile algorithm currently available. An implementation of MOB exists for weighted and ordinary least squares, generalized linear models and GLM-type models from the quasi families (Zeileis *et al.* [2008], for the latter two see also [Rusch and Zeileis, 2013]) as well as multinomial logit models, beta regression models, negative binomial models [Rusch *et al.*, 2013], the Bradley-Terry model [Strobl *et al.*, 2011], the Rasch model [Strobl *et al.*, 2010], the partial credit model and parametric survival models [Zeileis *et al.*, 2008] (see also Rusch and Zeileis [2013]). For GLM-type models MOB exceeds the versatility even of SUPPORT and GUIDE (see Table 2.1). The basic steps of the algorithm for GLM-type models are (for details see e.g. Zeileis *et al.* [2008] or Chapter 3) to fit a GLM-type model to all observations in the current node by setting the gradient of the log-likelihood (score function) to zero. Then the stability of the score function evaluated at the estimated parameter is assessed with generalized M-fluctuation tests [Zeileis and Hornik, 2007] with respect to every possible ordering of the values of each partitioning covariates. If there is significant instability for one or more covariate, the covariate associated with the highest instability is selected. If no significant instability is found, the algorithm stops. After a splitting variable has been selected, the split points are computed by locally optimising the sum of the log-likelihood or deviance for two rival segmentations by an exhaustive search over all pairwise comparisons of possible partitions. This is repeated recursively for each daughter node until no significant instability is detected or another stopping criterion is reached. This approach works the same way for practically all parametric models for which a score function is defined. The main properties of MOB are a) model fitting, split variable selection, splitting and pruning are based on the same objective function and b) split variable selection is unbiased. Conceptually, the MOB algorithm combines many advantages of the aforementioned algorithms. For example, like GUIDE or LOTUS, it uses unbiased split selection and allows for separation of node model and splitting variables. It rigorously extends the idea of MLRT of change point estimation and it employs the same objective function to induce the tree structure, fit the node models and prune the trees. Comparable to SUPPORT, MOB pre-prunes the trees by only selecting variables with significant instability. MOB with more than a single explanatory variable

in the node model or with interactions as splitting variables can be seen as employing multivariate splits along the lines of Gama [2004]. This way it also allows for oblique partitioning.

2.1.2. A Generic Algorithm for Unbiased Model Trees

Gama [2004] presented a generic algorithm to functional trees in supervised learning problems that lays out required steps when fitting model trees without getting into detail on how the node model is fitted, how split variables and split points are found and the partition are built. Trees constructed after this fashion however may lack the desirable properties of unbiasedness in variable selection, which is the distinguishing feature of procedures like GUIDE and MOB. Along the lines of Gama however, it is possible to formulate a generic algorithm that a) aims at unbiased splits (for specific circumstances) and b) allows many types of extended models in the partitions.

A Generic Algorithm Using notation from Section 1.2.1, let the extended model fitted in each segment b to a possibly multivariate observation $\mathbf{Y}$ (with p variables) be denoted by $\mathcal{M}_b(\mathbf{Y}), b = 1, \ldots, B$, with $\mathbf{Y} \in \mathcal{Y}$. The observations in each segment b are denoted by $\mathbf{Y}_i, i \in I_b$. The collection of all segment-specific models is $\mathcal{M}_\mathcal{B}(\mathbf{Y}) := \{\mathcal{M}_1(\mathbf{Y}), \ldots, \mathcal{M}_B(\mathbf{Y})\}$. The objective function $\Psi : \mathcal{Y} \to \mathcal{S}$, where $\mathcal{S}$ is e.g., $\mathbb{R}^k, k \leq \max(n, p)$ or more generally some feature space, is used to connect the model to the $i \in I_b$ observations which is denoted by $\Psi_i = \Psi(\mathbf{Y}_i)$. There usually is some suitable function $f : \mathcal{S} \to \mathbb{R}^h, 1 \leq h \leq k$ of Ψ_i, $f([\Psi_i]_{i \in I_b})$ that is optimised to actually fit the model to all observations. In parametric models, a common choice for f might be the sum, i.e., $f\left([\Psi_i]_{i \in I_b}\right) = \sum_{i \in I_b} \Psi_i$. The l covariates $Z_j, j = 1, \ldots, l$ are split candidate variables that span the space $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_l$. This space is to be partitioned into the $\{\mathcal{B}_b\}_{b=1,\ldots,B}$ with B segments. A generic algorithm could then look something like this:

1. Model fitting: A model $\mathcal{M}_b(\mathbf{Y})$ is fitted to the data $\mathbf{Y}_i, i \in I_b$ in the current node b (including the root node containing all observations at the beginning) by optimizing $f\left([\Psi(\mathbf{Y}_i)]_{i \in I_b}\right)$. The function $f(\cdot)$ might aggregate Ψ_i over the observations and $\Psi(\cdot)$ has to be a specific function depending on the desired node model. In particular this node model can be a parametric, a semiparametric or a nonparametric supervised or unsupervised model. For example, for MOB with GLM-type node model and estimation by maximum likelihood, this is the sum of the negative log-likelihood contributions of the observations to the fitted parametric model, i.e., $\Psi(\mathbf{Y}_i, \boldsymbol{\vartheta}) = -\log L(\boldsymbol{\vartheta}|\mathbf{Y}_i)$ and $f(\cdot) = \sum_{i \in I_b}(\cdot)$.

2. Split variable selection: This is the most crucial step for induction of trees that are unbiased. Some function $\lambda(\mathbf{Y}), \lambda : \mathcal{Y} \to \mathbb{R}^h$ of the observations that maps the observation onto the real line is needed. Preferably, $\lambda(\cdot)$ is also a function of Ψ_i to stay within a single framework for tree induction and model fitting and avoid the *ad hoc* nature [Murthy, 1997] of many tree algorithms. To ensure the most basic unbiased split variable selection, $\lambda(\cdot)$ must be invariant to the number of possible

		Model tree algorithm							
		FT	GUIDE	LMT	LOTUS	M5'	MLRT	MOB	SUPPORT
Tree structure	Pre-pruning		*		*			×	×
	Post-pruning	×	×	×	×	×	×	*	*
	Unbiased		×		×			×	
	Covariate type	all	all	all	all	all	all	all	metric
	Multiway splits	×		×					
Separate node model and splitting variables			×		×			×	
Adaptive node model		*	×	×	×	×			*
Type of node model	Gaussian	×	×			×	×	×	×
	Binomial (Logit)	×		×	×		*	×	×
	Binomial (other links)	*					*	×	
	Quasi-Binomial (Logit)	*						×	×
	Quasi-Binomial (other links)	*						×	
	Poisson	*	×				*	×	×
	Quasi-Poisson	*	×					×	×
	Gamma	*					*	×	×
	Inverse Gaussian	*					*	×	
	Negative Binomial	*	*				*	×	
	Beta	*					*	×	
	Multinomial Logit	×		×			*	×	
	Parametric Survival	*	×				*	×	×
	Longitudinal Gaussian	*	×				*	*	×
	General Maximum Likelihood	*					*	*	
	Generalized Estimating Equations	*	×					*	
	General Quasi-Likelihood	*	×					*	
	Robust (M-type)	*						*	
	Quantile	*	×						

Table 2.1.: Comparison of properties and applicability of different model tree algorithms. For the rows a × denotes if there already exists an implementation and ∗ denotes if an implementation is possible within the provided framework of the specific algorithm without changes (please note that Functional Trees (FT) are on a more abstract level than all the other specific algorithms). This is an updated version of Table 3 from Rusch and Zeileis [2013].

split points, i.e., a variable with hundred possible split points must not be preferred to a variable with two split points if both variables have no association with the response (see Loh and Shih [1997] or Kim and Loh [2001] for a more lengthy discussion of unbiased split variable selection). Then $\lambda_j([\mathbf{Y}_i]_{i\in I_b})$ or $\lambda_j([\Psi_i]_{i\in I_b})$ is assessed for every split variable candidate Z_j. There needs to be a decision rule which Z_j to select based on $\lambda_j(\cdot)$ or a function thereof. For example in MOB, $\lambda(\cdot)$ is a parameter stability test statistic, and the decision rule is to use the Z_j with the lowest p-value for $\lambda_j\left([\Psi_i]_{i\in I_b}\right)$. In GUIDE $\lambda(\cdot)$ is a χ^2 test statistic and again the Z_j with the lowest p-value for $\lambda_j\left([\mathbf{Y}_i]_{i\in I_b}\right)$ is chosen.

3. Split point selection: After the split variable Z_j is selected, we need to find the m split points for that variable. Usually a single split point is chosen (binary tree). One can approximate the optimal split by using an exhaustive search over all possible splits and compare the sum (or another function) of the objective function in both partitions, for example, $\sum_{b=1}^{B} f\left([\Psi(\mathbf{Y}_i)]_{i\in I_b}\right)$ (with the corresponding indices $I_b, b = 1, \ldots, B$) over all conceivable partitions $\{\mathcal{B}_b\}$. The split that minimises the sum (or possibly other function) of the objective function in the local partitions is then chosen.

4. Split: Split the node into $m + 1$ daughter nodes.

5. Recursion: Repeat the procedure recursively in each daughter node until a stopping criterion is reached.

This generic algorithm can have the property that split variable selection, split point selection and model fitting are all based on the same objective function. Also, the algorithm fulfills the basic condition necessary for unbiased variable selection, i.e., "independence" of the choice of split variable and split point. Classic algorithms like CART and C4.5 go through all splits on all variables and greedily use the locally best one, which makes them susceptible for chossing splitting variables with more splits ("biased").

Pruning Usually, trees are pruned. For a supervised learning problem this can help to avoid overfitting. For the generic algorithm from above, any type or pre- or post-pruning can be used (e.g. cost-complexity pruning as in CART [Breiman *et al.*, 1984], a significance level of a test statistic as in MOB [Zeileis *et al.*, 2008] or the generic pruning procedure by Gama [2004]). To stay within a single statistical framework and to avoid the appearance of being *ad hoc*, it is once again preferable to have the pruning procedure depend on $\Psi(\mathbf{Y}_i)$.

2.2. Implementation and Extension of MOB in R

The MOB algorithm has been introduced in Zeileis *et al.* [2008] and was accompanied by an implementation of MOB in R [R Development Core Team, 2012] in the package **party** [Hothorn *et al.*, 2012a]. The implementation already allowed to fit least squares and generalized linear models as well as the corresponding quasi likelihood models and the paper described the application of some those models. Rusch and Zeileis [2013] discussed the application of MOB with GLM-type node models in more detail and argued that for these kind of node models, MOB is currently the most versatile algorithm. This versatility is partly due to the general framework that the MOB algorithm provides and partly due to the specific implementation of MOB in R , which is designed to be very modular and can be extended relatively easily. For example, extensions to psychometric models (Bradley-Terry, Rasch and Partial Credit Models) are available in the package **psychotree** [Zeileis *et al.*, 2012b]. Extension is especially simple for GLM-type models.

Principles of how to extend the functionality have already been laid out in the package vignette [Zeileis *et al.*, 2012a] but are kept on a more general level. Therefore, in line with adding to the work done by Zeileis *et al.* [2008], in this section I describe the current design of the `mob` function from the package **party** and provide a detailed discussion of how to extend the current functionality to new node models by using a concrete example of an extension used in Rusch *et al.* [2013], negative binomial node models with unknown shape parameter. The latter can serve as a general manual for making additional node models available to the `mob` function (and, more generally, new StatModels).

2.2.1. MOB in R

The functionality for fitting model-based recursive partitioning models is designed to be modular. Accessible on the user level is the `mob` function for model fitting, `mob_control` for setting tuning parameters and controlling the behaviour of the fitting function and a number of S3 methods for objects of class `mob` for the following generics: `print`, `summary`, `residuals`, `fitted`, `coef`, `summary`, `sctest`, `logLik`, `deviance`, `weights` and `predict`. These functions will be briefly described in the following:

`mob` The main model fitting function. It uses R 's formula interface to specify the model tree to be fitted. The formula object is a multipart formula [Zeileis and Croissant, 2010] that specifies the functional form and is of the form `y` $\sim$ `x1+ ... +xk | z1+...+zl`, where the part with the `x` variable set is the node model (e.g. a GLM-type model) and the `z` variables are used for partitioning. The variable sets used left and right of `|` can be overlapping. Please note that on the left and right hand side of `|` any type of R formula syntax [see, e.g., Chambers and Hastie, 1992] can be used. In particular this applies to any kind of model linear in the parameters for the `x` on the left hand side and to include interactions of the `z` on the right hand side to allow oblique partitioning. To specify the type of data model and the according likelihood, the `model` statement is used. Here, an object of class StatModel [Hothorn *et al.*, 2012c] must be passed, e.g. the prototypical

`glinearModel`. Additional arguments can be passed as well, controlling the `fit` function in the fit slot of the StatModel object. These arguments must match either the arguments of the StatModel or the function it uses as its work horse, for example a `family` or a `link` or any other `glm.fit` argument to `glinearModel`. The `mob` function's behaviour can be controlled with the `control` argument that needs a list of class mob_control. Additional arguments are `data` (the name of the data frame where the `y`, `x` and `z` are to be found), `weights` (a vector of weights for fitting) and `na.action` which determines what is done if the data contain missing values. Please note that if an intercept only model is fitted in the nodes, the latter should be set to `na.action=NULL`.

`mob_control` Allows to specify meta parameters used for the fitted model tree such as `alpha` the global significance level for the parameter stability tests, the logical `bonferroni` if family-wise error correction should be applied, `minsplit` the minimum number of observations per node, `trim` the trimming parameter for parameter stability tests for metric variables, `objfun` a function for extracting the minimised value of the objective function from a fitted node (defaults to deviance and is also used for finding the locally optimal split in `z`), `breakties` a logical indicating if ties should be randomly broken for calculation of the parameter stability tests, `parm` the model parameters that should be included for the stability tests and `verbose` a logical specifying if fit information should be printed on screen.

`print` Prints the model tree structure and the node coefficients. If possible inherits from the fitted model object returned from the underlying model fitting function (e.g. from `glm`).

`summary` Prints a summary of the fitted model in each terminal node of the model tree. If possible inherits from the fitted model object returned from the underlying model fitting function.

`residuals` Extracts the residuals for each observation. If possible inherits from the fitted model object returned from the underlying model fitting function.

`fitted` Extracts the fitted values for each observation. If possible inherits from the fitted model object returned from the underlying model fitting function.

`coef` Extracts the node model coefficient for each terminal node. If possible inherits from the fitted model object returned from the underlying model fitting function.

`sctest` Extracts the parameter stability test statistics and p-values for each coefficient for the nodes.

`logLik` Extracts the log-likelihood of the model tree. If possible inherits from the fitted model object returned from the underlying model fitting function.

`deviance` Extracts the deviance of the model tree. If possible inherits from the fitted model object returned from the underlying model fitting function.

`weights` Extracts the weights used in the fitting process. If possible inherits from the fitted model object returned from the underlying model fitting function.

`predict` Predicts either the observations supplied as `newdata` as well as returns the fitted values for the fitted object or predicts the terminal node number of each observation. If possible inherits from the fitted model object returned from the underlying model fitting function.

2.2.2. Extending MOB in R

Currently, `mob` allows to use objects of class StatModel as defined in the package **modeltools** [Hothorn *et al.*, 2012b] for the node model. This ensures an object-oriented approach of providing a base functionality that can be applied to different model objects corresponding to different node models. This way the `mob` function can be extended relatively simple. Currently the available StatModels in **modeltools** are linear models (`linearModel`), generalized linear models (`glinearModel`) and survival models (`survReg`), which provide StatModel interfaces to `lm.fit`, `glm.fit` and `survreg` respectively. Additionally I have implemented experimental versions[1] of interfaces to `glm.nb` for negative binomial models with unknown shape parameter [`negbinModel`; Rusch *et al.*, 2013], for binary logistic regression in presence of quasi-complete separation [`safelogitModel`; Rusch *et al.*, 2012a] and an interface to `mlogit` for multinomial logistic regression, discrete choice models, extreme value models and rank-ordered logit models (`mlogitModel`).

In this section I describe how to extend `mob` functionality (up until version 1.0-1 of **party**[2]) to allow for other node models by using `negbinModel` as an example. For this, please note that the generics and methods for using and extending `mob` usually are S3 generics and methods and, unless clearly stated, method and generic refers to S3 method and generic. In a nutshell the following utilities are necessary[3] and have to be implemented (or borrowed):

1. Node model fitting function
   - A fitting function `foo` or `foo_fit` which fits the desired model and returns the object of class foo. It should be possible to supply a weights argument to `foo` or `foo_fit`.
   - A `weights` method for class foo to extract weights and a `estfun` method for class foo to extract the empirical estimation function value for each observation.
2. StatModel object

[1] They are available in the **mobtools** package [Rusch *et al.*, 2012b].
[2] There are plans to rewrite the functionality to allow even easier interfacing.
[3] Of course this is just one way of doing it. There are other ways, for example incorporating the code for fitting the model directly in the StatModel without interfacing a fitting function.

- An object of class StatModel that provides an interface to `foo` or `foo_fit`. The returned object from fitting the StatModel is of class bar.

3. `reweight` and additional functions
   - The `mob` function expects a `reweight` method for class bar that allows refitting the StatModel with new weights.
   - Implementations of methods for class bar for different generics, especially `summary`, `print`, `coef` and `predict` to make full use of the `mob` functionality. Optionally, but recommended.
   - For graphical display of the fitted model tree with `plot`, a panel-generating function, e.g., `node_foobarplot` should be written that generates a useful visualisation of the node model. Optionally, but recommended.

These steps are described in more detail in the next three sections.

Providing a Function for Fitting the Node Model

The first brick for extending `mob` is to provide a model fitting function. R base and contributed packages provide numerous fitting functions for many different models that can be used for this purpose. If no such implementation is available, one has to write its own. The fitting function can be either a function `foo_fit` that takes a design matrix `x` (e.g. from `model.matrix`) and the response vector `y` as arguments, `foo_fit(x,y,...)`, or a function that uses a formula interface, `foo(formula, data,...)`. The former is a bit cleaner to integrate with StatModel and faster as formulae need only be parsed once, but the latter probably can be more widely applied to the R ecosystem.

In principle the fitting function needs not follow any specific design rules. The only requirement for the objects of class foo returned by `foo` or `foo_fit` is that there are methods for class foo that allow extracting the empirical estimation function via `estfun` and allow extracting observation weights via `weights` (obviously called `estfun.foo` and `weights.foo`). However, integration is considerably easier if the object returned by `foo` or `foo_fit` has a similar structure to an object of class glm and if the function can take and use a `weights` argument, e.g., `foo_fit(x, y, weights,...)` or `foo(formula, weights, data,...)`.

The `estfun` method should extract a $n \times k$ matrix corresponding to the n observations and k parameters containing the empirical estimating functions. The columns should be named as in `coef(fooobject)` or `terms(fooobject)`, respectively. For example, for GLM and some GLM-type models, the `estfun` method for class glm from the package **sandwich** [Zeileis, 2006] can be used to extract the estimation function $r_i * x_i/\phi$, where r_i are the working residuals, the difference between the response and the linear predictor at convergence [Hardin and Hilbe, 2007], i.e. $r_i = (y_i - \hat{\mu}_i)\left(\frac{\partial \eta}{\partial \mu}\right)$ and ϕ being the scale parameter of the exponential family (1.2). The output of `estfun` should look similar to

```
R> set.seed(210485)
R> x <- runif(10)
```

```
R> y <- rnorm(10)
R> m <- glm(y~x)
R> estfun(m)

   (Intercept)         x
1       1.4692  1.397116
2       1.3959  0.002552
3       1.5662  0.552461
4       1.2166  0.756445
5      -3.2336 -1.424275
6      -2.5661 -0.986565
7      -2.6129 -0.213369
8      -0.6664 -0.664595
9       1.2681  0.186194
10      2.1629  0.394036
```

The second required method is a `weights` method that implements functionality that allows to determine which observations were used or not, i.e. allows observations to have weights of 0 or 1. Since we need only the working weights this function can be very simple if the fitting function already outputs these weights.

Additionally, it is useful to have a number of methods for standard generics implemented as well, such as `summary`, that the corresponding methods for class mob can inherit from. R functions that are modelled after `glm` usually fulfill the requirement and provide many additional functions to be inherited.

To illustrate, I use the example of negative binomial regression models with unknown shape parameter. The **MASS** package [Venables and Ripley, 2002] offers a fitting function `glm.nb` and a number of S3 methods for class negbin. A fitting function can be obtained simply by adapting Venables and Ripley's code from `glm.nb` to yield a function `glm.nb_fit` or to use `glm.nb` directly and build the necessary formula object in the fit slot of the StatModel.

The former strategy integrates more neatly into the idea of using a workhorse function to fit the model and can be more efficient as the response vector and design matrix have already been calculated. In our case the fitting function would look something like

```
R> glm.nb_fit <- function(x, y, weights, ...)
  {
   "<body>"
  }
```

with `x` being the design matrix, `y` the response vector and `weights` some observation weights. Please note that additional arguments `...` will be quite rich in this case. The term `"<body>"` of course stands for the whole inner code to fit the model.

Alternatively, one could use the other approach of using `glm.nb` directly. This has the advantage of only wrapping the original function in the StatModel which, especially

for fitting functions borrowed from other contributors, makes maintenance easier and acknowledges the original authors. For our example the function definition is:

```
R> glm.nb <- function (formula, data, weights, ...)
  {
   "<body>"
  }
```

Objects returned from `glm.nb_fit` or `glm.nb` are of class negbin and can use the S3 methods for class negbin from **MASS** [Ripley *et al.*, 2012]. They are also objects of class glm and therefore the `estfun` method for glm can be used as the function for extracting the empirical estimation functions

```
R> getS3method("estfun","glm")

function (x, ...)
{
    xmat <- model.matrix(x)
    xmat <- naresid(x$na.action, xmat)
    if (any(alias <- is.na(coef(x))))
        xmat <- xmat[, !alias, drop = FALSE]
    wres <- as.vector(residuals(x, "working")) * weights(x, "working")
    dispersion <- if (substr(x$family$family, 1, 17) %in% c("poisson",
        "binomial", "Negative Binomial"))
        1
    else sum(wres^2, na.rm = TRUE)/sum(weights(x, "working"),
        na.rm = TRUE)
    rval <- wres * xmat/dispersion
    attr(rval, "assign") <- NULL
    attr(rval, "contrasts") <- NULL
    res <- residuals(x, type = "pearson")
    if (is.ts(res))
        rval <- ts(rval, start = start(res), frequency = frequency(res))
    if (is.zoo(res))
        rval <- zoo(rval, index(res), attr(res, "frequency"))
    return(rval)
}
<environment: namespace:sandwich>
```

To extract weights, we need a `weights` method for negbin, `weights.negbin`, which is

```
R> weights.negbin <- function (object, ...)
 {
     res <- object$weights
     if (is.null(object$na.action))
```

```
        res
    else napredict(object$na.action, res)
}
```

If there are no methods `estfun` and `weights` for the class of objects returned by the fitting function, they need to be provided by the author.

Writing a StatModel object

If a fitting function, a `weights` method and an `estfun` method are available, an object of class StatModel [Hothorn *et al.*, 2012c] needs to be defined, which interfaces the fitting function. This is arguably the trickiest part, since the functionality of **modeltools** [Hothorn *et al.*, 2012b] is experimental and documentation of the functions, objects and concepts is rather terse. Basically, **modeltools** tries to provide tools to deal with statistical models in an object-oriented way which may provide a common ground for handling all type of models in R .

Among those tools is the class StatModel, which is a class that attempts to provide unified infrastructure and a clean representation of unfitted statistical models. This basically means that a StatModel already knows what type of model it is and how the data must look like, but it has not yet seen any data. Its aim is to provide a generic approach to fit models to data. Here, an unfitted model provides a function for data pre-processing (`dpp`, e.g. generating design matrices), a function for fitting the specified model to data (`fit`), and a function for computing predictions (`predict`). To extend node model functionality for `mob`, such a StatModel object has to be written. Some methods for generics such as `predict`, `fitted`, `print` and `model.matrix` are provided in the package to make use of the StatModel structure.

As an example[4] the code for a new StatModel for negative binomial models with unknown shape parameter, `negbinModel`, is:

```
negbinModel <- new("StatModel",
    capabilities = new("StatModelCapabilities"),
    name = "negative binomial GLM-type model",
    dpp = ModelEnvFormula,
    fit = function(object, weights = NULL, ...){
        if (is.null(weights)) {
            z <- glm.nb_fit(object@get("designMatrix"),
    object@get("response")[,1],
                            mustart=NULL, etastart=NULL,
                            control=glm.control(),
                            intercept=all(object@get("designMatrix")[,1] == 1),
                             ...)
        } else {
            z <- glm.nb_fit(object@get("designMatrix"),
```

[4]This or the `glinearModel` StatModel can serve as a general template for GLM-type models.

```
                            object@get("response")[,1],
                            w = weights,
                            mustart=NULL, etastart=NULL,
                            control=glm.control(),
                            intercept=all(object@get("designMatrix")[,1] == 1),
                            ...)
        }
        class(z) <- c("negbinModel", "negbin", "glm", "lm")
        z$offset <- 0
        z$contrasts <- attr(object@get("designMatrix"), "contrasts")
        z$terms <- attr(object@get("input"), "terms")
        z$predict_response <- function(newdata = NULL) {
            if (!is.null(newdata)) {
                penv <- new.env()
                object@set("input", data = newdata, env = penv)
                dm <- get("designMatrix", envir = penv, inherits = FALSE)
            } else {
                dm <- object@get("designMatrix")
            }
            pr <- z$family$linkinv(drop(dm %*% z$coef))
            return(pr)
        }
        z$addargs <- list(...)
z$ModelEnv <- object
        z
    },
    predict = function(object, newdata = NULL, ...)
        object$predict_response(newdata = newdata)
)
```

We can see that StatModel has the following slots: `name`, `dpp`, `fit` and `predict`. The slot `name` contains the name of the model as an object of class character

```
R> negbinModel@name
```

```
[1] "negative binomial GLM-type model"
```

The slot dpp is a function object that does data preprocessing and usually takes a formula as input. For the negbinModel this slot (and all the others except `fit` and `name`) is the same as in glinearModel and is a ModelEnvFormula[5].

```
R> negbinModel@dpp
```

[5]For brevity, this slot is not expanded because the code for `ModelEnvFormula` is not directly related to the extension of `mob` to new GLM-type node models. In particular, it does not need to be changed.

The slot `predict` contains a function object that computes predictions

```
R> negbinModel@predict

function (object, newdata = NULL, ...)
object$predict_response(newdata = newdata)
```

and the slot `capabilities` contains an object of class StatModelCapabilities.

```
R> negbinModel@capabilities

An object of class "StatModelCapabilities"
Slot "weights":
[1] TRUE

Slot "subset":
[1] TRUE
```

The most important slot for extending `mob` functionality is the `fit` slot. It is a function object that actually fits the model to the data. For new GLM-type node models, the `negbinModel` or `glinearModel` can be adapted. Changes usually have to be made only to the `fit` slot (where the `predict_response` function used in the `predict` slot also resides). Depending on whether the fitting function takes a design matrix and response variable as arguments (as in `glm.nb_fit`), or whether it takes a formula (as in `glm.nb`), the `fit` slot has to be adapted differently.

Interfacing of a Fitting Function that takes Design Matrix and Response Vector If we have a function that takes design matrix and response vector and the returned objects have the same object structure as glm objects, interfacing is rather straightforward and therefore recommended. Fitting happens with the `fit` function which in the above case is an interface to `glm.nb_fit` or generally to `foo_fit`. The main changes in adapting the functionality to allow a foo model to be fitted lies in this part

```
...
if (is.null(weights)) {
        z <- foo_fit(x = object@get("designMatrx''),
                     y = object@get("response")[,1], ...)
      }
    else{
        z <- foo_fit(x = object@get("designMatrix"),
                     y = object@get("response")[,1], weights = weights, ...)
      }
    class(z) <- c("fooModel","additionalClasses")
...
```

The function `foo_fit` takes a design matrix `x` and a response `y` and various other parameters. From the object returned by `dpp` the design matrix can be extracted via `object@get("designMatrix")` and the response `y` via `object@get("response")[,1]`. These just need to be passed on to `foo_fit`. Please note that the condition distinguishes between having supplied weights or not. For `mob` this is important as the model gets fitted recursively and flagging if an observation belongs to the current node is done by assigning them weights of 0 or 1. It is therefore encouraged to write the base fitting function `foo_fit` in such a way that it allows to have a `weights` argument passed (e.g. `foo_fit(x, y, weights,...)`).

In the `fit` slot of `negbinModel` from above, we can see how this would look like is we interface `glm.nb_fit`

```
R> negbinModel@fit

function (object, weights = NULL, trace = 0, ...)
{
    if (is.null(weights)) {
        z <- glm.nb_fit(object@get("designMatrix"),
                        object@get("response")[,1],
                        mustart = NULL, etastart = NULL,
                        control = glm.control(trace = trace),
                        intercept = all(object@get("designMatrix")[, 1] == 1),
                        ...)
    }
    else {
        z <- glm.nb_fit(object@get("designMatrix"),
                        object@get("response")[,1],
                        w = weights,
                        mustart = NULL, etastart = NULL,
                        control = glm.control(trace = trace),
                        intercept = all(object@get("designMatrix")[,1] == 1),
                        ...)
    }
    class(z) <- c("negbinModel", "negbin", "glm", "lm")
    z$offset <- 0
    z$contrasts <- attr(object@get("designMatrix"), "contrasts")
    z$terms <- attr(object@get("input"), "terms")
    z$predict_response <- function(newdata = NULL) {
        if (!is.null(newdata)) {
            penv <- new.env()
            object@set("input", data = newdata, env = penv)
            dm <- get("designMatrix", envir = penv, inherits = FALSE)
        }
        else {
```

```
            dm <- object@get("designMatrix")
        }
        pr <- z$family$linkinv(drop(dm %*% z$coef))
        return(pr)
    }
    z$addargs <- list(...)
    z$ModelEnv <- object
    z$trace <- trace
    z
}
```

Interfacing of a Fitting Function that takes a Formula Object If there is no fitting function that takes design matrix and response vector as arguments but needs a formula interface, e.g., `foo(formula, data,...)`, one can build the formula inside the `fit` slot of the StatModel. For this, one can use a function `extract_StatModelFormula`[6], which takes an object retrieved from `dpp` and rebuilds the formula

```
R> extractStatModelFormula <- function(object)
   {
     resnames <- names(object@get("response"))
     fmla <- as.formula(paste(resnames,
                              paste(as.character(object@formula$input),
                                    collapse="")))
     fmla
   }
```

The formula object returned from `extract_StatModelFormula` can then be passed to the main fitting body as

```
...
fmla <- extract_StatModelFormula(object)

if (is.null(weights)) {
        z <- foo(fmla, ...)
    }
    else {
        z <- foo(flma, weights = weights, ...)
    }
    class(z) <- c("fooModel", "additionalClasses")
...
```

Accordingly, we can write a new StatModel object for negative binomial regression that interfaces the `glm.nb` function directly, called `negbinModel2`, which only differs from `negbinModel` in the `fit` slot.

[6]For readability, it is an internal function that is only available to `negbinModel`, but can be defined outside the objects if there are a number of interfaces that use it.

```
R> negbinModel2@fit

function (object, weights = NULL, ...)
{
    dfx <- object@get("designMatrix")
    dfy <- object@get("response")
    df <- cbind(dfy, dfx)
    extractStatModelFormula <- function(object) {
        resnames <- names(object@get("response"))
        fmla <- as.formula(paste(resnames,
                          paste(as.character(object@formula$input),
                          collapse = "")))
        fmla
    }
    fmla <- extractStatModelFormula(object)
    if (is.null(weights)) {
        z <- glm.nb(fmla, data = df,
                    mustart = NULL, etastart = NULL,
                    control = glm.control(),
                    intercept = all(object@get("designMatrix")[,1] == 1),
                    ...)
    }
    else {
        z <- glm.nb(fmla, data = df,
                    w = weights,
                    mustart = NULL, etastart = NULL,
                    control = glm.control(),
                    intercept = all(object@get("designMatrix")[,1] == 1),
                    ...)
    }
    class(z) <- c("negbinModel", "negbin", "glm", "lm")
    z$offset <- 0
    z$contrasts <- attr(object@get("designMatrix"), "contrasts")
    z$terms <- attr(object@get("input"), "terms")
    z$predict_response <- function(newdata = NULL) {
        if (!is.null(newdata)) {
            penv <- new.env()
            object@set("input", data = newdata, env = penv)
            dm <- get("designMatrix", envir = penv, inherits = FALSE)
        }
        else {
            dm <- object@get("designMatrix")
        }
        pr <- z$family$linkinv(drop(dm %*% z$coef))
```

```
        return(pr)
    }
    z$addargs <- list(...)
    z$ModelEnv <- object
    z
}
```

Please note that within the `fit` function, I rebuild the data set as a data frame `df` by writing

```
R> dfx <- object@get("designMatrix")
R> dfy <- object@get("response")
R> df <- cbind(dfy,dfx)
```

and then call `glm.nb` thus

```
R> glm.nb(fmla,data=df)
```

to ensure that the variables are found in the correct environment, i.e., first searched for in the data frame `df`. Of course this comes at a cost if big data sets are processed. Also note that here the objects returned from `negbinmodel` or `negbinModel2` belong to class negbinModel. It is therefore useful to write S3 methods for different S3 generics that reproduce the functionality of S3 methods for glm or negbin.

