Tree-based methods for moderated regression with application to longitudinal data

Buergin, Reto Arthur

Tree-based methods for moderated regression with application to longitudinal data

by

Reto Bürgin

A thesis submitted to the
Geneva School of Economics and Management,
University of Geneva, Switzerland,
in fulfillment of the requirements for the degree of
PhD in Statistics

Members of the thesis committee:
Prof. Gilbert Ritschard, Supervisor, University of Geneva
Prof. Eva Cantoni, Chair, University of Geneva
Prof. Paolo Ghisletta, University of Geneva
Prof. Michel Oris, University of Geneva
Prof. Carolin Strobl, University of Zurich

Thesis No. Undefined
This document is the second submission to the dissertation committee.
March 2015

Acknowledgements

I want to thank all who have supported me and helped me to realize the dissertation. First, I am grateful to my family and my friends Nicola Schopfer and Roger Sutter. Many thanks also to my patient roommates in Geneva, in particular Christelle Ermont.

This dissertation would not have been possible without my scientific environment. Therefore, I want to thank my supervisor Gilbert Ritschard for the many discussions, the patience and the help for writing the articles. I also want to thank the dissertation committee, Eva Cantoni, Paolo Ghisletta, Michel Oris and Carolin Strobl, for reading and discussing the dissertation. Thanks also to Xavier de Luna and Beate Sick for reading and discussing the second chapter, Achim Zeileis for the email exchange concerning issues with R, and Rafael Lalive and Rainer Gabriel for the provided data. I also want to thank Andreas Ruckstuhl for having motivated my studies in data analysis, René Locher for having forwarding me the advertisement of the PhD position and Marianne Müller and Werner Stahel for having followed my diploma and master thesis. Thanks also to Dieter Baumberger, Andor Bariska, Martin Gubler, Marie-Madlen Jeitziner and Reto Schumacher for the collaborations during the dissertation. Finally, I'm grateful to all my colleagues from the LIVES project, the Institute for Demographic and Life Course Studies and the Research Center for Statistics. In particular, I want to thank my fellow students Pauline Adamopoulos, William Aeberhard, Giacomo Benini, Anne-Laure Bertrand, Maurizio Bigotta, Danilo Bolano, Alexis Gabadinho, Myriam Girardin, Alice Milivinti, Setareh Ranjbar, Emmanuel Rousseaux, Manuela Schicka, Laura Turbatu, Giannina Vaccaro, Jana Veselá and Jonathan Zufferey.

Finally, I'm grateful to the direction of LIVES for their great commitment with the PhD students, and the Swiss National Science Foundation for their financial support.

Abstract

The dissertation proposes contributions to graphical longitudinal data analysis and moderated regression analysis, in form of three self-contained articles. Special emphasis is placed on applications to longitudinal categorical data. For general use, the methods are implemented in freely available packages for the statistical software environment R.

The first article proposes a decorated parallel coordinate plot for longitudinal categorical data, featuring a jitter mechanism revealing the diversity of observed longitudinal patterns and allowing the tracking of each individual pattern, variable point and line widths reflecting weighted pattern frequencies, the rendering of simultaneous events, and different filter options for highlighting typical patterns. The proposed visual display has been developed for describing and exploring the order of event occurrences, but it can equally be applied to other types of longitudinal categorical data. Alongside the description of the principle of the plot, the scope of the plot is demonstrated by using data from the European Social Survey to learn orders in which family life events typically occur.

The second and the third articles focus on semi-parametric methods for moderated regression analysis. Linear regression models are combined with tree-based algorithms for learning whether and how selected coefficients of the predictor function vary by moderating variables. In particular, the developed algorithms partition the values spaces of the moderators to model the selected varying coefficients as piecewise constant functions.

The second article implements an algorithm for tree-structured varying coefficients in multivariate generalized linear mixed models for longitudinal data. As a special feature, the algorithm allows partitioning by time-varying moderators while maintaining the random effect component globally. The implementation includes an extension of a score-based coefficient constancy test for mixed models, which allows for unbiased and computationally efficient variable selection. Although the scope of the algorithm is quite general, the article focuses on its usage in an ordinal longitudinal regression setting. The potential of the algorithm is illustrated using data from the British Household Panel, to show how the effect of unemployment on longitudinal ordinal happiness varies across life circumstances.

The third article implements coefficient-wise partitioning for varying coefficients in generalized linear models. Coefficient-wise partitioning allows moderators to be selected separately by coefficient and coefficient-specific sets of moderators to be specified. Empirical evidence suggests that coefficient-wise partitioning potentially builds more accurate and/or more parsimonious models than competing tree-based algorithms are able to build. The article describes the developed algorithm and demonstrates the software implementation in R by applications on several real data sets.

Résumé

La thèse propose des contributions sous forme de trois articles autonomes portant sur l'analyse graphique de données longitudinales et l'analyse de régression avec facteurs de modération. Un accent particulier est mis sur l'application de ces développements à des données longitudinales catégorielles. Les méthodes proposées sont implémentées sous forme de librairies libres pour l'environment statistique R.

Le premier article propose un graphique à coordonnées parallèles pour données longitudinales catégorielles enrichi par un mécanisme de décalage des lignes permettant de rendre compte de la diversité des configurations des trajectoires tout en suivant les configurations individuelles, des épaisseurs variables des lignes et des points reflétant les fréquences pondérées des configurations séquentielles, la possibilité de visualiser des événements survenant simultanément, et enfin différentes possibilités de filtrer les séquences pour mettre en évidence les configurations séquentielles typiques. Le graphique a été développé dans le but de décrire et explorer l'ordre d'occurrence d'événements, mais il s'applique tout aussi bien à d'autres types de données longitudinales. En plus de décrire le principe du graphique, l'article démontre sa portée avec une étude de l'ordre typique des événements familiaux à partir de données de l'Enquête sociale européenne (ESS).

Les deuxième et troisième articles portent sur des méthodes semi-paramétriques pour l'analyse de régression modérée. Le principe général consiste à combiner des modèles de régression linéaire généralisé avec des algorithmes d'arbre de partitionnement pour apprendre si et comment certains coefficients de la fonction prédictive varient avec les variables de modération. En particulier, les algorithmes développés partitionnent l'espace des valeurs des modérateurs pour modéliser la variation des coefficients sous forme de fonctions constantes par morceaux.

Le deuxième article introduit un algorithme pour des coefficients variant selon une structure arborescente dans un modèle multivarié linéaire mixte généralisé pour données longitudinales. Une caractéristique importante de l'algorithme est de permettre le partionnement selon des modérateurs variant dans le temps tout en maintenant la composante aléatoire des coefficients globalement pour tout le modèle. La méthode proposée comprend une extension d'un test de constance des coefficients pour modèles mixtes fondé sur les scores. L'extension permet une sélection non biaisée et efficace sur le plan calculatoire. Bien que la portée de l'algorithme soit plus générale, l'article se focalise sur son utilisation dans le cadre de la régression longitudinale ordinale. Le potentiel de l'algorithme est illustré par une étude de comment l'effet du chômage sur le sentiment (ordinal) de bonheur varie selon les circonstances de vie. L'étude exploite des données du panel de ménages britannique (British Household Panel).

Le troisième article introduit une procédure avec un partitionnement propre à chaque coefficient variable. Le partitionnement propre à chaque coefficient permet de spécifier des modérateurs potentiels différents pour chaque coefficient et de sélectionner les plus pertinents pour chaque coefficient. Les résultats empiriques tendent à montrer que le

partitionnement propre à chaque coefficient génère des modèles plus précis et/ou plus parcimonieux que ceux produits par des algorithmes d'arbre concurrents. L'article décrit l'algorithme proposé et démontre l'utilisation du logiciel implémenté en R au travers de plusieurs applications sur des données réelles.

Contents

Introduction

Moderated regression involves modeling the dependence of selected relations on third variables. Let me begin with an application from the literature on welfare reforms.

Empirical example In 1990, the Austrian government introduced a change for their parental-leave (PL) system. In the former system, working women had the right to stay off work after childbirth up to 12 months and, thereafter, return to the same (or similar) job at the same employer. The 1990 PL reform extended the duration of this leave from 12 months to 24 months.

The question arises whether the extended leave discourages women to return to work. Here, using the subset of 2,650 women living in the regions Vorarlberg, Salzburg and Vienna, retrieved from the data prepared by Lalive and Zweimüller (2009), I study the impact of the reform on the proportions of women that did not return to work within the ten years after giving birth. Vorarlberg lies at the far west of Austria, Salzburg in the center and Vienna (the capital) at the far east.

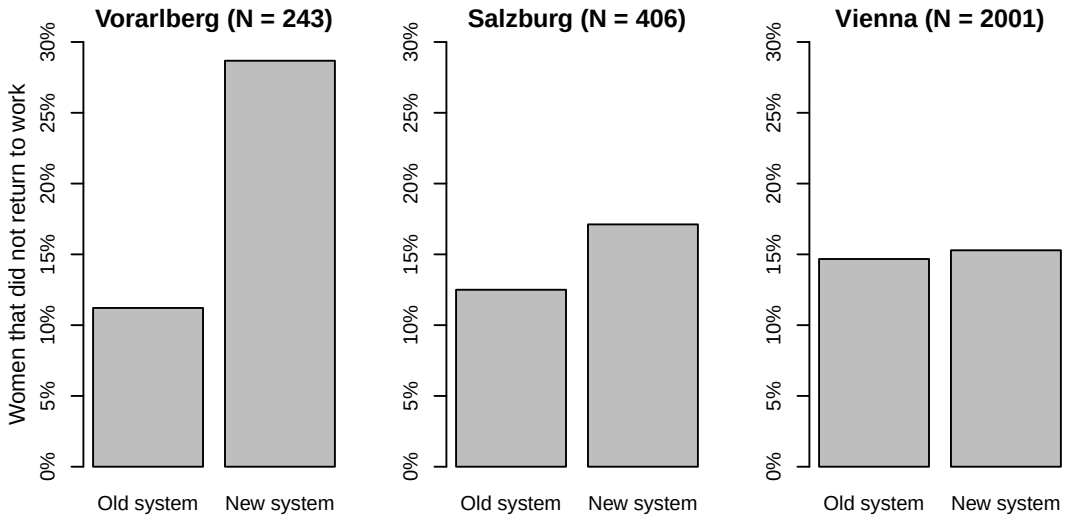

Figure 1: Effect of the Austrian 1990 parental-leave system reform. The bars lengths render the proportions of females that did *not* return to work in the 10 years after giving birth, by system and region.

Figure 1 shows the proportions of females that did not return to work, by PL system and region. In Vorarlberg, the PL reform increased the proportion by 17%, in Salzburg by 5% and in Vienna by 1%.

In this application, the relation of interest is that between the variables *PL system* and *not returned to work*. The relation may be studied by using a logistic regression model

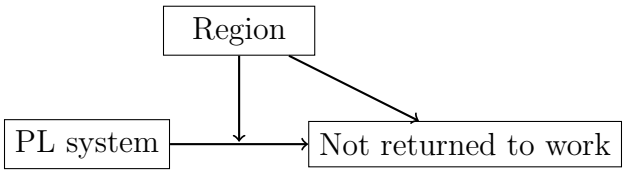

Figure 2: Path diagram of a model for the moderated effect of the 1990 Austrian parental-leave (PL) reform.

for *not returned to work* with only *PL system* in the linear predictor function. However, Figure 1 suggests to take account also of the moderation effect of region. This leads to the model shown in Figure 2. The horizontal arrow defines the interesting effect of the PL reform and the vertical arrow defines the moderation effect of region. With the diagonal arrow I also incorporate the direct effect of *region* on *not returned to work*, in order to take into account baseline differences between regions. However, you can see in Figure 1 that the baseline differences are not so crucial in this application. Specifically, in the old system, the proportion of women that did not return to work was 11% in Vorarlberg, 12% in Salzburg and 15% in Vienna. The model shown in Figure 2 could be implemented by specifying a linear predictor function with main effects for *PL system* and *region* and an interaction between the two predictors.

Main focus of research

The dissertation develops model building algorithms for regression problems including a response variable, a set of predictor variables of interest and a set of third variables. For instance, in the above example the response variable is *not returned to work*, the predictor of interest is *PL system* and the third variable is *region*. The aim of the algorithms is to incorporate the third variables into a given, parametric "basic" regression model that reflects the relations between the predictors of interest and the response variable.

The basic model is, as such, specified by the analyst and it formalizes the relations of interest. For the introductory example, it could be a logistic regression model for *not returned to work* with *PL system* in the predictor function. In gender pay gap studies (e.g. Arulampalam et al., 2007), it could be a Gaussian linear model for *hourly wage* with *gender* in the predictor function, or, in longitudinal studies on long term consequences of a hospital stay (e.g. Jeitziner et al., 2014), it could be a cumulative logit random intercept model for *pain* with *time elapsed since discharge* and individual specific intercepts in the predictor function. The dissertation focuses on generalized linear models and multivariate generalized linear mixed models (e.g. Fahrmeir and Tutz, 2001), which cover many models for cross-sectional and longitudinal data.

The third variables are not of primary importance, yet, their incorporation in the basic model potentially improves the estimation of the relations of interest, or avoids an omitted variable bias. Herein, I consider third variables to be potentially irrelevant (noisy) and assume that the functional forms for incorporating them in the predictor function are unknown. Therefore, to deal with third variables, I focus on statistical learning methods (e.g. Hastie et al., 2001; Berk, 2008) that perform variable selection, in order to build parsimonious models that can integrate nonlinearities and interactions while avoiding collinearities or overfitting.

Among the potential effects of third variables, the focus herein is set on their direct ef-

fects and particularly on their moderation effects (e.g. Hayes, 2013). Other effects of third variables include mediation or confounding effects, e.g., see Hayes (2013); Montgomery (2012). Moderation is a specific relational structure that describes how third variables affect the impacts of predictors on the response variable. For instance, moderation includes dynamic relations, e.g., how the effect of age on mortality has evolved over years (e.g. Holford, 1991). Furthermore, moderation includes differential effects, e.g., how the gender pay gap varies between industries or countries (e.g. Arulampalam et al., 2007), or how the effect of age on mortality varies between social groups. Finally, moderation includes spatial variation, e.g., how the effect of a welfare reform varies regionally. Because of the focus on moderation effects, I will generally refer to third variables as "moderators".

When using linear regression models, moderation effects can be implemented by multiplying the predictors of interest with the moderators and adding these product variables to the predictor function. The corresponding coefficients are then be interpreted as the variation of the coefficients of the predictors of interest across the values of the moderators. Note that, considering instead that the predictors of interest interact with the moderators, i.e., exhibit combination effects, may leads to the same model specification. In consequence, modeling moderation in linear models can be technically equivalent to modeling interaction, however, the way how the coefficients are interpreted is different.

While in linear regression it is assumed that the relevant moderators are known and with it the functional form of their impacts, herein I consider that the relevant moderators have to be selected from a set of potential moderators, and the functional forms of the moderation have to be estimated nonparametrically. This leads to so-called local regression models (e.g. Cleveland et al., 1992) or varying coefficient models (e.g. Hastie and Tibshirani, 1993). I will generally use the latter term.

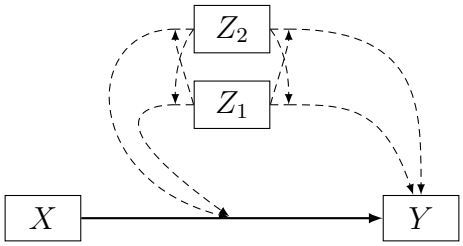

Figure 3: Path diagram for the considered model building problem. X, the predictor of interest; Y the response variable; Z_1 and Z_2 the moderators, *solid lines*, known relations; *dashed lines*, unknown relations.

Figure 3 illustrates the considered problem in situations with one predictor of interest, X, and two moderators, Z_1 and Z_2. The solid lines represent the basic model and the dashed lines the potential direct effects and moderation effects of Z_1 and Z_2. The problem is to decide which impacts are relevant, i.e., which of the dashed lines can be dropped, and to provide a functional description of the relevant impacts.

To tackle the described model building problem, I will consider tree-based algorithms (e.g. Belson, 1959; Morgan and Sonquist, 1963; Breiman et al., 1984) and in particular those variants combining regression trees with linear regression models (e.g. Quinlan, 1992; Chaudhuri et al., 1995; Alexander et al., 1996; Loh, 2002; Gama, 2004; Zeileis et al., 2008; Strobl et al., 2013; Wang and Hastie, 2014). The considered tree-based algorithms partition the value space of the moderators into strata such that the relations of interest differ between the strata, but are fairly constant within the strata. Accordingly, the built models fit the relations of interest separately by strata. While the tree-based approach

has certain drawbacks, particularly that it is a heuristic and may be instable regarding perturbations to the data, it has several advantages for the considered regression setting. It is scalable to many moderators, performs variable selection, can handle nonlinearities and interactions, provides a uniform framework to handle mixed scaled sets of moderators and yields easily readable outcomes in form of decision trees.

Data types

The dissertation considers three data types, among them two types of longitudinal data. Chapter 1 focuses on event sequence data, Chapter 2 on repeated measurement data and Chapter 3 on cross-sectional data.

Cross-sectional data

The simplest among the considered data types are cross-sectional data. Cross-sectional data measure variables referring to a given time point or period on a sample of individuals (or, more generally, sample units). Table 1 shows an extract of a fictitious cross-sectional data set in a vertical tabular form. The observations are reported in the rows and the variables *returned to work*, *PL system* and *region* in the columns.

Table 1: Extract of a fictitious cross-sectional data set.

Index	Individual	Returned to Work	PL System	Region
1	1	Yes	Old	Vienna
2	2	No	New	Vorarlberg
3	3	...	...	...

Cross-sectional data are commonly used for analyzing inter-individual differences. A limitation is that they do not convey information about changes across time. For example, one could compare the pain level of hospital patients with that of a reference group, but not whether the pain level of the hospital patients is increasing or decreasing at the time of measurement.

Longitudinal data

Longitudinal data are data that follow a sample of individuals over time. Two significant advantages arise out of this: Longitudinal data record information about temporal changes of inter-individual differences, but also about intra-individual changes. For example, one could study whether the pain level of hospital patients approaches that of the reference group as time elapses, but also typical individual trajectories, which may be characterized as “quick recovery” or “long term pain”.

Repeated measurement data

Repeated measurement data (e.g. Raghavarao and Padgett, 2014) record the same variables for multiple time points. They can be collected, for example, by yearly panel surveys where variables refer to the date of measurement. Special cases are cross-sectional data, which are a subset of observations corresponding to a single time point, and time series data, which are a subset of observations corresponding to a single individual. It is

generally desirable that the data record for each individual information from each measurement time point. However, because of attrition or sample rotations, real data are often unbalanced so that the number of observations varies between individuals.

Table 2: Extract of a fictitious, unbalanced repeated measurement data set.

Index	Individual	Year	Happiness	Employed	Gender
$(1, 1)$	1	1990	Very happy	Yes	Female
$(1, 2)$	1	1992	Very happy	Yes	Female
$(1, 3)$	1	1993	Moderately happy	no	Female
$(2, 1)$	2	1991	Rather happy	Yes	Male
$(2, 2)$	2	1992	...	...	...

Table 2 reports a fictitious, unbalanced repeated measurements data set in a vertical tabular form. It reports that individual 1 has participated three times, but not in the year 1991. Individual 2 has participated in 1991, but not in 1990.

The analysis of repeated measurements data can, but does not have to, be focused on temporal change. The time points can also refer to different conditions of measurement, and the focus be on changes across these conditions. For example, Chapter 2 considers how happiness changes at the transition to unemployment, using repeated measurement data of individuals who were observed under employment and unemployment.