Providing a `reweight` method

For the extension of `mob` to work, one additional function for the object returned from the StatModel has to be provided, namely a S3 method for the generic `reweight`. This has to do with the implementation of `mob`, which refits the model by supplying new weights to improve efficiency [Zeileis *et al.*, 2012a]. If the object returned from `fit(StatModel)` is of class bar this means to write `reweight.bar`. This is also why both the low-level fitting function and the StatModel should take a `weights` argument. The only thing `reweight` does is to call the `fit` method of the StatModel again with new weights. These weights are extracted with a method for the generic `weights` in the `mob` function. A `reweight` method for fooModel as described earlier only needs to define the `fit` method used for `fooModel` and to call it accordingly with the once before fitted model as argument. Hence the argument `object` is what gets returned after calling `fit(fooModel,dpp(fooModel,formula,data))`. The `reweight` method would then look something like this

```
R> reweight.fooModel <- function(object, weights, ...) {
     fit <- fooModel@fit
     do.call("fit", c(list(object = object$ModelEnv, weights = weights),
                      object$addargs))
   }
```

and continuing with the example for the negative binomial StatModel object from above, which is of class negbinModel, this is specifically

```
R> reweight.negbinModel <- function(object, weights, ...) {
     fit <- negbinModel@fit
     do.call("fit", c(list(object = object$ModelEnv, weights = weights),
                      object$addargs))
   }
```

This was the last brick in the wall and in principle the negative binomial MOB can be fitted now.

`predict` and other utilities

As mentioned the prediction function for possible `newdata` can be added to the `fit` slot and predictions can be extracted by calling the predict function in the `predict` slot. For example see `z$predict_response` from `negbinModel` above:

```
    z$predict_response <- function(newdata = NULL) {
        if (!is.null(newdata)) {
            penv <- new.env()
            object@set("input", data = newdata, env = penv)
            dm <- get("designMatrix", envir = penv, inherits = FALSE)
        }
        else {
            dm <- object@get("designMatrix")
        }
        pr <- z$family$linkinv(drop(dm %*% z$coef))
        return(pr)
    }
    z$addargs <- list(...)
    z$ModelEnv <- object
    z
}
```

It is good practise to have the predict function associated with the fitting function and to fit and predict in one go. This way the predict function is always associated with the returned object. In GLM-type models the predictions are calculated as $g^{-1}(\eta_i)$, and in many R functions a method for the generic `family` will be available that contains the inverse link function $g(\cdot)^{-1}$. Therefore, as in the `negbinModel` case, actually calculating predictions will reduce to the one-liner `pr <- z$family$linkinv(drop(dm %*% z$coef))`.

It is then prudent to write methods for negbinModel for standard generics as well

```
R> predict.negbinModel <- function(object, newdata = NULL, ...)
     object$predict_response(newdata = newdata)
```

```
R> fitted.negbinModel <- function(object, ...)
     object$predict_response()
R> print.negbinModel <- function(x, digits = max(3, getOption("digits") - 3),
                                 ...)
 {
     fam <- x$family$family
     substr(fam, 1, 1) <- toupper(substr(fam, 1, 1))
     cat(paste(fam, "GLM-type model with coefficients:\n"))
     print.default(format(coef(x), digits = digits),
                   print.gap = 2, quote = FALSE)
     invisible(x)
 }
R> summary.negbinModel <- function(object, dispersion = 1,
                                   correlation = FALSE, ...)
 {
     if(is.null(dispersion)) dispersion <- 1
     summ <- c(summary.glm(object, dispersion = dispersion,
                           correlation = correlation),
               object[c("theta", "SE.theta", "twologlik", "th.warn")])
     class(summ) <- c("summary.negbin", "summary.glm")
     summ
 }
R> model.matrix.negbinModel <- function(object, ...)
     object$ModelEnv@get("designMatrix")
```

Integration with `mob`

Once the StatModel and the `reweight` method have been written, integration with `mob` is straightforward. The necessary argument `model` must match the name of the StatModel object (`fooModel`), i.e. the function call is then

```
R> mob(formula, data, model=fooModel, ...)
```

Again, `...` stands for additional arguments to `mob` and particularly arguments to the fooModel fit function.

As proof-of-concept[7], I continue with a data set for the negative binomial example. Venables and Ripley provide the `quine` data set as an example. It contains records of children from Walgett, New South Wales, Australia, who were classified by culture, age, sex and learner status and their number of days absent from school in a particular school year was recorded. The response variable is the days absent ("Days"), which are overdispersed count data.

[7] A more thorough use of `mob` with `negbinModel` to answer a pertinent research question can be found in the next section.

```
R> library(MASS)
R> data(quine)
```

We can use recursive partitioning with a negative binomial model in the nodes to model the data. Since there is no *a priori* assumed node model, we can fit an intercept-only model in the nodes. The model is then a tree-structured analysis of deviance model in forward selection.

```
R> mob1 <- mob(Days ~ 1 | Sex + Age + Eth + Lrn, data = quine,
            model = negbinModel,
            control = mob_control(alpha = 0.05), na.action = NULL)
R> mob1

1) Eth == {A}; criterion = 0.997, statistic = 11.36
  2) Age == {F0, F1}; criterion = 0.962, statistic = 10.796
    3)*  weights = 33
Terminal node model
Negative Binomial(1.873) GLM-type model with coefficients:
(Intercept)
       2.74

  2) Age == {F2, F3}
    4)*  weights = 36
Terminal node model
Negative Binomial(1.5) GLM-type model with coefficients:
(Intercept)
       3.28

1) Eth == {N}
  5)*  weights = 77
Terminal node model
Negative Binomial(0.9186) GLM-type model with coefficients:
(Intercept)
        2.5
```

The resulting tree is visualised[8] in Figure 2.2.2.

We see that three partitions result. The importance of ethnicity and of age as well as their interaction is established. Sex plays no role whatsoever in the MOB model. We can look at a summary of the negative binomial distribution fitted in each partition. The mean μ_k of the distribution in partition k are given as `exp(coef(mob1))[k]` for $k = 1, 2, 3$ due to using the logarithm as the link function. The variance is $\mu_k + \mu_k^2/\theta_k$ (θ_k being the shape parameter).

[8] For intercept-only GLM for discrete data, there are panel-generating functions `node_ddistplot` to display the discrete distribution fitted in the node and `node_cdistplot` for some continuous distributions. They can be found in **mobtools**.

```
R> plot.BinaryTree(mob1, terminal_panel = node_ddistplot,
                   tp_args=list(dist="negbin"))
```

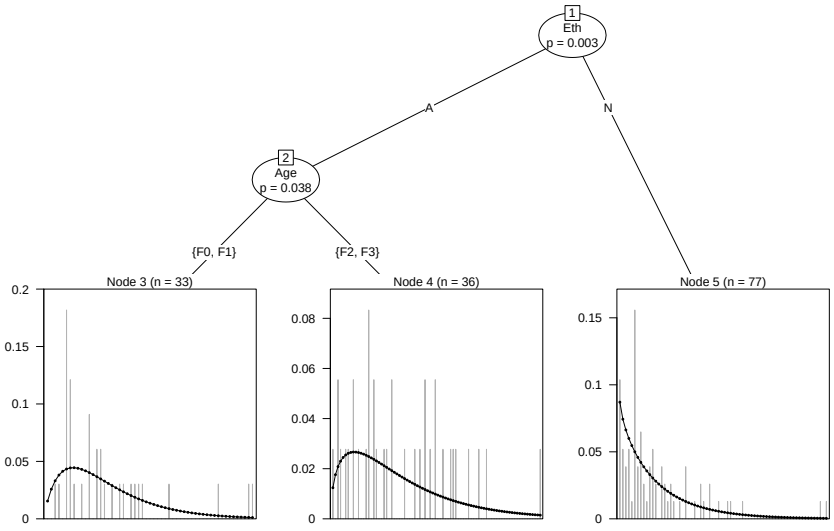

Figure 2.1.: The fitted negative binomial model tree for the quine data. The node model used was an intercept-only model.

```
R> summary(mob1)

$`3`

Call:
NULL

Deviance Residuals:
   Min      1Q  Median      3Q     Max
 -1.85    0.00    0.00    0.00    2.07

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.740      0.135    20.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(1.873) family taken to be 1)

    Null deviance: 34.69  on 32  degrees of freedom
Residual deviance: 34.69  on 32  degrees of freedom
AIC: 247.6

Number of Fisher Scoring iterations: 1


              Theta:  1.873
          Std. Err.:  0.487

 2 x log-likelihood:  -243.551

$`4`

Call:
NULL

Deviance Residuals:
   Min      1Q  Median      3Q     Max
 -2.96    0.00    0.00    0.00    1.64

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)     3.28       0.14    23.4   <2e-16 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(1.5) family taken to be 1)

    Null deviance: 40.92  on 35  degrees of freedom
Residual deviance: 40.92  on 35  degrees of freedom
AIC: 310.7

Number of Fisher Scoring iterations: 1


              Theta:  1.500
          Std. Err.:  0.366

 2 x log-likelihood:  -306.698

$`5`

Call:
NULL

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.210  -0.465   0.000   0.000   2.265

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.500      0.123    20.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(0.9186) family taken to be 1)

    Null deviance: 89.764  on 76  degrees of freedom
Residual deviance: 89.764  on 76  degrees of freedom
AIC: 548.9

Number of Fisher Scoring iterations: 1


              Theta:  0.919
          Std. Err.:  0.159
```

```
2 x log-likelihood:  -544.902
```

We see that the mean is lowest for the group `Eth=N`, with on average roughly 12.2 days of absence. The standard deviation is 13 days. The highest mean number of days absent, roughly 24.5, we find for children in the age groups `F2` and `F3` who are `Eth=A`. The standard deviation there is estimated to be 20.6 days.

2.2.3. Example: Effect of Private Insurance on the Demand for Health Care

Riphahn *et al.* [2003] investigated determinants for health care demands. One of the *foci* of their investigation lies in whether the choice of an add-on private health care insurance is positively related to the number of uses of the health care system. This question relates to two effects postulated in literature: The first being "adverse selection" , which means that people who expect rising health care expenditures in the future will purchase an add-on insurance and second the "moral hazard" phenomenon, which means that having a private or add-on insurance is associated with a more frequent use of the health care system. The "moral hazard" theory is derived within the model of Cameron *et al.* [1988] by assuming that if a price of a medical service is lower, it will be demanded more and therefore that there will be a higher demand of medical services for policies that are more generous (such as private or add-on insurances).

To investigate both effects, Riphahn *et al.* [2003] used a part of the German Socioeconomic Panel (GSOEP) to analyse the number of visits to a doctor in one of the last three months prior to the survey (`docvis`) and the number of visits to the hospital within a given calender year (`hospvis`), conditional on a number of sociodemographic variables. The authors chose to analyse the data with sophisticated uni- and bivariate random effects lognormal-Poisson models. Greene [2008] revisited the data and reanalysed them with two versions of a negative binomial count data model and compared their performance with a Poisson fixed effects model. For `docvis` neither Riphahn *et al.* [2003] nor Greene [2008] find a significant influence of add-on insurance on the number of doctor visits for males and females, conditional on the other variables. However, Riphahn *et al.* [2003] find a significant influence of add-on insurance on the number of visits to the hospital for males.

Regarding adverse selection, the significant effect for males and the fact that the non-significant point estimates for `docvis` and `hospvis` for females are positive, let Riphahn *et al.* [2003] conclude that the presence of adverse selection cannot be ruled out. Regarding moral hazard, their primary investigation of an influence of private insurance on the `docvis` and `hospvis` turns out non-significant. Nevertheless, the significant effect of having add-on insurance on `hospvis` in the male subsample keeps the authors from dismissing the existence of moral hazard with a similar argument as for adverse selection. In his negative binomial model based analyses, Greene [2008, 2007] does not discuss `hospvis` and states that "analysis of the count of hospital visits is left for further research" [Greene, 2007, p.30]. To pick up this ball and because of the importance of the results of modelling `hospvis` in the male subsample for the conclusions of Riphahn *et al.* [2003], I will reanalyse the hospital visits in the same fashion as Greene [2008] did for

`docvis` and extend the analysis to zero-inflated models as well as use a tree-structured negative binomial model as introduced in the previous section.

The data set[9] used by Riphahn *et al.* [2003], Greene [2008] is a panel data set from an unbalanced panel of 7293 families. The families were observed at one or more (up to seven) of the following years 1984, 1985, 1986, 1987, 1988, 1991, 1994. The overall number of observations is 27326. Riphahn *et al.* [2003] as well as Greene [2008] used 15 explanatory variables, `age` and its square in years (this is divided by 10 for numerical stability), health satisfaction (`hsat`, scale 1:worst-10:best), handicapped (`handdum`, yes=1, else=0), degree of handicap (`handper`, percent 0-100), `married` (yes=1, no=0), years of schooling (`educ`), net household income (`hhninc`, German Mark/1000), children under 16 in household (`hhkids`, yes=1, else=0), self-employed (`self`, yes=1, else=0), civil servant (`beamt`, yes=1, else=0), blue collar employee (`bluec`, yes=1, else=0), employed (`working`, yes=1, else=0), insured in public healthcare (`public`, yes=1, else=0) and if the person has an add-on private insurance (`addon`, yes=1, else=0). Riphahn *et al.* [2003] used random effects to account for the panel structure, whereas Greene [2008] included fixed dummy effects for the years 1985, 1986, 1987, 1988, 1991 and 1994. Both analyses split the sample according to gender and only the male subsample proves relevant for the conclusions in both sudies. Accordingly, I will only present results for males as well and almost exclusively focus on the estimated conditional effect of `addon`. To analyse the `hospvis` data in the fashion of Greene [2008], I first fit the fixed effects Poisson model to `hospvis`:

```
R> pois1 <- glm(hospvis ~ I(age/10) + I(age^2/10) + hsat + handdum +
               handper + married + educ + I(hhninc/1000) + hhkids +
               self + beamt + bluec + working + public + addon + year,
               data = maldat, family=poisson)
R> summary(pois1)

Call:
glm(formula = hospvis ~ I(age/10) + I(age^2/10) + hsat + handdum +
    handper + married + educ + I(hhninc/1000) + hhkids + self +
    beamt + bluec + working + public + addon + year, family = poisson,
    data = maldat)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.510  -0.527  -0.410  -0.334  18.524

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.13610    0.46647    0.29  0.77047
I(age/10)       0.10915    0.19982    0.55  0.58490
I(age^2/10)    -0.00115    0.00228   -0.50  0.61390
```

[9] `http://qed.econ.queensu.ca/jae/2003-v18.4/riphahn-wambach-million/`

```
hsat           -0.24128    0.00973  -24.81  < 2e-16 ***
handdum1       -0.04410    0.07759   -0.57  0.56982
handper         0.00339    0.00131    2.58  0.00976 **
married1       -0.04979    0.06710   -0.74  0.45810
educ           -0.08374    0.01335   -6.27  3.6e-10 ***
I(hhninc/1000)  0.03206    0.01518    2.11  0.03463 *
hhkids1         0.09653    0.05815    1.66  0.09692 .
self1          -0.02716    0.09721   -0.28  0.77992
beamt1         -0.08071    0.11160   -0.72  0.46956
bluec1          0.03789    0.06432    0.59  0.55578
working1       -0.09817    0.08045   -1.22  0.22235
public1        -0.14486    0.10235   -1.42  0.15696
addon1          0.57455    0.15580    3.69  0.00023 ***
year1985        0.43391    0.08292    5.23  1.7e-07 ***
year1986       -0.07402    0.09377   -0.79  0.42987
year1987        0.11176    0.09708    1.15  0.24965
year1988       -0.03154    0.08824   -0.36  0.72077
year1991       -0.14934    0.09324   -1.60  0.10923
year1994        0.06982    0.09358    0.75  0.45563
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11049  on 14242  degrees of freedom
Residual deviance: 10094  on 14221  degrees of freedom
AIC: 12579

Number of Fisher Scoring iterations: 6
```

Note that the pure effect for `addon` is estimated on the log scale as 0.575 with an associated standard error of 0.156. This corroborates the result of Riphahn *et al.* [2003] as this effect is significant at the 0.05 level. To investigate if the extra-Poisson variability leads to a vanishing effect, as Greene [2008] claims, I fit a negative binomial in the same fashion as in Greene [2008] (coined NB 2 there), which is

```
R> nb1 <- glm.nb(hospvis ~ I(age/10) + I(age^2/10) + hsat + handdum +
               handper + married + educ + I(hhninc/1000) + hhkids +
               self + beamt + bluec + working + public + addon + year,
               data = maldat)
R> summary(nb1)

Call:
glm.nb(formula = hospvis ~ I(age/10) + I(age^2/10) + hsat + handdum +
```

```
    handper + married + educ + I(hhninc/1000) + hhkids + self +
    beamt + bluec + working + public + addon + year, data = maldat,
    init.theta = 0.1267770066, link = log)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.790  -0.432  -0.359  -0.303   6.273

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.225996   0.691326    0.33   0.7437
I(age/10)       0.075133   0.302388    0.25   0.8038
I(age^2/10)    -0.000577   0.003475   -0.17   0.8681
hsat           -0.230792   0.015967  -14.45  < 2e-16 ***
handdum1       -0.052289   0.128638   -0.41   0.6844
handper         0.005677   0.002257    2.51   0.0119 *
married1        0.005911   0.101507    0.06   0.9536
educ           -0.090729   0.018803   -4.83  1.4e-06 ***
I(hhninc/1000)  0.015457   0.023580    0.66   0.5121
hhkids1         0.028820   0.087707    0.33   0.7425
self1           0.041655   0.139863    0.30   0.7658
beamt1         -0.142186   0.161429   -0.88   0.3784
bluec1          0.016404   0.094114    0.17   0.8616
working1       -0.071609   0.124994   -0.57   0.5667
public1        -0.134255   0.147923   -0.91   0.3641
addon1          0.661166   0.246912    2.68   0.0074 **
year1985        0.275955   0.130078    2.12   0.0339 *
year1986       -0.062711   0.135567   -0.46   0.6437
year1987        0.109404   0.152863    0.72   0.4742
year1988       -0.088166   0.130625   -0.67   0.4997
year1991       -0.217806   0.135947   -1.60   0.1091
year1994        0.023815   0.140837    0.17   0.8657
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(0.1268) family taken to be 1)

    Null deviance: 4086.6  on 14242  degrees of freedom
Residual deviance: 3655.8  on 14221  degrees of freedom
AIC: 9635

Number of Fisher Scoring iterations: 1
```

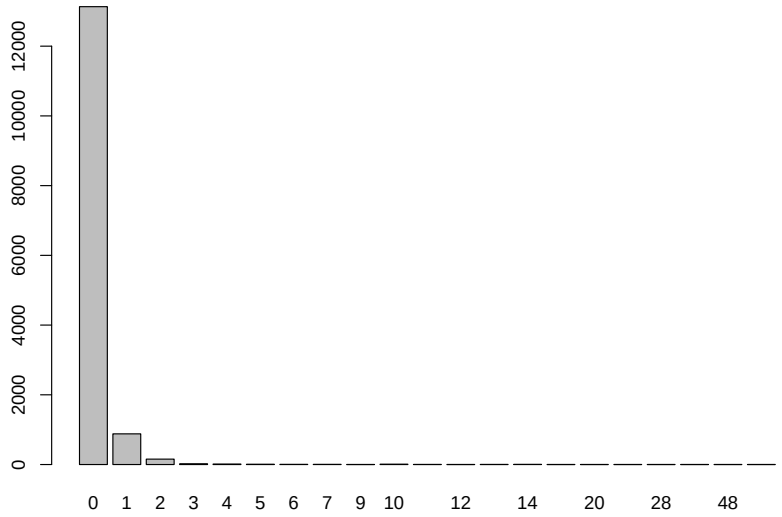

Figure 2.2.: The frequency of the number of hospital visits in the last year (`hospvis`).

```
          Theta:  0.12678
    Std. Err.:  0.00719

 2 x log-likelihood:  -9589.27600
```

Note that the point estimate on the log-scale for `addon` is now bigger, 0.661, with a standard error of 0.248. This increase may be due to chance, scale differences of the GLM or the more complicated estimation of the negative binomial model. One way or the other, the effect of `addon` is still significant, once again reinforcing the results of Riphahn *et al.* [2003].

Looking at the distribution of the dependent variable for males in Figure 2.2.3, it appears as if there is an unusual high amount of zeros. This can have an influence on the fit and parameter estimation of a Poisson or negative binomial model, as well as their bivariate or random effect counterparts. Hence, I will use a zero-inflated negative binomial to account for the excess number of zeros:

```
R> zi1 <- zeroinfl(hospvis ~ I(age/10) + I(age^2/10) + hsat + handdum +
                   handper + married + educ + I(hhninc/1000) + hhkids +
                   self + beamt + bluec + working + public + addon + year,
                   data = maldat, dist="negbin")
R> summary(zi1)
```

```
Call:
zeroinfl(formula = hospvis ~ I(age/10) + I(age^2/10) + hsat + handdum +
    handper + married + educ + I(hhninc/1000) + hhkids + self + beamt +
    bluec + working + public + addon + year, data = maldat, dist = "negbin")

Pearson residuals:
   Min     1Q Median     3Q    Max
-0.420 -0.257 -0.209 -0.171 53.616

Count model coefficients (negbin with log link):
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.945345   1.040409   -1.87  0.06151 .
I(age/10)       0.848870   0.473733    1.79  0.07315 .
I(age^2/10)    -0.010035   0.005433   -1.85  0.06474 .
hsat           -0.154483   0.020377   -7.58  3.4e-14 ***
handdum1       -0.009957   0.162981   -0.06  0.95129
handper         0.000483   0.002691    0.18  0.85755
married1        0.248767   0.150785    1.65  0.09898 .
educ           -0.073790   0.028972   -2.55  0.01087 *
I(hhninc/1000)  0.004502   0.029210    0.15  0.87752
hhkids1        -0.023109   0.136645   -0.17  0.86571
self1           0.400352   0.258649    1.55  0.12166
beamt1         -0.117652   0.280832   -0.42  0.67526
bluec1         -0.213340   0.152770   -1.40  0.16257
working1        0.087713   0.202143    0.43  0.66435
public1         0.154205   0.244602    0.63  0.52841
addon1          0.419948   0.365146    1.15  0.25011
year1985        0.689605   0.187390    3.68  0.00023 ***
year1986        0.106371   0.198168    0.54  0.59143
year1987        0.418806   0.203353    2.06  0.03945 *
year1988        0.145340   0.186462    0.78  0.43571
year1991        0.018572   0.200186    0.09  0.92608
year1994        0.181726   0.203273    0.89  0.37132
Log(theta)     -1.524973   0.111613  -13.66  < 2e-16 ***

Zero-inflation model coefficients (binomial with logit link):
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     -8.4796     2.1620   -3.92  8.8e-05 ***
I(age/10)        2.4689     1.0140    2.43    0.015 *
I(age^2/10)     -0.0298     0.0119   -2.50    0.012 *
hsat             0.2316     0.0418    5.54  3.0e-08 ***
handdum1         0.1939     0.7146    0.27    0.786
handper         -0.0345     0.0197   -1.75    0.080 .
```

```
married1         0.7120     0.3650    1.95    0.051 .
educ             0.0256     0.0538    0.48    0.634
I(hhninc/1000)  -0.0495     0.0538   -0.92    0.358
hhkids1         -0.2005     0.2570   -0.78    0.435
self1            0.6671     0.3832    1.74    0.082 .
beamt1          -0.0309     0.5344   -0.06    0.954
bluec1          -0.6035     0.2977   -2.03    0.043 *
working1         0.4705     0.5276    0.89    0.373
public1          0.7269     0.4815    1.51    0.131
addon1          -0.4719     0.6407   -0.74    0.461
year1985         1.0037     0.4405    2.28    0.023 *
year1986         0.5386     0.4732    1.14    0.255
year1987         0.9337     0.6861    1.36    0.174
year1988         0.5666     0.4563    1.24    0.214
year1991         0.5653     0.4951    1.14    0.254
year1994         0.4409     0.5042    0.87    0.382
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Theta = 0.218
Number of iterations in BFGS optimization: 69
Log-likelihood: -4.75e+03 on 45 Df
```

What is most interesting is that for the zero-inflated negative binomial model, the effect of `addon` decreases and can now not be assumed to be significantly different from zero at the 0.05 level. To assess adequacy of the models, one can compare their fit with Akaike's Information Criterion [AIC; Akaike, 1974] and the Bayesian Information Criterion [BIC; Schwartz, 1978]. Here, it is not clear if the negative binomial model and or its zero-inflated counterpart is more suitable for the data. AIC favors the zero-inflated negative binomial model

```
R> AIC(pois1,nb1,zi1)

      df   AIC
pois1 22 12579
nb1   23  9635
zi1   45  9592
```

and BIC the standard negative binomial model.

```
R> AIC(pois1,nb1,zi1,k=log(dim(maldat)[1]))

      df   AIC
pois1 22 12746
nb1   23  9809
zi1   45  9933
```

It might be that a negative binomial model (either with or without excess zeros) is still too restrictive for a data set of this complexity. There may very well be still more heterogeneity present that was overlooked by both negative binomial models and the Poisson model. Additionally, all three models above and the analyses after the fashion of Riphahn *et al.* [2003] and Greene [2008] used an additive model for the relationship between the counts and the explanatory variables. This means interaction patterns have been largely ignored.

As interactions are automatically discovered in partition based models and because they can learn additional heterogeneity if present in the data, one can use a negative binomial model tree to see if and how the relationship between `addon` and the number of hospital visits is mediated by the other explanatory variables and if there are subgroups in the sample of males that show differential behaviour (which basically means additional heterogeneity). To see this, I use a negative binomial node model where `addon` is used as a categorical predictor for the counts of `hospvis`. All other variables, including year, are used for partitioning.

```
R> mob1 <- mob(hospvis ~ addon | year + age + hsat + handdum + handper +
              hhninc + hhkids + educ + married + working + bluec + self +
              public + beamt, data = maldat, model = negbinModel, trace=1,
              control = mob_control(alpha=0.01,minsplit=50,verbose=TRUE))
```

A visualisation of the tree can be found in Figure 2.2.3. We see that there are nontrivial interactions between the explanatory variables and the simple negative binomial model of `hospvis` explained by `addon`. With 8 partitions with different mean and shape parameters, the presence of additional heterogeneity is obvious. Most importantly, in light of the question as to whether there is evidence for "moral hazard" or "adverse selection" for hospital visits in the male subset, it cannot generally be concluded to be the case. In the tree model, there is only a single significant difference of the number of hospital visits between those with an add-on insurance and those without and it occurs for the subgroup in node 13. We can check the magnitude of the effect of `addon` on the original scale for each partition.

```
R> round(exp(coef(mob0.01)),digits=2)
```

```
   (Intercept) addon1
3         0.62   0.00
6         0.49   1.15
8         0.18   1.03
9         0.34     NA
12        0.22   0.00
13        0.08   4.17
14        0.44   0.46
15        0.08   1.87
```

We see that in node 13, the estimated effect is 4.17. There also is a possibly substantial

```
R> plot(mob0.01,terminal_panel=node_bivplotscaled(mob0.01,func=sqrt))
```

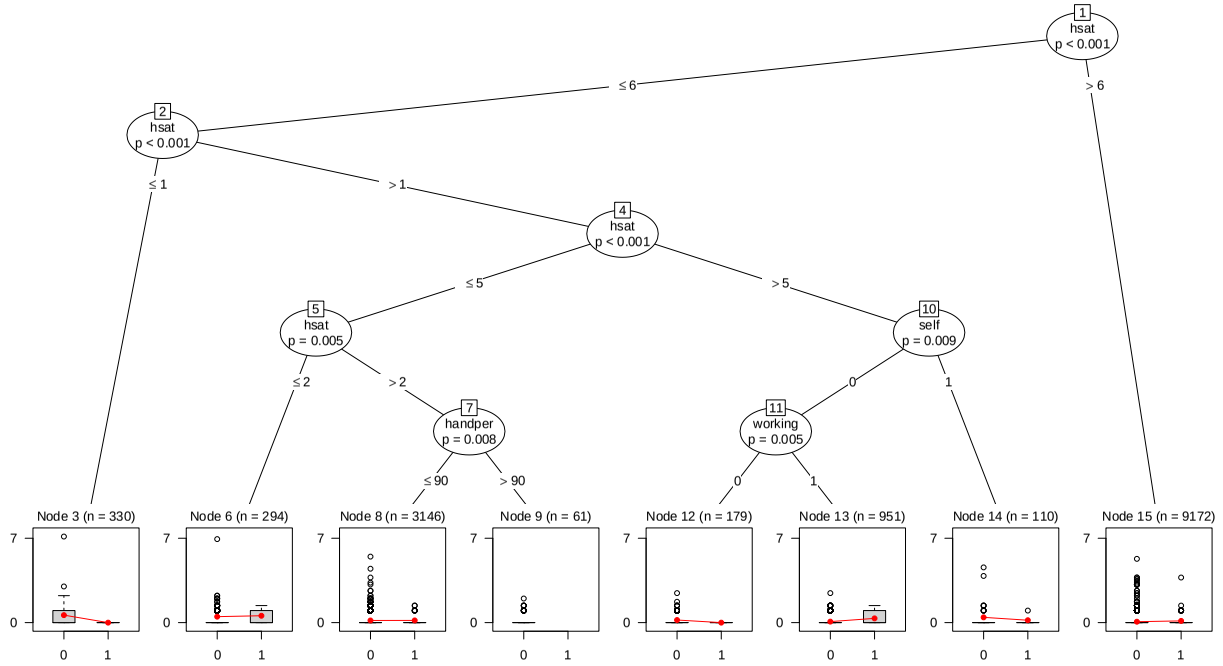

Figure 2.3.: The tree-structured negative binomial model with `addon` as predictor in the nodes. The significance value used for the parameter stability tests is 0.01. The y-axis of the node plots is on the square root scale for better legibility of the plots.

effect estimated as 1.87 in node 15, but it is not big enough to be regarded as significant (on the log scale point estimate and standard error are 0.62 and 0.38).

Let us now take a closer look at segment 13, the segment with the significant effect.

```
R> summary(mob0.01,node=13)

Call:
NULL

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.684   0.000   0.000   0.000   4.001

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.470      0.127  -19.42   <2e-16 ***
addon1         1.429      0.613    2.33     0.02 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.304) family taken to be 1)

    Null deviance: 295.96  on 950  degrees of freedom
Residual deviance: 290.74  on 949  degrees of freedom
AIC: 580.8

Number of Fisher Scoring iterations: 1


              Theta:  0.304
          Std. Err.:  0.115

 2 x log-likelihood:  -574.829
```

Node 13 consists of 951 males. The men in this partition have in common that they rate their health satisfaction with 6, are working but are not self-employed. For this subgroup, those with an add-on insurance visit hospitals significantly and substantially more often than their non-privately insured counterparts. It is the only segment with significant differences between men with add-on insurance and men without. This means overall there seems to rarely exist a significant positive association between `hospvis` and `addon`, with the exception of a single group. We can check this by assessing parameter stability only for the parameter for `addon` and not over the intercept as well.

```
R> mob(hospvis ~ addon | year + age + hsat + handdum + handper +
     hhninc + hhkids + educ + married + working + bluec + self +
```

```
     public + beamt, data = maldat, model = negbinModel,
     control = mob_control(alpha=0.1,parm=2,verbose=TRUE))

----------------------------------------
Fluctuation tests of splitting variables:
           year   age  hsat handdum handper hhninc hhkids  educ

statistic 6.808 2.246 1.165  1.3944   3.329  3.408 2.0196 3.447
p.value   0.997 1.000 1.000  0.9776   1.000  1.000 0.9058 1.000

          married working  bluec   self  public beamt

statistic 0.03831  0.4235 0.1041 0.6078 0.02196 6.405
p.value   1.00000  1.0000 1.0000 0.9997 1.00000 0.148

Best splitting variable: beamt
Perform split? no
----------------------------------------
1)*  weights = 14243
Terminal node model
Negative Binomial(0.0913) GLM-type model with coefficients:
(Intercept)       addon1
     -2.064        0.327
```

Not surprisingly, no instability is detected by using a stability test significance level of less than 0.1 and the resulting tree is just a root node. Therefore, partitioning in the other tree is mostly driven by differences in the intercept, i.e., the mean number of visits for people with no add-on insurance.

Based on the partitioned negative binomial model for the male subgroup, we can conclude that similar to what Greene [2008, 2007] and Riphahn *et al.* [2003] showed for `docvis` and males and `docvis` and `hospvis` for females, there seems to be mostly no significant effects at the 0.05 level of having add-on insurance on the number of hospital visits, except for a single segment. This is in contrast to the results obtained by using fixed effect Poisson or negative binomial models, where overall there is a significant effect of `addon`. The partitioned model suggests that a possible existence of a "moral hazard" or "adverse selection" as detected by the nonpartitioned models holds only for a very specific group. This adds an interesting additional interpretation to the findings of Riphahn *et al.* [2003] and Greene [2008]: The significant effect of `addon` found for males that made them hesitate to conclude that there is no "moral hazard" or "adverse selection", can largely be explained by using a zero-inflated model or by heterogeneity and interactions in a partitioned model. The partitioned model allows to identify the group that may have led to the overall significant effect of `addon` in the analysis by Riphahn *et al.* [2003].