Event sequence data

Event sequences (e.g. Ritschard et al., 2009) are longitudinal data that record dates or ages at which selected events occurred. Because events, such as getting married, do not persist but occur at a unpredictable time point, they are often collected by retrospective surveys. Table 3 shows a fictitious event sequence data set in the vertical tabular form, considering the four events "Graduating university", "First employment", "First marriage" and "First child". It reports that individual 1 has graduated university at 25, found a first employment and married at 27 and became mother with 30.

Table 3: Extract of a fictitious event sequence data set.

Index	Individual	Event	Age	Rank	Gender
1	1	Graduating university	25	1	Female
2	1	First employment	27	2	Female
3	1	First marriage	27	2	Female
4	1	First child	30	3	Female
5	2	...	...	...	...

A specific aspect of event sequence analysis is the order in which events occurred. For this, Chapter 1 will work with rank order of occurrences of events. The column "Rank" of Table 3 presents a possible construction. For individual 1, it assigns rank 1 to the earliest event "Graduating university"; rank 2 to the simultaneous events "First employment" and "First marriage", and rank 3 to the latest event "First child". An alternative construction would be to assign rank 0 to the latest event and numbering in the descending order.

Contents

The Chapters 1, 2 and 3 present the three proposed articles in their chronological order of writing. The initial focus of research was on developing graphical methods for longitudinal categorical data. The main result out of this research is the plot of Chapter 1. When testing the plot, I found particularly interesting its usage for studying how longitudinal patterns vary across cohorts and countries, or, more generally speaking, how intra-individual relations between time and the categorical target variable vary across social subgroups. This brought me to put the emphasis on moderated relations. Chapter 2 presents the first result from the explicit research in this direction, developing a tree-based approach for moderated regression models to longitudinal ordinal categorical data. Finally, the extensive work with tree-based methods brought me to the coefficient-wise partitioning extension, which is presented in Chapter 3. The implementation refers to cross-sectional data to simplify the presentation.

Table 4: Overview of the contents of the Chapters 1, 2 and 3 by keywords. Symbols: ✓, is a focus; (✓), is not a focus, but is related or a special case; ⋆, in appendix.

		Chapter		
Aspect	Keyword	1	2	3
Data	Cross-sectional data	(✓)	(✓)	✓
	Repeated measurement data	⋆	✓	
	Event sequence data	✓		
Analysis	Moderated relations	(✓)	✓	✓
	Descriptive statistics	✓		
	Exploratory data analysis	✓	✓	✓
	Subgroup analysis	✓	✓	✓
	Regression analysis		✓	✓
Methods	Visualizations	✓	(✓)	(✓)
	Semi-parametric regression models		✓	✓
	Tree-based algorithms		✓	✓
Evaluation	Real data applications	✓/⋆	✓	✓
	Simulations studies		✓/⋆	⋆
Application	R-codes	⋆	⋆	✓/⋆

Table 4 overviews the contents of the Chapters 1, 2 and 3 by keywords, including references to the corresponding Appendices A, B and C. In the following, I summarize the main ideas and the contributions to the literature.

Chapter 1 The Chapter "A decorated parallel coordinate plot for categorical longitudinal data"[1] develops a parallel coordinate plot (e.g. Yang, 2003) for rendering longitudinal categorical patterns. The contributions are jittering, line weighting and color filtering techniques to allow identifying typical longitudinal patterns while rendering at the same time the diversity of the observed patterns. The article focuses on the usage of the plot for describing and exploring the order of event occurrences.

[1] Up to minor modifications, this chapter is the article: Bürgin, R. and G. Ritschard (2014). A Decorated Parallel Coordinate Plot for Categorical Longitudinal Data. *The American Statistician 68*(2), 98-103.

Illustration Although Chapter 1 does not explicitly refer to moderated regression, the decorated parallel coordinate plot proves useful as a tool for describing or exploring moderated relations, or differences between subgroups, respectively. Figure 4 illustrates the plot for the empirical example of Chapter 2. The focus of interest is on whether and how the effect of a transition from employment to unemployment on happiness is moderated by variables such as gender, age, etc. The used data are a subset including 1,487 respondents from the British Household Panel survey (e.g. Taylor et al., 2010) who experienced a switch from employment to unemployment between two consecutive waves.

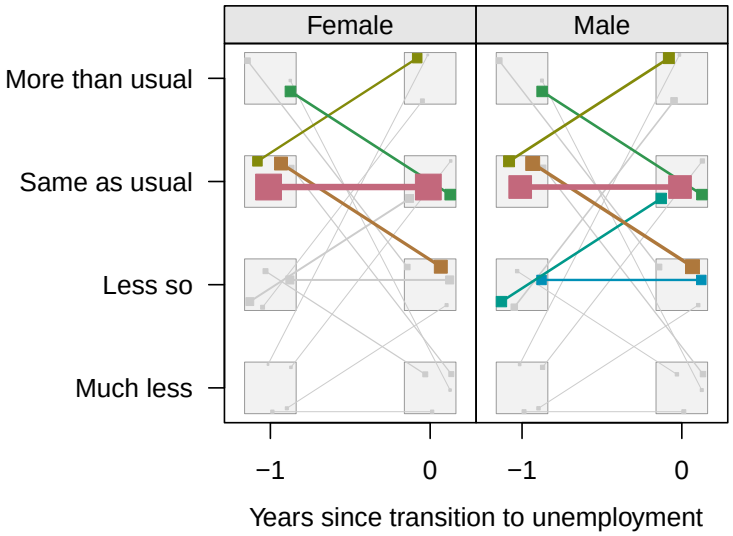

Figure 4: Happiness patterns at the transition from employment to unemployment, by gender. Highlighted lines describe transition patterns with frequency above 5%. The y-axis gives the ordinal levels of happiness.

Figure 4 shows the observed happiness patterns at the transition to unemployment. The plots split females and males to study whether gender moderates the effect of the transition. It can be seen that respondents most often answer with category "Same as usual" twice in succession. The most frequent intra-individual change presents the pattern "Same as usual" to "Less so". The moderation effect is that intra-individual changes are more frequent for males than for females. For example, "Less so" to "Same as usual" is more frequent for males than for females.

Chapter 2 The Chapter "Tree-based varying-coefficient regression for longitudinal ordinal responses"[2] proposes a tree-based algorithm for varying coefficients in multivariate generalized linear mixed models for longitudinal responses. The contribution is to combine and extend the technique for partitioning and tree size selection of Zeileis et al. (2008); and the technique for incorporating a tree structure into a mixed model of Hajjem et al. (2011) and Sela and Simonoff (2012); for general longitudinal varying coefficient regression. Although the algorithm is quite general, the focus is placed on its usage in an longitudinal ordinal categorical regression setting.

[2]Up to minor modifications, this chapter is the article: Bürgin, R. and G. Ritschard (2015). Tree-Based Varying-Coefficient Regression for Longitudinal Ordinal Responses. *Computational Statistics & Data Analysis 86* 65-80.

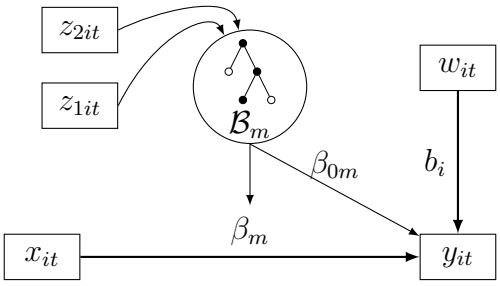

Figure 5: Chapter 2: Schematic representation of a mixed model with tree-structured varying coefficients. x_{it}, the predictor measured on individual i at time t; y_{it}, the longitudinal response; z_{1it} and z_{2it}, the moderators; w_{it}, the predictor corresponding to the random coefficient b_i. $\mathcal{B}_m$, $m = 1, \ldots, M$, are strata from a partition of the value space of Z_1 and Z_2. β_m is the fixed effect of x_{it} on y_{it} in stratum $\mathcal{B}_m$, and β_{0m} the fixed effect of stratum $\mathcal{B}_m$ on y_{it}.

The idea is to partition the value space of the moderators into M strata, $\{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$, and to estimate the fixed coefficients of the basic mixed model separately for each stratum. Figure 5 shows a schematic representation of a fitted model in a setting with one fixed coefficient predictor, X, one random coefficient predictor, W, and two moderators, Z_1 and Z_2. The tree component in the middle assigns the observations z_{1it} and z_{2it} to a stratum $\mathcal{B}_m$ and, accordingly, the model determines the strata-specific fixed coefficient β_m as the linear effect of x_{it} on y_{it}. The direct effects of the moderators are incorporated by the coefficients β_{0m}, which correspond to dummy variables for the strata $\{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$. The random coefficients, b_i, are used to take into account intra-individual correlations and could include individual-specific intercepts or slopes over time. Herein, they are not considered as to be moderated, which is an simplifying assumption. Indeed, one could consider to regress the random coefficients or their distributional parameters on the moderators. However, such extensions would likely result in models which are difficult to fit or interpret.

Chapter 3 The Chapter "Coefficient-wise tree-based varying coefficient regression with vcrpart"[3] proposes building a separate partition for each varying coefficient. The algorithm is implemented for generalized linear models and is based on the partitioning and tree size selection techniques of Breiman et al. (1984) or Wang and Hastie (2014), respectively.

Figure 6 shows a schematic representation of a model with coefficient-specific partitions in a setting with one predictor X, and two moderators, Z_1 and Z_2. Unlike the model shown in Figure 5, the one here incorporates two partitions, one for the coefficient of the predictor X, $\{\mathcal{B}_{11}, \ldots, \mathcal{B}_{1M_1}\}$, and a second for the direct effects, $\{\mathcal{B}_{21}, \ldots, \mathcal{B}_{2M_2}\}$. This allows moderators to be selected separately by coefficient and coefficient-specific sets of moderators to be specified, which is the principal contribution of the article. For example, Z_2 may affects the response directly but does not moderate the effect of X. In this case, the algorithm may selects both Z_1 and Z_2 to construct the partition for the direct effect, but only Z_1 to construct the partition for the moderation effect on X.

[3]Up to minor modifications, this chapter is the article: Bürgin, R. and G. Ritschard (2015). Coefficient-Wise Tree-Based Varying Coefficient Regression with vcrpart. Revised (from pre-screening stage) manuscript submitted to Journal of Statistical software for publication.

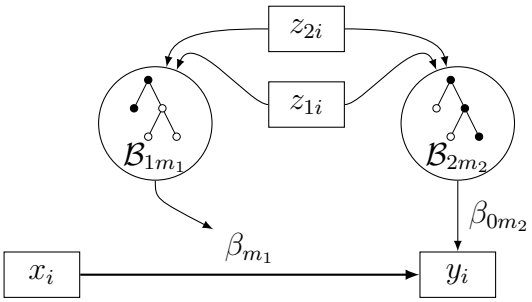

Figure 6: Chapter 3: Schematic representation of a fitted model with coefficient-wise partitions. x_i, the predictor measured on individual i; y_i the measured response; z_{1i} and z_{2i} the measured moderators.

Introductory literature An introduction to graphical categorical data analysis is provided by Friendly (2000), and Gabadinho et al. (2011) propose a series of visualizations for longitudinal categorical data. The Chapters 2 and 3 expect basic knowledge about recursive partitioning, generalized linear models and mixed models. For an introduction to recursive partitioning, see for example Loh (2008); Strobl et al. (2009); Loh (2014). Consistency properties of recursive partitioning, which are not discussed herein, are found in Breiman et al. (1984, Chap. 12), Devroye et al. (1996, Chap. 20) and Kim et al. (2007). Generalized linear models are explained, for instance, by McCullagh and Nelder (1989) and mixed models by Fahrmeir and Tutz (2001); Skrondal and Rabe-Hesketh (2004). For specific information on ordinal mixed models, see for example Agresti (2010); Tutz (2012). For a general overview of regression models for longitudinal data, see for example Molenberghs and Verbeke (2005).

Software

The developed methods are implemented in the packages **TraMineR**[4] (Gabadinho et al., 2011) and **vcrpart**[5] (Bürgin, 2015) for the freely available software environment R (R Core Team, 2014).

Table 5: Overview of software implementations.

Chapter	R Package	Main function
1	**TraMineR**	`seqpcplot`
2	**vcrpart**	`tvcolmm`
3	**vcrpart**	`tvcglm`

Table 5 overviews the main functions. The `seqpcplot` function implements the decorated parallel coordinate plot of Chapter 1, and the `summary` function allows to extract the frequencies of unique sequence patterns from the output of the `seqpcplot` function. `tvcolmm` allows fitting cumulative logit mixed models with tree-structured fixed effect components, and `tvcglm` implements the coefficient-wise partitioning approach for generalized linear models. The **vcrpart** provides for the output from these two functions several methods, such as the `summary` function to read the fitted coefficients, the `plot` function

[4]See also http://cran.r-project.org/web/packages/TraMineR/ and http://mephisto.unige.ch/traminer/.
[5]See also http://cran.r-project.org/web/packages/vcrpart/.

to draw decision trees and partial dependencies and the `predict` function for predicting responses of new observations.

Connection with the LIVES project

The dissertation results from research work carried out within the framework of the Swiss National Center of Competence in Research LIVES *Overcoming Vulnerability: Life Course Perspectives*[6] and specifically within the individual project IP 14 *Measuring Live Sequences and the Disorder of Lives*. The LIVES project is financed by the Swiss National Science Foundation.[7]

The proposed articles are intended to provide statistical methods for the research fields of the LIVES project. Concretely, the illustrations on family life event histories (Sec. 1.3), the effect of unemployment on happiness (Sec. 2.3.1, cf. Oesch and Lipps (2013)) and the effect of parental leave duration on return to work (Sec. 3.4.3, cf. Lalive and Zweimüller (2009)) refer to the thematic scope or to research of members of LIVES. Moreover, in connection with the illustration on family life event histories, I have also collaborated with colleagues of the LIVES project to prepare the article "The transition of the sequencing of family life events in Europe: a cross-regional perspective" (Bürgin et al., 2015).

Bibliography

Agresti, A. (2010). *Analysis of Ordinal Categorical Data* (2 ed.). Wiley Series in Probability and Statistics. New Jersey, USA: John Wiley & Sons.

Alexander, W. P., S. D. Grimshaw, and P. William (1996). Treed Regression. *Journal of Computational and Graphical Statistics 5*(2), 156–175.

Arulampalam, W., A. L. Booth, and M. L. Bryan (2007). Is There A Glass Ceiling Over Europe? Exploring the Gender Pay Gap Across the Wage Distribution. *Industrial and Labor Relations Review 60*(2), 163–186.

Belson, W. A. (1959). Matching and Prediction on the Principle of Biological Classification. *Applied Statistics 8*(2), 65–75.

Berk, R. A. (2008). *Statistical Learning from a Regression Perspective*. Springer Series in Statistics. New York, USA: Springer-Verlag.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. New York, USA: Wadsworth.

Bürgin, R. (2015). *vcrpart: Tree-Based Varying Coefficient Regression for Generalized Linear and Ordinal Mixed Models*. R package version 0.3-3, URL http://cran.r-project.org/web/packages/vcrpart/.

Bürgin, R., R. Schumacher, and G. Ritschard (2015). The Transition of the Sequencing of Family Life Events in Europe: a Cross-Regional Perspective. Manuscript submitted to the LIVES Working Paper series for publication.

[6] See http://www.lives-nccr.ch/.
[7] See http://www.snf.ch/.

Chaudhuri, P., L. Wen-Da, W.-Y. Loh, and C.-C. Yang (1995). Generalized Regression Trees. *Statistica Sinica 5*, 641–666.

Cleveland, W. S., E. Grosse, and W. M. Shyu (1992). Local Regression Models. In J. M. Chambers and T. J. Hastie (Eds.), *Statistical Models in S*, Chapter 8, pp. 309–376. Pacific Grove, USA: Wadsworth & Brooks/Cole.

Devroye, L., L. Györfi, and G. Lugosi (1996). *A Probabilistic Theory of Pattern Recognition*. Stochastic Modelling and Applied Probability. New York, USA: Springer-Verlag.

Fahrmeir, L. and G. Tutz (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Friendly, M. (2000). *Visualizing Categorical Data*. Cary, USA: SAS Institute.

Gabadinho, A., G. Ritschard, N. S. Müller, and M. Studer (2011, 4). Analyzing and Visualizing State Sequences in R with TraMineR. *Journal of Statistical Software 40*(4), 1–37.

Gama, J. (2004). Functional Trees. *Machine Learning 55*(3), 219–250.

Hajjem, A., F. Bellavance, and D. Larocque (2011). Mixed Effects Regression Trees for Clustered Data. *Statistics & Probability Letters 81*(4), 451–459.

Hastie, T. and R. Tibshirani (1993). Varying-Coefficient Models. *Journal of the Royal Statistical Society B 55*(4), 757–796.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Hayes, A. F. (2013). *Introduction to Mediation, Moderation, and Conditional Process Analysis: A Regression-Based Approach*. New York, USA: Guilford Press.

Holford, T. R. (1991). Understanding the Effects of Age, Period, and Cohort on Incidence and Mortality rates. *Annual Review of Public Health 12*(1), 425–457.

Jeitziner, M.-M., S. Zwakhalen, R. Bürgin, V. Hantikainen, and J. Hamers (2014). Long-Term Consequences of Pain, Anxiety and Agitation for Critically Ill Older Patients After an Intensive Care Unit Stay. *Journal of Clinical Nursing*. Forthcoming.

Kim, H., W.-Y. Loh, Y.-S. Shih, and P. Chaudhuri (2007). Visualizable and Interpretable Regression Models with Good Prediction Power. *IIE Transactions 39*(6), 565–579.

Lalive, R. and J. Zweimüller (2009). Does Parental Leave Affect Fertility and Return-to-Work? Evidence from Two Natural Experiments. *The Quarterly Journal of Economics 124*(3), 1363–1402.

Loh, W.-Y. (2002). Regression Trees With Unbiased Variable Selection and Interaction Detection. *Statistica Sinica 12*(2), 361–386.

Loh, W.-Y. (2008). Classification and Regression Tree Methods. In F. Ruggeri, R. Kenett, and F. W. Faltin (Eds.), *Encyclopedia of Statistics in Quality and Reliability*, pp. 315–323. Chichester, UK: John Wiley & Sons.

Loh, W.-Y. (2014). Fifty Years of Classification and Regression Trees. *International Statistical Review 82*(3), 329–348.

McCullagh, P. and J. Nelder (1989). *Generalized Linear Models* (2 ed.). Monographs on Statistics and Applied Probability. London, UK: Chapman and Hall.

Molenberghs, G. and G. Verbeke (2005). *Models for Discrete Longitudinal Data*. Springer Series in Statistics. New York, USA: Springer-Verlag.

Montgomery, D. C. (2012). *Design and Analysis of Experiments* (8 ed.). John Wiley & Sons.

Morgan, J. N. and J. A. Sonquist (1963). Problems in the Analysis of Survey Data, and a Proposal. *Journal of the American Statistical Association 58*(302), 415–434.

Oesch, D. and O. Lipps (2013). Does Unemployment Hurt Less if There is More of it Around? A Panel Analysis of Life Satisfaction in Germany and Switzerland. *European Sociological Review 29*(5), 955–967.

Quinlan, J. R. (1992). Learning with Continuous Classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, Singapore, pp. 343–348. World Scientific.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL http://www.R-project.org.

Raghavarao, D. and L. Padgett (2014). *Repeated Measurements and Cross-Over Designs*. New Jersey, USA: John Wiley & Sons.

Ritschard, G., A. Gabadinho, M. Studer, and N. S. Müller (2009). Converting Between Various Sequence Representations. In Z. Ras and A. Dardzinska (Eds.), *Advances in Data Management*, Volume 223 of *Studies in Computational Intelligence*, pp. 155–175. Berlin, Germany: Springer-Verlag.

Sela, R. and J. S. Simonoff (2012). RE-EM trees: A Data Mining Approach for Longitudinal and Clustered Data. *Machine Learning 86*(2), 169–207.

Skrondal, A. and S. Rabe-Hesketh (2004). *Generalized Latent Variable Modelling*. Interdisciplinary Statistics. Florida, USA: Chapman and Hall.

Strobl, C., J. Kopf, and A. Zeileis (2013). Rasch Trees: A New Method for Detecting Differential Item Functioning in the Rasch Model. *Psychometrika*, 1–28. Forthcoming.

Strobl, C., J. Malley, and G. Tutz (2009). An Introduction to Recursive Partitioning. *Psychological methods 14*(4), 323–348.

Taylor, M. F., N. B. John Brice, and E. Prentice-Lane (2010). *British Household Panel Survey User Manual Volume A: Introduction, Technical Report and Appendices*. Colchester, UK: University of Essex.

Tutz, G. (2012). *Regression for Categorical Data*. Cambridge Series in Statistical and Probabilistic Mathematics. New York, USA: Cambridge Series in Statistical and Probablilistic Mathematics.

Wang, J. C. and T. Hastie (2014). Boosted Varying-Coefficient Regression Models for Product Demand Prediction. *Journal of Computational and Graphical Statistics 23*(2), 361–382.

Yang, L. (2003). Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates. In V. Kumar, M. Gavrilova, C. Tan, and P. L'Ecuyer (Eds.), *Computational Science and Its Applications - ICCSA 2003*, Volume 2668 of *LNCS*, Berlin, Germany, pp. 21–30. Springer-Verlag.

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

Chapter 1

A decorated parallel coordinate plot for categorical longitudinal data

Abstract This article proposes a decorated parallel coordinate plot for longitudinal categorical data, featuring a jitter mechanism revealing the diversity of observed longitudinal patterns and allowing the tracking of each individual pattern, variable point and line widths reflecting weighted pattern frequencies, the rendering of simultaneous events, and different filter options for highlighting typical patterns. The proposed visual display has been developed for describing and exploring the order of event occurrences, but it can be equally applied to other types of longitudinal categorical data. Alongside the description of the principle of the plot, we demonstrate the scope of the plot with a real data set.[1]

1.1 Introduction

The article introduces an original way of plotting a set of event sequences such as the successions of life events describing professional careers or family trajectories. The plot is intended for identifying the typical order of occurrence of the events in the considered sequences while rendering at the same time the diversity of the observed sequencing patterns. Although the plot can be used for any kind of categorical sequences, it is specifically designed for rendering events that, unlike states for example, do not have durations, can simultaneously occur at the same time point, and whose position in the sequence does not convey other time information than the order of occurrence.

As a first illustration of the proposed plot, Figure 1.1 renders the sequencings of *family life events* up to 45 years old. The plotted sequences come from the European Social Survey (2006) and concern 487 Scandinavians born between 1930 and 1939. In the left panel, events are aligned on their order of occurrence in the sequence. Each line represents a unique observed order pattern and the line width reflects the frequency of the pattern. Looking at the thickest line, we learn that the most frequent pattern is to first experience "Leaving home" (rank 1), later "First union" and "First marriage" the same year (rank 2), and later again "First childbirth" (rank 3). The lines are jittered to avoid overlapping and to help with identifying typical patterns; only patterns with a minimal support – here 5% – are colored. The diversity of all observed patterns is rendered through the remaining bleached out patterns. To facilitate the tracking of distinct patterns, there are gray arrangement zones at the intersection of the x – rank order of occurrence – and y – event label – coordinates, and the events in an order pattern are represented by solid

[1] A second application, descriptive statistics of the used data and R-codes are available in Appendix A.

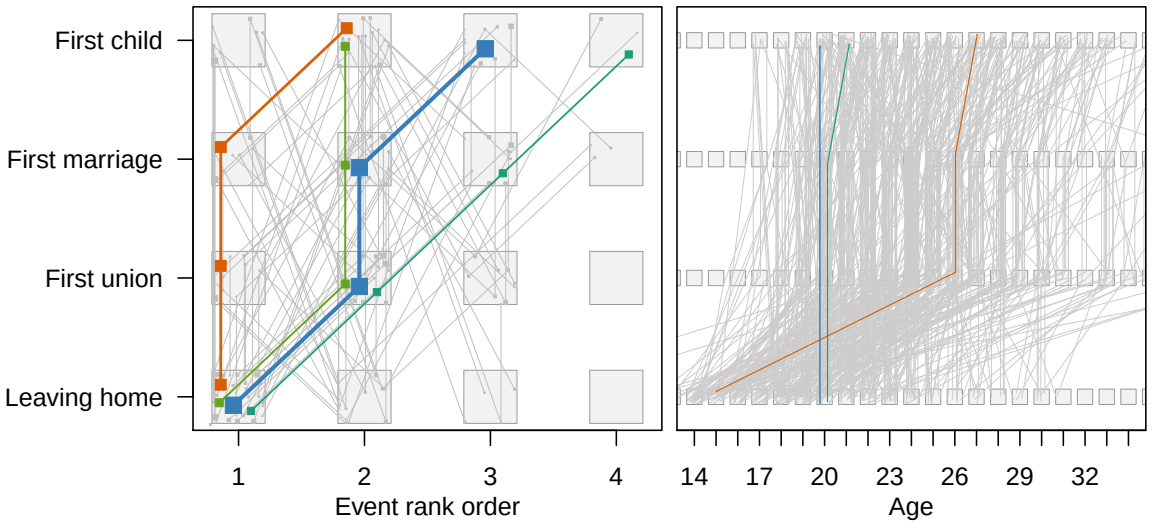

Figure 1.1: Parallel coordinate plot of Scandinavian *family life events* of the 1930-39 birth cohort. *Left panel*, alignment on rank orders of occurrence of the events; *right panel*, alignment on event time stamps. Patterns with frequency below the minimum support of 5% in the left panel and 1% in the right panel are grayed out.

squares occupying the same position inside the successively crossed arrangement zones. Simultaneous events, i.e., events occurring during the same year of age as "First union" and "First marriage" in the most frequent pattern, share the same rank and are connected by a vertical line.

The left panel only accounts for the order of the events. Therefore, each pattern may represent people who experience the events in the corresponding order but not necessarily at the same ages. As shown in the right panel of Figure 1.1, accounting for the timing information dramatically increases the pattern diversity and makes it more difficult to identify typical patterns. The three highlighted patterns are the only ones reaching a 1% support.

The proposed graphical method has been developed to achieve three main objectives: (i) identification of standard patterns with possible simultaneous events; (ii) ability to render the entire diversity of the observed patterns; and (iii) suitability for group comparisons.

The three features of the proposed plot – its ability to render the diversity of observed sequence patterns, to highlight standard patterns and to compare groups – can, for instance, prove useful in life course studies on the de-standardization of life trajectories (e.g. Elzinga and Liefbroer, 2007; Widmer and Ritschard, 2009). The diversity of sequence patterns may be computed from pairwise dissimilarities by means of discrepancy measures, as shown in Studer et al. (2011), and compared between cohorts to study changes over time. The pairwise dissimilarities can be obtained for example with optimal matching for state sequences, and a method as the one described in Ritschard et al. (2013) for event sequences. By simultaneously rendering the diversity with the typical patterns and their frequencies, our plot enriches the information provided by discrepancy measures. This is demonstrated in Section 1.3, where our plot clearly exhibits an increase in diversity of family life event sequence patterns across Scandinavian cohorts.

The literature proposes several methods for rendering categorical sequences. Bar,

mosaic or association plots (Hartigan and Kleiner, 1984; Friendly, 2000) are helpful to render distributions of categorical data and highlight the association between pairs of categorical variables. For example, the barplots in Figure 1.2 render how the events "Leaving home", "First union", "First marriage" and "First child" are distributed among the rank orders of occurrence for the Scandinavian 1930-39 birth cohort. The drawback of these plots is that they neither render individual sequence patterns nor their diversity.

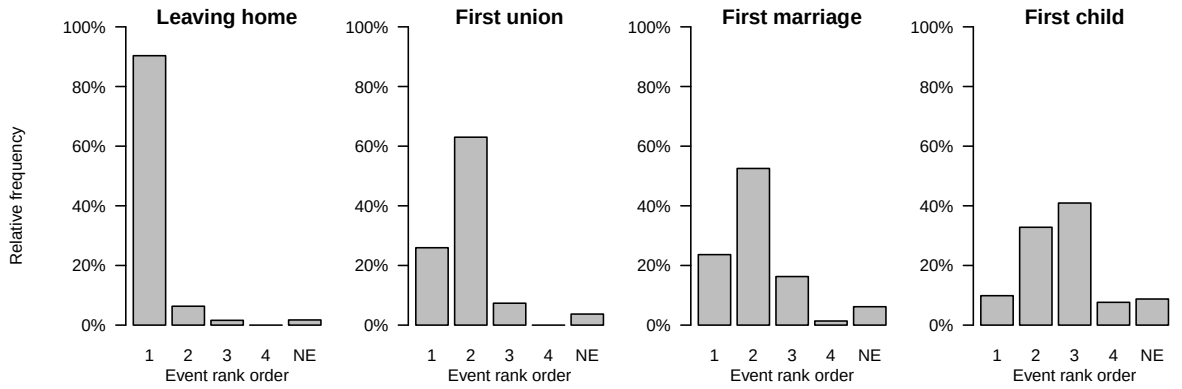

Figure 1.2: Barplots of the distributions of Scandinavian *family life events* of the 1930-30 birth cohort across the event rank orders. "NE" gives the relative frequency of people who did not experience the corresponding event.

Alternatively, by considering the event occurring at each successive position as a categorical variable, a set of sequences can be seen as a series of categorical variables and the successions of events at the successive positions rendered by means of parallel coordinate plots. The plot consists of reporting the position in the sequence (or time point) on the x-axis and assigning a vertical coordinate to each event-category. Each unique sequence pattern is then visualized as a polyline connecting the successive events in the order they appear in the sequence. Varying line widths can be used to visualize the support of each event-to-event segment. Examples of categorical parallel coordinate plots are hammock plots (Schonlau, 2003) and parallel sets (Kosara et al., 2006) and an explicit example of parallel coordinate for sequential patterns can be found in Yang (2003).

The left panel of Figure 1.7 shows a basic version of the categorical parallel plot for the *family life event* sequences of the Scandinavian 1950-59 birth cohort. It can be seen that lines often overlap, which makes it impossible to track single patterns. Figure 1.3 implements the hammock plot and the parallel sets for the Scandinavian 1930-39 birth cohort, by using the gparallel R package (Hofmann and Vendettuoli, 2013). We merged simultaneous events to new event categories, e.g., "FU+FM" reflects the simultaneous occurrence of "First union" and "First marriage". The vertical bars stack relative frequencies of event occurrences at the successive event rank orders, which do not necessarily sum up to 100% because the observed sequence patterns have different lengths. The line segments render the frequency of event-to-event pairs, the implementation of which is the only difference between the two plots. The parallel sets and the hammock plot are useful to visualize the distribution of events at the successive positions and the frequency of event-to-event pairs. However, both plots require a technique to deal with simultaneous events, such as merging them to new event categories, and they not allow to unambiguously track individual sequence patterns with more than two positions.

Among plots specifically designed for sequence data, there are various plots for state sequences (Brzinsky-Fay et al., 2006; Gabadinho et al., 2011). These plots essentially

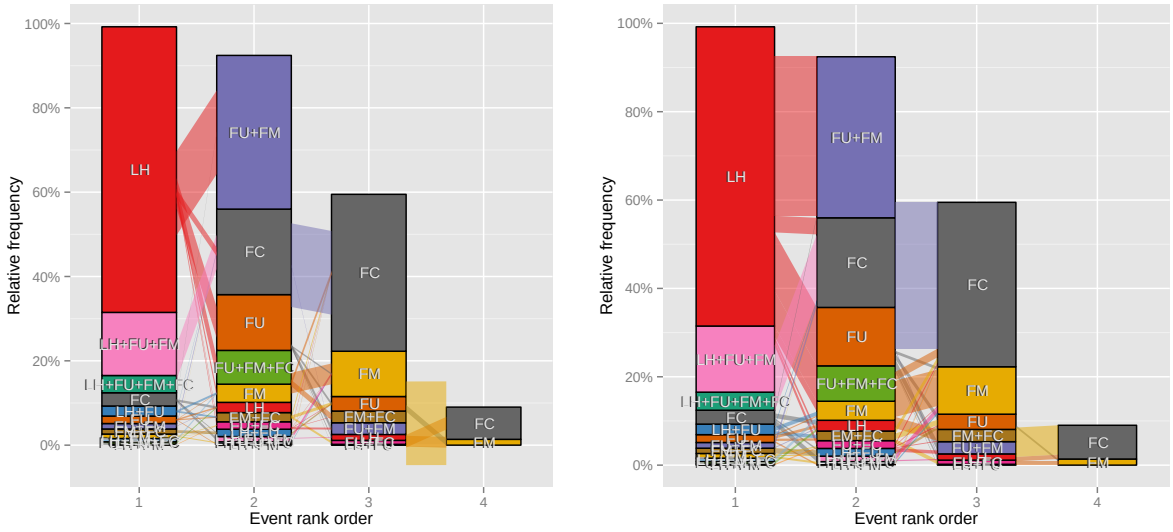

Figure 1.3: Alternative categorical parallel coordinate plots for visualizing the Scandinavian *family life event* sequences of the 1930-39 birth cohort. *Left*, the hammock plot; *right*, the parallel sets. Abbreviations: *LH*, "Leaving home"; *FU*, "First union"; *FM*, "First marriage"; *FC*, "First child".

render the duration of the states and do not apply for sequences made of elements such as events that do not have durations. Alongside the already mentioned parallel coordinate plot, there are two further types of graphics that can potentially be applied to any kind of categorical sequences including event sequences. Graphics of the first type, known as *life lines* or *calendar plots*, arrange color-coded event symbols along horizontal lines (Wang et al., 2010; Wongsuphasawat et al., 2011).

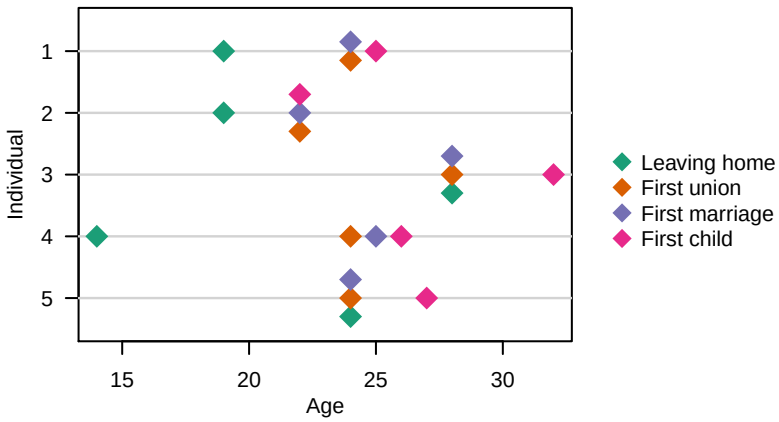

Figure 1.4: Calendar plot for five of the Scandinavian *family life event* sequences of the 1930-39 birth cohort.

Figure 1.4 shows the principle of calendar plots for five individuals of the Scandinavian 1930-39 birth cohort. Each line renders one sequence by placing symbols at the ages at which events occured. Simultaneous events are rendered by vertically stacking up the event symbols. A drawback of calendar plots is that the results become quickly unreadable when many sequences have to be displayed.

The second type of plots are *directed graphs* (Hébrail and Cadalen, 2000; Huzurbazar,

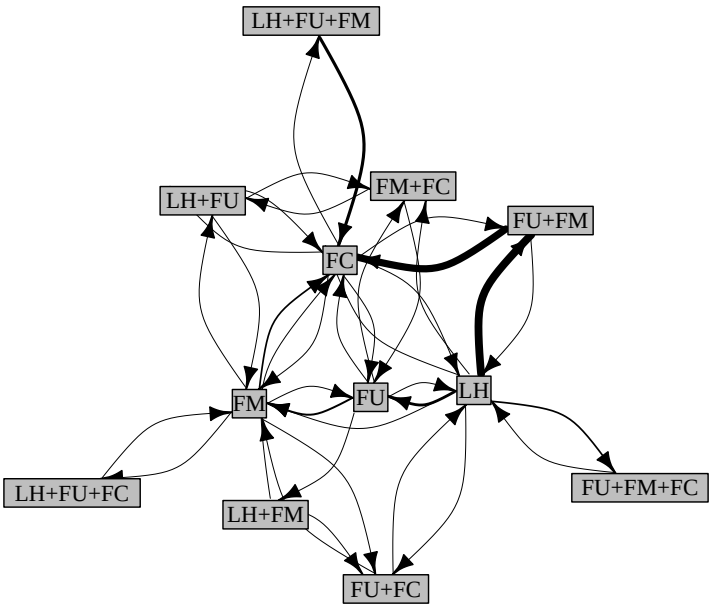

Figure 1.5: Directed graph visualization of the Scandinavian *family life event* sequences of the 1930-39 birth cohort. Nodes represent events or combinations of simultaneous events. Widths of the directed line segments render the frequencies of successions of (simultaneous) events. Experiencing no event or experiencing all four events simultaneously is not illustrated.

2004), such as the graphical representation of a flowgraph, that connect event nodes with directed line segments along the event order. Figure 1.5 implements a directed graph for the Scandinavian 1930-39 birth cohort, by using the R igraph package (Csardi and Nepusz, 2006). Each node represents an event or a combination of simultaneous events, and the widths of the arrows render the frequency of directed event-to-event pairs. For example, it can be seen that "Leaving home" (LH) is commonly followed by the simultaneous occurence of "First union" and "First marriage" (FU+FM). The plot does however leave open whether "Leaving home" is the first event in the sequence. Therefore, a disadvantage of directed graphs is that they do not allow to unambiguously track entire individual sequence patterns.

The decorated parallel coordinate plot proposed in this article extends the parallel coordinate principle with the following main features: (i) algorithmically controlled *jittering*; (ii) possibility to merge *embeddable* sequences; and (iii) filter instruments and criteria to improve the exploratory power of the plot. The plot can also render weighted frequencies of the sequence patterns and cases experiencing no event.

The article is organized as follows. In the upcoming section, we provide additional details on the algorithmically controlled jittering and discuss options for improving plot readability. Subsequently, we extend the family life event example to illustrate the plot capacities including its suitability for comparison purposes. Finally, we address practical issues regarding the plot usage, its scope and limits and conclude by summarizing our findings.

1.2 Jittering, embedding and filtering mechanisms

The basic principle of the proposed plot has been explained in the introduction. This section gives additional details regarding the jittering arrangement, embeddable sequence patterns and filtering criteria.

Jittering arrangement The jittering arrangement is defined within the light gray rectangular *arrangement zone* replicated at each grid point. A distinct location in this zone is assigned to each sequence pattern. For example, the thickest line in the left panel of Figure 1.1 goes through solid squares located at the bottom-center in each crossed arrangement zone. The placing procedure first assigns a solid square of size proportional to the (possibly weighted) sample frequency to each order pattern. Next, a random location is successively assigned to the squares. Location is allocated in decreasing order of the size of the squares and so that squares do not touch each other. In case the remaining space is insufficient, the size of all solid squares are proportionally reduced to make them all fit in the zone. The plot is then finalized by drawing connecting lines between the successive squares belonging to the same pattern. The widths of the line segments are adjusted to the pattern frequency but are slightly thinner than the event-squares for readability. Simultaneous events appear as vertical segments. To maintain the line-continuity in these cases, we connect the precedent event with the lowest event of the vertical segment and the subsequent event with the highest one (or optionally conversely). In the exceptional case where the same event would occur several times at the same position, the multiple occurrence would be reflected by a "sunflower" inscribed in the concerned square. Finally, *zero-event sequences*, i.e., empty sequences corresponding to cases that do not experience any event, are reflected by a square outside the bottom-left arrangement zone.