We looked at four different types of fixed effect models and the results were partly contradictory. To assess adequacy of the models by using information criteria, the results are unambiguous as to what is the best model:

```
R> AIC(pois1,nb1,zi1,mob0.01)

         df   AIC
pois1    22 12579
nb1      23  9635
zi1      45  9592
mob0.01 30  9570

R> AIC(pois1,nb1,zi1,mob0.01,k=log(dim(maldat)[1]))

         df   AIC
pois1    22 12746
nb1      23  9809
zi1      45  9933
mob0.01 30  9796
```

Clearly, the Poisson fixed effects model is out of the question according to both criteria. There is substantial extra-Poisson heterogeneity present as well as substantial excess zeros. Both criteria favour the tree-structured model. Even though it does not allow for excess zeros, it appears to explain the data well and at the same time stays parsimonious, especially compared to the zero-inflated model[10]. The tree model does account for additional heterogeneity and because of fitting negative binomial models to subsets of the data manages to alleviate the excess zero problem of a global negative binomial model. Of course, the latter it does less well than the zero-inflated model, but that in turn has problems with the additional heterogeneity. Regarding the question of "moral hazard" or "adverse selection", both models - the zero-inflated negative binomial model and the partitioned negative binomial model - are unanimous: The data do not support a general significant effect of `addon` on `hospvis`, even for males.

[10]The zero-inflated model could be made more parsimonious by using less predictors for the zero-inflated component. That is left for further research.

Bibliography

Ahn H (1994a). "Tree-structured Exponential Regression Modeling." *Biometrical Journal*, **36**, 43–61.

Ahn H (1994b). "Tree-structured Extreme Value Model Regression." *Communications in Statistics - Theory and Methods*, **23**, 153–174.

Ahn H (1996a). "Log-gamma Regression Modeling Through Regression Trees." *Communications in Statistics - Theory and Methods*, **25**, 295–311.

Ahn H (1996b). "Log-normal Regression Modeling Through Recursive Partitioning." *Computational Statistics & Data Analysis*, **21**, 381–398.

Ahn H, Chen J (1997). "Tree-structured Logistic Model for Over-dispersed Binomial Data with Application to Modeling Developmental Effects." *Biometrics*, **53**, 435–455.

Ahn H, Loh W (1994). "Tree-Structured Proportional Hazards Regression Modeling." *Biometrics*, **50**, 471–485.

Aitkin M, Francis B, Hinde J, Darnell R (2009). *Statistical Modelling in R*. Oxford University Press, Inc., New York.

Akaike H (1974). "A new look at the statistical model identification." *IEEE Transactions on Automatic Control*, **19**(6), 716–723.

Blei D, Jordan M, Ng A (2003). "Latent Dirichlet Allocation." *The Journal of Machine Learning*, **3**, 993–1022.

Breiman L, Friedman J, Olshen R, Stone C (1984). *Classification and Regression Trees*. Wadsworth, California.

Cameron A, Trivedi T, Milne F Piggot J (1988). "A Microeconomic Model for the Demand of Health Care and Health Insurance in Australia." *Review of Economic Studies*, **55**, 85–106.

Chambers J, Hastie T (1992). *Statistical Models in S*. Chapman & Hall, London.

Chan K, Loh W (2005). "LOTUS. An Algorithm for Building Accurate and Comprehensible Logistic Regression Trees." *Journal of Computational and Graphical Statistics*, **13**, 826–852.

Chaudhuri P, Huang M, Loh W, Yao R (1994). "Piecewise-polynomial Regression Trees." *Statistica Sinica*, **4**, 143–167.

Chaudhuri P, Lo W, Loh W, Yang C (1995). "Generalized Regression Trees." *Statistica Sinica*, **5**, 641–666.

Chaudhuri P, Loh W (2002). "Nonparametric Estimation of Conditional Quantiles using Quantile Regression Trees." *Bernoulli*, **8**, 561–576.

Choi Y, Ahn H, Chen J (2005). "Regression Trees for Analysis of Count Data with Extra Poisson Variation." *Computational Statistics & Data Analysis*, **49**, 893–915.

Clarke B, Fokoue E, Zhang H (2009). *Principles and Theory of Data Mining and Machine Learning*. Springer, New York.

Efron B, Tibshirani R (1994). *An Introduction to the Bootstrap*. Monographs on Statistics & Applied Probability. Chapman & Hall/CRC, London.

Friedman J, Hastie T, Tibshirani R (2000). "Additive Logistic Regression: A Statistical View of Boosting." *The Annals of Statistics*, **38**(2), 337–374.

Gama J (2004). "Functional Trees." *Machine Learning*, **55**, 219–250.

Greene W (2007). "Functional Form and Heterogeneity in Models for Count Data." *Working Paper 07–10*, Leonard N. Stern Economics Working Papers, Department of Economics, Leonard N. Stern School of Business, New York University, New York. URL `http://ssrn.com/abstract=986620`.

Greene W (2008). "Functional Forms for the Negative Binomial Model for Count Data." *Economics Letters*, **99**, 585–590.

Hardin J, Hilbe J (2007). *Generalized Linear Models and Extensions*. 2nd edition. Stata Press, College Station, Texas.

Hastie T, Tibshirani R, Friedman J (2009). *Elements of Statistical Learning*. 2nd edition. Springer, New York.

Hothorn T, Hornik K, Strobl C, Zeileis A (2012a). *party: A laboratory for recursive partitioning*. R package version 1.0-1, URL `http://CRAN.R-project.org/package=party`.

Hothorn T, Leisch F, Zeileis A (2012b). *modeltools: Tools and Classes for Statistical Models*. R package version 0.2-19, URL `http://CRAN.R-project.org/package=modeltools`.

Hothorn T, Leisch F, Zeileis A (2012c). *StatModel-class: Class "StatModel"*. Modeltools 0.2-19 Documentation.

Hunt E, Marin J, Stone P (1966). *Experiments in Induction*. Academic Press, New York.

Kim H, Loh W (2001). "Classification Trees with Unbiased Multiway Splits." *Journal of the American Statistical Association*, **96**, 589–604.

Landwehr N, Hall M, Eibe F (2005). "Logistic Model Trees." *Machine Learning*, **59**, 161–205.

Lee S (2005). "On Generalized Multivariate Decision Trees by using GEE." *Computational Statistics & Data Analysis*, **49**, 1105–1119.

Loh W (2002). "Regression Trees with Unbiased Variable Selection and Interaction Detection." *Statistica Sinica*, **12**, 361–386.

Loh W (2008). *Regression by Parts: Fitting Visually Interpretable Models with GUIDE*, volume 3 of *Handbook of Computational Statistics*, pp. 447–469. Springer, New York.

Loh W (2009). "Improving the Precision of Classification Trees." *The Annals of Applied Statistics*, **3**, 1710–1737.

Loh W, Shih Y (1997). "Split Selection Methods for Classification Trees." *Statistica Sinica*, **7**, 815–840.

Loh W, Zheng W (2012). "Regression Trees for Longitudinal and Multiresponse Data." *The Annals of Applied Statistics*. Forthcoming.

Malchow H (2008). *Political Targeting*. 2nd edition. Predicted Lists, LLC.

McCullagh P, Nelder J (1989). *Generalized Linear Models*. 2nd edition. Chapman & Hall, London.

Morgan J, Sonquist J (1968). "Problems in the Analysis of Survey Data, and a Proposal." *Journal of the American Statistical Association*, **58**, 415–434.

Murthy S (1997). "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey." *Data Mining and Knowledge Discovery*, **2**, 345–389.

Potts D, Sammut C (2005). "Incremental Learning of Linear Model Trees." *Machine Learning*, **61**, 5–48.

Quinlan R (1992). "Learning with Continuous Classes." In *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pp. 343–348. World Scientific, Singapore.

Quinlan R (1993). *C 4.5: Programs for Machine Learning*. Morgan Kaufmann Publ., San Mateo, California.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org/`.

Riphahn R, Wambach A, Million A (2003). "Incentive Effects in the Demand for Health Care: A Bivariate Panel Count Data Estimation." *Journal of Applied Econometrics*, **18**(4), 387–405.

Ripley B, Hornik K, Gebhardt A, Firth D (2012). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-17, URL `http://CRAN.R-project.org/package=MASS`.

Rusch T, Hofmarcher P, Hatzinger R, Hornik K (2011). "Modeling Mortality Rates In The WikiLeaks Afghanistan War Logs." *Report 112*, Research Report Series, Institute for Statistics and Mathematics, WU (Vienna University of Economics and Business), Vienna. URL `http://epub.wu.ac.at/id/eprint/3210`.

Rusch T, Hofmarcher P, Hatzinger R, Hornik K (2013). "Model trees with topic model pre-processing: An approach for data journalism illustrated with the WikiLeaks Afghanistan war logs." *The Annals of Applied Statistics*. Forthcoming.

Rusch T, Lee I, Hornik K, Jank W, Zeileis A (2012a). "Influencing Elections with Statistics: Targeting Voters with Logistic Regression Trees." *Report 117*, Research Report Series, Institute for Statistics and Mathematics, WU (Vienna University of Economics and Business), Vienna. URL `http://epub.wu.ac.at/3458/`.

Rusch T, Zeileis A (2013). "Gaining Insight with Recursive Partitioning of Generalized Linear Models." *Journal of Statistical Computation and Simulation*. doi:10.1080/00949655.2012.658804. Forthcoming, URL `http://eeecon.uibk.ac.at/~zeileis/papers/Rusch+Zeileis-2012.pdf`.

Rusch T, Zeileis A, Hothorn T, Leisch F (2012b). *mobtools: A collection of StatModels and of utilities for extending mob.* R package version 0.0-1.

Schober C, Rusch T (2010). "Studie zur Messung der Wirkung der Schuldnerberatung in Österreich auf Gläubiger - Zusatzauswertungen zu einem bestehenden Bericht [Study to Measure the Effect of Debt Advisory Service in Austria on Debtors - Complementary Analyses of an Existing Report]." Wien: NPO-Institut, WU (Wirtschaftsuniversität Wien).

Schwartz G (1978). "Estimating the Dimensions of a Model." *Annals of Statistics*, **6**, 461–464.

Sela R, Simonoff J (2012). "RE-EM Trees: A Data Mining Approach for Longitudinal and Clustered Data." *Machine Learning*, **86**, 169–207. 10.1007/s10994-011-5258-3, URL `http://dx.doi.org/10.1007/s10994-011-5258-3`.

Strobl C, Kopf J, Zeileis A (2010). "A New Method for Detecting Differential Item Functioning in the Rasch Model." *Technical Report 92*, Department of Statistics, Ludwig-Maximilians-Universität München. URL `http://epub.ub.uni-muenchen.de/11915/`.

Strobl C, Wickelmaier F, Zeileis A (2011). "Accounting for Individual Differences in Bradley-Terry Models by Means of Recursive Partitioning." *Journal of Educational and Behavioral Statistics*, **36**(2), 135–153. doi:10.3102/1076998609359791.

Su X, Wang M, Fan J (2004). "Maximum Likelihood Regression Trees." *Journal of Computational and Graphical Statistics*, **13**, 586–598.

Venables W, Ripley B (2002). *Modern Applied Statistics with S*. 4th edition. Springer, New York. ISBN 0-387-95457-0, URL `http://www.stats.ox.ac.uk/pub/MASS4/`.

Wang Y, Witten I (1997). "Induction of Model Trees for Predicting Continuous Classes." In *Proceedings of the Posters of the European Conference on Machine Learning*. University of Economics, Faculty of Informatics and Statistics, Prague, Czech Republic.

Zeileis A (2006). "Object-oriented Computation of Sandwich Estimators." *Journal of Statistical Software*, **16**(9), 1–16. URL `http://www.jstatsoft.org/v16/i09`.

Zeileis A, Croissant Y (2010). "Extended Model Formulas in R: Multiple Parts and Multiple Responses." *Journal of Statistical Software*, **34**(1), 1–13. URL `http://www.jstatsoft.org/v34/i01/`.

Zeileis A, Hornik K (2007). "Generalized M-Fluctuation Tests for Parameter Instability." *Statistica Neerlandica*, **61**(4), 488–508.

Zeileis A, Hothorn T, Hornik K (2008). "Model-Based Recursive Partitioning." *Journal of Computational and Graphical Statistics*, **17**(2), 492–514.

Zeileis A, Hothorn T, Hornik K (2012a). *Party with the mob*. R package vignette (version 1.0-1).

Zeileis A, Strobl C, Wickelmaier F, Kopf J (2012b). *psychotree: Recursive Partitioning Based on Psychometric Models*. R package version 0.12-1, URL `http://CRAN.R-project.org/package=psychotree`.

Zhang H, Singer BH (2010). *Recursive Partitioning and Applications*. 2nd edition. Springer, New York.

Part II.

Collected Research Articles

3. Gaining Insight with Recursive Partitioning of Generalized Linear Models

Gaining Insight with Recursive Partitioning of Generalized Linear Models

Thomas Rusch
WU Wirtschaftsuniversität Wien

Achim Zeileis
Universität Innsbruck

Abstract

Recursive partitioning algorithms separate a feature space into a set of disjoint rectangles. Then, usually, a constant in every partition is fitted. While this is a simple and intuitive approach, it may still lack interpretability as to how a specific relationship between dependent and independent variables may look. Or it may be that a certain model is assumed or of interest and there is a number of candidate variables that may non-linearly give rise to different model parameter values. We present an approach that combines generalized linear models with recursive partitioning that offers enhanced interpretability of classical trees as well as providing an explorative way to assess a candidate variable's influence on a parametric model. This method conducts recursive partitioning of a generalized linear model by (1) fitting the model to the data set, (2) testing for parameter instability over a set of partitioning variables, (3) splitting the data set with respect to the variable associated with the highest instability. The outcome is a tree where each terminal node is associated with a generalized linear model. We will show the method's versatility and suitability to gain additional insight into the relationship of dependent and independent variables by two examples, modelling voting behaviour and a failure model for debt amortization, and compare it to alternative approaches.

Keywords: model-based recursive partitioning, generalized linear models, model trees, functional trees, parameter instability, maximum likelihood.

1. Introduction

In many fields, classic parametric models are still dominant in statistical modelling and often rightly so. They demand some insight into the data generating process as well as a strong theoretical foundation to be applicable and as such force a researcher to be clear about the question she wants answered and to put a great deal of thought into collecting data and setting up the statistical model. They have the undeniable advantage to be interpretable in light of the research questions. Usually they pose restrictions on the relationship between the explanatory variables and the target variables. A very common restriction is to define the functional relationship between (transformations of) the independent and (transformations of) the dependent variables as linear. This gives rise to many parametric models, such as the classic linear model (Rao and Toutenburg 1997), generalized linear models (GLM, McCullagh and Nelder 1989) or, somewhat more generally, maximum likelihood (ML) models with linear predictors (LeCam 1990).

However, the linearity assumption for the coefficients of the predictor variables is precisely

This is an electronic version of an article published in *Journal of Statistical Computation and Simulation* available online doi:10.1080/00949655.2012.658804

what can sometimes appear to be too rigid for the whole data set, even if the model might fit well in a subsample. Especially with large data sets or data sets where knowledge about the underlying processes is limited, setting up useful parametric models can be difficult and their performance may not be sufficient. This is why a number of flexible methods that only need very few assumptions have recently been developed (sometimes collected under the umbrella terms "data mining" and "machine learning", Clarke, Fokoue, and Zhang 2009). Many of these methods are able to incorporate non-linear relationships or find the functional relationship by themselves and therefore can have higher predictive power in settings where classic models are biased or even fail. However, they may leave the researcher puzzled as to what the underlying mechanisms are, since many of them are either black box methods (e.g., random forests) or have a high variance themselves (e.g., trees). See Hastie, Tibshirani, and Friedman (2009) for a comprehensive discussion of some of the most popular of these methods and their advantages and disadvantages over classic parametric models.

In this paper we present an approach that integrates classic generalized linear models and maximum likelihood models with a linear predictor with a popular data mining method, recursive partitioning or trees. Trees have become a widely researched method since their first inception by Morgan and Sonquist (1968), see e.g., Breiman, Friedman, Olshen, and Stone (1984), Quinlan (1993), Hothorn, Hornik, and Zeileis (2006), Zhang and Singer (2010). Their biggest advantage is often seen in being simple to interpret and easy to visualize and at the same time allowing to incorporate high-order interactions and exhibiting higher predictive power than classic approaches. Over the last 20 years, effort went into combining parametric regression models with recursive partitioning (Chaudhuri, Lo, Loh, and Yang 1995). These approaches were sometimes coined hybrid, model or functional trees (Gama 2004) and include methods such as M5 (Quinlan 1993), SUPPORT (Chaudhuri, Huang, Loh, and Yao 1994), GUIDE (Loh 2002), LMT (Landwehr, Hall, and Eibe 2005) and LOTUS (Chan and Loh 2004). A recent proposal is model-based recursive partitioning (MOB, Zeileis, Hothorn, and Hornik 2008) which provides a unified framework for fitting, splitting and pruning based on M-estimation (including least squares and maximum likelihood as special cases).

Building upon the MOB framework, in what follows we explicitly present and discuss recursive partitioning of generalized linear and related models. The remainder of the paper is as follows: In Section 2 we discuss recursive partitioning of generalized linear models, from the basic idea of MOB in Section 2.1 and generalized linear models in Section 2.2 to the specific algorithm in Section 2.3. In Section 2.4 we discuss the extension to models with linear predictors that do not strictly belong to the class of GLM. In Section 3 we illustrate the usage of the algorithm for two data sets and how additional insight can be gained from this hybrid approach. Section 4 contains a comparative investigation into similarities and difference in applicability, properties and performance of the presented approach with alternative approaches from the literature. We conclude with a general discussion in Section 5.

2. Recursive partitioning of generalized linear models

2.1. Basic idea

Model-based recursive partitioning (Zeileis *et al.* 2008) looks for a piece-wise (or segmented) parametric model $\mathcal{M}_{\mathcal{B}}(Y, \{\boldsymbol{\vartheta}_b\}), b = 1, \ldots, B$ that may fit the data set at hand better than a

global model $\mathcal{M}(Y, \boldsymbol{\vartheta})$, where Y are observations from a space $\mathcal{Y}$. The existence of the real p-dimensional parameter vector in each segment $\boldsymbol{\vartheta}_b \in \Theta_b$ is assumed and their collection is denoted as $\{\boldsymbol{\vartheta}_b\}$. The partition $\{\mathcal{B}_b\}, b = 1, \ldots, B$ of the space $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_l$ spanned by the l covariates $Z_j, j = 1, \ldots, l$ gives rise to B segments within the data for which local parametric models $\mathcal{M}_b(Y, \boldsymbol{\vartheta}_b), b = 1, \ldots, B$ may fit better than the global model. All these local models have the same structural form, they only differ in terms of $\boldsymbol{\vartheta}_b$. Minimizing the objective function $\sum_{b=1}^{B} \sum_{i \in I_b} \Psi(Y_i, \boldsymbol{\vartheta}_b)$ (with the corresponding indices $I_b, b = 1, \ldots, B$) over all conceivable partitions $\{\mathcal{B}_b\}$ will result in the set of vectors of parameter estimates $\{\hat{\boldsymbol{\vartheta}}_b\}$. Technically this is difficult to achieve and a greedy forward search of selecting only one covariate in each step is suggested to approximate the optimal partition. In what follows, we will focus on generalized linear models (McCullagh and Nelder 1989) as the node model $\mathcal{M}(Y, \boldsymbol{\vartheta})$ and briefly extend it to other maximum likelihood models with linear predictors.

2.2. Generalized linear models

Let $Y = (y, \mathbf{x})$ denote a set of a response y and p-dimensional covariate vector $\mathbf{x} = (x_1, \ldots, x_p)$ with expected value $E(y) = \mu$. For $i = 1, \ldots, n$ independent observations, the distribution of each y_i is an exponential family with density (Aitkin, Francis, Hinde, and Darnell 2009)

$$f(y_i; \theta_i, \phi) = \exp\{[y_i\theta_i - \gamma(\theta_i)]/\phi + \tau(y_i, \phi)\} \tag{1}$$

Here, the parameter of interest (natural or canonical parameter) is θ_i, ϕ is a scale parameter (known or seen as a nuisance) and γ and τ are known functions. The n-dimensional vectors of fixed input values for the p explanatory variables are denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_p$. We assume that the input vectors influence (1) only via a linear function, the linear predictor, $\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ upon which θ_i depends. As it can be shown that $\theta = (\gamma')^{-1}(\mu)$, this dependency is established by connecting the linear predictor η and θ via the mean (Venables and Ripley 2002). More specifically, the mean μ is seen as an invertible and smooth function of the linear predictor, i.e.,

$$g(\mu) = \eta \text{ or } \mu = g^{-1}(\eta) \tag{2}$$

The function $g(\cdot)$ is called the link function. If the function connects μ and θ such that $\mu \equiv \theta$, then this link is called canonical and has the form $g = (\gamma')^{-1}$. Mean and variance for the n observations are given by

$$\mathrm{E}(y_i) = \mu_i = \gamma'(\theta_i) \qquad \mathrm{Var}(y_i) = \phi\gamma''(\theta_i) = V_i, \tag{3}$$

with $'$ and $''$ denoting the first and second derivatives respectively. Considering the GLM $\eta_i = g(\mu_i) = \boldsymbol{\beta}'\mathbf{x}_i$, the log-likelihood for n observations is given by Aitkin *et al.* (2009)

$$l(\theta, \phi; Y) = \sum_{i=1}^{n} [y_i\theta_i - \gamma(\theta_i)]/\phi + \sum_{i=1}^{n} \tau(y_i, \phi). \tag{4}$$

The score functions for $\boldsymbol{\beta}$ are then Aitkin *et al.* (2009)

$$S(\boldsymbol{\beta}, y_i) = \frac{\partial l(\boldsymbol{\beta}, \phi; Y)}{\partial \boldsymbol{\beta}} = \sum_i (y_i - \mu_i)\mathbf{x}_i / V_i g'(\mu_i), \tag{5}$$

and the information matrix is,

$$\begin{aligned} \mathcal{I}(\hat{\boldsymbol{\beta}}) &= -\frac{\partial^2 l(\boldsymbol{\beta}, \phi; Y)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} \\ &= -\sum_i \mathbf{x}_i \mathbf{x}_i' / V_i g_i'^2 - \sum_i (y_i - \mu_i) \mathbf{x}_i \mathbf{x}_i' (V_i g_i'' + V_i' g_i') / V_i^2 g_i'^3, \end{aligned} \tag{6}$$

with $g_i' = g'(\mu_i)$, $V_i' = \frac{dV_i}{d\mu_i}$ and $g_i'' = \frac{d^2 g(\mu_i)}{d\mu_i^2}$. In classic GLM the observed and expected information matrix has a block-diagonal structure so the cross-derivatives of $\boldsymbol{\beta}$ and ϕ are zero. Also, the structure of (5) shows that the MLE for $\boldsymbol{\beta}$ can be obtained independently of the nuisance parameter.

Asymptotically, the estimated parameter vector $\hat{\boldsymbol{\beta}}$ shows the same properties as other ML estimators (McCullagh and Nelder 1989) and is

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim \mathrm{N}_{p+1}(\mathbf{0}, \mathcal{I}(\hat{\boldsymbol{\beta}})^{-1}), \tag{7}$$

under standard regularity conditions.

2.3. Recursive partitioning algorithm

For GLM as described earlier, the algorithm of Zeileis *et al.* (2008) becomes:

1. Fit a generalized linear model (2) to all observations in the current node b. Hence, $\boldsymbol{\beta}_b$ is estimated by minimizing the negative of the log-likelihood (4). This can be achieved by setting the score function (5) to zero (which is admissible under mild regularity conditions) to yield the estimated parameter vector $\hat{\boldsymbol{\beta}}_b$.

2. Assess stability of the score function evaluated at the estimated parameter, $\hat{s}_i = S(\hat{\boldsymbol{\beta}}_b, y_i)$ with respect to every possible ordering of the values of each partitioning covariates $Z_j, j = 1, \ldots, l$ with generalized M-fluctuation tests (Zeileis and Hornik 2007). This yields a measure of instability of the parameter estimates for each covariate. If there is significant instability for one or more Z_j, select the Z_j associated with the highest instability. Here the p-value of the fluctuation test is used as a measure of effect size, the lower the p-value the higher the associated instability. If no significant instability is found, the algorithm stops. Please note that the significance level for the fluctuation tests has to be corrected for multiple testing to keep the global significance level, which can be achieved by a simple Bonferroni correction (Hochberg and Tamhane 1987).

3. After a splitting variable has been selected, the split points are computed by locally optimizing $-\sum_{k=1}^{K} l(\boldsymbol{\beta}_k, \phi; y_i \mathbb{1}_{[i \in I_k]})$ with $\mathbb{1}_{[\cdot]}$ denoting the indicator function. In principle this can be done for any number $K - 1$ of fixed or adaptively chosen splits that is less or equal to the number of observations in the current node. However, we restrict ourselves to binary splits, i.e., only one split point is chosen. This means we minimize $-l(\boldsymbol{\beta}_1, \phi; y_i \mathbb{1}_{[i \in I_1]}) - l(\boldsymbol{\beta}_2, \phi; y_i \mathbb{1}_{[i \in I_2]})$ for two rival segmentations with corresponding indices I_1 and I_2 by an exhaustive search over all pairwise comparisons of possible partitions.

4. This is then repeated recursively for each daughter node until no significant instability is detected or another stopping criterion is reached.

Parameter stability tests Step 2 in the algorithm above needs some additional details. As mentioned above, the parameter stability of the individual score function contributions with respect to the splitting variable Z_j is assessed by means of generalized M-fluctuation tests (Zeileis and Hornik 2007) for any ordering of the values of $Z_j, \sigma(Z_{ij})$. For a discussion of the empirical fluctuation process of the cumulative deviations of the score function $S(\hat{\boldsymbol{\beta}}_b, y_i)$ with respect to $\sigma(Z_{ij})$, $W_j(t, \hat{\boldsymbol{\beta}})$, and its asymptotical properties we refer to Zeileis and Hornik (2007) and Zeileis (2005). Depending on the nature of the covariate, we make use of two specific M-fluctuation tests for testing the null hypothesis of parameter stability for the empirical fluctuation process, $\lambda(W_j(\cdot)) = \lambda(W^0)$ where λ is a scalar functional and W_0 is a Brownian bridge. For continuous Z_j the *supLM* statistic (Andrews 1993) is used and for categorical covariates (factors) we employ the χ^2 statistic by Hjort and Koning (2002). The *SupLM* statistics is defined as

$$\lambda_{supLM}(W_j) = \max_{i=\underline{\iota},\ldots,\bar{\iota}} \left(\frac{i}{n} \cdot \frac{n-i}{n}\right)^{-1} \left|\left| W_j\left(\frac{i}{n}\right)\right|\right|_2^2, \tag{8}$$

where $[\underline{\iota}, \bar{\iota}]$ is the interval over which the potential instability point is shifted (typically defined by requiring some minimal segment size $\underline{\iota}$ and $\bar{\iota} = N - \underline{\iota}$). It is the maximization of single-shift LM statistics for all possible breakpoints in $[\underline{\iota}, \bar{\iota}]$. It has as its limiting distribution a squared, k-dimensional tied-down Bessel process (Zeileis *et al.* 2008). For categorical covariates we use

$$\lambda_{\chi^2}(W_j) = \sum_{c=1}^{C} \frac{|I_c|^{-1}}{n} \left|\left| \Delta_{I_c} W_j\left(\frac{i}{n}\right)\right|\right|_2^2, \tag{9}$$

where I_c is the set of indices of observations in category $c, c = 1, \ldots, C$ and $\Delta_{I_c} W_j$ is the increment of the empirical fluctuation process over the observations in category c. This test statistic is invariant to reordering of and within categories and captures instability for splitting data according to C categories. It has as its limiting distribution a χ^2-distribution with $df = k(C-1)$.

2.4. Beyond the GLM

One important property of standard GLM is that the parameter θ (or the parameter vector of the linear predictor) and the scale parameter ϕ are orthogonal (McCullagh and Nelder 1989). Estimates of parameters of the linear predictor $\hat{\boldsymbol{\beta}}$ are therefore (almost) independent of estimates of $\hat{\phi}$ under suitable limiting conditions (White 1982). Additionally, GLM assume that the explanatory variables do not affect the scale parameter ϕ at all (Aitkin *et al.* 2009). However, it is possible to extend the methodology used here beyond the standard GLM to incorporate (i) other distributions with non-orthogonal parameters such as the exponential distribution, the Weibull distribution or the extreme value distribution, or mixtures of exponential families such as the negative binomial distribution with unknown dispersion parameter and (ii) to use a linear predictor for the scale parameter for which parameter stability can also be assessed. In both cases, the node model $\mathcal{M}(Y, \boldsymbol{\vartheta})$ and the score functions will change. This has an effect on the asymptotic distribution of $\hat{\boldsymbol{\beta}}$, since we need to consider that we may deal with nuisance parameter estimation as well. See e.g., Aitkin *et al.* (2009) for inference with nuisance parameters. Apart from that however, the algorithm above still applies exactly the same way as long as an M-estimation approach (Huber 2009) such as maximum likelihood

is used for parameter estimation. This is because model fitting and the parameter stability tests and hence the algorithm employ M-estimation and the according asymptotics.

3. Gaining insight

3.1. Improved explanation with additional information

Due to its explorative character, model-based recursive partitioning can reveal patterns hidden within data modelled with GLM or provide further explanation of surprising or counter-intuitive results by incorporating additional information from other covariates. The tree-like structure allows the effects of these covariates to be non-linear and highly interactive as opposed to assuming a linear influence on the linked mean.

To illustrate, we use a data set from the 2004 general election in Ohio, USA. It was the presidential election of George W. Bush vs. John F. Kerry which took place on November 2nd, 2004 and saw Bush emerging as the winner with 34 more electoral seats than his adversary. Our sample consists of 19634 people from Ohio. We have aggregate voting records of each person, such as the overall number of times a person voted as well as the number of elections she was eligible to vote. Additionally, the data set includes a number of demographic, behavioural and institutional variables, such as each voter's age, gender, the party composition of the household ("partyMix"), the voter's rank ("householdRank", here the lower the number the higher the rank) and position in the household ("householdHead"), among others. We are interested in modelling the turnout of the 2004 general election on an individual level, i.e., has the person voted or not ("gen04").

In campaigning theory and voter targeting (e.g., Malchow 2008), past voting behaviour of a person is considered to be the strongest predictor of future voting behaviour. It is usually assumed that the more often a person went voting in the past, the more likely she is to do so in the upcoming election. Statistically this is a logistic regression problem with a binary dependent variable and therefore fits into the GLM framework. The number of attended elections is used as the predictor variable. It is important to note though, that the raw count of attended elections may be misleading because a higher count does not need to be the result of a person's general disposition to be more likely to vote. We therefore use the percentage of attended elections out of all elections a person was eligible to take part in to correct for possible bias. Figure 1 shows a spine plot of the data. It can be seen that the relationship is not monotonic but appears to be quadratic. This is not in accordance with intuition or the literature on voter targeting. One would expect a higher likelihood to vote for those who have a higher percentage of attended elections.

We fitted a global logistic regression model $\mathcal{M}(Y, \boldsymbol{\beta})$ with a quadratic effect of the predictor variable,

$$g(\mu) = \beta_0 + \beta_1 x + \beta_2 x^2 \tag{10}$$

where x is the percentage of attended elections ("percentAttended"). The estimated model parameters and goodness-of-fit values of the global model are displayed in Table 1. Interpolations of the predicted values were added to the spine plot in Figure 1. The initial observation could be confirmed by the model, the quadratic term turns out to be significant.

But why would people with a very high general attendance rate have a similarly low attendance

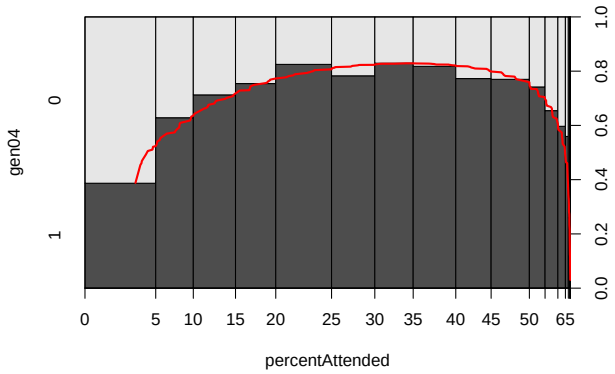

Figure 1: Spine plot of relative voting frequencies against the percent of attended elections out of all elections a person was eligible to. The solid black line is the interpolated prediction from a logistic regression model with a quadratic term for the predictor "percentAttended".

rate in the 2004 election as people who usually will attend elections rarely? And what people are they? We employ recursive partitioning of the logistic regression model in (10) to see if additional variables can shed more light on this phenomenon. We use a significance level of $\alpha = 0.05$ for the generalized M-fluctuation tests and force the minimum number of observations within each node to be at least 1060 (a fraction of about 8% of the overall data). The resulting tree is depicted in Figure 2 and the parameter estimates of the local models for the terminal nodes are given in Table 1.