Full-scaled real data sets will most often include a great number of distinct patterns and additional tricks may be necessary to distinguish patterns of interest in the plot. We propose two such adjustments.

Emphasizing interesting patterns The first option is to bleach out less interesting patterns and lay them in the background. The level of interest will typically be measured by the frequency of the pattern, but could as well be, for example, the inverse frequency if we are interested in atypical patterns, or some measure of the strength of association between the pattern and a target variable such as the sex, birth year or income of the concerned individuals. In Figure 1.1 patterns with support of 5% or higher are colored and all others are bleached out. Instead of the minimal support, we can also chose to highlight the minimum number of patterns such that their cumulated frequency reaches a given threshold. The latter would however only make sense for summable interest measures, and would not make sense for example for association measures.

Plotting only non-embeddable sequence patterns The second option aims at reducing the number of plotted lines without losing information and consists in drawing only *non-embeddable* sequence patterns. A sequence pattern S_1 is *embeddable* into a pattern S_2 if S_2 can be transformed into the exact form of S_1 by cutting an ending – or starting – substring from the sequence S_2. The *non-embeddable* patterns are those unique event order patterns which cannot be embedded into any other one.

The embedding is visualized by adjusting the line widths of shared partial line segments. For instance, in the right panel of Figure 1.7 where non-embeddable sequencing

patterns are plotted, we observe that the ending segment and square of the thick ascending diagonal line are slightly thinner than those at the start of the line, meaning that the line also represents shorter embedded patterns. Compared with the right panel in Figure 1.6 that plots the same data, we see that embedding shorter patterns in longer ones permits to reduce the number of drawn lines from 55 to 30.

The embedding trick raises two difficulties: first, the trick implies a technical ambiguity. Short event order patterns can often be embedded into more than one *non-embeddable* event order candidates. We suggest in that case to embed the patterns into the most frequent pattern among the available candidates. Doing so, instead of distributing them evenly over all candidates for example, will emphasize the commonness of the shared segments. Second, the interpretation becomes ambiguous when two or more event orders with both different start and end positions are embedded in the same non-embeddable event order pattern. For example, the three sequences A-B-B-*, *-B-B-C and A-B-B-C, where a "*" indicates an empty position, can be merged into the single non-embeddable sequence A-B-B-C with a weight of 2 for the paths A-B and B-C, and a weight of 3 for the path B-B. The same non-embeddable sequence results from the three sequences A-B-B-C, A-B-B-C and *-B-B-* and it is thus not possible to unambiguously retrieve the original sequences from the non-embedded sequence; hence the ambiguity. We recommend to use the embedding adjustment only with either left-aligned or right-aligned sequences.

Combining both adjustments Both tricks above can be applied together on the same plot. In that case, when one or more patterns have been embedded in a longer one, the entire non-embeddable event order pattern is highlighted whenever its most frequent segment fulfills the highlighting condition. As a consequence, some non-embeddable patterns which do not themselves reach the minimum interest level may be highlighted just because some other patterns were embedded in them.

1.3 An application: Family life event histories

In order to illustrate the practical scope of the proposed plot and especially its suitability for group comparison, we consider again the 487 Scandinavian family life trajectories of the 1930-39 birth cohort rendered in Figure 1.1 and compare them with the 885 trajectories collected for the 1950-59 cohort. All data come from the 2006 European Social Survey Round 3.[2]

When analyzing life events, a question of interest is whether typical sequencing patterns change or remain the same across age groups and an answer to this question is obtained by plotting side-side the trajectories of the different groups as in Figure 1.6. To facilitate the comparison, the same highlighting color and location in the arrangement zone are used in each of the groups when a pattern is present in several groups. For example, the pattern (Leaving home) − (First union, First marriage) − (First child) is displayed in blue and jittered up-left in both panels[3] of Figure 1.6.

The two plots in Figure 1.6 differ widely. The number of highlighted event orders with at least 5% support increases from four to eight and there are only two common highlighted patterns. The most typical pattern for cohort 1930-39 becomes much less

[2]Further descriptive statistics of the data set can be found in Appendix A.2.1.

[3]Due to the random factor in location and color assignments, the location and colors in Figure 1.6 differ from those in Figure 1.1.

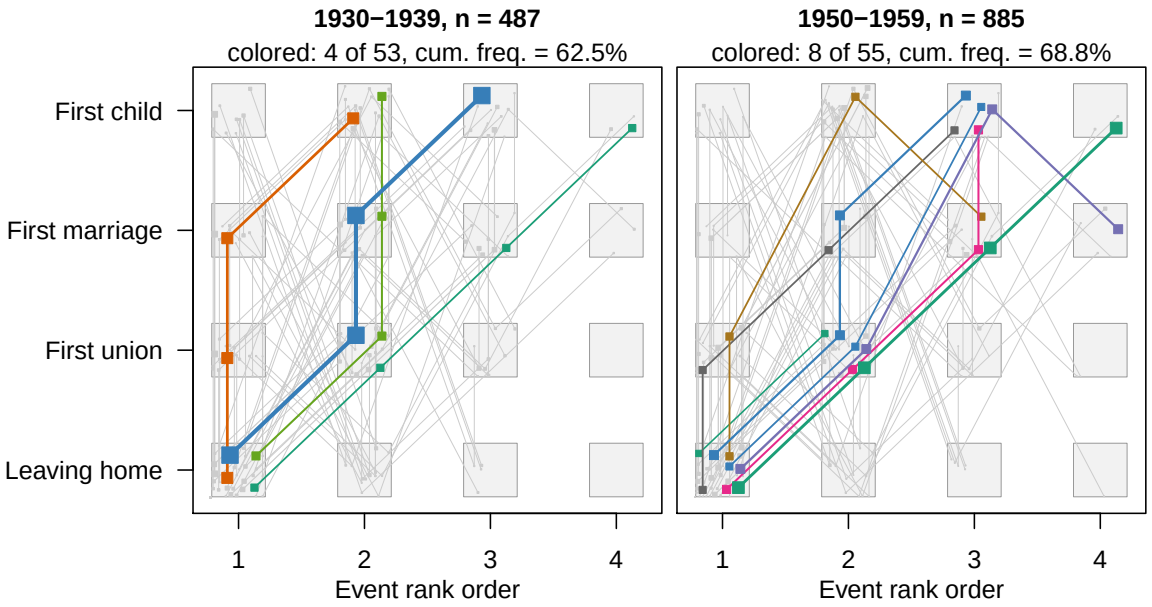

Figure 1.6: Cohort comparison of Scandinavian *family life event* orders. Highlighted lines describe order patterns with weighted frequency above 5%. No embedding.

frequent for the 1950-59 cohort where the most frequent pattern is the diagonal line, i.e., (Leaving home) − (First union) − (First marriage) − (First child). The cohort 1950-59 appears to be much less standardized. The number of frequent patterns increases from four to eight, while the cumulative frequency of the frequent patterns slightly increases from 62.5% to 68.8%. For the youngest cohort, there are also frequent patterns with "First child" without marriage or before "First marriage". In summary, the plot clearly exhibits how norms in the organization of life trajectories changed across cohorts.

The superiority of our extension over the basic parallel coordinate plot appears clearly when comparing the basic plot for the 1950-59 cohort shown in the left panel of Figure 1.7 with the plot in the right panel of Figure 1.6. In the basic plot, the plotted lines overlap, which makes it impossible to track single patterns. Even worse, basic parallel coordinates could be misleading regarding patterns actually not observed. For example, the pattern (First union, First child) − (Leaving home, First marriage) is not present in the data set while the plotted line segments may suggest it is. This problem does not occur with our extension because, as can be seen in Figure 1.6, the distinct sequence patterns are jittered and can be tracked by following up the corresponding event-squares similarly located in the arrangement zones.

The plot for the Scandinavian 1950-59 cohort can be slightly simplified with the embedding trick. The resulting plot is shown in the right panel of Figure 1.7. In that plot, the pattern (Leaving home)−(First union), for example, has been embedded into the pattern (Leaving home)−(First union)−(First marriage)−(First child), and both patterns are visualized by the same single line. The method reduces the total number of lines from 55 to 37 and the number of highlighted patterns from 8 to 6. Due to these changes, the square points within the gray zones have been rearranged, the widths of the event-squares and line segments adjusted, and colors newly reassigned. All these characteristics are therefore different from those in Figure 1.6.

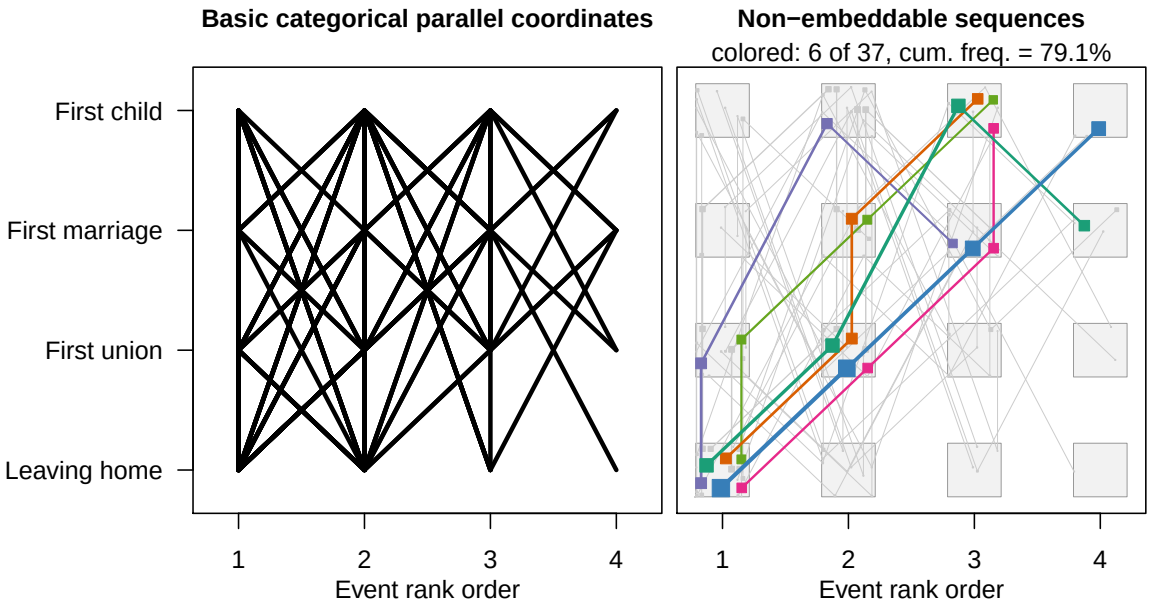

Figure 1.7: Alternative plots of the 1950-59 cohort. *Left panel*, basic parallel coordinate plot; *right panel*, non-embeddable event order patterns.

1.4 About the plot usage

The plot has been implemented in the TraMineR R package (Gabadinho et al., 2011). The `seqpcplot` function producing the plot offers a series of arguments for controlling, among others, the widths of the square-points and lines as well as their coloring, the filtering thresholds and position versus time alignment. The complete list of arguments is documented in the online help file of the `seqpcplot` function where the user also finds several examples.

The coordinate assignment for the event categories is basically arbitrary and could be for instance the alphabetical order. The readability of the solution will, however, most often depend on this coordinate order and could be improved by a suitable ordering. A meaningful solution is for example to arrange the event categories in their most frequently observed order of occurrence as in Section 1.3.

The default representation is obtained by aligning the successive elements in the sequences on their rank order of occurrence. A possible alternative is to align the states/events on their time of occurrence. By using time alignment we can render transition times. Practically, however, when the number of time positions increases the resulting graphic may become very cluttered because of the variability in the timing of similarly sequenced events. The right panel in Figure 1.1, for example, gives the time aligned representation of the Scandinavian *family life event* sequences o cohort 1930-39. The time-aligned plot exhibits a high diversity – essentially a timing diversity – of the trajectories which contrasts with the relatively low sequencing diversity shown in the left panel. We learn from the time-aligned plot that leaving home starts at about 14 years old, and that events "First union", "First marriage" and "First child" occur since age 17 but become much more frequent after 20 years old. Nevertheless, the plot looks cluttered and other plots such as survival curves or life and calendar lines (Wang et al., 2010; Wongsuphasawat et al., 2011) could be more appropriate for rendering the timing. By transforming event

sequences into state sequences – as explained in Ritschard et al. (2009) for example – we could also resort to plots for state sequences (Gabadinho et al., 2011) that explicitly render timing and durations.

Although there are no technical limitations to the scalability of the plot, increasing the number and/or length of the sequences or the alphabet size may impair the plot interest. The limitation is not that of the total number of sequences but that of the number of unique sequences. The number of unique sequences is linked with the sequence length and the size of the alphabet, i.e., the number of distinct events or states. The larger the alphabet, the less chances we have to find out a significant proportion of sequences sharing a common pattern. The same is true for the sequence length: the longer the sequence, the lower the chances of two sequences following a common pattern. The solution to discover regularities in case of a large alphabet would be to merge close elements of the alphabet. In case of long sequences, the solution could be to use a rougher time granularity which would transform the different sequencings of events occurring in a given window of time into a unique set of simultaneous events. To give an order of magnitude, the alphabet should not exceed about 10. Likewise, the plot may become hard to read when sequences contain more than 10 distinct successive elements. With shorter sequences we could afford a larger alphabet and reciprocally with a small alphabet we could afford longer sequences.

1.5 Conclusion

The decorated parallel coordinate plot proposed in this article and provided by the TraMineR R package (Gabadinho et al., 2011) is a powerful tool for exploring how elements are typically ordered in a set of sequences. The filtering mechanisms that dim out less interesting patterns together with the embedding trick, permit the most frequent patterns to be highlighted clearly while still rendering the entire diversity of the observed patterns. Moreover, replicated arrangement zones facilitate the tracking of individual jittered patterns. Although the plot is primarily designed for event sequences where only the rank order of occurrence of the events matters, the plot can also render time aligned events and be used with other types of categorical longitudinal data such as categorical panel data.

Bibliography

Brzinsky-Fay, C., U. Kohler, and M. Luniak (2006). Sequence Analysis with Stata. *The Stata Journal 6*(4), 435–460.

Csardi, G. and T. Nepusz (2006). The igraph Software Package for Complex Network Research. *InterJournal Complex Systems*, 1695.

Elzinga, C. H. and A. C. Liefbroer (2007). De-Standardization of Family-Life Trajectories of Young Adults: a Cross-National Comparison Using Sequence Analysis. *European Journal of Population / Revue européenne de Démographie 23*(3), 225–250.

European Social Survey (2006). ESS Round 3. Data File Edition 3.4, Norwegian Social Science Data Services, Norway – Data Archive and Distributor of ESS Data.

Friendly, M. (2000). *Visualizing Categorical Data*. Cary, USA: SAS Institute.

Gabadinho, A., G. Ritschard, N. S. Müller, and M. Studer (2011). Analyzing and Visualizing State Sequences in R with TraMineR. *Journal of Statistical Software 40*(4), 1–37.

Hartigan, J. A. and B. Kleiner (1984). A Mosaic of Television Ratings. *The American Statistician 38*(1), 32–35.

Hébrail, G. and H. Cadalen (2000). Visualisation et Classification Automatique de Parcours Professionnels. In *Actes des XXXIIe Journées de statistique, Fès, Maroc*, pp. 458–462.

Hofmann, H. and M. Vendettuoli (2013). *ggparallel: Variations of Parallel Coordinate Plots for Categorical Data*. R package version 0.1-1, URL http://CRAN.R-project.org/package=ggparallel.

Huzurbazar, A. V. (2004). *Flowgraph Models for Multistate Time-to-Event Data*. New Jersey, USA: John Wiley & Sons.

Kosara, R., F. Bendix, and H. Hauser (2006). Parallel Sets: Interactive Exploration and Visual Analysis of Categorical Data. *IEEE Transactions on Visualization and Computer Graphics 12*(4), 558 –568.

Ritschard, G., R. Bürgin, and M. Studer (2013). Exploratory Mining of Life Event Histories. In J. J. McArdle and G. Ritschard (Eds.), *Contemporary Issues in Exploratory Data Mining in Behavioral Sciences*, Chapter 9, pp. 221–253. Routledge.

Ritschard, G., A. Gabadinho, M. Studer, and N. S. Müller (2009). Converting Between Various Sequence Representations. In Z. Ras and A. Dardzinska (Eds.), *Advances in Data Management*, Volume 223 of *Studies in Computational Intelligence*, pp. 155–175. Berlin, Germany: Springer-Verlag.

Schonlau, M. (2003). Visualizing Categorical Data Arising in the Health Sciences Using Hammock Plots. In *Proceedings of the Section on Statistical Graphics, American Statistical Association; 2003, CD-ROM*.

Studer, M., G. Ritschard, A. Gabadinho, and N. S. Müller (2011). Discrepancy Analysis of State Sequences. *Sociological Methods and Research 40*(3), 471–510.

Wang, T. D., C. Plaisant, and B. Shneiderman (2010). Temporal Pattern Discovery Using Lifelines2. In *IEEE VisWeek 2010*, Salt Lake City, USA.

Widmer, E. and G. Ritschard (2009). The De-Standardization of the Life Course: Are Men and Women Equal? *Advances in Life course Research 14*(1–2), 28–39.

Wongsuphasawat, K., J. A. G. Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman (2011). LifeFlow: Visualizing an Overview of Event Sequences. In *Proceedings of the 2011 annual conference on Human Factors in Computing Systems (CHI), Vancouver, Canada, May 7-12, 2011*, pp. 1747–1756. New York: ACM.

Yang, L. (2003). Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates. In V. Kumar, M. Gavrilova, C. Tan, and P. L'Ecuyer (Eds.), *Computational Science and Its Applications - ICCSA 2003*, Volume 2668 of *LNCS*, Berlin, Germany, pp. 21–30. Springer-Verlag.

Chapter 2

Tree-based varying coefficient regression for longitudinal ordinal responses

Abstract A tree-based algorithm for longitudinal regression analysis that aims to learn whether and how the effects of predictor variables depend on moderating variables is presented. The algorithm is based on multivariate generalized linear mixed models and it builds piecewise constant coefficient functions. Moreover, it is scalable for many moderators of possibly mixed scales, integrates interactions between moderators and can handle nonlinearities. Although the scope of the algorithm is quite general, the focus is on its usage in an ordinal longitudinal regression setting. The potential of the algorithm is illustrated by using data derived from the British Household Panel Study, to show how the effect of unemployment on self-reported happiness varies across individual life circumstances.[1]

2.1 Introduction

Regression analysis for longitudinal responses addresses a wide range of applications, particularly in medical and social sciences. Siddall et al. (2003), for example, analyze long-term effects of injuries on repeatedly measured pain. Likewise, Oesch and Lipps (2013) use repeatedly measured well-being to examine the impact of the transition from employment to unemployment.

When carrying out longitudinal regression analysis, researchers are specifically interested in the impact of moderator variables on selected regression coefficients in order to enhance insights on the studied relation and/or to control for confounding variables. For example, the effect of an injury could depend on age, while that of unemployment could vary across social groups. Herein, we propose a method to learn such moderation in longitudinal data. The method combines a mixed model approach with a regression tree approach. Although the proposed method applies generally in the multivariate generalized linear mixed model (MGLMM) setting, we focus on its usage with longitudinal ordinally scaled responses such as pain or well-being.

The remainder of the article is organized as follows. The Sections 2.1.1 and 2.1.2 introduce the framework used in the present study and related works. Section 2.2 explains the method in detail. Section 2.3 illustrates its potential by using an empirical example and simulation studies and, finally, Section 2.4 concludes, including addressing the limitations of the proposed method and the software implementation.