Model	Node	$\hat{\beta}_0$ (se)	$\hat{\beta}_1$ (se)	$\hat{\beta}_2$ (se)	n	Dev	AIC
Global	-	−0.46 (0.03)	11.87 (0.29)	−17.32 (0.48)	19634	21948	21954
Segmented	2	$-\infty$ (−.−)	0.00 (−.−)	0.00 (−.−)	2180	0	6
	5	2.56 (0.38)	0.21 (1.87)	−6.53 (2.19)	2358	2126	2132
	7	0.42 (0.47)	14.09 (2.56)	−21.63 (3.22)	1277	808	814
	8	1.05 (0.41)	9.06 (2.17)	−15.36 (2.69)	1610	1170	1176
	10	−0.32 (0.08)	7.59 (1.17)	−4.16 (3.01)	1638	1991	1997
	13	−0.70 (0.06)	15.19 (0.77)	−19.10 (1.91)	4267	4602	4608
	14	0.16 (0.09)	12.23 (1.23)	−14.10 (3.04)	2222	1970	1976
	15	0.06 (0.14)	16.98 (1.35)	−17.82 (2.54)	4082	1565	1571

Table 1: Parameter estimates (standard errors in brackets) and goodness-of-fit statistics for the global logistic regression model and the terminal nodes of the piece-wise logistic regression model for the Ohio voter data. For legibility, $\hat{\beta}_1$ and $\hat{\beta}_2$ are given in units of the relative frequency. Please note that there are only non-voters in segment 2.

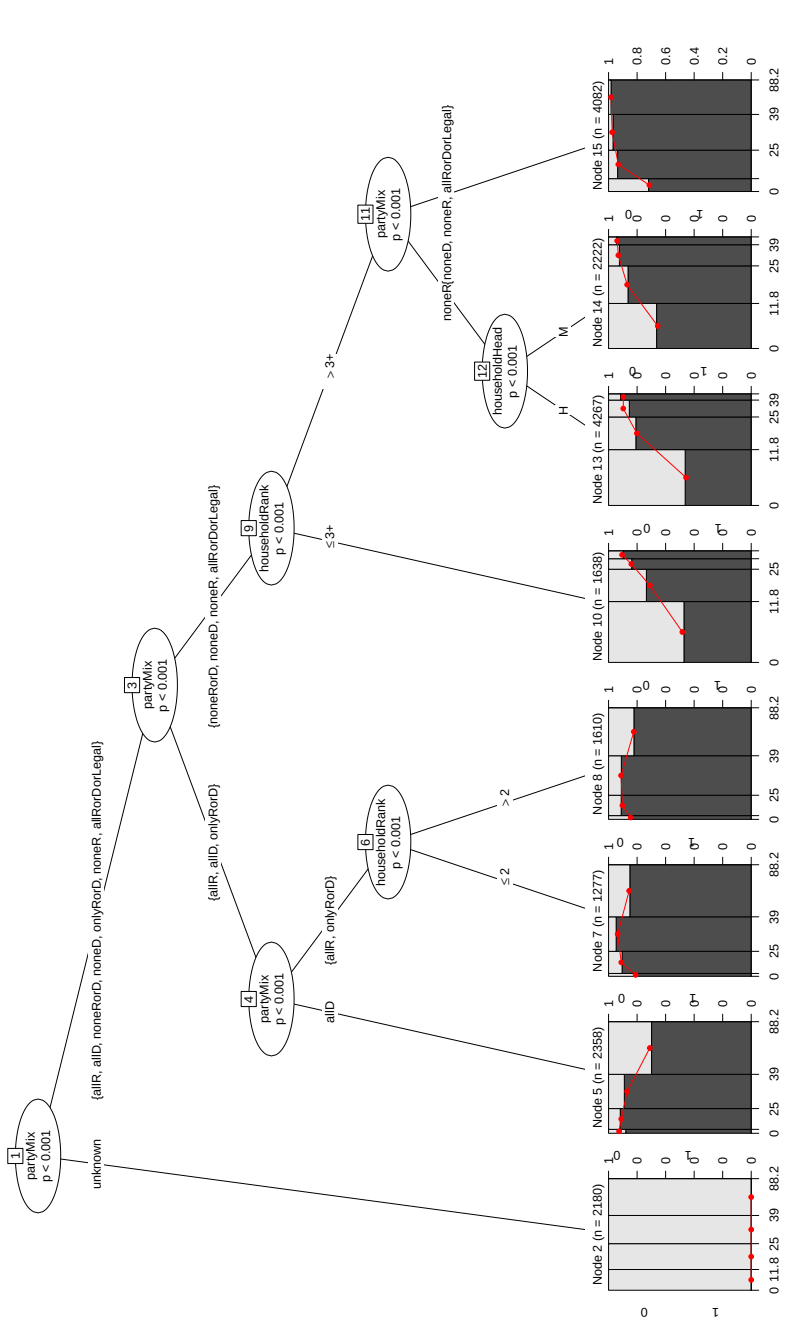

Figure 2: The resulting tree structure after partitioning the logistic regression model with linear predictor $\beta_0 + \beta_1 x + \beta_2 x^2$ where x is denoting the relative frequency of attended elections, "percentAttended". The terminal nodes display spine plots of the observed relative frequencies against the attended percentage for each partition with the solid lines connecting the predicted values from the logistic regression model.

The result from the partitioning algorithm ($\alpha = 0.05$ for the fluctuation tests) shows what or who may be responsible for the quadratic relationship between the percent of attended elections and the probability to vote. First there is a terminal node with people who did not vote at all. Please note that within this node we find (quasi-)complete separation[1](Albert and Anderson 1984). Second, the relationship is driven by the 5245 people whose household consists of members who are affiliated solely with the Democratic Party (node 5) and to a lesser extent by those affiliated solely with the Republican party (node 7) or whose household consists only of democrats and republicans (node 8). In other words, there are no independent voters in these households. Especially the segment of people whose household is composed entirely of Democrats ($n_5 = 2358$) contribute to the overall quadratic relationship seen in Figure 1. They show declining voting probability for people with a high general individual turnout and quite strongly so. While those people with a small to medium percentage of general attendance have fairly high voting probabilities that slightly increase for higher predictor values, those with a general attendance rate of 0.39 or more (nearly half of the segment) experience a sheer drop of voting probability.

For the other two segments, those whose household consists entirely of Republicans or of a mix of Republicans and Democrats ($n_7 = 1277$ and $n_8 = 1610$) this picture is less striking. Here, an attendance rate of about 0.1 to 0.4 is associated with the highest voting probability, whereas very rare voters ($x \leq 0.1$) and frequent voters ($x \geq 0.4$) have a similarly high voting probability that is slightly less than for the other people in the segment. Nodes 7 and 8 differ in the assigned rank in the household. The difference between these two nodes lies in the slightly higher overall voting probability and a higher probability for those with an attendance percentage between 25% and 40% for those with household ranks 1 and 2 (node 7).

On the other hand, the segments in terminal nodes $10, 13, 14$ and 15 indeed show a monotonically increasing voting probability for an increase of the predictor variable. This is in accordance with intuition and literature on political campaigning. Here, having at least one household member who identifies herself as "independent" is the key difference to the segments with an inverse U-shaped voting probability relationship with the percentage of attended elections. By using model-based recursive partitioning with additional covariate information, we are able to find an explanation as to why a quadratic effect has to be included into the logistic regression model. We can single out the observations that are responsible for this phenomenon and show that there are segments in which the assumed monotonic relationship is actually present.

3.2. Identifying segments with poor or good fit

Another area in which model-based recursive partitioning can be helpful is in identifying segments of the data for whom an *a priori* assumed model fits well. It may be that overall this model has a poor fit but that this is due to some contamination (for example merging two separate data files or systematic errors during data collection at a certain date). By using the described algorithm the data set might be partitioned in a way that enables us to find the segments that have poor fit and find segments for which the fit may be rather good (see also Juutilainena, Koskimäkia, Laurinena, and Röninga 2011 for an alternative for regression analysis).

[1]In this node the ML estimator does not exist. The algorithm has the positive effect of separating these observations from the rest, hence estimation in other nodes works well which might otherwise not be the case.

To illustrate this, we use data of debt amortization rates as a function of the duration of the enforcement. It can be expected that the longer the enforcement lasts, the higher amortization rate should be achieved. What is special about these data is that they came from two sources and were merged into a single data set. The merged data set consisted of $n = 165$ observations, with 75 observations from file "0" and 90 observations from file "1".

The structure of the statistical problem here is similar to a "time-to-event" analysis. We consider the amortization rate relative to the original claim as the metric variable whose hazard function we want to model. Failure to pay more, default, insolvency, bankruptcy or meeting the obligation are considered as the event "stopped paying". Additionally, we have the possibility of right censored observations if a person was lost to follow up. This lead us to using a Weibull Regression model which is an example of the type of models described in Section 2.4. Here, the scale parameter and the parameters of the linear predictor are not orthogonal and have to be estimated simultaneously.

Formally, following Venables and Ripley (2002), we model the hazard function $h(r)$, with r denoting a realization of the random variable of achieved amortization rate, R, which takes the form of

$$h(r) = \lambda^{\alpha}\alpha r^{\alpha-1} = \alpha r^{\alpha-1}\exp(\alpha\boldsymbol{\beta}^{\top}\boldsymbol{x}) \tag{11}$$

for the Weibull distribution. The parameter λ is modelled as an exponential function of the

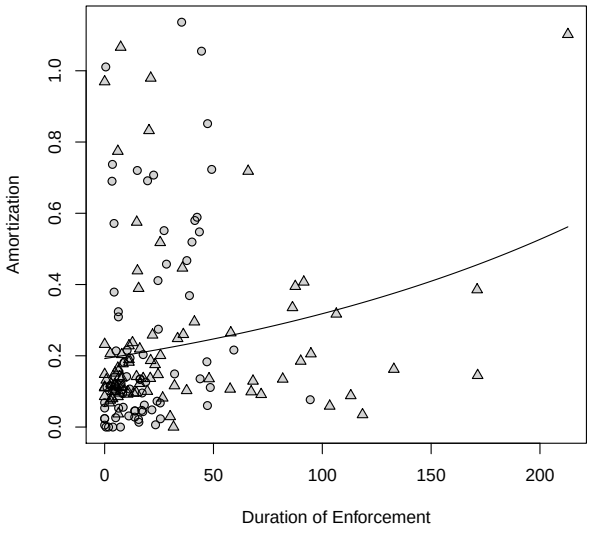

Figure 3: Scatterplot of duration of the enforcement and the achieved amortization rate until the event "failure to pay more" happened. The solid line represents the predicted values from the global Weibull regression model. Observations from file "0" are plotted as circles, those from file "1" as triangles.

Model	Node	$\hat{\beta}_0$ (se)	$\hat{\beta}_1$ (se)	Scale (sd)	n	log-lik
Global	-	−1.65 (0.12)	0.01 (0.00)	0.17 (0.06)	165	76.9
Segmented	3	−2.31 (0.33)	0.03 (0.01)	0.48 (0.11)	65	42.4
	4	1.95 (0.09)	0.01 (0.00)	−0.48 (0.09)	79	76.8
	5	−0.53 (0.22)	−0.00 (0.01)	−0.34 (0.19)	21	−4.6

Table 2: Parameter estimates (standard errors in brackets) and goodness-of-fit statistics for the global Weibull model and the terminal nodes of the segmented Weibull model for the debt amortization data.

covariates $\mathbf{x}$. In a loglinear model formulation this becomes

$$\log(R) = -\log\lambda + \frac{1}{\alpha}\log\epsilon \tag{12}$$

with ϵ being a disturbance term that is independent of $\boldsymbol{x}$ and w.l.o.g. exponentially distributed. In this particular example, $\boldsymbol{x}$ consists of an intercept and the duration of the enforcement.

A visualization of the data can be found in Figure 3. The point type corresponds to the different files, a circle for file "0" and the triangle for file "1". Additionally we include the predicted values from the global Weibull regression model. The results of the model fitting can be seen in Table 2.

What we can see here is that the model does not fit well. The log-likelihood for the regression model is 76.9 and for the intercept only model it is 75.1 which is not a significant difference at $\alpha = 0.05$ ($p = 0.054$). Apart from that it looks as though the Weibull regression is not really appropriate for the whole data set. However, one can see that a subset of the data may be appropriately modelled with the proposed relationship if it were not for the observations that have quite high amortization rates for a low enforcement duration. There are at least two possible explanations for such a lack of fit: (i) explanatory variables that were not considered in the model (misspecification) and (ii) data contamination. In this analysis it is quite likely that (i) has some effect. Hence we use information from other covariates in the subsequent recursive partitioning and gauge their influence. Inspection of Figure 3 however reveals something else. Observations that have high amortization rates for low duration time are mainly from file "0". Additionally the distribution of the enforcement duration in file "1" is more skewed (skewness 1.96 vs. 1.39) and has a much longer right tail. The same holds for the amortization rate. It looks as if merging of the two data sets could have led to a contamination as they are probably not comparable. We partition these based on the Weibull regression model from (11). As additional covariates that are used for partitioning we have the person's gender, liability at the begin of the enforcement, the current liability, the number of securities a person has as well as a person's collateralization ratio. We also include a dummy variable to flag which file the observation was from. The significance level for the parameter test is again 0.05.

The resulting tree can be found in Figure 4 and the estimated model in Table 2. We see that both suspicions from above can be confirmed. First, there is an additional variable, collateralization ratio, that seems to be relevant. Its inclusion leads to a segment where the influence of the duration is not significant. This is partly due to the small sample size in this node, but we can also see that the regression coefficient has a negative sign. It does

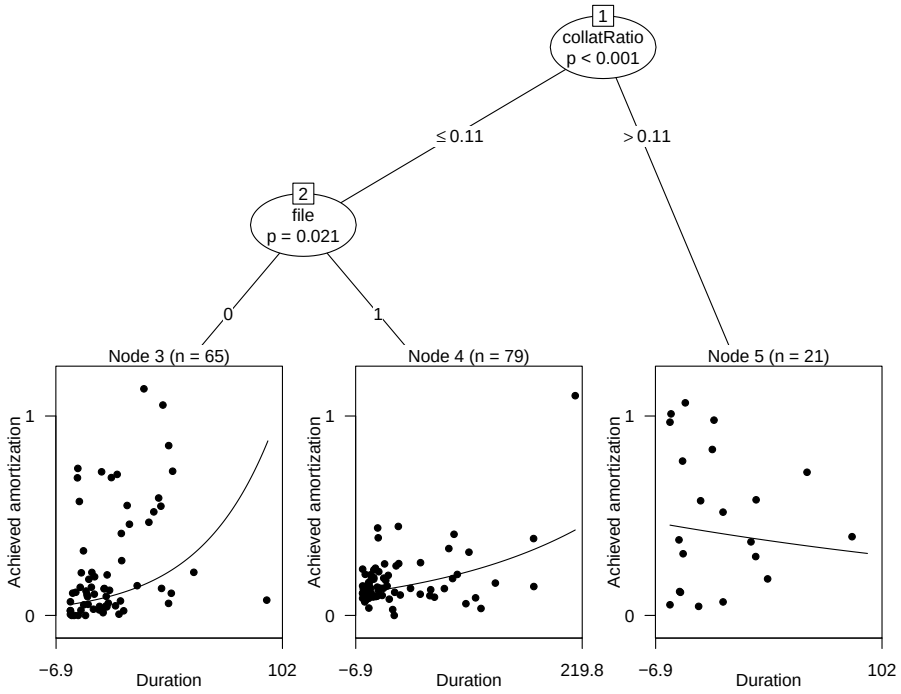

Figure 4: The recursively partitioned Weibull regression model of amortization rate explained by the duration of the enforcement. For each terminal node there is a scatterplot with the solid line representing the predicted values from the local model.

not appear as if there would be a positive relationship that we just do not detect but rather that there is no positive relationship at all. This makes sense, as the collateralization ratio is a measure of how many and how well diversified the securities of a person are and how high their value is. A person with a high collateralization ratio (two cars for example) may be able to amortize her debt very fast or at least it may not depend on the duration of the enforcement. It seems rather likely that a person with a high collateralization ratio who does not amortize her debt rather soon may have problems with or may refuse payment regardless of enforcement duration.

Second, for those with a collateralization ratio of less than 0.11, the algorithm points to a difference in the two data sources. For one data set, file "1", the Weibull regression actually fits rather well (node 4, log-likelihood of 76.8). Additionally, we have a significant positive influence of the explanatory variable. Please note however, that the coefficient and corresponding p-value is highly influenced by an outlier with amortization rate greater than 1. Removing this value leads to a much weaker association that is barely significant on a 5% level[2]. In

[2]If a semi-parametric Cox model is fitted, there is no significant influence.

node 3, for which all observations stem from file "0", we see an ill fit of the Weibull model with a log-likelihood of 45.1. It even looks as if the (significant) regression line is splitting the data in this node into two groups rather than explaining them. There seems to be heterogeneity in the data in this segment that cannot be explained by the regression model.

What we can see from this analysis however is that recursive partitioning of models can help us identify segments in our data for which the model may either fit well or may be inappropriate. Here, merging the data from file "0" with those in file "1" leads to some contamination of the merged data set. This contamination masks the acceptable fit for the subset of observations from file "1", a fact that is not necessarily clear from the non-segmented analysis. Most probably those two data sets were obtained individually and on different occasions or for different studies. They just happen to have similar variables in them. This goes to show once again that planning a study involves more than just collecting data.

4. Comparison to similar approaches

A number of model tree algorithms have been proposed in recent years. Table 3 gives an overview of different model tree algorithms, properties of the tree induction, which node models they can fit and the available software (R, R Development Core Team 2011, Weka, Hall, Frank, Holmes, Pfahringer, Reutemann, and Witten 2009, or author binaries) to fit them.

In the machine learning literature, tree algorithms with models in each node have been around at least since the M5 algorithm of Quinlan (1993) (see also Wang and Witten 1997 for the "rational reconstruction" M5') for linear models in nodes. Another algorithm is LMT (Landwehr *et al.* 2005), which allows trees with a boosted logistic node model for binary or multinomial responses. Gama (2004) proposed an abstract framework coined "functional trees", for building tree algorithms with univariate or multivariate splits and node models.

Several model tree algorithms were also suggested in the statistical literature, for example SUPPORT (Chaudhuri *et al.* 1994, 1995), which originally suggested smoothed or unsmoothed piece-wise polynomial models, and was subsequently extended to GLM-type and survival models (Ahn and Chen 1997; Choi, Ahn, and Chen 2005; Ahn and Loh 1994; Ahn 1994b,a, 1996b,a). Many of those model tree algorithms encompassed two novel ideas: (i) unbiasedness in the split variable selection, and (ii) separation of modelling and splitting variables. Two prominent examples are GUIDE (Loh 2002, 2009) and LOTUS (Chan and Loh 2004). While the former provides capabilities for fitting models to Gaussian responses, quantile regression (Chaudhuri and Loh 2002), Poisson models (Loh 2008), proportional hazard models and longitudinal models, the latter uses similar ideas for modelling binary responses. With the MLRT algorithm (Su, Wang, and Fan 2004), some effort also went into embedding regression trees into a rigorous statistical framework based on the likelihood as an objective function which can easily be extended to model trees (a similar idea for a very specific context has been proposed by Ichinokawa and Brodziak (2010)).

Conceptually, the MOB algorithm used in the present paper belongs to the statistically motivated algorithms and combines most advantages of the aforementioned algorithms. Like GUIDE or LOTUS, it uses unbiased split selection and allows for separation of node model and splitting variables. Similar to MLRT, it uses a rigorous statistical framework of employing the same objective function to induce the tree structure and fit the node models. Compa-

rable to SUPPORT, MOB pre-prunes the trees. Furthermore, MOB provides functionality for many different type of node models that even exceeds the versatility of SUPPORT and GUIDE. Analyzed within the "functional tree" framework, MOB with more than a single explanatory variable in the node model or with interactions as splitting variables can be seen as employing multivariate splits that allow for oblique partitioning which according to Gama (2004) is an advantage especially for large data sets. Moreover, the MOB algorithm can straightforwardly be extended to feature variable selection in the node model, post-pruning of the tree or smoothing of the piece-wise function (e.g., with Chandler and Johnson 2012).

4.1. Voting data revisited

To compare the performance of the presented approach to other algorithms, we reanalyze the Ohio voter data set 3.1 with the LMT and the LOTUS.

We fit LMT with Weka (Hall *et al.* 2009) for which we employ the RWeka interface (Hornik, Buchta, and Zeileis 2009). The trees are restricted to only allow for binary splits. As LMT does not allow for separation between variables employed for the node model and for splitting respectively, all prediction variables (including the square of "percentAttended") are supplied to the algorithm. This leads to a single root note (without any splits) and hence a global logistic model with 33 parameters. For the same data, MOB uses a tree with 7 splits and 3 parameters in the node model. LMT selects all those variables that are selected by the MOB plus some additional variables, leading to the large global logistic model. The overall classification accuracy of LMT for the training sample is 0.843 whereas the MOB has a classification accuracy of 0.840. Hence, the LMT is less parsimonious (33 vs. $24 = 3 \times 8$ parameters) while the predictive accuracy on the learning sample is only slightly higher (0.843 vs. 0.840). Additionally, the quadratic relationship between attendance percentage and voting probability is not as easily intelligible as compared to the MOB.

With the LOTUS binary, we fit a model with an analogous setup for node model and splitting variables compared to the MOB as specified in Section 3.1. A maximum number of 1060 observations per node is specified and we opt for no variable selection for the node model. Everything else is set to the LOTUS default values. The resulting pruned tree (0-SE) has 12 splits and each node model has 3 estimated parameters. Hence MOB fits a more parsimonous model tree ($3 \times 13 = 39$ parameters for LOTUS vs. $3 \times 8 = 24$ parameters for MOB). At the same time MOB achieves higher classification accuracy on the training sample (0.84 vs. 0.76). As is the case with LMT, splitting variables selected by LOTUS partly coincide with those selected by MOB. On the one hand, both algorithms select "householdRank" and "partyMix" quite often for splitting (MOB five times, LOTUS five times and high up in the tree hierarchy). On the other hand, the variables "dontPhone", "compOwner", "income" and "educationLevel" are chosen for splitting only by the LOTUS (and deeper down the tree hierarchy). The biggest difference of the LOTUS to the MOB tree is that the first split is due to observations labeled "unknown" and "noneRorD" for "partyMix". This leads to a left subtree with 5 additional leaves for the LOTUS. MOB selects the same variable but only splits off observations that have a value of "unknown", which are not partitioned further. To a depth of 3, the right subtrees after the first split for MOB and LOTUS are more or less similar in terms of splitting variables and split points and therefore explanation of the quadratic relationship is comparable for both methods.

Thus, all algorithms achieve a more or less similar classification accuracy. They all agree on

		Model tree algorithm							
		FT	GUIDE	LMT	LOTUS	M5'	MLRT	MOB	SUPPORT
Tree structure	Pre-pruning		*		*			×	×
	Post-pruning	×	×	×	×	×	×	*	*
	Unbiased		×		×			×	
	Covariate type	all	all	all	all	all	all	all	metric
	Multiway splits	×		×				*	
Separate node model and splitting variables			×		×			×	
Adaptive node model		*	×	×	×	×	*	*	*
Type of node model	Gaussian	×	×			×	×	×	×
	Binomial (Logit)	*		×	×		*	×	×
	Binomial (other links)	*					*	×	
	Quasi-Binomial (Logit)	*						×	×
	Quasi-Binomial (other links)	*						×	
	Poisson	*	×				*	×	×
	Quasi-Poisson	*						×	×
	Gamma	*					*	×	×
	Inverse Gaussian	*					*	×	
	Negative Binomial	*					*	×	
	Beta	*					*	×	
	Multinomial	*		×			*	*	
	Parametric Survival	*	×				*	×	×
	Longitudinal Gaussian	*	×				*	*	×
	General Maximum Likelihood	*					*	*	
	General Quasi-Likelihood	*						*	
	Robust (M-type)	*						*	
	Quantile	*	×						
Software			author	Weka	author	Weka		R	

Table 3: Comparison of properties and applicability of different model tree algorithms. For the rows a × denotes if there already exists an implementation and ∗ denotes if an implementation is possible within the provided framework of the specific algorithm without changes (please note that the FT algorithm is on a more abstract level than all the other specific algorithms). The last row lists the availability of software packages ("author" means binaries are publicly available from the author).

"percentAttended" and its square, "partyMix" and "householdRank" to be important variables. LMT chooses a large global regression model with a high predictive accuracy. MOB and LOTUS use a much simpler logistic model, but can achieve comparable accuracy to LMT through splits (especially MOB). For this, MOB needs a lower number of splits than LOTUS which makes the MOB results easier to interpret.

5. Conclusion

In this paper, we introduced recursive partitioning of generalized linear and related models as a special case of model-based recursive partitioning. We tried to illustrate how the algorithmic approach may lead to additional insight for a *a priori* assumed parametric model, especially if the underlying mechanisms are too complex to be captured by the GLM. As such, model-based recursive partitioning can automatically detect interactions, non-linearity, model misspecification, unregarded covariate influence and so on. As an exploratory tool, it can be used for complex and large data sets for which it has a number of advantages. On the one hand, compared to a global GLM, a MOB model tree can alleviate the problem of bias and model misspecification and provide a better fit. On the other hand, compared to tree algorithms with constants, the specification of a parametric model in the terminal nodes can add extra stability and therefore reduce the variance of the tree methods. Being a hybrid of trees and classic GLM-type models, the performance of MOB models usually lies between those two poles: They tend to exhibit higher predictive power than classic models but less than non-parametric trees (Zeileis *et al.* 2008). They add some complexity compared to classical model because of the splitting process but are usually more parsimonous than non-parametric trees. They show a slightly higher variance than a global model in bootstrap experiments, but much less than non-parametric trees (even pruned ones). Compared to other model tree algorithms, MOB often exhibits comparable predictive accuracy while at the same time being more parsimonous than direct competitors. Results from MOB trees are often easy to communicate and visualize. Additionally, MOB is currently the most versatile model tree algorithm and can be rigorously justified from a statistical point of view. We believe that the exploratory use of recursive partitioning of GLM-type models, particularly with the presented approach, is fruitful for researchers dealing with models with linear predictors to detect possible hidden structure and get a better grasp of what is really happening in the data at hand, especially if modelling with classical statistical methods reaches its limitations.

References

Ahn H (1994a). "Tree-Structured Exponential Regression Modeling." *Biometrical Journal*, **36**, 43–61.

Ahn H (1994b). "Tree-Structured Extreme Value Model Regression." *Communications in Statistics – Theory and Methods*, **23**, 153–174.

Ahn H (1996a). "Log-Gamma Regression Modeling through Regression Trees." *Communications in Statistics – Theory and Methods*, **25**, 295–311.

Ahn H (1996b). "Log-Normal Regression Modeling through Recursive Partitioning." *Computational Statistics & Data Analysis*, **21**, 381–398.

Ahn H, Chen J (1997). "Tree-Structured Logistic Model for Over-Dispersed Binomial Data with Application to Modeling Developmental Effects." *Biometrics*, **53**, 435–455.

Ahn H, Loh WY (1994). "Tree-Structured Proportional Hazards Regression Modeling." *Biometrics*, **50**, 471–485.

Aitkin M, Francis B, Hinde J, Darnell R (2009). *Statistical Modelling in R*. Oxford University Press, New York.

Albert A, Anderson JA (1984). "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models." *Biometrika*, **71**, 1–10.

Andrews DWK (1993). "Tests for Parameter Instability and Structural Change with Unknown Change Point." *Econometrica*, **61**, 821–856.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). *Classification and Regression Trees*. Wadsworth, California.

Chan KY, Loh WY (2004). "LOTUS – An Algorithm for Building Accurate and Comprehensible Logistic Regression Trees." *Journal of Computational and Graphical Statistics*, **13**, 826–852.

Chandler G, Johnson L (2012). "Automatic Locally Adaptive Smoothing for Tree-Based Set Estimation." *Journal of Statistical Computation and Simulation*. doi:10.1080/00949655.2011.613395.

Chaudhuri P, Huang MC, Loh WY, Yao R (1994). "Piecewise-Polynomial Regression Trees." *Statistica Sinica*, **4**, 143–167.

Chaudhuri P, Lo WD, Loh WY, Yang CC (1995). "Generalized Regression Trees." *Statistica Sinica*, **5**, 641–666.

Chaudhuri P, Loh WY (2002). "Nonparametric Estimation of Conditional Quantiles Using Quantile Regression Trees." *Bernoulli*, **8**, 561–576.

Choi Y, Ahn H, Chen JJ (2005). "Regression Trees for Analysis of Count Data with Extra Poisson Variation." *Computational Statistics & Data Analysis*, **49**, 893–915.

Clarke B, Fokoue E, Zhang HH (2009). *Principles and Theory of Data Mining and Machine Learning*. Springer-Verlag, New York.

Gama J (2004). "Functional Trees." *Machine Learning*, **55**, 219–250.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009). "The Weka Data Mining Software: An Update." *SIGKDD Explorations*, **11**(1).

Hastie T, Tibshirani R, Friedman J (2009). *Elements of Statistical Learning*. Springer-Verlag, New York, 2nd edition.

Hjort NL, Koning A (2002). "Tests for Constancy of Model Parameters over Time." *Nonparametric Statistics*, **14**, 113–132.

Hochberg Y, Tamhane AC (1987). *Multiple Comparison Procedures*. John Wiley & Sons, New York.

Hornik K, Buchta C, Zeileis A (2009). "Open-Source Machine Learning: R Meets Weka." *Computational Statistics*, **24**(2), 225–232.

Hothorn T, Hornik K, Zeileis A (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics*, **15**(3), 651–674.

Huber P (2009). *Robust Statistics*. John Wiley & Sons, Hoboken, 2nd edition.

Ichinokawa M, Brodziak J (2010). "Using Adaptive Area Stratifiaction to Standardize Catch Rates with Application to North Pacific Swordfish (Xiphias Gladius)." *Fish Res*, **106**, 249–260.

Juutilainena I, Koskimäkia H, Laurinena P, Röninga J (2011). "BUSDM – An Algorithm for the Bottom-Up Search of Departures from a Model." *Journal of Statistical Computation and Simulation*, **81**, 561–578.

Landwehr N, Hall M, Eibe F (2005). "Logistic Model Trees." *Machine Learning*, **59**, 161–205.

LeCam L (1990). "Maximum Likelihood – An Introduction." *ISI Review*, **58**, 153–171.

Loh WY (2002). "Regression Trees with Unbiased Variable Selection and Interaction Detection." *Statistica Sinica*, **12**, 361–386.

Loh WY (2008). "Regression by Parts: Fitting Visually Interpretable Models with GUIDE." In CH Chen, W Härdle, A Unwin (eds.), "Handbook of Data Visualization," Springer Handbooks of Computational Statistics, pp. 447–469. Springer-Verlag, New York.

Loh WY (2009). "Improving the Precision of Classification Trees." *Annals of Applied Statistics*, **3**, 1710–1737.

Malchow H (2008). *Political Targeting*. Predicted Lists, LLC, 2nd edition.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. Chapman and Hall, London, 2nd edition.

Morgan JN, Sonquist JA (1968). "Problems in the Analysis of Survey Data, and a Proposal." *Journal of the American Statistical Association*, **58**, 415–434.

Quinlan JR (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo.

R Development Core Team (2011). "R: A Language and Environment for Statistical Computing." http://www.R-project.org/.

Rao CR, Toutenburg H (1997). *Linear Models: Least Squares and Alternative Methods*. Springer-Verlag, New York, 2nd edition.

Su X, Wang M, Fan J (2004). "Maximum Likelihood Regression Trees." *Journal of Computational and Graphical Statistics*, **13**, 586–598.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S.* Springer-Verlag, New York, 4th edition.

Wang Y, Witten I (1997). "Induction of Model Trees for Predicting Continuous Classes." In "Proceedings of the posters of the European Conference on Machine Learning," University of Economics, Faculty of Informatics and Statistics, Prague, Czech Republic.

White H (1982). "Maximum Likelihood Estimation of Misspecified Models." *Econometrica*, **29**, 1–25.

Zeileis A (2005). "A Unified Approach to Structural Change Tests Based on ML Scores, F Statistics, and OLS Residuals." *Econometric Reviews*, **24**(4), 445–466.

Zeileis A, Hornik K (2007). "Generalized M-Fluctuation Tests for Parameter Instability." *Statistica Neerlandica*, **61**(4), 488–508.

Zeileis A, Hothorn T, Hornik K (2008). "Model-Based Recursive Partitioning." *Journal of Computational and Graphical Statistics*, **17**(2), 492–514.

Zhang H, Singer B (2010). *Recursive Partitioning and Applications.* Springer-Verlag, New York, 2nd edition.

Affiliation:

Thomas Rusch
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Augasse 2–6
1090 Wien, Austria
E-mail: Thomas.Rusch@wu.ac.at

Achim Zeileis
Department of Statistics
Faculty of Economics and Statistics
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: Achim.Zeileis@R-project.org
URL: http://eeecon.uibk.ac.at/~zeileis/

4. Model Trees with Topic Model Pre-Processing: An Approach for Data Journalism Illustrated with the WikiLeaks Afghanistan War Logs

Model trees with topic model pre-processing: An approach for data journalism illustrated with the WikiLeaks Afghanistan War Logs

Thomas Rusch
WU (Wirtschaftsuniversität Wien)

Paul Hofmarcher
WU Wirtschaftsuniversität Wien

Reinhold Hatzinger
WU Wirtschaftsuniversität Wien

Kurt Hornik
WU Wirtschaftsuniversität Wien

Abstract

The WikiLeaks Afghanistan war logs contain more than 76000 reports of incidents in the US-led Afghanistan war, covering the period from January 2004 to December 2009. The availability of such complex data and the potential to derive stories from them has shifted the focus of journalistic attention increasingly toward data driven journalism. In this paper we advocate the usage of modern statistical methods for problems of data journalism which may help journalistic work and lead to additional insight. Using the WikiLeaks Afghanistan war logs for illustration, we present an approach that allows to build intelligible statistical models for interpretable segments in the data, in this case to understand the fatality rates associated with different circumstances. Our approach combines pre-processing by Latent Dirichlet Allocation (LDA) with model-based recursive partitioning. LDA is used to process the natural language information contained in each report summary by estimating latent topics and assigning each report to one of them. Together with other variables these topic assignments subsequently serve as explanatory variables for modeling the reported number of fatalities. Modeling itself is carried out with recursive partitioning of negative binomial distributions. We identify segments with different fatality rates that correspond to a small number of topics and other variables as well as their interactions. Furthermore, we carve out the similarities between segments and connect them to stories that have been covered in the media. This gives an unprecedented description of the war in Afghanistan and serves as an example of how data journalism can benefit from modern statistical techniques.

Keywords: Afghanistan, count data, data base data, data journalism, fatalities, latent dirichlet allocation, model-based recursive partitioning, model trees, topic models, WikiLeaks.

1. Introduction

The analysis of fatalities in wars and armed conflicts is an eminent subject of scientific investigation. Most of them have been conducted in a historical context, often retrospectively estimating the number of and circumstances under which fatalities of war occurred. There are literally hundreds of historical investigations into numerous wars, see e.g. Garfield and Neugut (1991) for a review of the last 200 years.

Notwithstanding such efforts, contemporary systematic scientific investigation into the number of fatalities in wars are much rarer and more closely tied to the emergence of statistics and epidemiology as disciplines rather than to the discipline of history. Some of the first examples we could find were Marshall and Balfour (1838) or Nightingale (1863). While these investigations were still firmly rooted in descriptive statistics, statistical modeling of the number of fatalities was about to become imperative as Bortkiewicz (1898) published his seminal work on the use of the Poisson distribution for rare events which he motivated by the analysis of horse-kick deaths of Prussian soldiers. To our knowledge this was the first instance of a parametric and inferential approach to analyze fatalities of war. Contemporary investigations into the number and circumstances of casualties of war that made use of statistical modeling next to descriptive approaches increased much since then, for example Spiegel and Salama (2001), Thomas, Parker, Horn, Mole, Spiro, Hooper, and Garland (2001), Lakstein and Blumenfeld (2005) or Holcomb, McMullin, Pearse, Caruso, Wade, Oetyen-Gerdes, Champion, Lawnick, Farr, Rodriguez, and Butler (2007).

In the modern age their number seems to peak[1] arguably because data on war fatalities are much easier to come by. Recent work, for example for the war in Afghanistan, includes the studies on child casualties by Bhutta (2002) and on military fatalities by Bird and Fairweather (2007) or Bohannon (2011). Other recent work in this field has been done by Haushofer, Biletzki, and Kanwisher (2010); Degomme and Guha-Sapir (2010); Buzzell and Preston (2007); Burnham, Lafta, Doocy, and Roberts (2006).

In July 2010 the availability of data on a specific war became unprecedented, as whistleblower website WikiLeaks released a massive amount of military classified war logs from the Afghanistan war into the public. These documents constitute a "war diary" of the military operation in Afghanistan, containing a detailed description of what happened in each event for which a report was filed, including counts of killed and wounded people, local and administrative information, temporal and spatial information and a short written description of each particular incident. The documents themselves stem from a database of the US army and along the lines of WikiLeaks, they do not generally cover any top-secret operations or European or other operations of the International Security Assistance Force (ISAF). In total, the war logs consist of 76911 documents and cover the time period between January 2004 and December 2009. They provide an unprecedented view of the war in Afghanistan with an information abundance that has previously been unknown and has only been topped by the release of the Iraq war logs some months later.

Interestingly, the scientific community has been rather hesistant in approaching the data[2] (but see O'Loughlin, Witmer, Linke, and Thorwardson (2010); Conway (2010) for notable exceptions). In journalism and the media world however, the impact of the release was very strong. The German news magazine *Der Spiegel* wrote that the editors-in-chief of *Der Spiegel*, The New York Times and The Guardian were "unanimous in their belief that there is a justified public interest in the material" (Gebauer 2010) and the war diary was marked as the 21st century equivalent of the Pentagon Papers from the 1970s. However, while the Pentagon

[1] According to a quick survey in the ISI Web of Knowledge citation database, searching for "war casualties" found 1476 records, 840 of which were published after 2000. 580 of those were published no earlier than 2005.

[2] This might be due to concern about the legitimacy of publishing such data. However, a Congressional Research Service (CRS) expertise (Elsea 2011) considers the publication of such information lawful: "Thus, although unlawful acquisition of information might be subject to criminal prosecution with few First Amendment implications, the publication of that information remains protected." (p. 29). Even more so, the *usage* of the leaked data is generally considered legal, even if the publication would not be.

Papers have provided an aggregated view on the war in Vietnam, the WikiLeaks war diary is an account of the daily events in Afghanistan containing thousands of mosaic tiles describing incidents from the perspective of the US forces. They were written by different people and are sometimes accurate and sometimes possibly not. The war logs themselves neither contain information on strategic decisions nor do they provide a coherent, general picture of the war. Hence, each media outlet had to write its own stories based on the material (see O'Loughlin *et al.* 2010). This take on the WikiLeaks Afghanistan war logs has been praised as data-driven journalism in action (see Rogers 2010).

To elicit stories out of data is a contemporary issue for journalists especially when the amount of data is huge and cannot be processed easily by humans. This is where data journalism or data-base journalism, a type of journalism which allows stories to enfold from data, comes into play. This type of journalism uses statistical and computational methods to deal with the problem of processing large amounts of data (often in form of documents) and presenting them in an accessible form. For example, a popular approach is to narrow down the data by keyword searches with the goal to find a relevant subset that can be processed by a human reader. Another one is to count the frequency of words within documents to allow for a broad overview of the data or to extract additional information that can be used for telling a story without the need for directly reading or processing all data points (see e.g. Hofmarcher, Theußl, and Hornik 2011; Cohen, Hamilton, and Turner 2011). More advanced approaches may aim at clustering the documents into "similiar" sets of documents, e.g. via bag of words models (see Zhang, Jin, and Zhou 2010). This allows the journalist to find the story by reading just a few documents within each cluster. Mostly, a descriptive or visualizable result is the primary goal of such procedures, but in principle the analysis is not limited to that.

Regarding the Wikileaks Afghanistan war logs, all analyses so far, journalistic and scientific alike, have remained mostly on a descriptive level and therefore important insights from an inferential or modeling approach have not been gained. This could be due to the sheer bulk of the data. One of the peculiarities of the war log and its main challenge is that the data at hand stem from a database and that the information is captured in both numeric variables as well as written text. To neglect the written text in a statistical evaluation of such data sets would often come along with discarding important if not crucial information. Especially in the WikiLeaks data nearly all detailed information about the events is stored as written text. Thus it is essential for statistical evaluation to incorporate that information.

Modern statistical and data mining procedures provide tools to handle, analyze and model such data sets appropriately and therefore allowing a more thorough investigation. In this paper we will make exemplary use of such statistical learning procedures to analyze the number of fatalities in the war logs and to build statistical models. By combining two modern ideas, topic models and model-based recursive partitioning, our analysis allows to draw a bigger picture of the war from the thousands of mosaic tiles. In doing so, we present an approach that might be particularly suitable for data journalism, especially since in the end it provides palpable segments of data points characterized by a small number of parameters that directly relate to the question at hand.

The idea of our approach is as follows: Each single entry in the WikiLeaks war logs contains several variables but also a written report summary containing a short description of what happened in this particular incident. We are interested in extracting explanatory information from the reports, some type of meta information that aggregates reports with similar content. We achieve this by using Latent Dirichlet Allocation (LDA; Blei, Jordan, and Ng 2003) which

clusters written report summaries into latent topics. In a second step, we then use the generated topic assignments as further explanatory variables in modeling fatality rates in this data set. We use the provided fatality counts as our target variable. Since there is a high degree of overdispersion present, we model the number of fatalities with the negative binomial distribution (Lawless 1987). This enables us to estimate the average number of deaths per report appropriately. To allow for a flexible, non-linear functional relationship between explanatory variables and the fatality numbers, which also focuses on interactions, we employ the model-based recursive partitioning approach of Zeileis, Hothorn, and Hornik (2008).

The remainder of this paper is organized as follows: Section 2 contains a description of the WikiLeaks war logs. The methodological Section 3 presents the methods used in the present effort. The results are described and discussed in Section 4. We finish with conclusions in Section 5.

2. The Wikileaks Afghanistan War Logs

The release of 76911 individual war logs by `WikiLeaks.org` provides an unprecedented possibility to take a look at an ongoing war. The war logs cover the period from January 2004 to December 2009 and each event for which a report has been filed corresponds to a single document. Figure 1 displays the number of filed reports per month. While for the first years of the military operation we can find only a few hundreds of reports per month, this number increases up to more than 3500 in mid 2009.

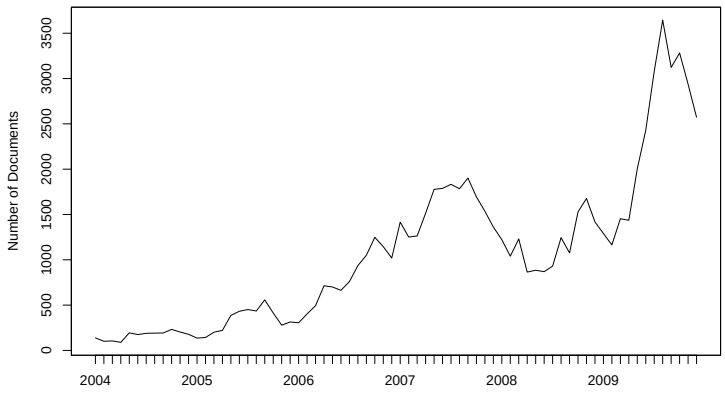

Figure 1: Monthly quantity of filed reports.

The report documents contain 32 columns with numerical and factor variables. They include four columns listing the number of "Civilian", "Enemy", "Friend" and "Host" fatalities within each report. The sum of the fatalities for each report serves as our target variable. Troops

Table 1: The number of casualties by group.

	Allied	Host	Civilian	ACF	Total
killed	1146	3796	3994	15219	24155
wounded	7296	8503	9044	1824	26667

fighting against coalition troops are referred to as "Enemies". We adopt the term "Anti-Coalition Fighters" (ACF) to describe this variable. The "Friends" column refers to ISAF forces including the NATO countries and the US military, while "Host" stands for local (Afghan) military and police. We subsume the former under "coalition troops" or "allied forces" and the latter under "Afghan or host forces".

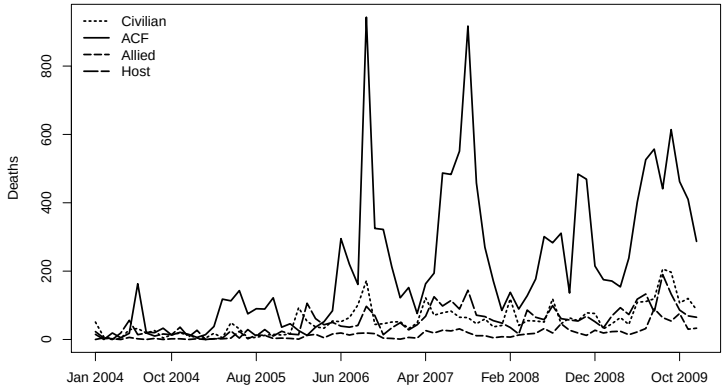

Figure 2: Monthly counts of fatalities by group.

Table 1 provides summary statistics for the casualties and Figure 2 displays a plot of the number of fatalities over time for each group during the observation period. In total we find 24155 fatalities in the war logs. 63% of the fatalities have been labeled as ACF. The second highest fatality number (16.54%) has been observed for civilians, closely followed by 15.72% Afghan soldiers and policemen and 1146 or 4.74% killed allied soldiers. Palpable are the two peaks for killed insurgents in late summer 2006 and 2007 in Figure 2. They account for 943 killed ACF fighters during September 2006 and for 917 in September 2007. The former peak corresponds to "Operation Medusa", an operation that had the aim to establish government control over areas of Kandahar province. The latter marks operations near Kandahar in an effort to remove insurgents who have returned to this area. Mid to late 2009 is the bloodiest period for civilians, coalition soldiers and ACF. Between May 2009 and December 2009 we observe 1056 (26.4%) out of 3994 civilian fatalities (see Table 1). In August 2009, during the period of the presidential elections (August 20) we observe 206 civilian victims and 190 killed ACF. For both groups, this has been the highest death toll within one month. Roughly the

same situation can be observed for allied soldiers. Here the monthly maximum of 90 deaths has happened in July 2009 and from May 2009 to December 2009 the data account for 346 (30.2%) killed allied soldiers.

Additional to the number of fatalities, the reports contain 28 columns with numerical and factor variables that serve as possible explanatory variables. We restrict ourselves to describing only those explanatory variables that were of special relevance for our analysis.

The factor `attackOn`, with its levels `FRIEND, NEUTRAL, ENEMY, UNKNOWN` encodes the US militaries point of view on whom an "attack" (action) has been directed during the incident. O'Loughlin *et al.* (2010, p. 474 ff) state that this variable seems to have been mislabeled and should have been named "attackBy". However, after inspection of a random sample of about 100 report summaries of the war logs we believe that `attackOn` does not contain information about who carried out a certain action but rather contains information about on whom the action described in the report has been directed. For instance, leaflets of Anti-Coalition Forces (ACF) calling for attacks against the US forces have been categorized as `attackOn=NEUTRAL`, fire fights between ACF and allied soldiers as `attackOn=ENEMY` and friendly fire has been labeled as `attackOn=FRIEND`.

The categorical variable `Dcolor` controls the display color of the message in the messaging system and map views. Messages relating to enemy activity have the color `red`, those relating to friendly activity have been colored `blue`, and `green` stands for neutral. This variable can be seen as the one encoding by whom an action has been carried out ("attackBy").

Another important variable for our analysis is `region`, roughly describing where an event took place. It has levels `RC NORTH, RC EAST, RC WEST, RC SOUTH, RC CAPITAL, UNKNOWN` and `NONE SELECTED` (RC stands for "Regional Command").

Last, there is `complexAttack`, a binary variable that encodes the complexity of an attack. The US military states an attack as complex if it has been well organized and executed, if soldiers have made use of heavy artillery and the troops have been able to withdraw from the battlefield in an organized fashion (see Roggio 2009).

The Report Summaries The variables described above, which may serve as explanatory variables for modeling the number of fatalities, only allow for a rather limited view into the events associated with each report and therefore the circumstances under which fatalities have happened. We can however find additional information about the context of the various incidents in the provided report summaries. These summaries contain a short verbal description of what has happened during the incident.

To give an example, on 19-Jul-2005 we can find the following report:

> On 19 July, at about 0730 hrs, a BBIED went off on an alleged suicide bomber targeting Enjeel district Chief of Police. As a result, the attacker was instantly killed, but no injures to anyone else was reported. Police investigation is ongoing.

The report summaries tell us the hows and whys of the mission in a very detailed way, something the other provided variables can not. Thus the report summaries and their content are at the core of evaluating the ongoings of this war as portrayed in the war logs as well as gaining insight into mortality in different situations. Disregarding these summaries in evaluating the war logs would be equivalent to discarding the most important information.

However, making use of this information is challenging. First, the summaries are plain natural language text which we need to process. Second, the sheer bulk of reports makes processing of the summaries by humans rather difficult. A person would have to read or process more than 76900 texts. If each summary takes a minute to read and file or process in any way, it would amount to approximately 1282 hours of work (or 160 work days if a work day consists of 8 hours).

There are three possible strategies to deal with such data: Either the reports are processed by crowdsourcing them to a high number of people. Or, if there is an *a priori* defined category system, one may classify the reports into these categories with a supervised approach. Both strategies were not feasible. Hence we used a technique that at the same time generates a category system and provides some kind of meta information to be used as explanatory variables by aggregating reports with similar content. These methods are known as "topic models" of which Latent Dirichlet Allocation (LDA; Blei *et al.* 2003) is a prime example.

3. Method

3.1. Using Topic Models To Build Explanatory Variables From Report Summaries

Latent Dirichlet Allocation (LDA) is a powerful document generative hierarchical model for clustering words into topics and documents into mixtures of topics. In LDA the topics are assumed to be uncorrelated (but see Blei and Lafferty 2007, for a version with correlated topics). Assuming that the similarity of the circumstances between reports is reflected in the words contained in the respective summaries, we can use LDA to assign reports based on their summaries to a number of topics lower than the number of documents. Hence, in this fashion we use the allocation of each report to (one or more) latent topic(s) as a task of complexity reduction or as a pre-processing step.

According to Blei and Lafferty (2009), topics are automatically discovered from the original texts and no *a priori* information about the existence of a certain theme is required. This means LDA generates the category system by itself. Only the number of topics for the whole set of documents has to be specified. The resulting topics are shared across the whole set of documents. Please note that in general the topic distribution of each report does only include non-zero probabilities.

Regarding the appropriateness of topic models for such a task, Chang, Boyd-Graber, Wang, Gerrish, and Blei (2009) presented results of a comparison of topic models with human classification. They concluded that "humans are able to appreciate the semantic coherence of topics and can associate the same documents with a topic that topic model does" (Chang *et al.* 2009, p. 8). Along similar lines, Griffiths and Steyvers (2004, p. 5228) note that "the extracted topics capture meaningful structure in the data, consistent with the class designations provided by the authors".

The Document Generative LDA Model

Following Blei and Lafferty (2009) and Blei (2012), LDA specifies the data-generating process as a probabilistic model, in which each document is a mixture of a set of topics and each word

in the document is chosen from the selected topic specific word distribution.

More formally, let q denote the size of a vocabulary (unique words within the considered corpus of documents) and let s be the number of topics $\boldsymbol{\beta}_t, t = 1, \ldots, s$. Each topic $\boldsymbol{\beta}_t$ is a q-dimensional symmetric Dirichlet distribution over the vocabulary with scalar parameter η. The only observed variables are words $\boldsymbol{w}_{1:h}$, where h denotes the number of documents and $w_{d,m} \in \{1, \ldots, q\}$ denotes the mth word of document d. The documents d, $d = 1, \ldots, h$ are sequences of those words of varying lengths q_d. Each document d is assigned to a topic with the assignment being denoted by z_d and and the topic assignment of each of its words $w_{d,m}$ is denoted by $z_{d,m}$. Each document is seen as a mixture of topics and hence each document has a vector of topic proportions denoted by $\boldsymbol{\pi}_d$ with $\pi_{d,t}$ denoting the proportion of topic t in document d. The distribution of $\boldsymbol{\pi}_d$ is a s-dimensional symmetric Dirichlet distribution with scalar parameter κ. Hence the generative model for LDA is

$$P(\boldsymbol{W}_{1:h}, \boldsymbol{\beta}_{1:s}, \boldsymbol{\pi}_{1:h}, \boldsymbol{Z}_{1:h}|\eta, \kappa) = \prod_{t=1}^{s} P(\boldsymbol{\beta}_t|\eta) \prod_{d=1}^{h} P(\boldsymbol{\pi}_d|\kappa) \left(\prod_{m=1}^{q_d} P(z_{d,m}|\boldsymbol{\pi}_d) P(W_{d,m}|\boldsymbol{\beta}_{1:s}, z_{d,m}) \right), \tag{1}$$

where the conditional distributions of the topic assignments and the words are assumed to be multinomial, i.e. $P(Z_{d,m}|\boldsymbol{\pi}_d) \sim \text{Multinomial}(\boldsymbol{\pi}_d)$ and $P(W_{d,m}|\boldsymbol{\beta}_{1:s}, z_{d,m}) \sim \text{Multinomial}(\boldsymbol{\beta}_{z_{d,m}})$. For estimation of the model we employed the variational EM-Algorithm, which has the effect that η can remain unspecified (see e.g. Grün and Hornik 2011). Since we use LDA to generate topics and assign each document to one of them, we need the posterior distribution of the latent topics, the topic assignment and the topic proportions given the documents,

$$P(\boldsymbol{\beta}_{1:s}, \boldsymbol{\pi}_{1:h}, \boldsymbol{Z}_{1:h}|\boldsymbol{w}_{1:h}, \eta, \kappa) = \frac{P(\boldsymbol{W}_{1:h}, \boldsymbol{\beta}_{1:s}, \boldsymbol{\pi}_{1:h}, \boldsymbol{Z}_{1:h})}{P(\boldsymbol{W}_{1:h})}, \tag{2}$$

and the conditional expectations $\hat{\boldsymbol{\beta}}_{t,u} = E(\boldsymbol{\beta}_{t,u}|\boldsymbol{w}_{1:h})$, $\hat{\boldsymbol{\pi}}_{d,t} = E(\boldsymbol{\pi}_{d,t}|\boldsymbol{w}_{1:h})$ as well as $\hat{z}_{d,t} = E(Z_d = t|\boldsymbol{w}_{1:h})$ with $u = 1, \ldots, q$.

In this analysis, we *a-priori* specified 100 latent topics to be estimated from the stop-word free corpus of stemmed words. In addition, we set the parameter κ of the symmetric Dirichlet distribution of the topic proportions to a very small value (0.001) in order to ensure that the estimated topic distribution for each document will assign a probability of nearly one to a single topic and very small probabilities to all other topics. This enables that the topic of each document is uniquely determined and allows to classify the documents into topics without loss of information by switching from soft to hard assignments. The resulting dummy variables that encode whether a document belongs to a topic or not then serve as possible explanatory variables for subsequent modeling of the fatality numbers.

3.2. Recursive Partitioning of Negative Binomial Distributions

To model the observations (number of fatalities per report) $Y_i, (i = 1, \ldots, n)$, with realisations y_i, we use trees with a pre-specified node model. These trees are flexible, non-linear algorithmic models that allow us to incorporate information of p observed explanatory variables $\mathbf{x}_i = (x_{1i}, \ldots, x_{pi})^T$. Here, the conditional distribution of Y, $D(Y|\cdot)$, is modeled as a partition function f depending on the state of p input vectors (explanatory variables), $\mathbf{x} = (x_1, \ldots, x_p)$, i. e.,

$$D(Y|\mathbf{x}) = D(Y|f(x_1, \ldots, x_p)) \tag{3}$$

where the function f partitions the overall covariate space $\mathcal{X}$ into a set of r disjoint segments $R_1, \ldots, R_r$ such that $\mathcal{X} = \bigcup_{k=1}^{r} R_k$ (Hothorn, Hornik, and Zeileis 2006). In each leaf R_k, a model for the conditional distribution is specified.

Our model for the conditional distribution $D(Y|\mathbf{x})$ within each segment $R_k, k = 1, \ldots, r$, is a negative binomial distribution with mean μ_k and dispersion parameter θ_k, i.e., having the probability mass function

$$P(Y = y|k; \mu_k, \theta_k) = \frac{\Gamma(y + \theta_k)}{\Gamma(\theta_k) y!} \left(\frac{\mu_k}{\mu_k + \theta_k}\right)^y \left(\frac{\theta_k}{\mu_k + \theta_k}\right)^{\theta_k} \tag{4}$$

with $y \in \{0, 1, 2, \ldots\}$, and $\Gamma(\cdot)$ denoting the gamma function. Mean and variance of Y for each segment R_k are given by (Lawless 1987)

$$\mathrm{E}(Y) = \mu_k \qquad \mathrm{Var}(Y) = \mu_k + \mu_k^2 \theta_k^{-1} \tag{5}$$

Please note that the above formulation pays dues to interpreting the negative binomial as a gamma mixture of Poisson distributions (Aitkin, Francis, Hinde, and Darnell 2009) and thus essentially being a Poisson model that can account for extra variation. It can be seen as a two-stage model for the discrete response Y in each segment R_k (cf. Venables and Ripley 2002),

$$Y|V \sim \mathrm{Poisson}(\mu_k V), \qquad \theta_k V \sim \mathrm{Gamma}(\theta_k). \tag{6}$$

Here V is an unobserved random variable having a gamma distribution with mean 1 and variance $1/\theta_k$. However, the marginal mean-variance identities for Y in (5) hold whenever V is a positive-valued random variable with mean 1 and variance θ_k^{-1} and V needs not necessarily be gamma-distributed (Lawless 1987). This node model integrates conceptually well with other approaches of using Poisson or Quasi-Poisson models to model fatalities. Using the negative binomial distribution however has the advantage over a Poisson model to account for extra variation and over Quasi-Poisson to integrate nicely into a maximum likelihood framework (see Venables and Ripley 2002). In principle, these other count data models might also be used as the node model. In fact, a Quasi-Poisson model tree approach for modeling overdispersed count data has been proposed by Choi, Anh, and Chen (2005). Their rationale is similar to ours, but we use negative binomial distributions to account for overdispersion and a tree algorithm that is unbiased in variable selection. The last point is very important for the correct interpretation of the tree structure (Loh and Shih 1997; Loh 2002; Kim and Loh 2001) and depends on the splitting procedure (Loh 2009).

Estimation

To estimate our proposed model, we employ the model-based recursive partitioning framework of Zeileis *et al.* (2008). We consider an intercept-only model estimated from a negative binomial likelihood which is then recursively partitioned based on the state of the partitioning covariates. For GLM-type models such as the negative binomial model, the algorithm is described in detail in Rusch and Zeileis (2011). This algorithm ensures that split variable selection is unbiased. Additionally, using trees has the advantage of inherent variable selection.

As tuning parameters for the tree algorithm we have the global significance level α of the generalized M-fluctuation tests (Zeileis and Hornik 2007) used for split variable selection and the minimum number of observations per node. Setting the former to low values can be regarded as pre-pruning to avoid overfit.

Eventually we get a classification of all observations into a set of segments or partitions $\mathcal{R} = \{R_1, \ldots, R_r\}$. The negative binomial distributions in these partitions are characterized by the parameter estimates $\hat{\mu}_k$ and $\hat{\theta}_k, k =, 1 \ldots, r$ and the estimated overall tree model by $\hat{\boldsymbol{\vartheta}} = ((\hat{\mu}_1, \hat{\theta}_1)^T, \ldots, (\hat{\mu}_r, \hat{\theta}_r)^T)$.

Pre-pruning the Trees To find sensible values for the significance level of the parameter stability test as well as for the minimal number of observations per node, we fitted different models using a grid of the two algorithm hyperparameters. Specifically, we used global significance levels α of 1×10^{-7}, 5×10^{-7}, 1×10^{-6}, 5×10^{-6}, 1×10^{-5}, 5×10^{-5}, 1×10^{-4}, 5×10^{-4}, 1×10^{-3}, 5×10^{-3}, 1×10^{-2} and 5×10^{-2}. Very low values for α were chosen because of the size of the data set (using significance levels of around 0.01 or higher might lead to spurious significances due to sample size). For the minimum number of observations per node we used values of 52, 100, 200, 300, 400, 500, 600 and 700. We then fitted a negative binomial model tree for all $12 \times 7 = 84$ combinations of hyperparameters and chose the tree that enabled the best explanation.

Interpretation of the Models

Basically interpretation happens on two levels: First, the level of the individual segments for which we get the estimated mean number of fatalities as well as the associated standard deviation. These fatality rates identify which segments come along with a higher or lower average death toll. Second, the level of the explanatory variables that define the segments. Here conclusions can be drawn about the specific circumstances that give rise to the different fatality rates in the segments. In case of topics as explanatory variables, we only look at which topics are selected for splitting and interpret them ex-post based on their most frequent words. Hence topics are used only for splitting without any further interpretation of or prior hypothesis about the underlying topic model. For readability we assign a unique name to each topic, but it should be kept in mind that those names are somewhat arbitrary. Since they are derive solely from the ten most frequent words as well as from looking at a random sample of assigned report summaries, they are necessarily neither exhaustive in their denotative and connotative meaning nor can they capture the circumstancial complexity of all assigned reports.

4. Results and Discussion

In our analysis the modeled response was the overall fatality number (sum of fatalities of civilians, the ACF, of Coalition troops and of Afghan police and soldiers). Detailed analyses for all groups separately can be found in Rusch, Hofmarcher, Hatzinger, and Hornik (2011).

Along the lines of the methodological procedure described above and to understand the fatality numbers associated with different circumstances, we first need the split information, i.e. which topics or further variables have been selected as splitting variables as well as where the split occured. Second, we need the estimated parameters of the segment-specific model, i.e. mean and dispersion. Accordingly, the split information is presented in Figure 3 and the logarithm of the estimated node model parameters in Table 4.1.

Regarding splits based on topics, a presentation of the selected estimated latent topics ten most frequent keywords and how many reports were assigned to them can be found in Table 3.

For instance, the report summary from Section 2 belongs to Topic 61, "Suicide and IED Bombing". In Table 3 the ten most frequent words of this (and all other topics) are displayed. Additionally, we can see in the first row of Table 3 (`numberDOC`) that overall 378 incidents were assigned to this topic.

In Figure 3 we visualize the negative binomial distribution in each terminal node by a parsimonious plot of the magnitudes of the mean and the standard deviation. The vertical line in each panel marks the location of the mean, the horizontal line shows the distance between zero and one standard deviation (cf. Friendly 2001). The height of the vertical line is the deviance divided by the degrees of freedom and indicates goodness of fit of the intercept-only model in the node. A smaller height means better fit.

We labeled the segments $k = 1, \ldots, r$ in an increasing order from right to left as they are displayed in the plot. This is of course arbitrary and should not imply a natural ordering of the k segments (terminal nodes). Each terminal node (leaf) k is associated with a negative binomial distribution with parameter estimates $\hat{\mu}_k$ and $\hat{\theta}_k$ and the vector of all parameters in the terminal nodes combined is the parameter vector of the final model. For each segment, Table 4.1 lists the segment number, parameter estimates and standard errors, degrees of freedom $(n_k - 1)$, deviance, the maximum number of fatalities and the percentage of incidents with no fatalities.

In what follows we discuss the results in more detail for some segments.

4.1. Fatalities in the War Logs

For all fatalities combined, we find $r = 14$ segments (with a global significance level for the fluctuation tests of $\alpha = 1 \times 10^{-4}$ and a minimum number of observations in each terminal node of 300). The resulting tree is depicted in Figure 3.

Overall the tree for the overall number of fatalities is dominated by fatalities of the ACF and of the civilian population. The tree itself is largely a combination of the trees for ACF and civilian fatalities alone (see Rusch *et al.* 2011). Our presentation will therefore mainly focus on ACF fatalities and civilian deaths, since those groups account for the highest number of deaths. Fatalities of allied forces and the troops of the host nation play a minor role for the overall number of deaths due to the comparatively small number of those fatalities (especially of allied forces) and the high congruency of civilian deaths and deaths of host nation troops[3].

The first three segments are dominated by reports listing high numbers of fatalities of the ACF. These reports belong either to "Task Force Reports (Bushmaster)" or are associated with incidents attributable to "Hostile Contacts ACF vs TF" in the South and elsewhere.

The first segment consists of $n_1 = 830$ incidents, with a maximum number of deaths of 101. 75.4% of the reports reported no fatalities. The average fatality number per report for this segment was $\hat{\mu}_1 = 2.18$ (2.1 for ACF alone). The 101 ACF deaths that mark the maximum death toll in this segment is the third highest death number in the whole war diary, as is

[3]For what follows, it should be noted that the entries in the database can be prone to data entry errors, mainly misclassification of fatalities to their respective group. For instance, the Kunduz Airstrike incident on 03-Mar-2009 lists 56 fatalities. All fatalities are stated to be "ACF fighters" in the war log. In the media however, the killed people were identified as being civilians (see guardian.co.uk 2010) who were invited by the Taliban to take fuel from stolen fuel trucks (see Amnesty International 2009). An allied airstrike against the fuel trucks killed those 56 civilians. This should be kept in mind, although generally there is a high congruency between the data in the WikiLeaks war log and other independent data sets (Bohannon 2011).