[1] Auxiliary calculations, a discussion of a random forest extension, supplementary simulation studies, descriptive statistics of the used data and R-codes are available in Appendix B.

2.1.1 Framework

The proposed algorithm extends multivariate generalized linear mixed models (e.g. Tutz and Hennevogl, 1996) by allowing the fixed coefficients to vary as nonparameterized functions of some moderator variables $Z_1, \ldots, Z_L$. Let $\mathbf{y}_{it}$ denote the $R\times 1$ response vector of individual i at time t, $i = 1, \ldots, N$, $t = 1, \ldots, N_i$. Denote by $\mathbf{X}_{it}$ and $\mathbf{W}_{it}$ the $Q\times P_{\boldsymbol{\beta}}$ and $Q\times P_{\mathbf{b}_i}$ design matrices associated with fixed coefficients $\boldsymbol{\beta}$ and (individual-specific) random coefficients $\mathbf{b}_i$, respectively. Further, denote by $\mathbf{z}_{it}$ the $L\times 1$ vector of moderators, also called *effect modifiers* in the literature (e.g. Hastie and Tibshirani, 1993). MGLMMs link the $Q\times 1$ predictor vector $\boldsymbol{\eta}_{it}$ with the conditional expectation $\boldsymbol{\mu}_{it} = E(\mathbf{y}_{it}|\mathbf{b}_i; \mathbf{X}_{it}, \mathbf{W}_{it}, \mathbf{z}_{it})$ via $\boldsymbol{\mu}_{it} \in \mathbb{R}^R \mapsto \boldsymbol{\eta}_{it} = \mathbf{g}(\boldsymbol{\mu}_{it}) \in \mathbb{R}^Q$, where $\mathbf{g}$ is a known link function. We aim to fit predictor functions of form

$$\mathcal{M} : \boldsymbol{\eta}_{it} = \mathbf{X}_{it}\boldsymbol{\beta}(\mathbf{z}_{it}) + \mathbf{W}_{it}\mathbf{b}_i, \quad \mathbf{b}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_b) \quad . \tag{2.1}$$

The fixed coefficients $\boldsymbol{\beta}(\cdot) = (\beta_1(\cdot), \ldots, \beta_{P_{\boldsymbol{\beta}}}(\cdot))^\top$ of $\mathcal{M}$ are *varying coefficients* that state that the linear effects of the elements of matrix $\mathbf{X}_{it}$ on the expectation of $\mathbf{y}_{it}$ are nonparameterized functions of $\mathbf{z}_{it}$. In the predictor function $\mathcal{M}$, the intercept coefficients are included in $\boldsymbol{\beta}(\cdot)$. Such *varying intercepts* are functions of $\mathbf{z}_{it}$ and estimate the direct effects of $\mathbf{z}_{it}$ on $E(\mathbf{y}_{it}|\cdot)$. In contrast to fixed coefficients, the individual-specific random coefficients $\mathbf{b}_i$ do not depend on $\mathbf{z}_{it}$ in $\mathcal{M}$. Such random coefficients are used to take into account the correlation between repeated responses and could include individual-specific intercepts or slopes over time. As stated in (Eq. 2.1), we assume here that the random coefficients are normally, identically and independently distributed with $\mathrm{E}(\mathbf{b}_i) = \mathbf{0}$ and $\mathrm{Var}(\mathbf{b}_i) = \boldsymbol{\Sigma}_b$.

MGLMMs include models with density functions of the multivariate exponential family that, with random coefficients $\mathbf{b}_i$, have the general form

$$f(\mathbf{y}_{it}|\mathbf{b}_i; \boldsymbol{\beta}, \phi) = \exp\left\{\frac{\mathbf{y}_{it}^\top\boldsymbol{\theta}_{it} - b(\boldsymbol{\theta}_{it})}{\phi} + c(\mathbf{y}_{it}, \phi)\right\} \quad , \tag{2.2}$$

with ϕ the dispersion parameter and $b(\cdot)$ and $c(\cdot)$ family-specific functions. $\boldsymbol{\theta}_{it}$ is the so-called vector of natural parameters. It is here defined as $\boldsymbol{\theta}_{it} = \mathbf{d}(\boldsymbol{\mu}_{it}) = \mathbf{d}(\mathbf{g}^{-1}(\mathbf{X}_{it}\boldsymbol{\beta}(\mathbf{z}_{it}) + \mathbf{W}_{it}\mathbf{b}_i))$, with $\mathbf{d}(\cdot)$ a known, vector-valued function. MGLMMs include, for instance, several univariate models such as the (Gaussian) linear mixed model or the Poisson mixed model. Here, we restrict the consideration of specific models to that of the cumulative logit mixed model, which really requires the multivariate form above.

The cumulative logit mixed model (CLMM) The cumulative logit model (e.g. McCullagh, 1980) is a popular and conceptually simple model for ordinal response variables Y taking ordered categorical values r in $\{1, \ldots, R\}$. It is motivated (e.g. Tutz, 2012) by assuming that Y is a coarse version of a latent continuous variable $Y^* = f(\cdot) + \varepsilon$, with $f(\cdot)$ a function of predictors and ε the error with distribution $\varepsilon \overset{i.i.d.}{\sim}$ Logistic$(0, 1)$. The connection between the observed ordinal and the latent variable is defined as: $Y = r \Leftrightarrow \theta_{r-1} < Y^* \leq \theta_r$; with $-\infty{=}\theta_0 < \theta_1 < \ldots < \theta_R{=}\infty$ the *threshold* coefficients.

The cumulative logit mixed model has been introduced by Hedeker and Gibbons (1994), and Tutz and Hennevogl (1996) exemplified it as a special case of MGLMMs. Here, the CLMM with varying coefficients is defined as follows: Let $\mathbf{y}_{it} = (y_{it1}, \ldots, y_{itR})^\top$ be the response vector of individual i at time t, which is coded as $y_{itr} = 1$ if $Y_{it} = r$ and $y_{itr} = 0$ if

$Y_{it} \neq r$. Assume that $\mathbf{y}_{it}$ is an outcome of a multinomial distribution with the conditional probabilities $E(\mathbf{y}_{it}|\mathbf{b}_i; \mathbf{x}_{it}, \mathbf{w}_{it}, \mathbf{z}_{it}) = \boldsymbol{\pi}_{it}$, with $\mathbf{x}_{it}$ and $\mathbf{w}_{it}$ the predictor vectors to be incorporated into the design matrices $\mathbf{X}_{it}$ and $\mathbf{W}_{it}$. The CLMM links the predictor $\boldsymbol{\eta}_{it}$ with the conditional probabilities $\boldsymbol{\pi}_{it}$ via $\eta_{itq} = g_q(\boldsymbol{\pi}_{it}) = \log((\pi_{it1}+\ldots+\pi_{itq})/(1-\pi_{it1}-\ldots-\pi_{itq})) = \text{logit}(P(Y_{it} \leq q))$ for $q = 1, \ldots, Q = R-1$. The predictor function is defined as

$$\mathcal{M}_{\text{CLMM}} : \begin{bmatrix} \eta_{it1} \\ \vdots \\ \eta_{itQ} \end{bmatrix} = \begin{bmatrix} 1 & & \mathbf{x}_{it}^\top \\ & \ddots & \mathbf{x}_{it}^\top \\ & 1 & \mathbf{x}_{it}^\top \end{bmatrix} \boldsymbol{\beta}(\mathbf{z}_{it}) + \begin{bmatrix} 1 & \mathbf{w}_{it}^\top \\ \vdots & \vdots \\ 1 & \mathbf{w}_{it}^\top \end{bmatrix} \mathbf{b}_i \ , \tag{2.3}$$

where the qth row determines the logits of responding with $\{1, \ldots, q\}$ rather than with $\{q+1, \ldots, R\}$. The first Q elements of $\boldsymbol{\beta}(\cdot)$ are the varying intercepts, or *varying thresholds* $\theta_1(\cdot), \ldots, \theta_{R-1}(\cdot)$ in terms of the latent variable motivation, that take into account the direct effects of the moderators $\mathbf{z}_{it}$. In order to maintain the order $P(Y_{it} \leq 1) \leq \ldots \leq P(Y_{it} \leq Q)$, these intercepts must satisfy $\beta_1(\mathbf{z}_{it}) \leq \ldots \leq \beta_Q(\mathbf{z}_{it}) \; \forall \; (i,t)$. Further, stacking the vectors $\mathbf{x}_{it}^\top$ and $(1, \mathbf{w}_{it}^\top)$ in the design matrices constraints the corresponding effects to be identical for all Q cumulative logits. This constraint, which considerably simplifies the model, is commonly called the *proportional odds assumption* (e.g. McCullagh, 1980) or *parallelism*. For the direct effects of $\mathbf{z}_{it}$, the proportional odds assumption is relaxed in $\mathcal{M}_{\text{CLMM}}$ since the corresponding varying intercepts are logit-specific. Therefore, $\mathcal{M}_{\text{CLMM}}$ can be seen as a *partial proportional odds model* (e.g. Tutz, 2012, Chap. 9.1.3). Note that if $R = 2$, $\mathcal{M}_{\text{CLMM}}$ is equivalent to a logistic mixed model.

The unknown varying coefficients $\boldsymbol{\beta}(\cdot)$ of the predictor function $\mathcal{M}$ (Eq. 2.1) are proposed to be approximated by a piecewise constant function, based on *model-based recursive partitioning*, which is conceptually similar to *regression trees* (e.g. Breiman et al., 1984). These two approaches can be distinguished by their aims: regression trees attempt to discover differences in the mean, while model-based recursive partitioning aims to discover differences in the model coefficients. While recursive partitioning has certain drawbacks, particularly that it is a heuristic and may be instable regarding small changes in the data, its advantages for statistical learning are hardly covered by the alternative methods to date (cf. Hastie et al., 2001, Sec. 10.7). Recursive partitioning is conceptually simple, can handle many inputs (moderators), nonlinearities and interactions, treats inputs of different scales (nominal, continuous etc.) uniformly and yields easily readable outcomes in the form of decision trees.

The algorithm proposed in this study builds on the *model-based recursive partitioning* algorithm (MOB, Zeileis et al., 2008), which provides a unified design for splitting and tree size selection based on M-estimation and which has been extended to various models (e.g. Rusch and Zeileis, 2012; Strobl et al., 2013). We aim to redesign MOB to fit $\mathcal{M}$ (Eq. 2.1) while preserving the algorithm's statistical properties. This redesign involves two adjustments relative to MOB. The first adjustment allows us to include time-varying moderators while maintaining the random effect component. Because MOB fits a tree with unconnected models at the terminal nodes, a split by a time-varying moderator can render impossible the connection between observations of the same individual. Inspired by the algorithms of Hajjem et al. (2011) and Sela and Simonoff (2012), our algorithm builds a closed model that consists of a tree-structured fixed effect component and a global random effect component. By doing so, the observations of an individual are connected with the single set of corresponding random coefficients, regardless of in which nodes these observations fall. The second adjustment tailors the coefficient constancy tests for the variable and tree size selection of MOB to our algorithm.

2.1.2 Related work

Literature on longitudinal varying coefficient regression refers primarily to spline or kernel regression techniques for modeling the fixed coefficients as functions of time. For example, Tutz and Kauermann (2003) and Zhang (2004) develop generalized linear mixed models with time-varying fixed coefficients, based on local polynomial regression, and Kauermann (2000) proposes an implementation for the marginal cumulative logit model. The tree-based approach for varying coefficients originates from combining linear models and regression trees, e.g., see Quinlan (1992) or Alexander et al. (1996). Wang and Hastie (2014) formalize their tree-based algorithm most explicitly as an approach for varying coefficient regression and provide an in-depth comparison of the tree-based and spline/kernel methods. One of the rare explicit tree-based techniques for longitudinal varying coefficient regression is that of Su et al. (2011), focusing on moderation on a single predictor.

Our research also intersects with the recent discussion on longitudinal regression trees based on mixed models. The first implementation may be that of Abdolell et al. (2002), fitting unconnected linear mixed models for subspaces of a single variable. The *mixed effects regression tree* (MERT, Hajjem et al., 2011) and *random effects/EM tree* (RE-EM Tree, Sela and Simonoff, 2012) algorithms aim to approximate general fixed effect components by a piecewise constant function. Similar to our approach, these algorithms fit closed models where only the fixed effect component is built algorithmically. Hajjem (2010) extends MERT for generalized linear mixed models. Eo and Cho (2014) propose with *mixed-effects longitudinal tree* (MELT) an implementation focusing on trends over time and building on the *generalized, unbiased interaction detection and estimation* algorithn (GUIDE Loh, 2002). Specifically, they fit a tree with unconnected linear mixed models that specify polynomials of time in the fixed effect component. Our contribution is extending the scope of longitudinal regression trees based on mixed models to general varying coefficient regression and proposing a new splitting procedure based on MOB (Zeileis et al., 2008). MERT and RE-EM Tree focus, in our terminology, on the case where only varying intercepts are specified and all covariates are assigned to vector $\mathbf{z}_{it}$. MELT, in turn, focuses on the case where $\mathbf{X}_{it}$ represents a polynomial expansion of time and the remaining covariates are assigned to vector $\mathbf{z}_{it}$. Unlike the above tree approaches, our algorithm does not include auto-correlated errors and, unlike MELT, it does not fit separate random coefficients for every terminal node.

2.2 Method

2.2.1 Piecewise constant approximation for varying coefficients

The algorithm approximates the varying coefficients $\boldsymbol{\beta}(\cdot)$ of $\mathcal{M}$ (Eq. 2.1) by using a vectorial piecewise constant function. Consider a partition of the value space $\mathcal{Z}_1 \times \ldots \times \mathcal{Z}_L$ of the L moderators $Z_1, \ldots, Z_L$ into M (terminal) nodes $\{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$. The approximating predictor function is

$$\widehat{\mathcal{M}} : \boldsymbol{\eta}_{it} = \sum_{m=1}^{M} 1(\mathbf{z}_{it} \in \mathcal{B}_m)\mathbf{X}_{it}\boldsymbol{\beta}_m + \mathbf{W}_{it}\mathbf{b}_i \ . \tag{2.4}$$

The right-hand side of $\widehat{\mathcal{M}}$ is linear and states that the elements of $\boldsymbol{\beta}(\cdot)$ may vary across nodes $\mathcal{B}_m$, but that they remain constant within nodes. The total vector of unknown

coefficients of $\widehat{\mathcal{M}}$ is $\boldsymbol{\gamma} := (\boldsymbol{\beta}^\top, \text{vec}(\boldsymbol{\Sigma}_b^{1/2})^\top)^\top$, with length $P_\gamma := MP_\beta + P_{\mathbf{b}_i}(P_{\mathbf{b}_i}+1)/2$. For some MGLMMs, there is also an additional dispersion parameter ϕ.

Estimation The predictor function $\widehat{\mathcal{M}}$ can be estimated by using the techniques for generalized linear mixed models (e.g. Tutz, 2012, Chap. 14.3). We focus on the direct maximization of the marginal log-likelihood equation by using numeric integration. To simplify the integral of that equation, the random coefficients $\mathbf{b}_i$ are standardized as $\mathbf{a}_i = \boldsymbol{\Sigma}_b^{-1/2}\mathbf{b}_i$ so that $\mathbf{a}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_Q)$. As a consequence, the Choleski decomposition $\boldsymbol{\Sigma}_b^{1/2}$ is estimated instead of $\boldsymbol{\Sigma}_b$. The marginal likelihood with standardized random coefficients is

$$\ell(\boldsymbol{\gamma}) = \sum_{i=1}^{N} \log L_i(\boldsymbol{\gamma}) = \sum_{i=1}^{N} \log \int \prod_{t=1}^{N_i} f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma})\phi(\mathbf{a}_i)\, d\mathbf{a}_i \ , \tag{2.5}$$

where $f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma})$ is the family-specific conditional density of $\mathbf{y}_{it}$ and $\phi(\cdot)$ is the multivariate normal density function. To maximize $\ell(\boldsymbol{\gamma})$ of (Eq. 2.5), we solve the score equations $\sum_{i=1}^{N} \frac{\partial \log L_i(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} := \sum_{i=1}^{N} \mathbf{u}_i(\boldsymbol{\gamma}) = \mathbf{0}$ for $\boldsymbol{\gamma}$, where

$$\mathbf{u}_i(\boldsymbol{\gamma}) = \frac{1}{L_i(\boldsymbol{\gamma})} \int \sum_{t=1}^{N_i} \frac{\frac{\partial}{\partial \eta} f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma}) \frac{\partial}{\partial \boldsymbol{\gamma}} \boldsymbol{\eta}_{it}}{f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma})} \prod_{t=1}^{N_i} f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma})\phi(\mathbf{a}_i)\, d\mathbf{a}_i \tag{2.6}$$

is a $P_\gamma \times 1$ vector. Our software solves these equations by using Fisher's scoring algorithm with Gauss-Hermite quadrature to approximate the integral in (Eq. 2.6). Note that the score equations can be expressed as the sum of the observation-scores, $\sum_{i=1}^{N} \mathbf{u}_i(\boldsymbol{\gamma}) = \sum_{i=1}^{N} \sum_{t=1}^{N_i} \mathbf{u}_{it}(\boldsymbol{\gamma})$, where

$$\mathbf{u}_{it}(\boldsymbol{\gamma}) = \frac{1}{L_i(\boldsymbol{\gamma})} \int \frac{\frac{\partial}{\partial \eta} f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma}) \frac{\partial}{\partial \boldsymbol{\gamma}} \boldsymbol{\eta}_{it}}{f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma})} \prod_{t=1}^{N_i} f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma})\phi(\mathbf{a}_i)\, d\mathbf{a}_i \ . \tag{2.7}$$

Estimating CLMMs For the CLMM, the conditional density of $\mathbf{y}_{it}$ is $f(\mathbf{y}_{it}|\mathbf{a}_i;\boldsymbol{\gamma}) = \prod_{r=1}^{R} \pi_{itr}^{y_{itr}} = \prod_{r=1}^{R} (\frac{e^{\eta_{itr}}}{1+e^{\eta_{itr}}} - \frac{e^{\eta_{it,r-1}}}{1+e^{\eta_{it,r-1}}})^{y_{itr}}$, where $\eta_{it0} = -\infty$ and $\eta_{itR} = \infty$. The used fitting function `olmm` of the R package **vcrpart** (Bürgin, 2015) solves the score equations (Eq. 2.6) by using initial values that respect the order $P(Y_{it} \leq 1) \leq \ldots \leq P(Y_{it} \leq Q)$. This procedure generally works well, but problems could arise if some response categories occur very rarely. Fahrmeir and Tutz (2001) mention that the procedure can fit inadmissible thresholds if these are very similar. To avoid this, they propose a reparameterization that could be considered to improve `olmm`. Further, Kosmidis (2014) points out that coefficients can diverge to infinity and therefore proposes an improved estimator. Ad hoc solutions for both problems could be to merge response categories or to specify a sufficiently large minimum node size (see Sec. 2.2.4). Finally, the number of quadrature points for approximating the integral in (Eq. 2.6) can impact the accuracy of the fit. Higher numbers increase the accuracy, however, at the cost of computational time. `olmm` allows to control the number of points manually and uses a default of seven.

2.2.2 Algorithm

The predictor function $\widehat{\mathcal{M}}$ (Eq. 2.4) is a good approximation for $\mathcal{M}$ (Eq. 2.1) if the true coefficient functions are fairly constant within the nodes. To find such nodes, we propose a

breath-first type algorithm (e.g. Russell and Norvig, 2003) that in each iteration splits one of the current M nodes into two. Splitting requires three selections in each step: a node; a moderator; and a split in the selected variable. The constraint is the maintenance of the global random coefficients, on the basis of which a closed model including all observations must be fitted at any stage.

Algorithm 1: Fitting tree-based varying coefficients in MGLMMs.

Input: $\alpha \in [0, 1]$, e.g., $\alpha = 0.05/L$

Initialize $\mathcal{B}_1 \leftarrow \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_L$ and $M \leftarrow 1$

repeat

1 Fit the MGLMM with the predictor function

$$\boldsymbol{\eta}_{it} = \sum_{m=1}^{M} 1(\mathbf{z}_{it} \in \mathcal{B}_m)\mathbf{X}_{it}\boldsymbol{\beta}_m + \mathbf{W}_{it}\mathbf{b}_i \ .$$

2 Test for the constancy of the coefficients $\boldsymbol{\beta}_m$ separately for each variable Z_l, $l = 1, \ldots, L$, in each node $\mathcal{B}_m$, $m = 1, \ldots, M$. This yields $L \times M$ p-values, $p_{11}, \ldots, p_{LM}$, for rejecting coefficient constancy.

if $p_{min} := \min(p_{11}, \ldots, p_{LM}) \leq \alpha$ **then**

3 Select the variable Z_l and node $\mathcal{B}_s$ where $p_{ls} = p_{min}$

foreach *unique candidate split* $\triangle_k$ *in* $\{z_{lit} : \mathbf{z}_{it} \in \mathcal{B}_s\}$ *dividing* $\mathcal{B}_s$ *into two nodes* $\mathcal{B}_{sk1}$ *and* $\mathcal{B}_{sk2}$ **do**

4 Compute $\hat{\ell}_{\triangle_k} = \max_{\boldsymbol{\gamma}} \ell_{\triangle_k}(\boldsymbol{\gamma})$ of the MGLMM

$$\boldsymbol{\eta}_{it} = \sum_{m \neq s}^{M} 1(\mathbf{z}_{it} \in \mathcal{B}_m)\mathbf{X}_{it}\boldsymbol{\beta}_m + \sum_{m=1}^{2} 1(\mathbf{z}_{it} \in \mathcal{B}_{skm})\mathbf{X}_{it}\boldsymbol{\beta}_{sm} + \mathbf{W}_{it}^{\top}\mathbf{b}_i \ .$$

end

5 Split $\mathcal{B}_s$ by $\hat{\triangle} = \arg\max_{\triangle_k} \hat{\ell}_{\triangle_k}$ and set $M \leftarrow M + 1$.

end

until $p_{min} > \alpha$

Algorithm 1 summarizes the proposed algorithm. Varying coefficients are fitted separately on an increasing number of small nodes until the tests in Step 2 accept coefficient constancy, for all moderators in all nodes. These tests are also used in each step to select the node and variable simultaneously, while the split in the variable is selected by using exhaustive search.

In Section 2.2.3 it turns out that the constancy tests for Step 2 must be adjusted, while splitting entirely based on exhaustive search (e.g. Wang and Hastie, 2014) could be applied straightforwardly. We implement these tests for statistical and computational reasons. Statistically, the variable selection based on these tests is not biased towards moderators with many splits, as it is with exhaustive search (cf. Hothorn et al., 2006). Computationally, the advantage is that with such tests the algorithm must refit the model for the splits in the selected variable and node only. By contrast, full exhaustive search requires refitting the model for the splits in all moderators and all nodes, the number of which increases in each iteration.

2.2.3 Coefficient constancy tests for variable, node and tree size selection

Coefficient constancy tests have been studied extensively in econometrics (e.g. Nyblom, 1989; Andrews, 1993). Although these tests, often called *structural change tests*, have been developed to examine coefficient constancy over time, they naturally extend to other variables. For our purposes, it is computationally convenient to focus on score-based tests, such as the *M-fluctuation* tests of Zeileis and Hornik (2007), which merely require us to estimate the model under the H_0 hypothesis of coefficient constancy. Specifically, we want to use the observation-scores $\hat{\mathbf{u}}_{it} := \mathbf{u}_{it}(\hat{\boldsymbol{\beta}})$ of (Eq. 2.7), which allows testing fixed coefficient constancy with respect to both time-varying and time-invariant moderators. Thereby, the remaining coefficients $\boldsymbol{\Sigma}_b$ and ϕ are treated as nuisance parameters. In the following, we summarize the M-fluctuation tests for multivariate generalized linear models (without random coefficients) and introduce two preparatory steps for their use in Algorithm 1. The first step linearly transforms the observation-scores $\hat{\mathbf{u}}_{it}$ to remove intra-individual correlations. The second step extracts and mean-centers the subsets of these scores to apply the tests nodewise. The aim of both steps is to ensure that the transformed observation-scores have approximately the same first two moments and covariances as have the scores of models without random coefficients. While asymptotic aspects are not considered, a comprehensive simulation study is presented in Section 2.3.2.

2.2.3.1 Coefficient constancy tests for multivariate generalized linear models

For a complete description of these M-fluctuation tests, see Zeileis and Hornik (2007). Here, we summarize the M-fluctuation for multivariate generalized linear models. Let $\mathbf{y}_i$, $i = 1, \ldots, N$ be the $R{\times}1$ response vectors and $\mathbf{X}_i$ the corresponding $Q{\times}P_\beta$ design matrices. Assume that $\mathbf{y}_i$ given $\mathbf{X}_i$ follows a distribution of the multivariate exponential family, and that the conditional expectation is determined by $\mathbf{g}(E(\mathbf{y}_i|\mathbf{X}_i)) = \mathbf{X}_i\boldsymbol{\beta}_i$, with $\mathbf{g}$ a known link function. In particular, we test $H_0 : \boldsymbol{\beta}_i = \boldsymbol{\beta}_1$ for all i against the alternative that the coefficients $\boldsymbol{\beta}_i$ change with the values of a variable Z. Using M-fluctuation tests for this approach requires estimating the model under H_0, namely maximizing the likelihood or solving the score equations $\sum_{i=1}^{N} \boldsymbol{\psi}_i(\boldsymbol{\beta}_1) = \mathbf{0}$ for $\boldsymbol{\beta}_1$. By using the fitted model, the cumulative process of the estimated scores along the values of Z,

$$\boldsymbol{\Psi}_N(\tau) = \frac{1}{\sqrt{N}} \sum_{i=1}^{\lfloor \tau N \rfloor} \hat{\boldsymbol{\psi}}_{\sigma(z_i)} \qquad (0 \leq \tau \leq 1) \ , \tag{2.8}$$

is examined for divergences from its expectation $\mathbf{0}$. The $\hat{\boldsymbol{\psi}}_i = \boldsymbol{\psi}_i(\hat{\boldsymbol{\beta}}_1)$ are the estimated scores and $\sigma(z_i)$ is the ordering permutation giving the antirank of observation z_i in vector $(z_1, \ldots, z_N)$. $\boldsymbol{\Psi}_N$ is computed as a P_β-dimensional sequence of length $N + 1$ that starts and ends with zero. Assuming that under H_0: (i) $\mathrm{E}(\hat{\boldsymbol{\psi}}_i) = \mathbf{0}\ \forall\ i$; (ii) $\mathrm{Var}(\hat{\boldsymbol{\psi}}_i) = \mathrm{Var}(\hat{\boldsymbol{\psi}}_1)\ \forall\ i$; and (iii) $\mathrm{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) = \mathrm{Cov}(\hat{\boldsymbol{\psi}}_1, \hat{\boldsymbol{\psi}}_2)\ \forall\ i \neq i'$; which requires that the predictors are *stationary* over the tested variable (cf. Hjort and Koning, 2002), it can be derived (see B.1.1) that $\mathrm{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) = -\frac{1}{N-1}\mathrm{Var}(\hat{\boldsymbol{\psi}}_1)$ for $i \neq i'$ and, consequently, $\mathrm{Cov}(\boldsymbol{\Psi}_N(\tau_1), \boldsymbol{\Psi}_N(\tau_2)) = \frac{\lfloor N\tau_1 \rfloor (N - \lfloor N\tau_2 \rfloor)}{N(N-1)}\mathrm{Var}(\hat{\boldsymbol{\psi}}_1)$ for $\tau_1 < \tau_2$. Moreover, under regularity conditions, $\boldsymbol{\Psi}_N$ can be shown (e.g. Zeileis and Hornik, 2007) to converge under H_0 to a limit process $\boldsymbol{\Psi}^0$ as N tends to infinity. This limit process has covariance $\mathrm{Cov}(\boldsymbol{\Psi}^0(\tau_1), \boldsymbol{\Psi}^0(\tau_2)) = \tau_1(1 - \tau_2)\mathrm{Var}(\boldsymbol{\psi}(\boldsymbol{\beta}_1))$, where $\mathrm{Var}(\boldsymbol{\psi}(\boldsymbol{\beta}_1))$ is the variance of scores

at the true $\boldsymbol{\beta}_1$. In other words, $\boldsymbol{\Psi}_N$ converges to a linear transformation of $P_{\boldsymbol{\beta}}$ independent Brownian bridges $\mathbf{B}^0$. Likewise, the *standardized cumulative score process*

$$\breve{\boldsymbol{\Psi}}_N(\tau) = \hat{\mathbf{J}}^{-1/2}\boldsymbol{\Psi}_N(\tau) \qquad (0 \leq \tau \leq 1) \ , \tag{2.9}$$

where $\hat{\mathbf{J}}$ is an estimate for $\text{Var}(\boldsymbol{\psi}(\boldsymbol{\beta}_1))$, typically $\hat{\mathbf{J}} = \frac{1}{N}\sum_{i=1}^{N}\hat{\boldsymbol{\psi}}_i\hat{\boldsymbol{\psi}}_i^\top$, can be shown to converge to $P_{\boldsymbol{\beta}}$ independent Brownian bridges. To construct a test, a suitable scalar statistic $\lambda(\breve{\boldsymbol{\Psi}}_N)$ is applied, the H_0 distribution of which is simply the limiting distribution of $\lambda(\mathbf{B}^0)$.

Test statistics Our algorithm adopts the statistics used in MOB (Zeileis et al., 2008). For continuous variables, we use the Lagrange multiplier statistic "supLM" of Andrews (1993), which is designed to capture coefficient shifts at a single, unknown cutpoint. It is defined as

$$\lambda_{supLM}\left(\breve{\boldsymbol{\Psi}}_N\right) = \max_{i=\underline{i},\ldots,\bar{i}}\left(\frac{i}{N}\cdot\frac{N-i}{N}\right)^{-1}\left\|\breve{\boldsymbol{\Psi}}_N\left(i/N\right)\right\|_2^2 \ , \tag{2.10}$$

i.e., as the maximum of the squared L_2 norm of $\breve{\boldsymbol{\Psi}}_N$ in interval $[\underline{i}, \bar{i}]$ (e.g., $[\lceil N/10\rceil,\ N - \lceil N/10\rceil]$). Asymptotically, λ_{supLM} is distributed as the supremum of a squared, $P_{\boldsymbol{\beta}}$-dimensional tied-down Bessel process $\sup_\tau(\tau(1-\tau))^{-1}\left\|\mathbf{B}^0(\tau)\right\|_2^2$.

For categorical variables, we use the χ^2-type statistic of Hjort and Koning (2002), which is designed to capture overall between-category coefficient variation. For variables with categories c in $\{1,\ldots,C\}$, it is defined as

$$\lambda_{\chi^2}\left(\breve{\boldsymbol{\Psi}}_N\right) = \sum_{c=1}^{C}\frac{1}{N_c N}\left\|\Delta_c\left(\breve{\boldsymbol{\Psi}}_N(i/N)\right)\right\|_2^2 \ , \tag{2.11}$$

where N_c is the number of observations in category c and $\Delta_c(\breve{\boldsymbol{\Psi}}_N)$ is the increment of $\breve{\boldsymbol{\Psi}}_N$ over the observations of category c. Under H_0, λ_{χ^2} is χ^2-distributed with $P_{\boldsymbol{\beta}}(C-1)$ degrees of freedom.

2.2.3.2 Pre-decorrelating the observation-scores of MGLMMs

Substituting the scores $\hat{\boldsymbol{\psi}}_i$ in (Eq. 2.8) with the scores $\hat{\mathbf{u}}_{it}$ of (Eq. 2.7) is misleading. While the $\hat{\boldsymbol{\psi}}_i$'s depend on each other only via the constraint $\sum_i\hat{\boldsymbol{\psi}}_i = \mathbf{0}$, the $\hat{\mathbf{u}}_{it}$'s are additionally intra-individually linked via (Eq. 2.7). Therefore, $\text{Cov}(\hat{\mathbf{u}}_{it}, \hat{\mathbf{u}}_{it'})$ for $t \neq t'$ is hardly equal to $\text{Cov}(\hat{\mathbf{u}}_{it}, \hat{\mathbf{u}}_{i't'})$ for $i \neq i'$. To illustrate an outcome from using the *raw* scores $\hat{\mathbf{u}}_{it}$ in M-fluctuation tests and to motivate the *pre-decorrelation* transformation below, we consider the following case example: We repeatedly (5,000 times) generated responses $\mathbf{y}_{it}$ with $i = 1,\ldots,50$ and $t = 1,\ldots,10$, from the logistic mixed model: $\mathcal{M}_{ex} : \text{logit}(P(Y_{it}=1)) = \beta_0 + b_i$, with $\beta_0 = 0$ and $b_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$; fitted the true model $\mathcal{M}_{ex}$ on these data; and computed $\breve{\Psi}$ of (Eq. 2.9) from the raw scores $u_{it}(\hat{\beta}_0)$ and from the pre-decorrelated scores $u^*_{it}(\hat{\beta}_0)$. Specifically, to compute $\breve{\Psi}$, we first cumulated scores with indices $t = 1,\ldots,5$, and then scores with indices $t = 6,\ldots,10$. Since we fit the true model on the data, the computed processes $\breve{\Psi}$ should be distributed as a Brownian bridge.

Figure 2.1 compares the variance of a Brownian bridge with the variance of the simulated processes $\breve{\Psi}$, based on the raw scores (left) and the pre-decorrelated scores (right).

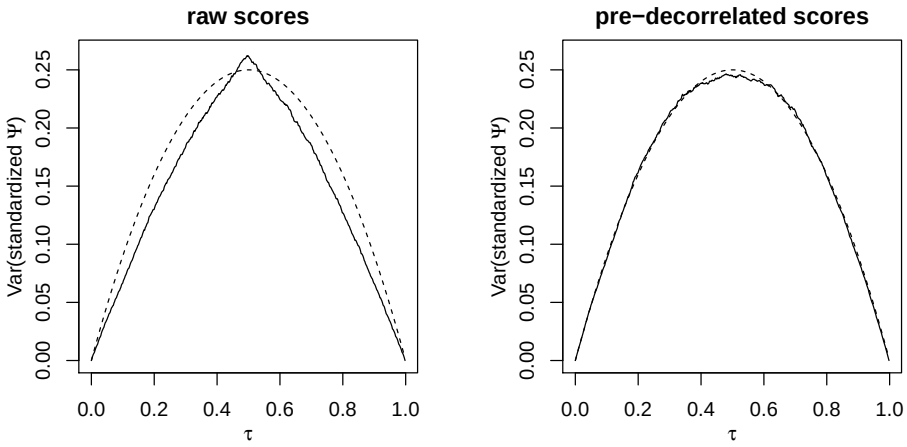

Figure 2.1: Case example: variance of standardized cumulative score processes $\breve{\Psi}$. *Solid lines*, variance of simulated processes based on the raw scores (left panel) and based on the pre-decorrelated scores (right panel); *dashed lines*, the variance of a Brownian bridge. In the right panel the lines cover each other.

The plots suggest that the processes based on the pre-decorrelated scores are distributed as a Brownian bridge, but not the processes based on the raw scores. Further experiments revealed that the variance pattern of processes from raw scores depends on the cumulative order, and that the triangular pattern above is a special case.[2]

The need for adjusting coefficient constancy tests based on standardized cumulative score processes $\breve{\mathbf{\Psi}}_N$ (Eq. 2.9) has previously been discussed in connection with misspecified models or estimation techniques that do not require a fully specified likelihood (e.g., generalized estimating equations, Liang and Zeger, 1986). For example, Chan et al. (2013) discuss the "supLM" test of Andrews (1993) for (misspecified) probit models when the responses are serially correlated. A solution for both cases (e.g. Zeileis and Hornik, 2007) is to use an adjusted estimator for the covariance matrix of scores J in (Eq. 2.9). For example, for time series data the heteroskedasticity and autocorrelation consistent covariance estimator of Andrews and Monahan (1992) may be used. However, adjusting J will not solve our problem of intra-individual correlation between the scores $\hat{\mathbf{u}}_{it}$ of (Eq. 2.7). Adjusting J will merely scale the variance of the process $\breve{\mathbf{\Psi}}_N$, whereas our problem is, as demonstrated in Figure 2.1, that the shape of the variance of $\breve{\mathbf{\Psi}}_N$ from the scores $\hat{\mathbf{u}}_{it}$ can be different than that of a Brownian bridge.

The proposed pre-decorrelated scores $\hat{\mathbf{u}}^*_{it}$ are computed by using the linear within-individual transformation

$$\hat{\mathbf{u}}^*_{it} = \hat{\mathbf{u}}_{it} + \mathbf{T} \sum_{t'=1, t' \neq t}^{N_i} \hat{\mathbf{u}}_{it'} \ , \tag{2.12}$$

where $\mathbf{T}$ is the $MP_\beta \times MP_\beta$ transformation matrix, such that under H_0

$$\mathrm{E}(\hat{\mathbf{u}}^*_{it}) = \mathbf{0} \ \forall \ i, t \ , \tag{2.13}$$

$$\mathrm{Var}(\hat{\mathbf{u}}^*_{it}) = \mathrm{Var}(\hat{\mathbf{u}}^*_{11}) \ \forall \ i, t \ \text{ and} \tag{2.14}$$

$$\mathrm{Cov}(\hat{\mathbf{u}}^*_{it}, \hat{\mathbf{u}}^*_{it'}) = \mathrm{Cov}(\hat{\mathbf{u}}^*_{it}, \hat{\mathbf{u}}^*_{i't''}) = -\frac{1}{\sum_{i=1}^{N} N_i - 1} \mathrm{Var}(\hat{\mathbf{u}}^*_{11}) \ , \tag{2.15}$$

[2] Appendix B.4.1 shows the variance of standardized cumulative score processes of further scenarios.

for all $(i,t) \neq (i,t')$ and $(i,t) \neq (i',t'')$. The transformation forces the expectation, the variance and the covariance of $\mathbf{u}^*_{it}$'s to comply with those of the $\boldsymbol{\psi}_i$'s, see assumptions (i)–(iii) on page 33. Therefore, if such a matrix $\mathbf{T}$ exists, we can assume that the covariance of processes $\breve{\boldsymbol{\Psi}}_N$ (Eq. 2.9) based on the $\mathbf{u}^*_{it}$'s is the same as that based on the $\boldsymbol{\psi}_i$'s.