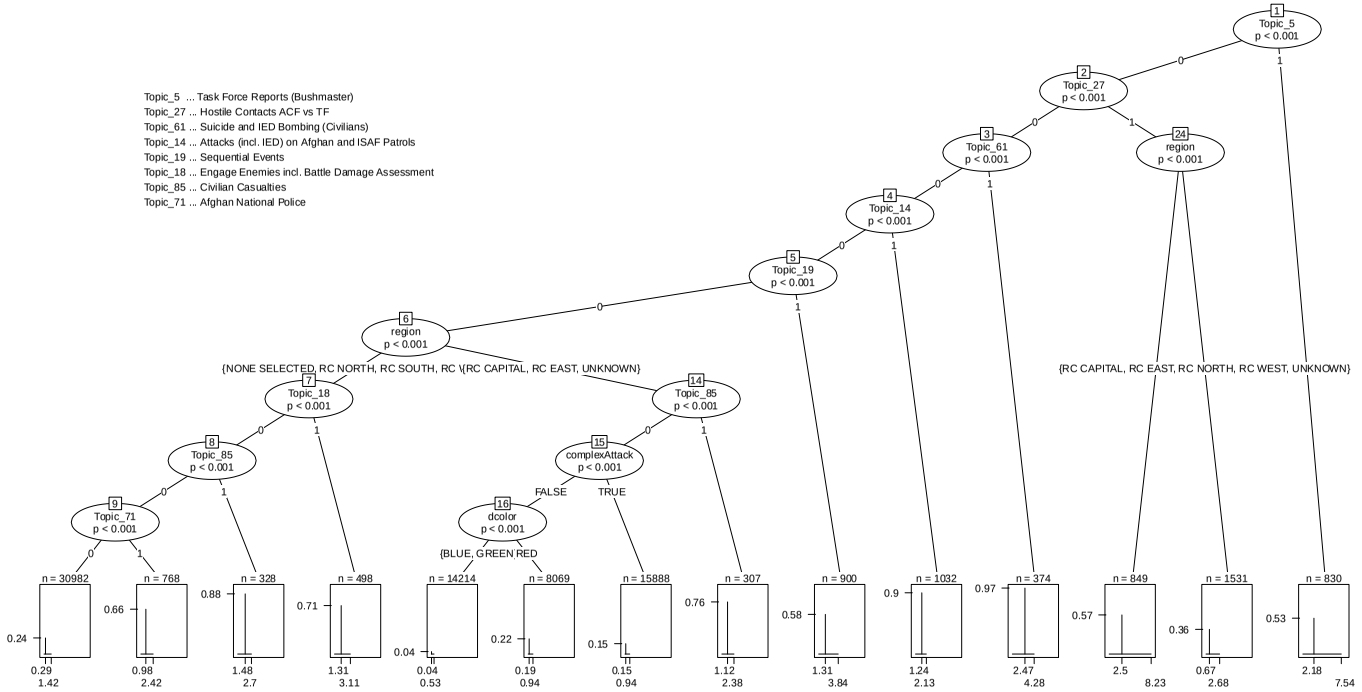

Figure 3: The negative binomial model tree for the combined fatalities. In the terminal nodes the vertical line marks the mean, the horizontal line the length between zero and one standard deviation and the height of the vertical line is the deviance divided by the degrees of freedom.

Table 2: Segment-wise statistics for all fatalities combined. The first column refers to the segment. For each segment we listed the logarithm of the estimated mean ($log(\hat{mu}_k)$), its standard error ($se(log(\hat{mu}_k))$), the estimated dispersion parameter ($\hat{\theta}_k$) and its standard error ($se(\hat{\theta}_k)$), the degrees of freedom (df), the residual deviance (Dev), the highest number of fatalities reported (max) and the percentage of reports with zero fatalities (%zero).

Segment	$log(\hat{\mu}_k)$	$se(log(\hat{\mu}_k))$	$\hat{\theta}_k$	$se(\hat{\theta}_k)$	df	Dev	max	%zero
R_1	0.779	.120	.089	.007	829	436.36	101	75.4
R_2	−0.399	.102	.069	.006	1530	554.37	68	84.8
R_3	0.917	.113	.096	.008	848	486.90	186	72.4
R_4	0.904	.090	.386	.038	373	361.19	36	42.8
R_5	0.215	.053	.468	.037	1031	926.77	31	53.8
R_6	0.269	.098	.128	.011	899	523.48	70	73.1
R_7	0.114	.121	.275	.039	306	234.08	43	63.2
R_8	−1.882	.049	.032	.002	15887	2418.40	25	94.6
R_9	−1.635	.054	.055	.003	8068	1801.90	28	92
R_{10}	−3.227	.113	.006	.001	14213	513.4	67	98.7
R_{11}	0.269	.106	.205	.022	497	353.50	56	66.3
R_{12}	0.389	.101	.373	.046	327	288.75	35	52.7
R_{13}	−0.016	.089	.199	.019	767	504.83	21	70.2
R_{14}	−1.238	.028	.048	.001	30981	7324.10	80	91

the mean fatality rate. All in all 1808 deaths are reported in this segment, 1712 of those are categorized as ACF. This segment is characterized by reports that belong to Topic 5 "Task Force Reports (Bushmaster)". Table 3 displays the most frequent words in the summaries of this and subsequent topics. For Topic 5 they were "task force", "fire", "close", "track", "insurgencies", "bushmaster", "isaf". Inspection of report summaries from this topic suggests that this segment refers to reports by US task forces (TF) with a focus on actions of task force unit "Bushmaster". TF "Bushmaster" is a task force consisting of Afghans and American green beret soldiers, the latter being a synonym for the United States Army Special Forces. According to Wikipedia they have "six primary missions: unconventional warfare, foreign internal defense, special reconnaissance, direct action, hostage rescue, and counter-terrorism. The first two emphasize language, cultural, and training skills in working with foreign troops. Other duties include combat search and rescue (CSAR), security assistance, peacekeeping, humanitarian assistance, humanitarian de-mining, counter-proliferation, psychological operations, manhunts, and counter-drug operations" (Wikipedia 2011). The topic mainly describes events or fights connected with this and other TF, including detention of individuals, fights and espionage.

The next two segments are governed by Topic 27 "Hostile Contacts ACF vs TF" and differ in terms of the region they took place. They describe incidents where task forces or ground troops had enemy contact in fire fights taking place (individual combat with small arms, see Table 3). Excluded from this topic are reports from Topic 5. Incidents assigned to this topic are further split according to the region where the events took place. The right branch in Figure 3 contains events around Kabul (`RC CAPITAL`), `RC EAST`, `RC WEST`, `RC NORTH` and `UNKNOWN` regions, as collected in segment R_2 which might be called "Hostile Contact ACF vs TF (not in the South)". These are associated with a death rate of $\hat{\mu}_2 = 0.671$ (0.6 for ACF

alone). Of these 1531 incidents the maximum number of fatalities is 68 and 84.8% reported no fatalities.

Of the reports belonging to Topic 27 "Hostile Contact ACF vs TF", the 849 events that happened in the South of Afghanistan (mainly provinces Kandahar and Helmand, `RC SOUTH`) show a much higher estimated fatality rate of $\hat{\mu}_3 = 2.501$ (2.4 for the ACF alone). This is the highest estimated death rate of the whole analysis. It can be explained by the South, especially the province of Kandahar, being Taliban heartland and their stronghold. It is therefore heavily attacked by coalition troops (see O'Loughlin *et al.* 2010). This result of higher death rates for incidents happening in the South is recurrent for all groups of fatalities (see Rusch *et al.* 2011). The segment "Hostile Contact ACF vs TF (South)" contains among others events that took place during Canadian-led "Operation Medusa", which began on September 2, 2006 and lasted until September 17 (see Wikipedia 2010). Reports in this segment (R_3) have a maximum number of fatalities of 186 on September 9, 2006. This report (its incident being part of "Operation Medusa") notes 181 killed ACF fighters, one killed coalition force soldier and four killed Afghan soldiers 10 km southwest of Patrol Base Wilson, in Kandahar province's volatile Zhari district. This is the highest number of killed ACF fighters (or overall death) in the whole data within a single war log entry. Moreover, segment R_3 is generally the segment with the highest ACF fatalities (see Rusch *et al.* 2011). Still for 72.4% of the documents in this segment no fatalities are reported.

The next three segments we discuss consist of incidents that are characterized by a high death toll of the civilian population mainly resulting from actions of the ACF.

First, there is Topic 61 "Suicide and IED Bombing" with corresponding segment R_4. It describes incidents that were related to suicide bombing attacks or other attacks with improvised explosive devices (IED) such as cars (cf Table 3). For example, one report assigned to Topic 61 and dated with 18-Feb-2008 reports 30 killed civilian due to a suicide bomb attack near Kandahar. It also includes reports where explosives were found or seized. The segments' $n_4 = 374$ reports list fatalities in 57.2% of the cases which makes it the only segment with a median death number higher than 0. The maximum number of killed people is 36. Accordingly, the estimated mean death rate for this segment is $\hat{\mu}_4 = 2.471$ (1.12 for civilians alone, the second highest civilian fatality rate). It is the second highest overall death rate per incident, closely matching the results from R_3. However, in R_4 "Suicide and IED Bombing" fatalities are mostly civilians or Afghan police forces, whereas deaths in R_3 "Hostile Contacts ACF vs TF (South)" are mostly ACF fighters. In R_4 we observe 924 deaths, 420 are civilian, followed by 246 killed afghan soldiers and 233 killed ACF fighters.

The next segment is R_7 "Civilian Casualties (East, Capital and unknown regions)" with an overall average number of fatalities of $\mu_7 = 1.12$. These are those $n_7 = 307$ incidents in the East, Capital or unknown region associated with Topic 85 "Civilian Casualties". In Table 3 we see the clear context of civilian fatalities of this topic. Out of the ten most frequent terms of this topic, six are synonyms respectively acronyms of civilians. These are: "ln" (local national), "local(s)", "civilian", "lns" (local nationals), "child", "nationals". The other four terms suggest a clear connection to casualties, namely "wound", "injur" (injury), "kill", "hospit" (hospital). The maximum number of fatalities in this segment is 43 and there are 63.2% of reports that list no fatality at all.

Segment R_{12} (governed by events from Topic 85 "Civilian Casualties" happening in the South, North, West or in a non-specified region) has an estimated mean of $\hat{\mu}_{12} = 1.476$. The per-

centage of reports without killings is 52.7% and the highest death toll is 35. The governing topic, Topic 85, appeared before as the governing topic of R_7. Therefore R_{12} and R_7 are corresponding topic-wise and only differ in terms of their location. It is interesting to see that R_{12} has a higher fatality number per incident, most probably due to events in the south. Incidents in Kabul and the East (R_7) are associated with lower death numbers and a higher percentage of reports with zero deaths. However, the report with the highest fatality number for this topic is part of R_7, describing an attack on the Indian Embassy in Kabul where 42 civilians and one Taliban were killed.

When looking at civilian fatalities alone (see Rusch *et al.* 2011), incidents from Topic 85 "Civilian Casualties" have the overall highest observed civilian death toll for action of the ACF, either against civilians or where civilians are "collateral damage" (on average 1.7 deaths per incident). Hence, incidents from this topic as well as incidents in Topic 61 "Suicide and IED Bombing" have in common that the attacks were overwhelmingly carried out by the ACF and were directed at places where there is a high number of the civilian population present, such as buses, bazars or markets. In contrast, for incidents which refer to actions of ISAF troops also belonging to Topic 85 "Civilian Casualties", we have about 25% of the former rate (0.41 deaths per incident, the fourth highest overall rate for civilians). Thus ACF action is associated with a fourfold increase in expected civilian fatalities for reports belonging to this topic. It is a clear and consistent finding that actions of the ACF come along with a higher civilian death toll than actions of the allied forces. Generally, when analyzing civilian fatalities alone, most resulting segments with high civilian fatality rates have in common that they are connected to attacks by the ACF often with improvised explosive devices (see also Bohannon 2011).

Topic 14 "Attacks (incl. IED) on Afghan and ISAF patrols" gives rise to segment R_5 with an average number of deaths per incident of $\hat{\mu}_5 = 1.241$ (0.32 for the civilian population and 0.51 for Afghan troops). In total, we observe 1287 deaths in the $n_5 = 1032$ reports (53.8% of whom had no deaths reported) in this segment. It is somewhat hard to identify the governing topic with an unique theme like before but inspecting a sample of report summaries indicates that this topic collects reports which describe explosions of IED or smaller fights or incidents following attacks by the ACF mainly with Afghan and some ISAF forces that were patrolling, resulting battle damage assessment (bda) and medical evacuation. Most victims in this segment are therefore Afghan soldiers (529), but we also observe 326 killed civilians, 170 ACF and 262 killed allied soldiers.

The last segment we discuss is governed by fatalities for the host nation troops. In the regions `RC NORTH`, `RC SOUTH`, `RC WEST` or unspecified regions, Topic 71 "Afghan National Police" gives rise to segment R_{13} with nearly one death per incident on average ($\hat{\mu}_{13} = 0.984$). Of the $n_{13} = 768$ events 70.2% did not result in deaths. Topic 71 can be categorized as describing events with an involvement of the Afghan National Police (ANP). Often, these were attacks on ANP checkpoints or police stations or police patrols. When looking at the fatalities of Afghan troops alone, the south is once again connected with a higher fatality rate for incidents belonging to Topic 71 (0.78 vs. 0.4).

It should also be noted (and that finding is consistent throughout all the fatality groups) that segments containing by far the largest number of reports have on average relatively low death rates per incident. For all fatalities, these are segments R_{14}, R_{10} and R_8 with $\mu_{14} = 0.29$, $\mu_{10} = 0.04$ and $\mu_8 = 0.15$. They contain about 80% of all reports. Hence most of the every day happenings in this war come along with a low death toll. Only in case of certain events this

number increases. This increase is mainly connected to either fights between allied forces and the Taliban and other ACF groups (leading to high ACF fatality numbers) or characterized by attacks by the ACF who aim at or tolerate civilian casualties (leading to high civilian or Afghan troop fatality numbers).

5. Conclusions

Undoubtedly, innovations like the internet have changed the supply of potential data of interest. For science as well as journalism, it is unavoidable to gather, manage and process this bulk of information. Central to this is reading, interpreting and understanding text documents with the aid of automated procedures. The foreseeable increase of available written information, e.g. in the world wide web, will even increase the need for such methods. At least partly, this has nourished data journalism where the database becomes the center of journalistic work. This paper illustrates how modern statistical procedures can provide aid in extracting relevant information from bulks of written text documents or from a database and how they may help in processing and structuring the information to facilitate interpretation of the data, as has been the primary goal of statistical modeling ever since.

Text mining tools and topic models were used to analyze written text from the WikiLeaks war diary automatically by assigning overarching themes to the single documents. This allowed to get a view on the data which is hard to obtain by manual processing and that may even discover connections between documents which may not be at all obvious. The assignment of topics to the single documents offered the opportunity to use those topics as explanatory variables in further data analysis. One has to bear in mind, however, that the assignment of documents to topics is by far not absolute and that it can be difficult to interpret the meaning of latent topics, especially if they are to be named (as is often the case with unsupervised techniques). At any rate, we saw that explanatory variables generated by pre-processing with LDA proved to be very important in subsequent modeling whereas the variables that were already available played a minor role. Hence, discarding the information stored in the report summaries would have lead to completely different models or interpretation.

Model-based trees were then used to model the data flexibly and accurately as well as for providing an intuitive association of circumstances and fatalities. A representative data model (here the negative binomial distribution) was used to relate the observations to the question at hand. Instead of simply calculating the arithmetic mean of the dependent variable, the underlying model takes a whole likelihood for overdispersed count data into account when estimating mean fatality rates, which is suitable for the description of rare events. Pre-pruning with an inferential splitting procedure led to a tree that has useful explanatory power as well as fits the data at hand very well (usually better than would be expected[4]). The model-based approach we chose offered additional insight as to how the fatality rates for specific incidents looked like, something that has not been done so far for this war.

This clearly illustrates the high potential that text mining procedures on the one hand and model-based recursive partitioning on the other have for a wide range of possible applications in socio-economic sciences (see, e.g. Kopf, Augustin, and Strobl 2010) as well as data journalism, especially if the data stem from a database or consist of both numerical variables and written text which has to be analyzed.

[4]Residual deviance was often much smaller than the degrees of freedom.

Acknowledgement

The authors want to thank Bettina Grün and Achim Zeileis for useful discussions and expert advice.

A. Frequent Terms of the Topics

In Table 3 a list of the ten most frequent terms for each topic as well as their occurence for different fatality groups and the number of documents assigned to them can be found.

B. Computational Details

All calculations have been carried out with the statistical software R 2.12.0-2.14.1 (R Development Core Team 2011) on `cluster@WU` (FIRM 2011). Topic models were estimated with the extension package **topicmodels** 0.0-7 (Grün and Hornik 2011). Further packages used were **slam** 0.1-18 and **tm** 0.5-4.1. Recursive partitioning infrastructure was provided by the function `mob()` (Zeileis *et al.* 2008) from the package **party** 0.9-99991. Further packages used were **strucchange** 1.4-3. The negative binomial family of models used for `mob()` is based on the implementation of `glm.nb()` in package **MASS** 7.3-7 (Venables and Ripley 2002) and can be obtained from the corresponding author.

Table 3: The ten most frequent terms of the estimated latent topics and the number of documents assigned. A × denotes that this topic serves as a split variable for the mentioned subgroup as well.

	Topic 5	Topic 14	Topic 18	Topic 19	Topic 27	Topic 61	Topic 71	Topic 85
numberDOC	830	1035	508	900	2382	378	1288	638
CIVILIAN		×				×	×	×
ACF	×		×	×	×			
ISAF		×				×		
HOST		×				×	×	
	tf	wia	engag	updat	fire	suicid	anp	ln
	bushmast	ie	bda	att	enemi	bomber	cp	wound
	fire	kia	ground	aaf	contact	deton	attack	local
	forc	strike	damag	pax	tf	vest	event	civilian
	isaf	bda	mm	saf	tic	attack	close	hospit
	close	cat	fire	event	element	nds	qrf	kill
	track	medevac	ah	contact	acm	khowst	isaf	injur
	friend	struck	compound	station	arm	explos	checkpoint	lns
	insurg	vehicl	kill	vc	receiv	svbi	ie	child
	event	isaf	pid	fire	saf	kill	wia	nation

References

Aitkin M, Francis B, Hinde J, Darnell R (2009). *Statistical Modelling in R*. Oxford University Press, New York.

Amnesty International (2009). "Afghanistan: German Government must investigate deadly Kunduz Airstrikes." http://www.amnesty.org/en/news-and-updates/news/afghanistan-german-government-must-investigate-deadly-kunduz-airstrikes-20091030. [Online; accessed 07-March-2011].

Bhutta ZA (2002). "Children of war: the real casualties of the Afghan conflict." *British Medical Journal*, **324**, 349–352.

Bird SM, Fairweather CB (2007). "Military fatality rates (by cause) in Afghanistan and Iraq: a measure of hostilities." *International Journal of Epidemiology*, **36**, 841–846.

Blei D (2012). "Probabilistic Topic Models." *Communications of the ACM*, **55**(4), 77–84. Online available at http://cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext[accessed a preliminary version of 07-Sep-2011].

Blei DM, Jordan MI, Ng A (2003). "Latent Dirichlet Allocation." *The Journal of Machine Learning*, **3**, 993–1022.

Blei DM, Lafferty JD (2007). "A correlated topic model of SCIENCE." *Annals of Applied Statistics*, **1**(1), 17–35.

Blei DM, Lafferty JD (2009). "Topic Models." In A Srivastava, M Sahami (eds.), *Text Mining: Classification, Clustering, and Applications*. Chapman & Hall/CRC Press.

Bohannon J (2011). "Counting the Dead in Afghanistan." *Science*, **331**(6022), 1256–1260.

Bortkiewicz L (1898). *Das Gesetz der kleinen Zahlen [The law of small numbers]*. B.G. Teubner, Leipzig. URL http://www.archive.org/download/dasgesetzderklei00bortrich/dasgesetzderklei00bortrich.pdf.

Burnham G, Lafta R, Doocy S, Roberts L (2006). "Mortality after the 2003 invasion of Iraq a cross-sectional cluster sample survey." *Lancet*, **368**, 1421–1428.

Buzzell E, Preston SH (2007). "Mortality of American Troops in the Iraq War." *Population and Development Review*, **33**, 555–566.

Chang J, Boyd-Graber J, Wang C, Gerrish S, Blei DM (2009). "Reading Tea Leaves: How Humans Interpret Topic Models." In *Neural Information Processing Systems*.

Choi Y, Anh H, Chen J (2005). "Regression trees for analysis of count data with extra poisson variation." *Computational Statistics & Data Analysis*, **49**, 893–915.

Cohen S, Hamilton JT, Turner F (2011). "Computational Journalism: How computer scientists can empower journalists, democracy's watchdogs, in the production of news in the public interest." *Communciations of the ACM*, **54**(10), 66–71. URL http://cacm.acm.org/magazines/2011/10/131400-computational-journalism/fulltext.

Conway D (2010). "Wikileaks Afghanistan Data." http://www.drewconway.com/zia/?p=2226. [Online; accessed 07-March-2011].

Degomme O, Guha-Sapir D (2010). "Patterns of mortality rates in Darfur conflict." *Lancet*, **375**(9711), 294–300.

Elsea J (2011). *Criminal Prohibitions on the Publication of Classified Defense Information*. Congressional Research Service, Washington, DC.

FIRM (2011). "Cluster@WU." http://www.wu.ac.at/firm/cluster_folder. [Online; accessed 24-March-2011].

Friendly M (2001). *Visualizing categorical data*. SAS publishing, Cary, North Carolina.

Garfield RM, Neugut AI (1991). "Epidemiologic Analysis Of Warfare - A Historical Review." *Journal of the American Medical Association*, **266**, 688–692.

Gebauer M (2010). "Explosive Leaks Provide Image of War from Those Fighting It." http://www.spiegel.de/international/world/0,1518,708314,00.html. [Online; accessed 07-March-2011].

Griffiths TL, Steyvers M (2004). "Finding scientific topics." *PNAS*, **101**, 5228–5235.

Grün B, Hornik K (2011). "topicmodels: An R Package for Fitting Topic Models." *Journal of Statistical Software*, **40**(13), 1–30. URL http://www.jstatsoft.org/v40/i13/.

guardiancouk (2010). "Afghanistan war logs: 56 civilians killed in Nato bombing." http://www.guardian.co.uk/world/afghanistan/warlogs/826B488C-EA6F-A132-511610DB68C2EDBD. [Online; accessed 07-March-2011].

Haushofer J, Biletzki A, Kanwisher N (2010). "Both sides retaliate in the Israeli-Palestinian conflict." *PNAS*, **107**(42), 17927–17932. doi:10.1073/pnas.1012115107.

Hofmarcher P, Theußl S, Hornik K (2011). "Do Media Sentiments Reflect Economic Indices?" *Chinese Business Review*, **10**, 487–492.

Holcomb JB, McMullin NR, Pearse L, Caruso J, Wade CE, Oetyen-Gerdes L, Champion HR, Lawnick M, Farr W, Rodriguez S, Butler FK (2007). "Causes of death in US Special Operations Forces in the global war on terrorism - 2001-2004." *Annals of Surgery*, **245**, 986–991.

Hothorn T, Hornik K, Zeileis A (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics*, **15**, 651–674.

Kim Y, Loh W (2001). "Classification trees with unbiased multiway splits." *Journal of the American Statistical Association*, **96**, 589–604.

Kopf J, Augustin T, Strobl C (2010). "The Potential of Model-Based Recursive Partitioning in the Social Sciences – Revisiting Ockham's Razor." *Technical report*, Ludwig-Maximilians University, Munich.

Lakstein D, Blumenfeld A (2005). "Israeli army casualties in the second Palestinian uprising." *Military Medicine*, **170**, 427–430.

Lawless JF (1987). "Negative Binomial and Mixed Poisson Regression." *The Canadian Journal of Statistics*, **15**, 209–225.

Loh WY (2002). "Regression trees with unbiased variable selection and interaction detection." *Statistica Sinica*, **12**, 361–386.

Loh WY (2009). "Improving the Precision of Classification Trees." *Annals of Applied Statistics*, **3**, 1710–1737.

Loh WY, Shih YS (1997). "Split selection methods for classification trees." *Statistica Sinica*, **7**, 815–840.

Marshall H, Balfour TG (1838). *Statistical Report on the Sickness, Mortality, & Invaliding among the troops in the West Indies*. W. Clowes and Sons, London. URL http://books.google.com/books/download/Statistical_report_on%_the_sickness_morta.pdf?id=Vb4NAAAAQAAJ&output=pdf&sig=ACfU3U07DzW5FQxOyeWXLyS%bp-PuuaVvEQ.

Nightingale F (1863). *Notes on Hospitals*. 3rd edition. Longman, Green, Longman, Roberts, and Green, London. URL http://books.google.com/books/about/Notes_on_hospitals.html?id=k_w5uPm0D-cC.

O'Loughlin J, Witmer FD, Linke AM, Thorwardson N (2010). "Peering into the Fog of War: The Geography of the Wikileaks Afghanistan War Logs, 2004–2009." *Eurasian Geography and Economics*, pp. 1–24.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statstical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rogers S (2010). "Wikileaks' Afghanistan war logs: how our datajournalism operation worked." http://www.guardian.co.uk/news/datablog/2010/jul/27/wikileaks-afghanistan-data-datajournalism. [Online; accessed 07-March-2011].

Roggio B (2009). "US, Afghan troops beat back bold enemy assault in eastern Afghanistan." http://www.longwarjournal.org/archives/2009/10/us_afghan_troops_bea.php.

Rusch T, Hofmarcher P, Hatzinger R, Hornik K (2011). "Modeling Mortality Rates In The WikiLeaks Afghanistan War Logs." *Technical Report 112*, Research Report Series, Institute for Statistics and Mathematics, WU Wirtschaftsunversität Wien, Vienna.

Rusch T, Zeileis A (2011). "Gaining Insight with Recursive Partitioning of Generalized Linear Models." *Technical Report 109*, Research Report Series, Institute for Statistics and Mathematics, WU Wirtschaftsunversität Wien, Vienna.

Spiegel PB, Salama P (2001). "War and mortality in Kosovo, 1998-99: an epidemiological testimony." *Lancet*, **355**, 2204–2209.

Thomas TL, Parker AL, Horn WG, Mole D, Spiro TR, Hooper TI, Garland FC (2001). "Accidents and injuries among US Navy crewmembers during extended submarine patrols, 1997 to 1999." *Military Medicine*, **166**, 534–540.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer, New York. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Wikipedia (2010). "Operation Medusa — Wikipedia, The Free Encyclopedia." http://en.wikipedia.org/wiki/Operation_Medusa. [Online; accessed 20-December-2010].

Wikipedia (2011). "Special Forces (United States Army) — Wikipedia, The Free Encyclopedia." http://en.wikipedia.org/wiki/Special_Forces_(United_States_Army). [Online; accessed 05-June-2011].

Zeileis A, Hornik K (2007). "Generalized M-Fluctuation tests for parameter instability." *Statistica Neerlandica*, **61**, 488–508.

Zeileis A, Hothorn T, Hornik K (2008). "Model-based recursive partitioning." *Journal of Computational and Graphical Statistics*, **17**, 492–514.

Zhang Y, Jin R, Zhou ZH (2010). "Understanding bag-of-words model: a statistical framework." *International Journal of Machine Learning and Cybernetics*, **1**, 43–52.

Affiliation:

Thomas Rusch
Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien
Augasse 2–6
1090 Wien, Austria
E-mail: Thomas.Rusch@wu.ac.at

5. Influencing Elections with Statistics: Targeting Voters with Logistic Regression Trees

Influencing Elections with Statistics: Targeting Voters with Logistic Regression Trees

Thomas Rusch
WU (Wirtschaftsuniversität Wien)

Ilro Lee
University of New South Wales

Kurt Hornik
WU (Wirtschaftsuniversität Wien)

Wolfgang Jank
University of South Florida

Achim Zeileis
Universität Innsbruck

Abstract

Political campaigning has become a multi-million dollar business. A substantial proportion of a campaign's budget is spent on voter mobilization, i.e., on identifying and influencing as many people as possible to vote. Based on data, campaigns use statistical tools to provide a basis for deciding who to target. While the data available is usually rich, campaigns have traditionally relied on a rather limited selection of information, often including only previous voting behavior and one or two demographical variables. Statistical procedures that are currently in use include logistic regression or standard classification tree methods like CHAID, but there is a growing interest in employing modern data mining approaches. Along the lines of this development, we propose a modern framework for voter targeting called LORET (for logistic regression trees) that employs trees (with possibly just a single root node) containing logistic regressions (with possibly just an intercept) in every leaf. Thus, they contain logistic regression and classification trees as special cases and allow for a synthesis of both techniques under one umbrella. We explore various flavors of LORET models that (a) compare the effect of using the full set of available variables against using only limited information and (b) investigate their varying effects either as regressors in the logistic model components or as partitioning variables in the tree components. To assess model performance and illustrate targeting, we apply LORET to a data set of 19,634 eligible voters from the 2004 US presidential election. We find that augmenting the standard set of variables (such as age and voting history) together with additional predictor variables (such as the household composition in terms of party affiliation and each individual's rank in the household) clearly improves predictive accuracy. We also find that LORET models based on tree induction outbeat the unpartitioned competitors. Additionally, LORET models using both partitioning variables and regressors in the resulting nodes can improve the efficiency of allocating campaign resources while still providing intelligible models.

Keywords: campaigning, classification tree, get-out-the-vote, logistic regression, model tree, model-based recursive partitioning, political marketing, voter identification, voter segmentation, voter targeting.

1. Introduction

"Decisions are made by those who show up", said President Bartlet, a character from a popular

TV show, *The West Wing*. The character in the show used the line to motivate a college audience to voice their opinion by showing up at the polls. Getting eligible voters to actually vote ("get-out-the-vote"; GOTV) is an important goal in countries with a democratic political system and a lot of resources are spent on achieving that goal. Take the 2008 US presidential race for example. In that year, the world witnessed the amount of money raised and spent reaching unprecedented heights. By spending over USD 1 billion, the Obama and McCain campaigns tried to persuade and mobilize voters to engage in the political process by casting their vote on November 4th. However, even with monumental campaign effort, and large out-laying of resources, only 61.7% of eligible voters did cast their ballot.

1.1. Campaigning, mobilization and turnout in the United States

The impact of partisan campaigning or nonpartisan get-out-the-vote efforts on mobilization and turnout has been subject to numerous scientific investigations over the last 20 years see e.g., Whitelock, Whitelock, and van Heerde (2010); Baek (2009); Karp and Banducci (2007); Steel, Pierce, and Lovrich (1998); Finkel (1993); Gelman and King (1993). Starting from an early 'minimal effect' hypothesis (i.e., the idea that political campaigns only marginally mobilize, persuade or convert voters), the general sentiment nowadays is that campaigning does indeed have measurable effects on mobilization of (core) supporters (Holbrook and McClurg 2005; Hillygus and Jackman 2003). This mobilization, in turn, has been shown to have an effect on increasing overall turnout and on getting additional votes for a specific candidate (Holbrook and McClurg 2005; Cox and Munger 1989).

As a result, campaigns are spending huge amounts of money on mobilizing voters. Despite this spending, campaigns often fail to mobilize voters for the campaign's cause. Take the United States for example, where the "professionalization" (Muller 1999) of campaigning has had its origin[1] (Plasser 2000). Arguably, nowhere else is political campaigning a bigger business then in the United States. However, despite increased political consultancy and the hundreds of millions of campaign spending, the average voter turnout since 1980 during the Presidential election years has only been 56%; see also Table 1.

Table 1 shows voter turnout, total spending of presidential candidates since 1980 as well as total spending adjusted for inflation at 2008 CPI (consumer price index) rates (i.e., real expenditures). Figure 1 shows the bivariate relationship between turnout and the logarithm of real total campaign expenditures per eligible voter along with a fitted linear regression line. While some caution is warranted when interpreting a linear regression fitted to just 8 observations, there is clearly a positive association. The 2004 and 2008 elections saw especially increased expenditures per voter accompanied by a noticable increase in voter turnout. Given the relationship between campaign spending and turnout, campaigns are well advised to spend money on mobilizing voters (Baek 2009; Hall and Bonneau 2008). However, as campaigns increasingly face limited resources and budget constraints (in addition to public sentiment against excessive spending during times of economic hardship), it is important to allocate resources as efficiently as possible.

From a marketing point of view, voter mobilization is a two-step process (cf. Goldstein and Ridout 2002). In the first step, campaigns need to craft measures that best motivate people

[1] The professionalization of politial campaigning spread form the US to many democratic countries all over the world (Sussman and Galizio 2003). Accordingly we will focus on the US system but the ideas are easily generalizable to other democratic countries as well.

Year	Turnout (in %)	Expenditures (in mill. USD)	Real expenditures (at 2008 rates)
2008	61.7	1,324.7	1,324.7
2004	60.1	717.9	818.2
2000	54.2	343.1	429.0
1996	51.7	239.9	329.2
1992	58.1	192.2	295.0
1988	52.8	210.7	383.5
1984	55.2	103.6	214.7
1980	54.2	92.3	241.2
Mean	56.0	403.1	504.4
Sd	3.6	422.0	381.6
Min	51.7	92.3	214.7
Max	61.7	1,324.7	1,324.7

Table 1: Individual and aggregated turnout rate (votes for highest office divided by the voting-eligible population) for presidential elections in the United States and the money spent by all candidates (in million USD). The fourth column lists the real expenditures (inflation-adjusted at 2008 rates). Source: McDonald (2012) and `http://www.opensecrets.org/` and `http://www.bls.gov/data/inflation_calculator.htm`.

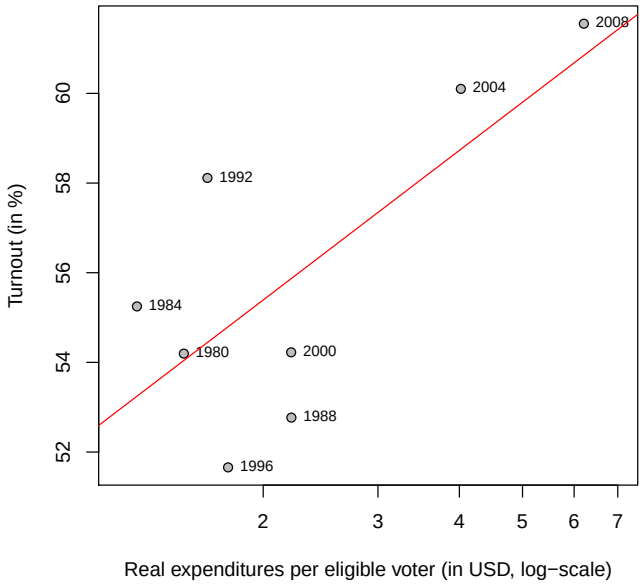

Figure 1: The relationship between real expenditures per eligible voter (in USD, log-scale) and turnout (in %) in the US since 1980 along with fitted linear regression (solid line). The model has $R^2 = 0.52$ and its slope implies an expected increase of turnout by 0.5 percentage points for a 10% increase of expenditures per eligible voter.