Balanced data For balanced data where $N_i = N_1 \; \forall \; i$, the scores are symmetrical in the sense that every score $\hat{\mathbf{u}}_{it}$ relates to $N_1 - 1$ "internal" counterparts $\{\hat{\mathbf{u}}_{it'} : t \neq t'\}$ via (Eq. 2.7) and the constraint $\sum_{i,t} \hat{\mathbf{u}}_{it} = 0$; and to $(N-1)N_1$ "external" counterparts only via $\sum_{i,t} \hat{\mathbf{u}}_{it} = 0$. Therefore, we assume that under H_0: (iv) $\mathrm{E}(\hat{\mathbf{u}}_{it}) = \mathbf{0} \; \forall \; (i,t)$; (v) $\mathrm{Var}(\hat{\mathbf{u}}_{it}) = \mathrm{Var}(\hat{\mathbf{u}}_{11}) \; \forall \; (i,t)$; (vi) $\mathrm{Cov}(\hat{\mathbf{u}}_{it}, \hat{\mathbf{u}}_{it'}) = \mathrm{Cov}(\hat{\mathbf{u}}_{11}, \hat{\mathbf{u}}_{12}) \; \forall \; i, \; t \neq t'$; and (vii) $\mathrm{Cov}(\hat{\mathbf{u}}_{it}, \hat{\mathbf{u}}_{i't'}) = \mathrm{Cov}(\hat{\mathbf{u}}_{11}, \hat{\mathbf{u}}_{21}) \; \forall \; t, t', \; i \neq i'$. Under these assumptions, $\mathbf{T}$ is found by solving $\widehat{\mathrm{Cov}}(\hat{\mathbf{u}}^*_{11}, \hat{\mathbf{u}}^*_{12}) - \widehat{\mathrm{Cov}}(\hat{\mathbf{u}}^*_{11}, \hat{\mathbf{u}}^*_{21}) = \mathbf{0}$, see B.1.2 for details. The resulting multiple quadratic equation depends on N_1, $\widehat{\mathrm{Var}}(\hat{\mathbf{u}}_{11})$, $\widehat{\mathrm{Cov}}(\hat{\mathbf{u}}_{11}, \hat{\mathbf{u}}_{12})$ and $\widehat{\mathrm{Cov}}(\hat{\mathbf{u}}_{11}, \hat{\mathbf{u}}_{21})$ and can be solved numerically, e.g., with Newton's method.

Unbalanced data For unbalanced data, the scores $\hat{\mathbf{u}}_{it}$ are not symmetrical in the sense above. Therefore, the assumptions (v)–(vii) would hardly hold. To use the solution for $\mathbf{T}$ for balanced data, we construct a balanced score matrix by recomputing the scores of individuals i with $N_i < N_{max} = \max_{i'} N_{i'}$ under the inclusion of $N_{max} - N_i$ imputed values. The imputation procedure is described in B.1.3, where the crucial point is the generation of response values by means of the model under H_0. Denote by $\hat{\tilde{\mathbf{u}}}_{i1}, \ldots, \hat{\tilde{\mathbf{u}}}_{iN_i}$ the recomputed scores for individual i and by $\hat{\tilde{\mathbf{u}}}_{i,N_i+1}, \ldots, \hat{\tilde{\mathbf{u}}}_{iN_{max}}$ the scores corresponding to the imputed observations. The pre-decorrelation (Eq. 2.12) for *incomplete* individuals yields

$$\hat{\mathbf{u}}^*_{it} = \hat{\tilde{\mathbf{u}}}_{it} + \mathbf{T} \left(\sum_{t'=1, t' \neq t}^{N_i} \hat{\tilde{\mathbf{u}}}_{it'} + \sum_{t'=N_i+1}^{N_{max}} \hat{\tilde{\mathbf{u}}}_{it'} \right) \; . \tag{2.16}$$

Matrix $\mathbf{T}$ is based on the raw scores $\hat{\mathbf{u}}_{it}$ of individuals with $N_i = \max_{i'} N_{i'}$ and the recomputed scores $\hat{\tilde{\mathbf{u}}}_{it}$ of individuals with $N_i < \max_{i'} N_{i'}$.

The proposed solution for unbalanced data perturbs the tests because of the randomness involved in the imputation. To account for this, we repeat the entire test procedure (e.g., five times) and use the average of the resulting p-values.

2.2.3.3 Nodewise tests

Step 2 in Algorithm 1 processes the coefficient constancy tests separately for each variable $Z_1, \ldots, Z_L$ in each node $\mathcal{B}_1, \ldots, \mathcal{B}_M$. The nodewise implementation has two advantages: (i) it is a computationally convenient to select the node to split, and (ii) it eliminates the dependency between the node predictor "$1(\mathbf{z}_{it} \in \mathcal{B}_m)$" and the variables $Z_1, \ldots, Z_L$ that violates the *stationarity* assumption of page 33.

The procedure for testing coefficient constancy regarding a variable Z_l in a node $\mathcal{B}_m$ involves five steps. First, we compute the $N_T = \sum_{i=1}^{N} N_i$ pre-decorrelated scores $\hat{\mathbf{u}}^*_{it}$. Second, we extract from the obtained $N_T \times MP_\beta$ score matrix $\hat{\mathbf{U}}^*$ and the $N_T \times 1$ vector $\mathbf{z}_l$ the N_m observations corresponding to node $\mathcal{B}_m$. Third, to ensure that the sum of scores is zero and that the tests are independent across nodes (see B.1.4), we mean-center the score matrix by column. Fourth, we compute $\breve{\boldsymbol{\Psi}}_{N_m}$ of (Eq. 2.9) by substituting the $\hat{\boldsymbol{\psi}}_i$'s of Section 2.2.3.1 with the elements of the column-centered node score matrix. Finally,

we extract the P_β columns of $\check{\boldsymbol{\Psi}}_{N_m}$ corresponding to $\boldsymbol{\beta}_m$ and compute the test statistic and the p-value.

2.2.4 Further details

Splitting Step 4 of Algorithm 1 cycles through the unique candidate splits in the values of the selected moderator Z_l in the selected node $\mathcal{B}_s$. Splits for ordinal or continuous moderators are based on rules of the form $\{\text{is } z_{lit} \leq \zeta_k?\}$, with ζ_k the unique values in the set $\{z_{lit} : \mathbf{z}_{it} \in \mathcal{B}_s\}$. For nominal moderators, we use rules of the form $\{\text{is } z_{lit} \in \zeta_k?\}$ where the ζ_k's are groupings of the categories in $\{z_{lit} : \mathbf{z}_{it} \in \mathcal{B}_s\}$. Thereby, to have sufficient observations to estimate the nodewise coefficients, we evaluate by default only those splits that yield nodes with a minimum size of 50 observations. For computational efficiency, our software also implements the split reduction techniques of Wang and Hastie (2014) that provide control on the maximum number of evaluated splits at each iteration.

Tree size The significance threshold α is the principal tuning parameter to control the tree size. Conventionally, this parameter is interpreted as the probability of a type I error, i.e., the probability of falsely rejecting coefficient constancy in a node. To account for the multiple test setting, a nodewise Bonferroni correction is applied, and for a 5% value probability a value of 0.05 divided by the number of moderators would be used. An alternative to determine α, which is not investigated in more detail here, is the use of cross-validation (e.g. Hastie et al., 2001, Sec. 7).

Alternative specifications Alternative fixed effect components to those in $\mathcal{M}$ (Eq. 2.1) can be specified by means of simple modifications. For instance, single fixed coefficients – without moderation – can be integrated by omitting them from the splitting procedure. In the latter case, the component $\mathbf{X}_{it}\boldsymbol{\beta}(\mathbf{z}_{it})$ of $\mathcal{M}$ would be decomposed as $\mathbf{X}_{1it}\boldsymbol{\beta}_1(\mathbf{z}_{it}) + \mathbf{X}_{2it}\boldsymbol{\beta}_2$. This approach can be useful to define a non-zero mean for a random slope.

Time-varying moderators Time-varying covariates such as the education level are common in longitudinal studies, e.g., see the empirical example below (Table 2.1). To allow such time-varying covariates, we use a closed model approach that ensures that the random effect component is maintained when splitting by time-varying moderators. However, the inclusion of time-varying moderators may raise interpretability problems. For example, when focusing on varying trends over time as do Eo and Cho (2014), splits in time-varying variables mean that individuals can switch between different static trends, which could be difficult to communicate. In such cases, it may be better to omit the time-varying moderators, or to summarize them as time-invariant variables, e.g., see Eo and Cho (2014, Sec. 2.5).

2.3 Results

2.3.1 Empirical example

To illustrate the scope of the method, we study the effect of the transition from employment to unemployment on self-reported *happiness* (on a scale of 1="Much less", 2="Less

so", 3="Same as usual" and 4="More than usual") by using data derived from the British Household Panel Survey (Taylor et al., 2010). Specifically, we extracted a subset of cases from the first 18 yearly waves (1991–2008). This subset includes those respondents who experienced at least one switch from (self-) employment to unemployment between two consecutive waves. To isolate the effect of the transition, we consider for each retained respondent a single trajectory formed by the up-to-three-year employment period before the unemployment spell and the up-to-three-year unemployment spell that followed employment. The individual periods therefore include between two and six observations. For example, the period of a respondent who was first a student, then worked for two years, then was unemployed for a year, and then found another job would consist of the two years of employment and the year of unemployment. Alternatively, the period of a respondent that worked for 12 years before being unemployed for five would consist of the last three years of employment and the first three of unemployment. If a respondent experienced multiple transitions, only the first was retained.[3] The used data include $1,487$ respondents and a total of $5,054$ observations.

To estimate the effect of the transition to unemployment, we use cumulative logit mixed models for *happiness*, Y, including the dummy coded fixed coefficient predictor *unemployed*, UE, and, to take into account intra-individual correlation, respondent-specific random intercepts.

Table 2.1: Moderator variables for the analysis of the effect of unemployment on happiness. Abbreviations: ti = time-invariant; tv = time-varying.

	Variable	Label	Characteristics	
1	Gender	GENDER	ti	0, Female; 1, Male
2	Age	AGE	tv	16, ..., 64 years
3	Education	EDU	tv	0, Lower; 1, Upper; 2, Tertiary
4	Lives with spouse	SPINHH	tv	0, No; 1, Yes
5	Household income	HHINC	tv	0.55 ..., 4.65 (equivalence scale)
6	Regional unemp.	UEREG	tv	0.05, ..., 10.2%
7	Sectoral unemp.	UESEC	tv	0, ..., 13.6%
8	Financial situation	FISIT	tv	0, Finding it very difficult; ...; 4, Living comfortably
9	spouse has job	SPJB	tv	0, No partner; 1, No; 2, Yes
10	Marital status	MASTAT	tv	0, Never married; 1, Married; ...; 5, Separated
11	Head of household	HOH	tv	0, No, 1; Yes
12	Number of children	NCHILD	tv	0, ..., 7
13	Resp. for child < 16	RACH16	tv	0, No; 1, Yes
14	Time unemployed	TUE	tv	-3, ..., 2 years

We use our algorithm to select and incorporate variables that moderate the effect of *unemployed* and/or have a direct effect on *happiness*. Following Oesch and Lipps (2013, see below) and own considerations, we retained the 14 variables listed in Table 2.1.[4] First, we consider that the variables 1–13 potentially moderate the effect of *unemployed* and/or affect *happiness* directly. This leads us to the varying coefficients CLMM

$$\mathcal{M}_1 : \text{logit}(P(Y_{it} \leq q)) = \beta_q(\mathbf{z}_{it}) + \text{UE}_{it}\beta_4(\mathbf{z}_{it}) + b_i, \;\; b_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma_b) \;\; , \tag{2.17}$$

[3] Appendix B.6.1 shows a couple of observed trajectories.

[4] Descriptive univariate statistics of these variables can be found in Appendix B.6.2.

for $q = 1, 2, 3$, where $\mathbf{z}_{it} = (\text{GENDER}_{it}, \ldots, \text{RACH16}_{it})^\top$ is the 14×1 vector of moderators. In $\mathcal{M}_1$, the direct effects of the moderators are estimated by the varying intercepts $\beta_1(\cdot)$, $\beta_2(\cdot)$ and $\beta_3(\cdot)$ and the moderation effects by the varying coefficient $\beta_4(\cdot)$. We fitted $\mathcal{M}_1$ by using $\alpha = 0.05$ plus the Bonferroni correction. The computation time was 45 seconds with a 3.5GHz processor.

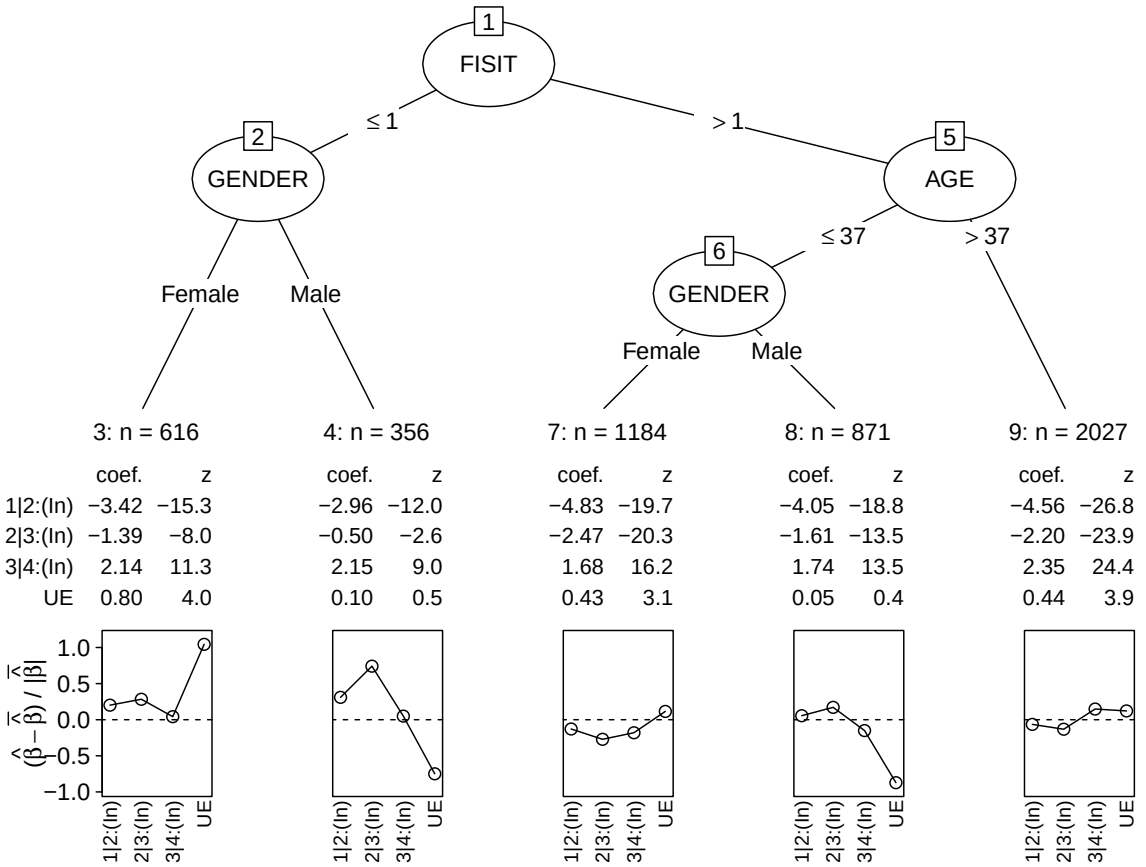

Figure 2.2: *Top* fitted tree structure; *middle*, nodewise coefficients β_{m1} to β_{m4} with the corresponding z-values; *bottom*, relative differences between nodewise coefficients and the associated node-size weighted average coefficients $\bar{\hat{\boldsymbol{\beta}}} = (-4.28, -1.95, 2.05, 0.39)^\top$. The selected moderators are *financial situation* (FISIT), *gender* and *age*.

Figure 2.2 shows the fitted tree structure and the nodewise coefficients of the fit for model $\mathcal{M}_1$. The node panels report the nodewise estimates for the varying coefficients and the corresponding z-values, where $z = \hat{\beta}/\widehat{\text{Sd}}(\hat{\beta}_{mp})$. The estimated standard errors $\widehat{\text{Sd}}(\hat{\beta}_{mp})$ are based on the expected Fisher information matrix and do not account for the error of the model selection procedure. The plots on the bottom show the relative difference between the nodewise coefficients and the corresponding sample-average coefficients. The estimated variance of the random intercepts, which is not shown in Figure 2.2, is $\hat{\Sigma}_b = 1.31$. The algorithm selects 3 of the 13 considered variables and partitions the data into 5 nodes. After the root node, it splits successively the nodes 5, 6, and 2. In the root node, all variables except *regional unemployment*, *sectoral unemployment*, *head of household* and *number of childs* show Bonferroni-corrected p-values below 0.05 in the coefficient constancy tests.

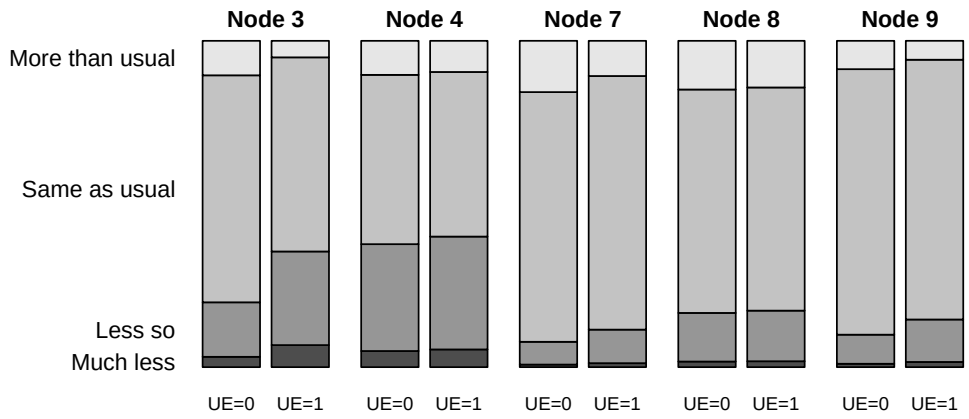

Figure 2.3: Fitted model for $\mathcal{M}_1$: Predicted conditional distributions (with $b_i = 0$) of happiness during employment (UE = 0) and unemployment (UE = 1).

The fitted model for $\mathcal{M}_1$ can be analyzed by using Figure 2.2 or the predicted distributions (conditional on $b_i = 0$) in Figure 2.3. Here, we focus on Nodes 3 and 4 that include respondents in awkward *financial situations* and where the effect of *unemployed* is considerably moderated by *gender*. For females (Node 3), the cumulative logits are estimated to increase by 0.8 at the transition (corresponding to an odds ratio of $e^{0.8} = 2.2$), while for males (Node 4) the cumulative logits are estimated to increase by 0.1 (odds ratio of $e^{0.1} = 1.1$). This finding does not mean that the male respondents are happier than females; rather, the high intercepts in Node 4 indicate that the corresponding respondents are generally less happy than others, whether employed or not (direct effect).

The fitted model can also be expressed by an explicit formula. For example, let us consider the logits for whether Respondent 1 replies with one of the two lower *happiness* categories "Much less" and "Less so". The estimated fixed coefficients are shown in Figure 2.2, and the posterior mean estimate (e.g. Tutz, 2012, Chap. 14.3.2) for the random effect of Respondent 1 is $\hat{b}_1 = -0.10$. The predictor function can then be written as

$$\text{logit}(P(Y_{it} \leq 2)) = \begin{cases} -1.39 + 0.80 \cdot \text{UE}_{it} - 0.10 & \textbf{if FISIT} \leq 1 \text{ and Female} \\ -0.50 + 0.10 \cdot \text{UE}_{it} - 0.10 & \textbf{if FISIT} \leq 1 \text{ and Male} \\ -2.47 + 0.43 \cdot \text{UE}_{it} - 0.10 & \textbf{if FISIT} > 1 \text{ and AGE} \leq 37 \text{ and Female} \\ -1.61 + 0.05 \cdot \text{UE}_{it} - 0.10 & \textbf{if FISIT} > 1 \text{ and AGE} \leq 37 \text{ and Male} \\ -2.20 + 0.44 \cdot \text{UE}_{it} - 0.10 & \textbf{if FISIT} > 1 \text{ and AGE} > 37. \end{cases} \tag{2.18}$$

The predictor functions for the first and the third logit are obtained by substituting the intercepts of (Eq. 2.18). The considered Respondent 1 is a female person, is over 50 years old and evaluates her financial situation with either 0 and 1. Therefore, empirically, only the first equation applies to this person.