	2008	2006	2004
Expenditures (in USD)	2,501,605	2,231,941	1,848,822
Votes received	164,562	71,651	113,040
Cost per vote (in USD)	15.20	31.15	16.36

Table 2: Election costs for Congressman Barrow (Georgia's 12th congressional district). Source: Secretary of State, Georgia, http://www.opensecrets.org/.

to turn up at the polls, i.e., to assure the effectiveness of mobilization. In the second step, campaigns need to identify suitable people that should be subjected to these measures (also known as voter targeting).

The first step includes decisions on which marketing measures to use. Since many measures lack in effectiveness, numerous studies have been designed on this subject, investigating diverse measures such as TV ads, canvassing and face-to-face contacting, telephone calls or negative campaigning (e.g., Green and Gerber 2008; Hansen and Bowers 2009; Ridout 2009; Lau, Sigelman, and Rovner 2007; Gillespie 2010).

In the second step of the marketing process, campaigns need to *identify* the "right" recipients for these marketing messages. To our knowledge, there has only been little work on this topic in the literature, some notable exceptions including Wielhouwer (2003); Parry, Barth, Kropf, and Jones (2008) or Murray and Scime (2010). Identifying the right people to target is important because it reduces wasteful spending (e.g., targeting a person who is very unlikely, or not even eligible, to vote would be considered extremely wasteful) and allows campaigns to efficiently allocate their limited resources.

As point in case consider Table 2 which shows how much money has been spent by the campaign of Congressman Barrow of Georgia's 12th congressional district, and how many votes he received in three consecutive election years (in the US, members of the house of representatives get elected every two years). During each of the three elections, the campaign spent similar amounts of money, however, in 2006 it cost the campaign double the amount of money for each vote it received. Assuming that the campaign was targeting roughly the same voters in all three elections, one cannot help but wonder whether it would have been better to target a lower number of people in 2006 (a midterm election year where turnout is generally lower). The identification of people who only vote in general elections might have helped in order to spend the available resources more efficiently. For such a precise identification of voters, statistics offers a number of suitable tools.

1.2. Voter targeting in get-out-the-vote (GOTV) campaigns

In his standard source on political targeting, Malchow (2008, p. 1) defines voter targeting as "... the process by which a campaign predicts which voters it needs to persuade to win." These voters include those who are undecided as well as those who are in favor of the issue at hand, at least in principle, but who need some encouragement to turnout. Malchow (2008) opines that efficient identification and prediction of which voter should be targeted is going to be one of the future major issues in campaigning. This is also reflected in his alternate definition of targeting as being "the process of determining which voters you need for victory and identifying them as efficiently as possible" (Malchow 2008, p. 7).

This goal of voter targeting shares similar objectives with that of consumer targeting in mar-

keting. However, there are several structural reasons as to why political campaign marketing differs from that of consumer marketing. Following Quelch (2008) these are: (i) the lower number of choices for voters in general elections than for consumers, (ii) that voters have to live with the majority's decision which might dampen their enthusiasm and (iii) that most of the voters only get to vote every couple of years on a fixed date while consumers can usually decide when to when and where purchase. Additionally, (iv), singling out a niche may work fine for marketing a product, but politicians cannot win by targeting just a single segment as they need to get the majority of votes. This may be the reason why political marketing is generally considered to be less successful than consumer marketing.

How is targeting carried out?

Campaigns basically try to mobilize voters who (however loosely) identify themselves with any party or candidate. They do not necessarily try to convince voters to cast their ballot for a specific candidate. Thus they may simply aim at increasing the number of people who show up at polls. Malchow (2008) describes targeting for turnout as a targeting procedure for which the campaign needs to know or predict the likelihood that a voter will actually vote, regardless of whether it is for persuasion or mobilization purposes, as well as making a strategic decision which range of prediction is of interest.

To make such predictions, campaigns are employing many different techniques, some of which are founded in statistical reasoning. This also pertains to the campaigns gearing up for the 2012 presidential election which are showing a strong interest in statistics for decision making. President Obama's campaign is actively seeking for data miners to join his campaign for reelection[2][3][4]. In addition, not only the incumbent is seeking help from statistics, but also some of his challengers such as the Texas Governor, Rick Perry[5][6]. The increased media coverage of the importance of statistics in election campaigns supports this effect.

When targeting voters, campaigns rely on data that are either public or proprietary. Public data offer a limited number of variables such as aggregate number of turnout, while data sets from proprietary sources often contain much richer information. Usually the most important variables that are collected are records of the individual voting history. The aptitude of voting history as a predictor for future election attendance has already been established (Denny and Doyle 2009) and consequently it is the gold standard in targeting (Malchow 2008). However there might be predictive power in additional variables that are often ignored.

Traditionally, campaigns have relied on simple deterministic rules for choosing who to target, e.g., using information from the last four comparable elections as the main predictors for future voting behavior. Intuitively, someone who voted in all four out of the last four elections is seen as a most likely voter whereas someone who did not vote in any of the four elections is very unlikely to vote now. However, forecasting the behavior of persons with other patterns (i.e., who voted sometimes but not always) is not clear in this very simple setup.

This has sparked interest in adopting probabilistic approaches in place of deterministic rules

[2] http://www.cnn.com/2011/10/09/tech/innovation/obama-data-crunching-election/index.html

[3] http://andrewgelman.com/2011/10/data-mining-efforts-for-obamas-campaign/

[4] http://www.politico.com/blogs/bensmith/0711/Obama_campaign_hiring_data_mining_scientists.html

[5] http://thecaucus.blogs.nytimes.com/2011/08/22/rick-perrys-scientific-campaign-method/

[6] http://www.campaignsandelections.com/magazine/us-edition/267417/technology-bytes-perryand39s-social-scientists-and-obamaand39s-data-brigade.thtml

based solely on the voting history. For instance, Malchow (2008) promotes a linear probability model as well as tree-like models such as CHAID (Kass 1980) for political microtargeting. Murray and Scime (2010) suggest decision trees as well. Other state-of-the-art approaches that are used include logistic or probit regression.

When using probabilistic models for GOTV targeting, campaigns are interested in assigning each voter an individual probability to show up at election day. Based on this estimated probability, it stands to reason that using targeting plans on people with a value around 0.5 is worthwhile (Malchow 2008), whereas targeting people with predicted probabilities near 0 or 1 is considered a waste. This is in accordance with results on how to best allocate campaign resources in general (Brams and Davis 1973; Snyder 1989), namely spending more resources on highly contested seats or states where the race is tight. In fact, a person with a predicted probability near zero is almost definitely not going to vote, regardless of how compelling the mobilization message is. A person with a predicted probability of one is going to turn out at the polls anyways, without the need for extra persuasion. In both cases, targeting those people would not lead to an increase in turnout, yet it would consume resources and hence be wasteful. However, voters with a predicted probability in a "targeting range" around 0.5 may be "convincable" to show up at the polls using the right incentive. Malchow (2008) suggests a targeting range of $[0.3, 0.7]$. Clearly, we can be hopeful to sway a person with a probability of voting of say, 0.35 as long as we get the right message to her. On the other hand, while a person with a probability of 0.68 might be going to vote without being targeted specifically, it should not hurt to encourage her a bit more.

1.3. A new unified framework for voter targeting

In this paper we propose a new and flexible statistical framework for voter segmentation that generalizes two standard models currently used in political targeting. In fact, our framework encompasses logistic regression as well as classification trees and allows for a combination of both within the same model. We refer to the resulting framework as logistic regression tree (LORET) models. LORET models are very flexible in that, in their simplest form, they reduce to a majority vote (May 1952) or naive Bayes (Hand and Yu 2001) model; on the other hand, they also allow regression-like modeling of predictors as well as tree-like partitioning of the sample space under the same umbrella. We investigate LORET models of varying degrees of flexibility, and compare them with a particular focus on the benefits that they provide for political decision makers. We apply LORET to a novel data set of Ohio voters and find that, depending on the nature of the race, different statistical methods lead to relevant differences in how campaign budgets are best allocated.

This paper is organized as follows. In Section 2, we present a statistical framework for voter targeting that combines logistic regression models with recursive partitioning. Section 3 describes a case study for which we apply the methods. There, we explain how we evalute the framework and investigate properties of our targeting approach from a campaign's point of view. The corresponding results can be found in Section 4. We finish with conclusions and some general remarks in Section 5.

2. LORET: Modeling and predicting voting behavior

Currently, campaigns employ methods like logistic regression or tree-based methods for voter prediction and targeting (Malchow 2008). Using this as a backdrop, we introduce a general framework, logistic regression trees (LORET), that encompasses and extends these approaches. Briefly, the idea is the following: Instead of fitting a global logistic regression model to the whole data, one might fit a collection of local regression models to subsets or segments of the data (i.e., a segmented logistic regression model) in order to obtain a better fit and higher predictive accuracy. Since usually the "correct" segmentation is not known, it needs to be learned from the data, for example by using recursive partitioning methods.

In what follows we start with the general formulation for logistic regression models for one or more segments and then show how for more than one segment, the segmentation can be learned with recursive partitioning.

2.1. Segmented logistic regression

Let $y_i \in \{0, 1\}$ denote a Bernoulli random variable for the i-th observation, $i = 1, \ldots, N$, and $\mathbf{x}_i$ denote a p-dimensional covariate vector $(x_{i1}, \ldots, x_{ip})$. Let us assume there are r (known or estimated) disjoint segments in the data. For each segment $k = 1, \ldots, r$, we can then specify a logistic regression model for the relationship between y and $x_1, \ldots, x_p$ within that segment,

$$P(y_i = 1|\mathbf{x}_i; \boldsymbol{\beta}^{(k)}) = \pi_i = \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)})}{1 + \exp(\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)})}, \tag{1}$$

where $k = k(i)$ is the segment to which observation i belongs and π_i denotes the probability to belong to class "1". The segment-specific parameter vector is $\boldsymbol{\beta}^{(k)}$ and its estimates are referred to as $\hat{\boldsymbol{\beta}}^{(k)}$, which can be easily obtained (given the segmentation) via maximum likelihood (see e.g., McCullagh and Nelder 1989). Based on the associated estimated probabilities, classification can then be done by

$$\hat{y}_i(c_0) = \begin{cases} 1 & \text{if } \hat{\pi}_i \geq c_0 \\ 0 & \text{if } \hat{\pi}_i < c_0. \end{cases} \tag{2}$$

where $c_0 \in [0, 1]$ is a specific cutoff value (but could, in principle, also be specified to be different for different segments).

If there is only a single segment (i.e., a root node and hence a known segmentation), LORET in (1) reduces to a logistic regression model. Here the conditional distribution of the response variable y is estimated given the status of p covariates. Evaluation of the logistic model at the estimated parameter vector $\hat{\boldsymbol{\beta}}$ yields the predicted probabilities, $\hat{\pi}_i$. If the model uses no covariates as regressors, it further reduces to a majority vote (May 1952) or naive Bayes model (Hand and Yu 2001), i.e., a logistic regression model with only an intercept or simply the relative frequency of class "1" transformed to the logit scale. The upper row in Figure 2 illustrates majority vote and logistic regression on an artificial set of data. The former fits a single constant, the latter a single logistic function of x to the entire data.

If there are more than one segment and the segmentation were known, then LORET can still be simply seen as estimating a maximum likelihood model from a binomial likelihood in each segment. This time however, one needs to specify interactions between factors corresponding to the segments and the cofficients of a logistic regression model to estimate the LORET.

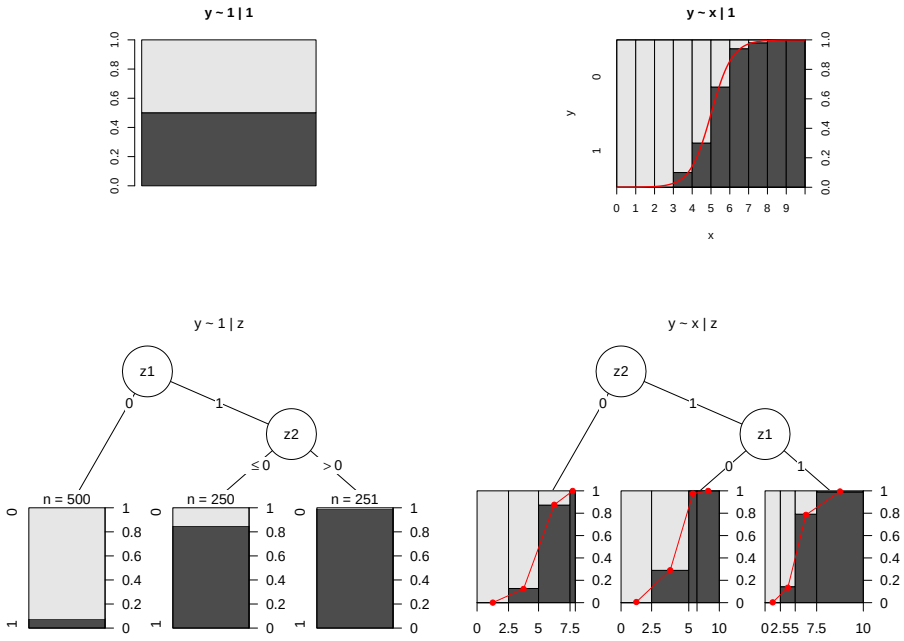

Figure 2: A visualization of the different cases of LORET. In the upper left panel there is the $y \sim 1 \,|\, 1$ LORET, fitting a constant. In the upper right the $y \sim x \,|\, 1$ LORET (logistic regression) is displayed, which is a single function of x for the whole data set. The lower left panel displays a $y \sim 1 \,|\, z$ LORET where the data set is partitioned based on the state of predictor variables z and in each partition a constant is fitted. In the lower right panel, the $y \sim x \,|\, z$ LORET can be found. Here the data set is again partitioned based on z but this time a logistic function of x is fitted in the partitions. Hence it combines the $y \sim 1 \,|\, z$ and $y \sim x \,|\, 1$ LORET.

If the segmentation is unknown, however, it needs to be learned from the data. Two popular approaches for this are mixture models (e.g., mixtures of experts or latent class regression) or some type of algorithmic search method. Recursive partitioning (often called a tree, Zhang and Singer 2010) is a popular example of the latter. Trees are usually induced by splitting the data set along a function of the predictor variables into a number of partitions or segments. The segments are usually chosen by minimizing an objective function (e.g., a heterogeneity measure or a negative log-likelihood) for each segment. The procedure is then repeated recursively for each resulting partition. This approach approximates real segments in the data and yields a segmentation for which maximum likelihood estimation of parameters in each segment can be carried out, as is done in LORET.

Method	Regressor variables	Partitioning variables	Schema
Majority vote	none	none	$y \sim 1 \,\vert\, 1$
Logistic regression	yes	none	$y \sim x \,\vert\, 1$
Classification tree	none	yes	$y \sim 1 \,\vert\, z$
Model tree	yes	yes	$y \sim x \,\vert\, z$

Table 3: Various instances of LORET.

2.2. Recursive partitioning

Let us assume we have an additional, ℓ-dimensional covariate vector $\mathbf{z} = (z_1, \ldots, z_\ell)$. Based on these covariates we learn the segmentation, i.e., we search for r disjoint cells that partition the predictor subspace. Depending on whether the logistic model used for y in each segment has any covariates or just a constant, there are two algorithmic approaches we can use: classification trees and trees with logistic node models.

Classification trees

If the logistic model is an intercept-only model and we have a number of partitioning variables $z_1, \ldots, z_\ell$, then LORET can be estimated as a classification tree. An illustration of a classification tree can be found in the lower left panel of Figure 2, where the data is first partitioned into three subsets and a intercept-only model is fitted to each subset separately. Hence in each terminal node the model is a constant. A wide variety of algorithms have been developed to fit classification trees; among them are: CHAID (Kass 1980), CART (Breiman, Friedman, Olsen, and Stone 1984), C4.5 (Quinlan 1993), QUEST (Loh and Shih 1997), CTree (Hothorn, Hornik, and Zeileis 2006) and many others. In this paper, we use CART and CTree, which are examples of a biased and an unbiased tree algorithm, respectively.

Trees with logistic node models

If there are partitioning variables $\mathbf{z} = (z_1, \ldots, z_\ell)$ as well as regressor variables for the logistic node model $\mathbf{x} = (x_1, \ldots, x_p)$, we get the most general type of LORET, which is a "model tree". The model is illustrated in the lower right panel in Figure 2. Like in a classification tree, the data is first partitioned into subsets. However, in contrast to a classification tree, separate logistic regressions with regressors are employed in each terminal node. Thus, the resulting model tree essentially combines data-driven partitioning like a classification tree with model-based prediction in a single approach. Different algorithms have been proposed to estimate model trees with logistic node models, including: SUPPORT (Chaudhuri, Lo, Loh, and Yang 1995), LOTUS (Chan and Loh 2004), LMT (Landwehr, Hall, and Eibe 2005), and MOB (Zeileis, Hothorn, and Hornik 2008). In what follows, we will use the MOB algorithm with a logistic node model for estimating the most general version of LORET, as it proved to have good properties for these kind of data (Rusch and Zeileis 2012).

To simplify notation and to stress the similarities, we will use a simple schema to refer to the different LORET types (cf. Table 3): The majority vote model will be refered to as $y \sim 1 \,\vert\, 1$, the global logistic regression model as $y \sim x \,\vert\, 1$, the classification tree model as $y \sim 1 \,\vert\, z$ and the full LORET model as $y \sim x \,\vert\, z$.

3. Case study: Ohio voters 2004

To illustrate our targeting framework, we use a unique set of data from the state Ohio in the US. Most US states collect and report voter registration information but the data is not readily and easily accessible (US Election Assistance Comission 2010). The collection of voter registration data is done at the county level and most of the states aggregate the data. However, due to technical and resource limitations, political campaigns often turn to political and marketing data providers who add value by collecting, maintaining, and updating the voter registration data. The voter registration data would typically include name, address, phone, gender, party affiliation, age, vote history (elections that each voter voted), and ethnicity (in many of the southern states). The data providers not only add value by standardizing the data that is collected from each state or county, they also add other potentially relevant behavioral information such as income, type of occupation, education, presence of children, property status (rental or owning), and charities that the person donated to. We use such a proprietary data set which was provided by Aristotle, Inc., one of the leading campaign application and data providers in the industry.

Our data set consists of $20,000$ eligible voters from Ohio. Ohio has proven to be an important state because in every election since 1964, the winner of that state has ultimately won the presidency. Also since 2000, the presidential vote difference between the Republican and Democratic candidates has been 4% or lower. Thus Ohio has been considered one of the top "battleground" states in every recent election.

3.1. Data description

Our set of data includes a total of 77 variables, many which are socio-demographic categorical variables like gender, job category or education level. The data set also contains records on past voting behavior from 1990 to 2004 in general elections, primary or presidential primary elections and other elections (all coded as binary variables – i.e., voted or not). We added three composite or aggregate variables: the raw count of elections a person attended, the number of elections a person attended since registering and the relative frequency of attended elections since registering. After removal of missing values and inconsistent entries, there are a total of $N = 19634$ records with 80 variables per record. Our target variable is the voting behavior ("yes"/"no") in the 2004 presidential election. This election is considered to be unusual in the campaign's high emphasis on face-to-face voter mobilization within neighborhoods and social networks (Middleton and Green 2008) as well as the sharp increase in turnout (see Table 1) and is therefore particularly well suited for illustration.

3.2. Two sets of predictors: Voting history only vs. kitchen sink data

As pointed out earlier, campaigns relied on very limited information when it comes to political targeting. While some of the literature on voter targeting also recommends taking into account a person's age (Malchow 2008; Karp, Banducci, and Bowler 2008), the most commonly and often solely used piece of information is the person's voting history (Malchow 2008). Thus, one of the goals of this study is to investigate the potential benefits of including additional information (besides a person's voting history) into the targeting model. To that end, we compare and contrast two sets of predictors:

- The first set employs the standard information used by many campaigns, which is also

LORET	Regressor variables	Partitioning variables	Partitioning algorithm
$y \sim 1 \mid 1$	none	none	–
$y \sim s \mid 1$	s	none	–
$y \sim s + e \mid 1$	$s + e$	none	–
$y \sim 1 \mid s$	none	s	CART, CTree
$y \sim 1 \mid s + e$	none	$s + e$	CART, CTree
$y \sim s \mid e$	s	e	MOB

Table 4: LORET versions combined with the two variable groups and the algorithms used to estimate the partition. The standard variable set of age and voting history is labeled "s" and the set of additional variables with "e" (hence all variables together are "$s + e$").

recommended in literature. The standard variables used by campaigns are a person's voting history, recorded over the the last four elections, and age. We call this set "s" for "standard".

- The second set contains all other variables available, i.e., 'the "kitchen sink". In our case this includes variables like gender, occupation, living situation, party affiliation, party makeup of the household ("partyMix"), position within the family ("hhRank" and "hhHead"), donations for various causes, education level, relative frequency of attended elections so far ("attendance") and many others. These variables constitute a set of additional variables, labeled with "e" for "extended".

3.3. Model specification

The combination of the two variable sets with the different LORET models leads to model specifications as displayed in Table 4. The models either employ only the standard set of variables or the combination of the standard and the extended set. For unpartitioned models, the parameters are estimated with maximum likelihood. If a partition is induced, we learn it with three different algorithms (CART, CTree and MOB) depending on the nature of the node model. Please note that if age is specified as a parameter in the logistic model part (i.e., for models $y \sim s \mid 1$, $y \sim s+e \mid 1$ and $y \sim s \mid e$), a quadratic effect will be used (cf. Rusch and Zeileis 2012). If age is included as a partitioning variable we use the untransformed variable since partitioning algorithms are invariant to monotone transformations such as taking squares.

All recursive partitioning algorithms that we employ allow for tuning with metaparameters. These tuning parameters can be used to avoid overfitting of the tree algorithms and control how branchy the tree becomes. Quite generally it can be said that the less branchy a tree is, the less prone it is to overfitting. In the algorithms we used, a higher number of observations per node, a lower tree depth, and a stricter split variable selection criterion all lead to smaller trees. At the same time our specification should grant enough flexibility for the algorithm to approximate a complex non-linear relationship in the data.

For CART the maximal depth of the tree and the minimum number of observation per node (minsplit) are available to control the tree appearance. We use a maximal tree depth of 7 and a minsplit of 100 (which corresponds to roughly 0.5% of the observations). For CTree and MOB the significance level of the association or stability tests respectively and the minimum number of observation per node can be used to tune the algorithm. We employ a global significance

level of $\alpha = 1 \times 10^{-6}$. This appeared sensible since the high number of observations might easily lead to spurious significance that is mainly due to the sample size. Hence we reduce the probability of "false positive" selection of a split variable or split point by specifying a low significance level. For minsplit we use 100 for CTree (the same as for CART) and 1000 for MOB (which enables reliable estimation of the node model).[7]

3.4. Model evaluation

We compare the different LORET specifications in terms of their ability to predict potential voters accurately and to allow for efficient targeting. Of particular interest is how data-driven approaches like trees compare to the model-driven approach of logistic regression and whether the combination of the two can lead to substantial improvements. We measure the performance of all models with different learning and test sets using different data- and domain-driven criteria. These criteria include standard measures from the data mining literature (such as predictive accuracy and ROC curves), and measures that arise from an election campaign and voter targeting practitioner point of view. We elaborate on each of these in more detail below. Additionally, we put emphasis on the interpretability of the models and model parameters that result from applying the LORET framework.

Learning and test samples via bootstrapping

We employ the benchmarking framework of Hothorn, Leisch, Zeileis, and Hornik (2005) to evaluate and compare different models via bootstrapping (see also Efron and Tibshirani 1993; Hastie, Tibshirani, and Friedman 2009). That is, we fit a model based on a learning set of size N which is sampled randomly (with replacement) from the entire set of data. The fitted model is then used to predict the out-of-bag test set which consists of observations that were not part of the learning sample. Ten folds of learning and test samples, $f = 1, \ldots, 10$ were used. To provide a further benchmark, we also train and evaluate all models on the whole data set. This allows us to gauge the tendency of a model to overfit as well as how close out-of-bag and in-sample performance are.

Measuring predictive accuracy

For each method, we assess the classification accuracy (acc_f) on each test set f at a given cutoff value $c_0 = 0.5$[8]. To estimate overall predictive accuracy, we use the average over all bootstrap samples $\overline{acc}$. When using the full data set as training and test set (i.e., within-sample performance), we denote the accuracy by acc_0.

Furthermore, we use the ROC curve for model comparison. It displays the false positive rate vs. the true positive rate. For a given threshold value, we average the ROC curves across all bootstrap samples. The area under the ROC curve for bootstrap sample f, auc_f, serves as a cutoff-independent measure of classification accuracy and we calculate it via the Wilcoxon statistic (Wilcoxon 1945). Once again, we average it over all bootstrap samples, $\overline{auc}$ and use

[7] The results were not sensitive to the choice of metaparameters. For CART, we explored depths from 3 to 20. For the global significance levels of CTree and MOB, we explored values of 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05 and 0.1. For the minimum number of observations a node must contain we explored values of 20, 50, 100, 150, 200, 250 and 500 for all methods. For these choices of depth, number of observations per node and significance level, the results were very similar.

[8] For simplicity, we use the same cutoff value of 0.5 for all segments k.

auc_0 to denote the in-sample area under the curve. For all the classification measures above, higher values imply better predictive capability. By using simultaneous pairwise confidence intervals (using Tukey's all-pairwise comparison contrasts and controlling for the family-wise error rate, cf. Hothorn, Bretz, and Westfall 2008), we assess whether differences in predictive accuracy (between two models) are significant or not. To account for the dependence structure of bootstrap samples, we center the accuracies beforehand (see Hothorn *et al.* 2005).

Measuring targeting effectiveness

While the above measures are interesting from a statistical point of view, a campaign may also want to gauge the monetary gain from applying LORET for targeting. Hence we investigate the targeting effectiveness in a simulated targeting environment.

A targeting range (such as $[0.3, 0.7]$) will contain both voters and non-voters. That is, it will contain individuals who will vote regardless of whether we target them with a customized message or not – and, as we have argued earlier, spending resources on such individuals might be a waste. However, the targeting range will also contain individuals who would not have voted out of their own motivation, but who, with the help of the right targeting message at the right time, will change their mind and will go to the polling stations after all. We will refer to these latter individuals broadly as "non-voters." Spending resources on non-voters is not wasteful, especially if there is a chance of swaying them. Thus, a targeting method is most effective, if – for a given targeting range – it identifies the largest number of non-voters and at the same time the lowest number of voters. We therefore assess the cost-benefit of a targeting method in the following way:

Since we know the outcome for the data at hand, we can treat each training/test sample as a possible targeting situation and compare costs for the presented methodology. We assign a monetary value to convincing a real non-voter to attend an election and see how the different LORET models fare in terms of overall cost. To do this we use a linear cost-benefit function for every method m which can be set up for each test sample f, $f = 1, \ldots, 10$.

Let o denote the number of individuals identified in the targeting range (i.e., with predicted probabilities within $[0.3, 0.7]$). We target each of these individuals (e.g., by mail, telephone, email, etc) which incurs a cost of c per individuum. Thus, the overall cost of targeting all o individuals equals $o \times c$. Let us assume that out of these o targeted individuals, n were non-voters. Let us assume further that our targeting efforts are effective in the sense that they turn a fraction v of all non-voters into a voter. In other words, while there are n non-voters, our targeting actions turns $n \times v$ of them into voters. Turning non-voters into voters can be assumed to carry a monetary benefit and we denote that benefit by b. Thus, the overall benefit of targeting equals $n \times v \times b$. This leads to a cost-benefit equation of the form

$$s = (n \times v \times b) - (o \times c) \tag{3}$$

Here, s stands for either the loss (if s is negative and hence $o \times c$ bigger than $n \times v \times b$) or gain (if s is positive and hence $o \times c$ smaller than $n \times v \times b$) of targeting. Notice that o and n depend on the chosen LORET version, so we index it with the superscript m. In addition, each test sample is different hence they also depend on the bootstrap sample f. Thus, let $o_f^{(m)}$ denote the number of individuals which model m applied to bootstrap sample f predicts to be in the targeting range. Similarly, let $n_f^{(m)}$ denote all the non-voters contained in $o_f^{(m)}$. We compute the cost-benefit of model m for our hypothetical targeting situation by computing

the average over all bootstrap samples.

$$\bar{s}^{(m)} = \frac{1}{F}\sum_{f=1}^{F}(n_f^{(m)} \times v \times b) - (o_f^{(m)} \times c) \tag{4}$$

For each model m, we explore $\bar{s}^{(m)}$ over a range of plausible values for v, b and c.

We also investigate the break-even point, b_0, i.e., the minimum benefit value of turning a non-voter into a voter for which, at a given targeting cost per person and a given effectiveness, the overall cost-benefit equals zero. We calculate it via the identity

$$b_{0f}^{(m)} = \frac{o_f^{(m)} \times c}{n_f^{(m)} \times v}, \tag{5}$$

which is proportional to the ratio of people in the targeting range and the number of real non-voters for a given ratio of targeting cost and targeting effectiveness $\left(b_{0f}^{(m)} \propto \frac{o_f^{(m)}}{n_f^{(m)}}\right)$. Again we average over all bootstrap samples f to get $\bar{b}_0^{(m)}$.

4. Results

In this section, we compare the methods from Section 2 using the measures described in Section 3.

4.1. Predictive accuracy

Looking at Figure 3 which shows boxplots of the predictive accuracy for the bootstrap samples as well as the within-sample accuracy (denoted by a cross) at a cutoff value of 0.5, one can see quite clearly how the different models from Table 4 behave. First, using both variable sets (the standard set and the extended set together) leads to a huge improvement in predictive accuracy as compared to just using the standard set. Interestingly, the improvement of using both the "s" and "e" variables over using only "s" is bigger than the improvement of using only "s" over using no covariates at all (cf. Figure 3). Second, LORET versions that employ recursive partitioning feature a better performance than global regression models alone. This holds for either using only the standard variable set as well as the combination of extended and standard set. This can also be seen in Figure 4 which displays the average classification accuracies as a function of different cutoff values in the upper panel and the mean ROC curves in the lower panel (averaged over the $F = 10$ out-of-bag samples).

Table 5 gives a detailed summary of the different performance measures for all models. The benchmark of the naive model $y \sim 1 \,|\, 1$ is an average prediction accuracy of $\overline{acc} = 70.36\%$ and an average AUC of $\overline{auc} = 0.5$, averaged over all test sets.

Global logistic regression models $y \sim s \,|\, 1$ and $y \sim s + e \,|\, 1$ display improved performance ($\overline{acc} = 74.97\%$ and $\overline{auc} = 0.740$ for the standard set and $\overline{acc} = 84.57\%$ and $\overline{auc} = 0.886$ for the combined set) with a huge improvement of the model that uses both variable sets.

Both classification tree algorithms, CART and CTree, used to estimate $y \sim 1 \,|\, s$ and $y \sim 1 \,|\, s + e$ result in a generally better performance compared to logistic regressions, both on the

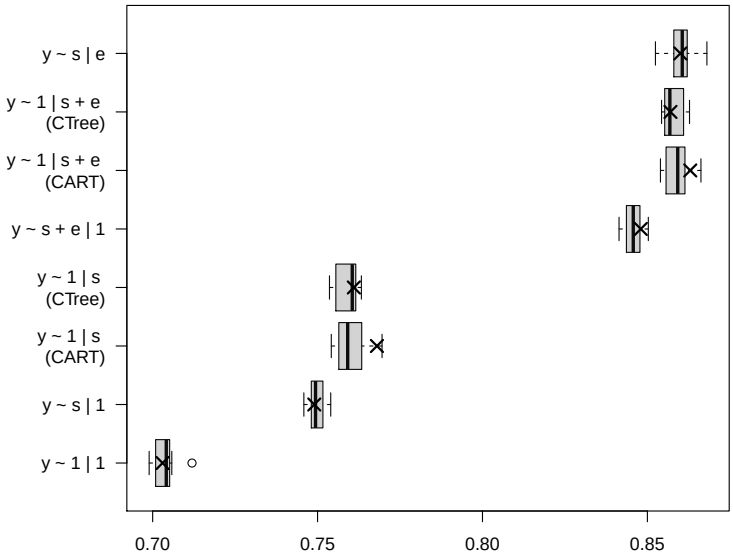

Figure 3: Boxplots of accuracies for all 10 out-of-bag samples for each LORET instances. The cross denotes the within-sample prediction accuracies of each model (acc_0).

standard set of predictors as well as for combining the standard and the extended set. Their performance peaks for the combined set with values of $\overline{acc} = 85.96\%$ and $\overline{auc} = 0.878$ for $y \sim 1 \,|\, s + e$ (CART) and $\overline{acc} = 85.78\%$ and $\overline{auc} = 0.898$ for $y \sim 1 \,|\, s + e$ (CTree).

For the LORET that uses the standard set of predictors as the model in the terminal nodes of the tree and the extended set of predictors for partitioning, i.e., $y \sim s \,|\, e$ result values of $\overline{acc} = 85.98\%$ and $\overline{auc} = 0.906$, respectively. Notice that this model yields the best mean AUC and, at this cutoff, the highest mean accuracy.

The performance differences of models using only standard variables and models employing both the standard and the extended variable sets are evident (see Table 5 and Figure 3). Makin use of the additional variables leads to highly improved performance.

However, the differences among the models employing the combined set themselves (especially between global logistic regression model and partitioned models) are not that strong. Therefore, to establish that these performance differences are not just due to chance, we calculated simultaneous 95%-confidence intervals of all pairwise performance differences between the models that use the combined set of variables based on their accuracy as well as AUC. The former can be found in the upper panel of Figure 5, the latter in the lower panel. We can see that the global logistic regression model performs significantly worse compared to the partitioned models. The tree methods perform best in terms of the accuracy and there are no significant differences amongst them. In contrast, in terms of the cutoff free measure AUC the $y \sim s \,|\, e$ LORET significantly outperforms all other methods.

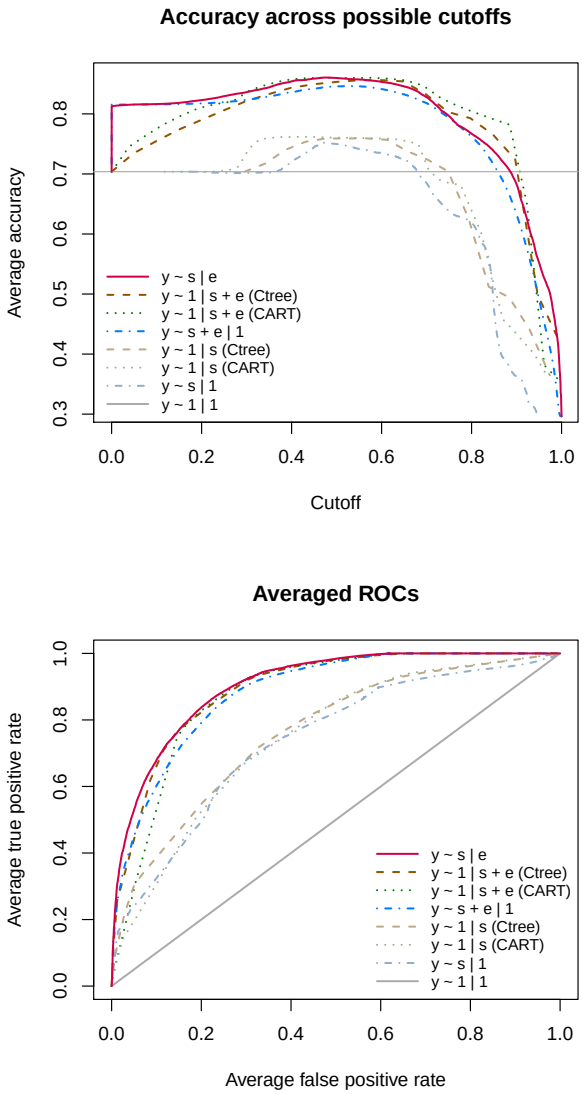

Figure 4: Performance indicators for different models. The upper panel displays features the average accuracies for the range of different cutoffs for the various LORET instances (for majority vote the average accuracy is displayed as a constant). The lower panel features the averaged receiver operating characteristic (ROC) curve for the different models. Threshold averaging has been used for all methods except majority vote.

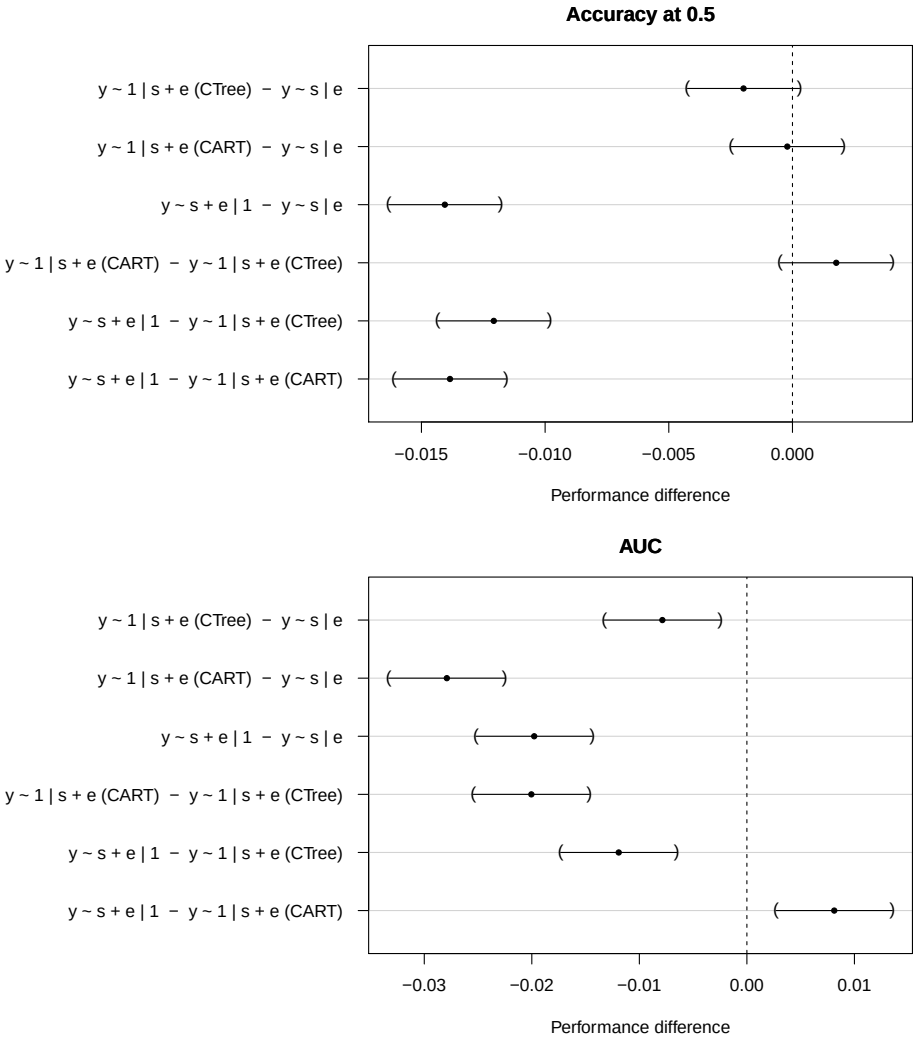

Figure 5: Simultaneous pairwise confidence intervals of the differences of mean accuracies at a cutoff 0.5 over all 10 out-of-bag samples (upper panel) and differences of the average area under the ROC curve (AUC) over all 10 out-of-bag samples (lower panel) for all methods employing the combination of the standard and extended variable set.

Method	Bootstrap samples					Full sample			
	$\overline{acc}$	$se(acc)$	$\overline{auc}$	p	$\tilde{r}$	acc_0	auc_0	p_0	r_0
$y \sim 1 \,\vert\, 1$	0.704	0.004	0.500	1	1.0	0.703	0.500	1	1
$y \sim s \,\vert\, 1$	0.750	0.002	0.740	8	1.0	0.749	0.739	8	1
$y \sim 1 \,\vert\, s$ (CTree)	0.759	0.004	0.765	1	15.0	0.761	0.762	1	14
$y \sim 1 \,\vert\, s$ (CART)	0.760	0.005	0.745	1	28.5	0.768	0.746	1	27
$y \sim s + e \,\vert\, 1$	0.846	0.003	0.886	57	1.0	0.848	0.888	57	1
$y \sim 1 \,\vert\, s + e$ (CTree)	0.858	0.003	0.898	1	18.0	0.857	0.898	1	18
$y \sim 1 \,\vert\, s + e$ (CART)	0.860	0.004	0.878	1	23.5	0.863	0.886	1	23
$y \sim s \,\vert\, e$	0.860	0.004	0.906	8	9.5	0.860	0.909	8	8

Table 5: Summary of performance indicators for each LORET instance. For the bootstrap samples, $\overline{auc}$ means the area under the ROC curve averaged over all 10 out-of-bag test sets. $\overline{acc}$ is the overall classification accuracy averaged over all test sets and $se(acc)$ its standard error. Complexity is given as the number of estimated parameters per segment (terminal node) p and the median number of segments $\tilde{r}$. For the full sample models (fitted and evaluated on all observations), the accuracy is given by acc_0, the AUC by auc_0 and the number of terminal nodes and cofficients in each node by r_0 and p_0, respectively.

4.2. Cost-benefit analysis

We evaluate the cost-benefit equation in (4) for each of our candidate models[9]. To that end, we investigate a range of scenarios for c (the cost of targeting a single person), b (the monetary benefit of turning a non-voter into a voter) and v (the effectiveness of a targeting message, that is, the proportion of non-voters that it will convert to voters). In fact, for c we investigate values of USD 5 and 15 as examples of low and high targeting costs. This is reasonable, since the 2008 Obama campaign spent roughly USD 8 on each vote President Obama got. Furthermore, we assume that the effectiveness v of a campaign can be either 0.3 or 0.1. While 0.3 is probably quite optimistic, a value of 0.1 would only require mobilizing every 10th non-voter to go to the polls. Putting a number on the monetary benefit b of turning a non-voter into a voter is the biggest challenge. In fact, b might be very small for campaigns that are expected to win in a landslide (i.e., for campaigns where one or two extra voters do not make any difference). However, for campaigns that expect a very close race, b might be extremely large. One example from recent history is the 2000 presidential election. In that election, George W. Bush won the State of Florida (and subsequently the presidency) from Al Gore by a margin of about only 500 votes (see, e.g., Agresti and Presnell 2002). Clearly, in such tight races, campaigns would put an extremely large value on b. In our analysis, we investigate values of b ranging between USD 0 and USD 500.

Figure 6 shows the results. The abscissa refers to different values of b; on the ordinate we find $\bar{s}^{(m)}$ as defined in (4). Notice that positive values of $\bar{s}^{(m)}$ correspond to a monetary gain; negative values indicate losses. Figure 6 displays scenarios for the four different combinations of c and v, starting with $c = 5$ and $v = 0.3$ (top left panel) and ending with $c = 15$ and $v = 0.1$ (bottom right panel).

[9]We only evaluate it for models based on the complete set of predictors since we have found in the previous section that using the standard set of predictors only leads to suboptimal performance.

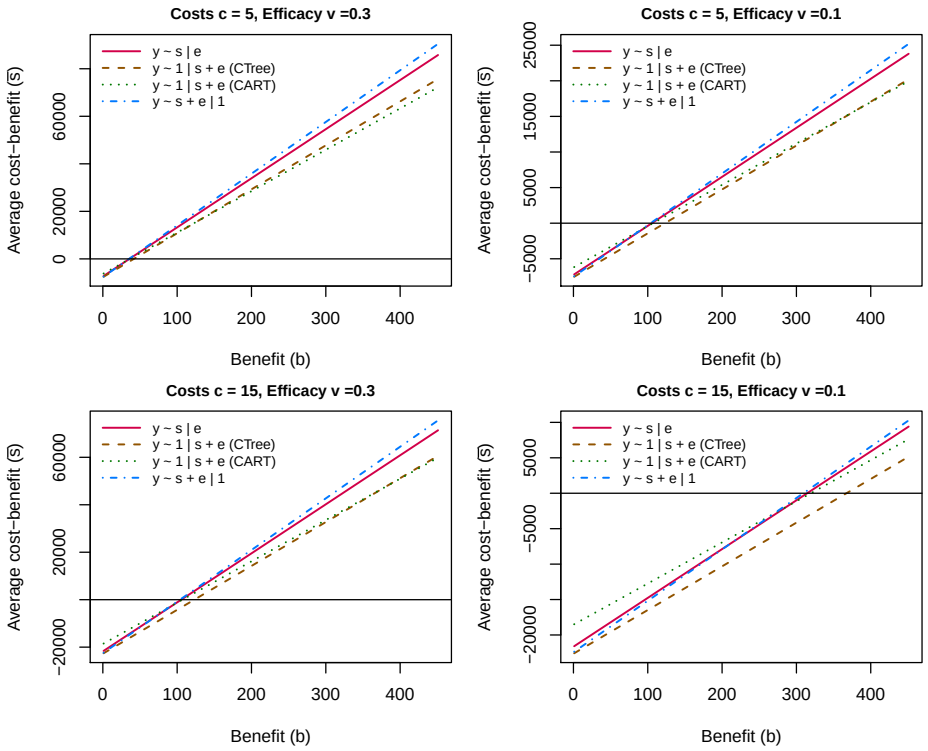

Figure 6: Average linear cost-benefit functions for different versions of LORET. The assumed costs c were USD 5 and USD 15 and the assumed efficacy v of the targeting measure was 0.3 and 0.1. The monetary benefit for turning a non-voter into a voter is depicted on the x-axis, the overall loss ($\bar{s} < 0$) or gain ($\bar{s} > 0$) of the targeting is displayed on the y-axis. The targeting range was $[0.3, 0.7]$.

We can see that the slopes of the cost-benefit function is lowest for classification trees $y \sim 1 \,|\, s+e$. For the CART-based classification tree however, the intercept is highest. This means that for a small benefit b of turning a non-voter into a voter and for a high cost c of targeting, a CART-based classification tree will perform best (i.e., leads to the lowest cost), but only in the loss region (i.e., the lowest loss occurs). With increasing values of b, $y \sim s + e \,|\, 1$ and $y \sim s \,|\, e$ both perform increasingly better – notice the much larger slope which suggests that both methods are especially valuable when there is a large benefit in turning a non-voter into a voter (such as in a tight races). Here, the global logistic regression model eventually performs best for high values of b. The implication for election campaigns is that the LORET framework can be used as a toolbox to increase monetary efficiency of voter targeting, tailor-made for different circumstances. Exactly how it should be used depends on the nature of the race.

Regarding the break even point (which is proportional to $\frac{o_f}{n_f}$, the ratio of people in targeting

range to the number of real non-voters for a given ratio of cost and effectiveness), we calculate the mean of the $\frac{o_f}{n_f}$ as $\bar{b}_0 = 2.06$ for $y \sim s+e \,|\, 1$, $\bar{b}_0 = 2.09$ for $y \sim s \,|\, e$, $\bar{b}_0 = 2.15$ for $y \sim 1 \,|\, s+e$ (CART) and $\bar{b}_0 = 2.46$ for $y \sim 1 \,|\, s + e$ (CTree) under the assumption of USD 1 targeting cost and perfect effectiveness of the targeting measures (i.e., $v = 1$). Hence, *ceteris paribus*, targeting with $y \sim s + e \,|\, 1$ amortizes targeting costs fastest, closely followed by $y \sim s \,|\, e$.

4.3. Interpretability of LORET models

Apart from being able to provide a high classification accuracy, the LORET framework allows to fit interpretable and easily intelligible models that provide further insight into the dynamics of voting behavior relevant for voter targeting. This is one of the major strengths of this approach as compared to "black-box" methods with high predictive capabilities. As point in case, consider the most general LORET, $y \sim s \,|\, e$. Since it has the highest accuracy and AUC and enables efficient targeting for a high benefit of turning non-voters around, we fit it to the whole data set to shed more light on its performance and the turnout of our sample. A table of the decision rules and the coefficients for the logistic regression model in each terminal node can be found in Table 6.

We can see that the segmentation is driven by only four variables, the party composition of the household for each voter ("partyMix"), the relative frequency of attended elections ("attendance"), the rank of the individual in the household ("hhRank", with "1" being highest and "3+" being lowest) and if the person is the head ("H") or a member ("M") of the household ("hhHead"). Hence most partitioning variables are concerned with the household structure rather than with individual-level variables. This is in accordance with literature on the importance of the household for voting behavior (e.g., Cutts and Fieldhouse 2009). We can further see that for all of those individuals for whom "partyMix" is unknown, the probability to vote is zero (actually a case of quasi-complete separation, Albert and Anderson 1984).

The segmentation gives rise to different logistic models that provide additional targeting suggestions for a campaign. We find substantial heterogeneity in the data set as to how voting history influences the outcome. For instance, in node 7 (people who attended elections quite often so far) we see that a higher turnout in earlier elections is associated with a relatively low probability to vote in 2004. Hence these people usually cast their ballot, but for some reason they did less so in 2004. This appears to be a segment that would have been ready for targeting.

The influence of age is also interesting. We specified a quadratic effect and see that, apart from node 10, the estimated probability increases with increasing age just to slow down and reverse. This turning point is rather high for nodes 7, 8 and 10 (70 to more than a 100 years) but substantial in nodes 12 (53.5 years) and 13 (51.1 years). For node 10 it is even at an age of 42. Node 10 is special insofar as it contains young people that have a low rank in the household.

5. Conclusions

In this paper a framework for voter targeting has been proposed, that combines ideas of trees with the idea of logistic regression, coined LORET. The performance of different specifications of LORET with different algorithms in terms of predictive accuracy as well as intelligibility of the models for an exemplary data set has been investigated. Furthermore, a simple linear

Node	Partitioning variables				Regressor variables							
	partyMix	attend.	hhRank	hhHead	const.	gen00	gen01	gen02	gen03	ppp04	age	age^2
2	unknown	–	–	–	$-\infty$	0.000	0.000	0.000	0.000	0.000	0.000	0.000
					(–.–)	(–.–)	(–.–)	(–.–)	(–.–)	(–.–)	(–.–)	(–.–)
6	allD	≤ 0.48	–	–	0.508	0.840	-1.474	0.287	-0.750	0.442	0.054	-0.038
					(0.623)	(0.269)	(0.212)	(0.212)	(0.212)	(0.231)	(0.024)	(0.022)
7	allR, onlyRorD	≤ 0.48	–	–	0.427	0.740	-0.465	0.756	-0.075	0.708	0.011	-0.004
					(0.660)	(0.239)	(0.174)	(0.185)	(0.177)	(0.169)	(0.028)	(0.027)
8	allR, allD,	> 0.48	–	–	2.760	0.277	-1.164	0.352	-1.890	-0.952	0.035	-0.017
	onlyRorD				(0.948)	(0.339)	(0.352)	(0.379)	(0.604)	(0.354)	(0.025)	(0.021)
10	noneRorD, noneD,	–	–	3+	4.057	0.781	0.591	1.249	1.520	0.677	-0.250	0.272
	noneR, legal				(0.797)	(0.128)	(0.203)	(0.165)	(0.214)	(0.212)	(0.052)	(0.076)
12	noneRorD, noneD,	–	$< 3+$	H	-3.630	1.415	-0.010	1.521	2.218	1.694	0.116	-0.108
	noneR, legal				(0.339)	(0.079)	(0.111)	(0.105)	(0.167)	(0.223)	(0.013)	(0.012)
13	noneRorD, noneD,	–	$< 3+$	M	-1.868	1.217	0.086	1.081	1.700	1.603	0.079	-0.078
	noneR, legal				(0.428)	(0.113)	(0.148)	(0.133)	(0.193)	(0.262)	(0.019)	(0.021)

Table 6: A tabular representation of the terminal nodes for the $y \sim s \,|\, e$ LORET for the whole Ohio voter data set. The first column lists the terminal node numbers. The next four columns list the partitioning variables (party mix, attendance, household rank, and household head) and the split point (if any). The last eight columns list the coefficients (upper row) and standard errors (lower row) for the fitted logistic models in the nodes. Please note that the values for the quadratic effect of age have been multiplied by 100 for readability.

cost-benefit analysis of targeting within this framework has been illustrated.

We find that the flexibility introduced by the tree structure leads to more accurate predictions. Furthermore, the framework enables the use of different targeting strategies for different situations. It is easy to understand or communicate to people who are familiar with logistic regression and/or trees and as such the framework is well suited for the purpose of voter targeting.

Regarding the special cases of LORET, a tree with a logistic node model (estimated with the MOB algorithm) may be the most useful default version. For our data, it has the best cutoff-independent predictive accuracy (measured by AUC) and the highest predictive accuracy (at a cutoff of 0.5). Additionally it has the advantage of being easily intelligible and of providing insight for refined targeting. As a result, decisions based on the $y \sim s \,|\, e$ LORET are easy to communicate to campaigns that already use logistic regression. Furthermore it has good potential for cost-efficient targeting, at least based on our sample.

The other instances of LORET, however, are not without merit either. Specifically, a LORET of the $y \sim 1 \,|\, s + e$ type is a good choice if it is not clear how the functional form in the nodes should look like or if there is no standard set of variables to be used in the terminal nodes. Here the nonparametric nature of classification trees show their advantage. If the focus of targeting lies in reducing targeting costs alone, logistic regression and model trees allow most flexible resource allocation and hence may lead to most efficent targeting. For our data set, targeting based on the $y \sim s + e \,|\, 1$ LORET performed best in the cost-benefit analysis. Therefore, even a LORET with just a root node can come in handy.

With the benefits analysed above, one would consider how to incorporate this technique into the overall campaign strategy. Although it is outside of the scope of this study, it needs to be pointed out that it is important for the campaigns to implement any GOTV programs on the likely supporters of the campaign if the intention is to increase the turnout of the supporters. There are three ways that campaigns target likely supporters. First, campaigns use voting results data per precinct from previous elections and gather a general understanding of the demographic and geographic profile of potential supporters. Second, more commonly, they conduct polls with representative samples. The additional benefit of running the polls is that the campaign can be more specific in profiling potential supporters and issues that would motivate them to turn out to vote. Third, campaigns use short surveys over the phone or go door to door interviewing voters to identify individuals who are supporters as well as potential supporters. The primary benefit of using this method is that campaigns can have specific individual level identification of potential supporters. This would also give campaigns the ability to customize communications to each individual. Once the campaigns have better knowledge of the potential voters profile and the likelihood of them voting, campaigns can maximize the return for each dollar spent targeting potential voters by communicating on issues that matter to them and only target voters who are likely to turn out to vote.

Another use of this modelling technique would be to suppress potential supporters of the opposition. This is often called negative campaigning or using "dirty tricks", but it is logical for campaigns to use this method to target voters who might fit into a profile that classify them as potential but not strong supporters of the campaign's opponent. Common ways the campaigns often incorporate this strategy would be to send negative attack message about the opponent to discredit the opponent's character or even distort facts to create confusion. Another tactic that a campaign could use is to assist or send anonymous support for another candidate

that shares the similar political philosophy. For example, for the 1992 presidential election, Ross Perot was an independent candidate; however he had a great amount of support from mostly republican party supporters. The democratic candidate Bill Clinton benefited from Perot dividing the republican electorate. In 2000, the democratic campaign was faced with the similar problem. Ralph Nader was an independent presidential candidate that attracted support from primarily democrats. The republican candidate George W. Bush benefited from it as his opponents had to campaign for the same pot of voters.

The bottom line is that this framework does not change the commonly used campaign tactics but it would influence campaign strategy because it is a more precise tool that would allow campaigns to target the recipients of positive or negative messages more accurately and efficiently which would give more options. With the LORET framework, campaigns have a flexible and versatile toolbox for GOTV targeting that can be customized to meet the requirements at hand.

For further research and practical application, it is possible to improve aspects of interest in GOTV campaigns. For example, it might be fruitful to use techniques such as artificial neural networks or ensembles of tree methods to improve predictive accuracy[10]. Regularized logistic regression models might prove to be a sensible alternative to the tree approach, especially in terms of interpretability and variable selection. It could also be interesting to improve the cost-benefit aspect by defining an appropriate objective function that explicitly incorporates the targeting costs which can then be minimized to yield LORET models that use these specific loss functions rather than the standard ones.

Computational details

All calculations have been carried out with the statistical software R 2.12.0–2.14.1 (R Development Core Team 2011). Logistic regression was fitted with the `glm()` function. Recursive partitioning infrastructure was provided by the packages **party** for `mob()` (Zeileis *et al.* 2008) and `ctree()` (Hothorn *et al.* 2006), as well as **rpart** (Therneau and Atkinson 1997; Therneau, Atkinson, and Ripley 2011) for CART. We used the **ROCR** package (Sing, Sander, Beerenwinkel, and Lengauer 2005, 2009) for calculating and plotting performance measures and ROC curves and **multcomp** (Hothorn *et al.* 2008) for the simultaenous confidence intervals.

References

Agresti A, Presnell B (2002). "The 2000 Presidential election in Florida: Misvotes, undervotes, overvotes." *Statistical Science*, **17**(4), 436–440.

Albert A, Anderson JA (1984). "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models." *Biometrika*, **71**(1), 1–10.

Baek M (2009). "A Comparative Analysis of Political Communication Systems and Voter Turnout." *American Journal of Political Science*, **53**(2), 376–393.

[10]We used random forests and logistic model trees with boosting in the nodes during the course of the study. On this data set their performance was not significantly better than the performance of the LORET models.

Brams SJ, Davis MD (1973). "Resource-Allocation Models in Presidential Campaigning: Implications for Democratic Representation." *Annals of the New York Academy of Sciences*, **219**, 105–123.

Breiman L, Friedman JH, Olsen RA, Stone CJ (1984). *Classification and Regression Trees*. Wadsworth, Pacific Grove, CA.

Chan KY, Loh WY (2004). "LOTUS: An Algorithm for Building Accurate and Comprehensible Logistic Regression Trees." *Journal of Computational and Graphical Statistics*, **13**, 826–852.

Chaudhuri P, Lo WD, Loh WY, Yang CC (1995). "Generalized Regression Trees." *Statistica Sinica*, **5**, 641–666.

Cox G, Munger M (1989). "Closeness, Expenditures, and Turnout in the 1982 U.S. House Elections." *American Political Science Review*, **83**, 217–231.

Cutts D, Fieldhouse E (2009). "What Small Spatial Scales Are Relevant as Electoral Contexts for Individual Voters? The Importance of the Household on Turnout at the 2001 General Election." *American Journal of Political Science*, **53**, 726–739.

Denny K, Doyle O (2009). "Does Voting History Matter? Analysing Persistence in Turnout." *American Journal of Political Science*, **53**(1), 17–35.

Efron B, Tibshirani R (1993). *An Introduction to the Bootstrap*. Chapman & Hall/CRC, Boca Raton, FL.

Finkel S (1993). "Reexamining the 'Minimal Effects' Model in Recent Presidential Elections." *Journal of Politics*, **55**, 1–21.

Gelman A, King G (1993). "Why Are American Presidential Election Polls so Variable when Votes Are so Predictable?" *British Journal of Political Science*, **23**, 409–519.

Gillespie A (2010). "Canvasser Affect and Voter Response: Results From National Focus Groups." *American Politics Research*, **38**(4), 718–758.

Goldstein K, Ridout TN (2002). "The Politics of Participation: Mobilization and Turnout over Time." *Political Behavior*, **24**(1), 3–29.

Green DP, Gerber AS (2008). *Get out the Vote: How to Increase Voter Turnout*. 2nd edition. Brookings Institution, Washington DC.

Hall MG, Bonneau CW (2008). "Mobilizing Interest: The Effects of Money on Citizen Participation in State Supreme Court Elections." *American Journal of Political Science*, **52**(3), 457–470.

Hand DJ, Yu K (2001). "Idiot's Bayes – Not so Stupid after All?" *International Statistical Review*, **69**, 385–399.

Hansen BB, Bowers J (2009). "Attributing Effects to a Cluster-Randomized Get-out-the-Vote Campaign." *Journal of the American Statistical Association*, **104**(487), 873–885.

Hastie T, Tibshirani R, Friedman JH (2009). *Elements of Statistical Learning*. 2nd edition. Springer-Verlag, New York.

Hillygus DS, Jackman S (2003). "Voter Decision Making in Election 2000: Campaign Effects, Partisan Activation, and the Clinton Legacy." *American Journal of Political Science*, **47**(4), 583–596.

Holbrook TM, McClurg SD (2005). "The Mobilization of Core Supporters: Campaigns, Turnout and Electoral Composition in United States Presidential Elections." *American Journal of Political Science*, **49**, 689–703.

Hothorn T, Bretz F, Westfall P (2008). "Simultaneous Inference in General Parametric Models." *Biometrical Journal*, **50**(3), 346–363.

Hothorn T, Hornik K, Zeileis A (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics*, **15**(3), 651–674.

Hothorn T, Leisch F, Zeileis A, Hornik K (2005). "The Design and Analysis of Benchmark Experiments." *Journal of Computational and Graphical Statistics*, **14**, 675–699.

Karp JA, Banducci SA (2007). "Party Mobilization and Political Participation in New and Old Democracies." *Party Politics*, **13**(2), 217–234.

Karp JA, Banducci SA, Bowler S (2008). "Getting out the Vote: Party Mobilization in a Comparative Perspective." *British Journal of Political Science*, **38**, 91–112.

Kass GV (1980). "An Exploratory Technique for Investigating Large Quantities of Categorical Data." *Journal of the Royal Statistical Society C*, **29**(2), pp. 119–127.

Landwehr N, Hall M, Eibe F (2005). "Logistic Model Trees." *Machine Learning*, **59**, 161–205.

Lau RR, Sigelman L, Rovner IB (2007). "The Effects of Negative Political Campaigns: A Meta-Analytic Reassessment." *American Political Science Review*, **69**(4), 1176–1209.

Loh WY, Shih YS (1997). "Split Selection Methods for Classification Trees." *Statistica Sinica*, **7**, 815–840.

Malchow H (2008). *Political Targeting*. 2nd edition. Predicted Lists, LLC, Sacramento, CA.

May KO (1952). "A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decisions." *Econometrica*, **20**, 680–684.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2nd edition. Chapman and Hall, New York.

McDonald M (2012). "Turnout Rates, 1980–2010." United States Election Project. `http://elections.gmu.edu/` [accessed 2012-02-16].

Middleton JA, Green DP (2008). "Do Community-Based Voter Mobilization Campaigns Work even in Battleground States? Evaluating the Effectiveness of MoveOn's 2004 Outreach Campaign." *Quarterly Journal of Political Science*, **3**(1), 63–82.

Muller MG (1999). "Electoral Campaigning as an Occupation – The Professionalization of Political Consultants in the United States." *Politische Vierteljahresschrift*, **40**(1), 198–199.

Murray GR, Scime A (2010). "Microtargeting and Electorate Segmentation: Data Mining the American National Election Studies." *Journal of Political Marketing*, **9**(3), 143–166.

Parry J, Barth J, Kropf M, Jones ET (2008). "Mobilizing the Seldom Voter: Campaign Contact and Effects in High-Profile Elections." *Political Behavior*, **30**(1), 97–113.

Plasser F (2000). "American Campaign Techniques Worldwide." *Harvard International Journal of Press-Politics*, **5**(4), 33–54.

Quelch J (2008). "How Political Marketing Can Learn from Consumer Marketing." http://blogs.hbr.org/quelch/2008/01/how_political_marketing_can_le.html.

Quinlan JR (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishing, San Mateo, California.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Ridout TN (2009). "Campaign Microtargeting and the Relevance of the Televised Political Ad." *Forum – A Journal of Applied Research in Contemporary Politics*, **7**(2), 1–13.

Rusch T, Zeileis A (2012). "Ganining Insight with Recursive Partitioning of Generalized Linear Models." *Journal of Statistical Computation and Simulation*. Forthcoming.

Sing T, Sander O, Beerenwinkel N, Lengauer T (2005). "ROCR: Visualizing Classifier Performance in R." *Bioinformatics*, **21**(20), 3940–3941.

Sing T, Sander O, Beerenwinkel N, Lengauer T (2009). *ROCR: Visualizing the Performance of Scoring Classifiers*. R package version 1.0-4, URL http://CRAN.R-project.org/package=ROCR.

Snyder JM (1989). "Goals and the Allocation of Campaign Resources." *Econometrica*, **57**(3), 637–660.

Steel BS, Pierce JC, Lovrich NP (1998). "Public Information Campaigns and 'At-Risk' Voters." *Political Communication*, **15**(1), 117–133.

Sussman G, Galizio L (2003). "The Global Reproduction of American Politics." *Political Comunication*, **20**(3), 309–328.

Therneau TM, Atkinson EJ (1997). "An Introduction to Recursive Partitioning Using the rpart Routine." *Technical Report 61*, Section of Biostatistics, Mayo Clinic, Rochester. URL http://www.mayo.edu/hsr/techrpt/61.pdf.

Therneau TM, Atkinson EJ, Ripley BD (2011). *rpart: Recursive Partitioning*. R package version 3.1-50, URL http://CRAN.R-project.org/package=rpart.

US Election Assistance Comission (2010). "The Impact of the National Voter Registration Act of 1993 on the Administration of Elections for Federal Office 2009–2010."

Whitelock A, Whitelock J, van Heerde J (2010). "The Influence of Promotional Activity and Different Electoral Systems on Voter Turnout: A Study of the UK and German Euro Elections." *European Journal of Marketing*, **44**(3-4), 401–420.

Wielhouwer PW (2003). "In Search of Lincoln's Perfect List – Targeting in Grassroots Campaigns." *American Politics Research*, **31**(6), 632–669.

Wilcoxon F (1945). "Individual Comparisons by Ranking Methods." *Biometrics Bulletin*, **1**, 80–83.

Zeileis A, Hothorn T, Hornik K (2008). "Model-Based Recursive Partitioning." *Journal of Computational and Graphical Statistics*, **17**(2), 492–514.

Zhang H, Singer B (2010). *Recursive Partitioning and Applications*. 2nd edition. Springer-Verlag, New York.

Affiliation:

Thomas Rusch
Institute for Statistics and Mathematics
WU (Wirtschaftsuniversität Wien)
Augasse 2–6
1090 Wien, Austria
E-mail: Thomas.Rusch@wu.ac.at