Predictive performance To evaluate the performance of the algorithm in this application, we compare the negative log-likelihood prediction errors of fits of $\mathcal{M}_1$ and fits of two reference cumulative logit random intercept models: $\mathcal{M}_2$, a basis model and $\mathcal{M}_3$, a sophisticated model. The prediction errors of fits for $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ were estimated by using cluster bootstrap (Field and Welsh, 2007). We generated 250 bootstrap samples

$(\mathcal{D}_1^*, \ldots, \mathcal{D}_{250}^*)$ from the total data $\mathcal{D}$, and fitted each model on each bootstrap sample. The bootstrap samples were drawn by randomly selecting $1,487$ respondents with replication from $\mathcal{D}$, and retaining the repeated observations corresponding to the selected respondents. Let $\widehat{\mathcal{M}}_{jk}^*$ be a fit for model $\mathcal{M}_j$, $j = 1, 2, 3$, based on the bootstrap sample $\mathcal{D}_k^*$, $k = 1, \ldots, 250$. Let $f_{\widehat{\mathcal{M}}_{jk}^*}(\mathbf{y}_{it}|b_i = 0)$ be the conditional density of $\mathbf{y}_{it}$ in $\widehat{\mathcal{M}}_{jk}$ with b_i set to its expected value 0. The negative log-likelihood prediction error of $\widehat{\mathcal{M}}_{jk}^*$ is computed as

$$\text{err}(\widehat{\mathcal{M}}_{jk}^*) = \frac{1}{N_{\{\mathcal{D}\setminus\mathcal{D}_k^*\}}} \sum\nolimits_{i,t\in\{\mathcal{D}\setminus\mathcal{D}_k^*\}} - \log f_{\widehat{\mathcal{M}}_{jk}}(\mathbf{y}_{it}|b_i = 0) \ , \tag{2.19}$$

where $\{\mathcal{D} \setminus \mathcal{D}_k^*\}$ is the set of observations of $\mathcal{D}$ that does not appear in $\mathcal{D}_k^*$ and $N_{\{\mathcal{D}\setminus\mathcal{D}_k^*\}}$ is the size of this set of observations. Below, we will examine the pairwise differences $\text{err}(\widehat{\mathcal{M}}_{1k}^*) - \text{err}(\widehat{\mathcal{M}}_{2k}^*)$ and $\text{err}(\widehat{\mathcal{M}}_{1k}^*) - \text{err}(\widehat{\mathcal{M}}_{3k}^*)$.

$\mathcal{M}_2$: The basis model First we compare the prediction error of fits for $\mathcal{M}_1$ and fits for the simple cumulative logit model

$$\mathcal{M}_2 : \text{logit}(P(Y_{it} \leq q)) = \beta_q + \text{UE}_{it}\beta_4 + b_i \ .$$

$\mathcal{M}_2$ keeps the varying coefficients of $\mathcal{M}_1$ constant and, thus, ignores the variables of Table 2.1. Therefore, the comparison of $\mathcal{M}_2$ with $\mathcal{M}_1$ evaluates the ability of our algorithm to learn moderation or direct effects.

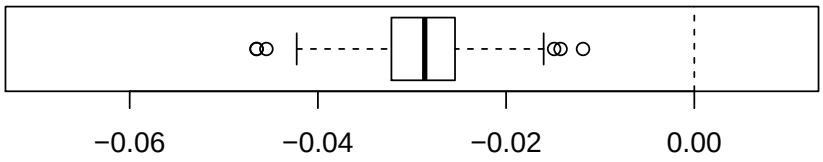

Figure 2.4: Boxplot for 250 pairwise differences $\text{err}(\widehat{\mathcal{M}}_{1k}^*) - \text{err}(\widehat{\mathcal{M}}_{2k}^*)$ comparing the prediction error of fits for $\mathcal{M}_1$ and $\mathcal{M}_2$.

Figure 2.4 shows the boxplot of the computed differences $\text{err}(\widehat{\mathcal{M}}_{1k}^*) - \text{err}(\widehat{\mathcal{M}}_{2k}^*)$ in a boxplot. It can be seen that fits for $\mathcal{M}_1$ outperform, without exception, fits for $\mathcal{M}_2$, indicating that the algorithm significantly improves the model in this application.

$\mathcal{M}_3$: A linear CLMM with direct and moderation effects We also wanted to compare fits for model $\mathcal{M}_1$ with fits for a more sophisticated model. Inspired by the study of Oesch and Lipps (2013), we consider the CLMM

$$\begin{aligned}\mathcal{M}_3 : \text{logit}(P(Y_{it} \leq q)) &= \beta_q + \text{GENDER}_{it}\beta_4 + \sum\nolimits_{j=0}^{1} 1(\text{GENDER}_{it}{=}j)\times \\ \Big[\overline{\text{AGE}}_{it}\beta_{5,j} &+ \overline{\text{AGE}}_{it}^2\beta_{6,j} + 1(\text{EDU}_{it}{=}1)\beta_{7,j} + 1(\text{EDU}_{it}{=}2)\beta_{8,j} + \text{SPINHH}_{it}\beta_{9,j} + \\ \log \text{HHINC}_{it}\beta_{10,j} &+ \text{UE}_{it}\beta_{11,j} + 1(\text{TUE}_{it}{=}{-}1)\beta_{12,j} + \text{UEREG}_{it}\beta_{13,j} + \\ &(\text{UE}_{it}{\times}\text{UEREG}_{it})\beta_{14,j} + \text{UESEC}_{it}\beta_{15,j} + (\text{UE}_{it}{\times}\text{UESEC}_{it})\beta_{16,j}\Big] + b_i \ .\end{aligned}$$

In their study of the effect of unemployment on well-being, Oesch and Lipps estimate separate models for females and males. Equivalently, we specify in $\mathcal{M}_3$ the interaction between

gender and all included covariates. For *age* (standardized, linear and squared), *education* (dummies for levels 1 and 2), *lives with spouse* (SPINHH) and the logarithm of *household income* we include only direct effects. Because Oesch and Lipps assume that well-being is different in the year before becoming unemployed, we add the dummy "$1(\text{TUE}_{it}{=}-1)$". For *regional unemployment* (UEREG) and *sectoral unemployment* (UESEC), we specify direct and interaction effects with *unemployment* (UE). Doing so integrates the hypothesis of Oesch and Lipps that suggests that "unemployment hurts less if there is more of it around". Although there remain differences between $\mathcal{M}_3$ and the model of Oesch and Lipps, the predictor functions of the two models are fairly comparable.

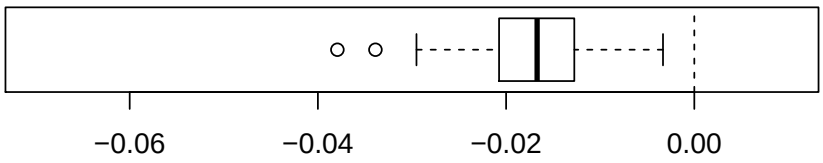

Figure 2.5: Boxplot for 250 pairwise differences $\text{err}(\widehat{M}^*_{1k}) - \text{err}(\widehat{M}^*_{3k})$ comparing the prediction error of fits for $\mathcal{M}_1$ and $\mathcal{M}_3$.

The 250 computed differences $\text{err}(\widehat{\mathcal{M}}^*_{1k}) - \text{err}(\widehat{\mathcal{M}}^*_{3k})$ shown in Figure 2.5 demonstrate that fits for $\mathcal{M}_1$ outperform fits for $\mathcal{M}_3$. The median of -0.017 is here lower than the median (-0.029) observed for the difference between $\mathcal{M}_1$ and $\mathcal{M}_2$. Moreover, when measuring the complexity of $\mathcal{M}_1$ by the median of the number of coefficients plus the number of splits, and that for $\mathcal{M}_3$ by the (constant) number of coefficients, $\mathcal{M}_1$ is with 25 vs 29 also less complex. The superiority of $\mathcal{M}_1$ over $\mathcal{M}_3$ can be explained as follows: first, the (subjective) *financial situation*, which is a good predictor (cf. Figure 2.2), is not included in $\mathcal{M}_3$. Second, our algorithm can benefit from technical differences, e.g., it incorporates the direct effects of moderators via the logit-specific varying intercepts rather than via the proportional odds effects. When incorporating the *financial situation* variable into $\mathcal{M}_3$ as a predictor with logit-specific effects, the median difference changes to 0.007 in favor of $\mathcal{M}_3$, the latter being however much more complex with a total of 41 coefficients. Anyway, the comparison made here does in no way invalidate hypothesis-driven model building, it just demonstrates that our algorithm is able to select relevant variables and builds parsimonious, understandable and competitive models.

Comparison with MOB Since our algorithm is a redesign of MOB (Zeileis et al., 2008), it is interesting to compare the results of the two algorithms. In Appendix B.3 we therefore use MOB to fit model $\mathcal{M}_1$, without random intercepts. The resulting tree structure is similar to that of Figure 2.2 from Algorithm 1. Specifically, the tree structure from MOB also includes as terminal nodes the nodes 3, 4 and 9, but node 6 is partitioned into seven terminal nodes instead of into two.

2.3.2 Simulation studies

The following simulation studies focus exclusively on the implemented coefficient constancy tests. Because the remaining parts of the algorithm, including the likelihood-based exhaustive search, do not fundamentally differ from other tree-based algorithms such as MOB, they are not studied here. The most important conclusions from the simulation studies are as follows:

- Under coefficient constancy, the implemented tests achieve fairly accurate type I errors. Specifically, the type I errors obtained with pre-decorrelation are more accurate than those without. This finding indicates that the variable selection process of the algorithm is approximately unbiased.
- As expected, the power of the implemented tests increases with increasing moderation strengths and number of observations. The imputation for unbalanced data slightly deteriorates the power of the tests.
- The power for variable selection of the implemented tests seems to be lower than that of the (slower) likelihood-based grid search approach. By contrast, they are more powerful than the M-fluctuation tests for a model that ignores intra-individual correlation.

The examined scenarios consider the coefficient constancy tests for six moderators, namely $Z_1, \ldots, Z_6$, that can be distinguished by their degree of intra-individual correlation (uncorrelated vs correlated vs time-invariant) and their scale (continuous vs categorical). Each scenario was repeated 2,000 times. As explained in Section 2.2.3.1, the testing procedure is based on the "supLM" statistic of Andrews (1993) for continuous moderators and on the χ^2-type statistic of Hjort and Koning (2002) for categorical moderators.

Generating the simulation data First, the values of the six moderators are generated by $z_{lit} = g_l(\tilde{z}_{1i} + \tilde{z}_{2it})$, where $\tilde{Z}_{1i} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_1)$ and $\tilde{Z}_{2it} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_2)$. For Z_1, Z_3, and Z_5, g_l is the identity function, while for Z_2, Z_4, and Z_6, g_l divides the values into four nominal categories $\{A, B, C, D\}$ based on their sample quartiles. For Z_1 and Z_2, we use $\sigma_1 = 0,\ \sigma_2 = 1$, (time-varying, uncorrelated); for Z_3 and Z_4, we use $\sigma_1 = 1,\ \sigma_2 = 1/2$ (time-varying, correlated); and for Z_5 and Z_6, we use $\sigma_1 = 1,\ \sigma_2 = 0$ (time-invariant). Second, the values x_{it} are generated by $X_{it} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$. Finally, the generated predictor and moderators are used to draw responses y_{it} with values in $\{1, 2, 3\}$ by using the model $\mathcal{M}_{\text{sim}}$

$$\mathcal{M}_{\text{sim}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}[\delta \cdot 1_{(z_{lit} \in \mathcal{B}_l)}] + b_i, \quad b_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1) \ ,$$

with $\beta_1 = -1$ and $\beta_2 = 1$. Model $\mathcal{M}_{\text{sim}}$ states that the coefficient of x_{it} is an indicator function with an amplitude δ for one of the six moderators. The node $\mathcal{B}_l$ is defined as $\mathcal{B}_l = \mathbb{R}^+$ for the continuous moderators Z_1, Z_3, and Z_5 and as $\mathcal{B}_l = \{C, D\}$ for the nominal moderators Z_2, Z_4, and Z_6.

2.3.2.1 Type I errors

Root node tests First, we set $\delta = 0$ (no moderation) and use $\mathcal{M}_{\text{root}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}\delta + b_i$ as the model under H_0. Table 2.2 reports the resulting type I errors for a nominal level of 5%, for different numbers of individuals and observations per individual, with and without pre-decorrelation.[5]

The test errors based on these decorrelated scores are close to the theoretical 5%, particularly for large N's. By comparison, the errors of the naive tests based on the raw scores (values in brackets) are systematically too small for moderators that have high intra-individual correlation.

[5] Q-Q plots of the p-values are shown in Appendix B.5. Results of the analogous simulation study for models with random slopes and unbalanced data are given in the Appendices B.4.2 and B.4.3.

Table 2.2: Relative frequencies of Type I errors in coefficient constancy tests for a nominal level of 5%. Values in brackets correspond to tests without pre-decorrelating the scores. Abbreviations: ii-cor = intra-individual correlation; cont = continuous, cat = categorical.

N/N_i	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
ii-cor	0	0	$\sim 2/3$	$\sim 2/3$	1	1
Scale	cont	cat	cont	cat	cont	cat
50/5	.042 (.042)	.051 (.054)	.038 (.027)	.044 (.038)	.036 (.018)	.039 (.020)
50/10	.050 (.050)	.042 (.041)	.042 (.024)	.044 (.030)	.036 (.014)	.040 (.010)
100/5	.039 (.039)	.056 (.054)	.046 (.032)	.053 (.038)	.039 (.018)	.050 (.020)
100/10	.050 (.047)	.050 (.048)	.054 (.024)	.047 (.027)	.046 (.018)	.041 (.010)
500/5	.054 (.054)	.044 (.044)	.057 (.036)	.050 (.036)	.054 (.024)	.056 (.018)
500/10	.053 (.052)	.042 (.044)	.052 (.030)	.045 (.028)	.061 (.023)	.052 (.012)

Nodewise tests Now, we set $\delta = 1$ and test the influencing variable Z_l within node $\mathcal{B}_l$, by using $\mathcal{M}_{\text{sim}}$ as the model under H_0. The simulation is performed for $Z_1, \ldots, Z_6$ as the influencing variable in $\mathcal{M}_{\text{sim}}$. The tests should accept H_0 because the coefficient of x is constantly $\delta = 1$ within $\mathcal{B}_l$.

Table 2.3: Type I errors for the nodewise coefficient constancy tests for a nominal level of 5%. Values within brackets correspond to the tests without pre-decorrelating the scores.

N/N_i	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
50/5	.038 (.040)	.048 (.047)	.036 (.033)	.041 (.038)	.020 (.023)	.042 (.036)
100/5	.050 (.049)	.048 (.048)	.045 (.038)	.038 (.037)	.044 (.028)	.040 (.029)
500/5	.040 (.040)	.050 (.048)	.054 (.042)	.049 (.042)	.052 (.028)	.063 (.036)

Table 2.3 shows the observed type I errors for a nominal level of 5%, for varying N's and a fixed $N_i = 5 \; \forall \; i$. The results are similar to those in Table 2.2, confirming that nodewise testing works. The effect of the small N's is more pronounced than that in Table 2.2 because the nodes $\mathcal{B}_l$ enclose only about half the data.

2.3.2.2 Power and comparisons

To evaluate the power of our test implementation, we generate data from $\mathcal{M}_{\text{sim}}$ for varying moderation strengths $\delta = \{0, 0.1, \ldots, 0.5\}$. All tests use $\mathcal{M}_{\text{root}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}\delta + b_i$ as the model under H_0.

Power for balanced and unbalanced data First, we use Z_3 (correlated, continuous) as the influencing variable to generate the data and as a moderator in the tests. The power is evaluated for scenarios (a) and (b). (a) uses balanced data where $N_i = 5 \; \forall \; i$ and N varies between (a.1) $N = 50$, (a.2) $N = 100$, and (a.3) $N = 150$. (b) uses unbalanced data where (b.1) $N = 140$ with $N_i = 5$ for individuals $i = 1, \ldots, 40$ and $N_i = 3$ for individuals $i = 41, \ldots, 140$, and (b.2) $N = 242$ with $N_i = 10$ for individuals $i = 1, 2$ and $N_i = 2$ for individuals $i = 3, \ldots, 242$. Therewidth, the imputation will increase the number of observations in (b.1) from 500 to 700 and in (b.2) from 500 to 2420 observations. For both (b.1) and (b.2), a single imputation is used to adjust the pre-decorrelation.
Figure 2.6 shows the moderation strength δ against the relative frequency of p-values below 0.05. As expected, the power of the tests increases as δ increases and as the

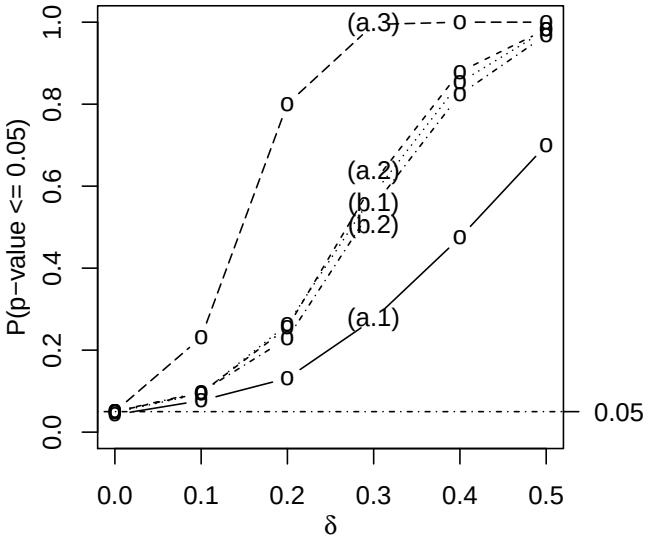

Figure 2.6: Power of tests on Z_3 for increasing moderation strengths δ. The figure shows the relative frequencies for p-values below 0.05 for scenarios (a) and (b); (a) uses balanced data where $N_i = 5$ and (a.1) $N = 50$, (a.2) $N = 100$, and (a.3) and $N = 150$; (b) uses unbalanced data, where in (b.1) N_i is 3 or 5 and $N = 140$ and in (b.2) N_1 is 2 or 10 and $N = 242$.

number of individuals N increases. The unbalanced scenarios (b.1) and (b.2) can be compared with (a.2), which also includes 500 observations. It can be seen that the power slightly decreases with increasing numbers of imputed observations. Considering that in scenario (b.2) the imputation enlarges the score matrix from 500 to 2420 entries, the loss of power is surprisingly low. A plausible explanation for this is that the imputed scores are dropped after the pre-decorrelation transformation and therefore their impact is limited. Nevertheless, in practice, the data may manually be balanced out to avoid the power of the tests to be deterioted by the imputation.

Variable selection In this last scenario, we use our tests to select between the moderators Z_2, Z_4, and Z_6, where (alternately) one of these moderates β_3. For the comparison, we also use the likelihood-based exhaustive search and M-fluctuation tests with the cumulative logit model without random coefficients to select $\mathcal{M}_{\text{clm}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}\delta$ under H_0. In all scenarios, $N = 100$ and $N_i = 5 \; \forall \; i$.
Figure 2.7 shows the frequencies of selecting the true moderator. Both selection schemes based on CLMMs are unbiased. The exhaustive search is unbiased because all three moderators have the same number of splits. This selection method performs best, followed by our test implementation. The tests based on the model $\mathcal{M}_{\text{clm}}$ without random coefficients have lower power and they are biased towards the intra-individually correlated moderator Z_6.

2.4 Conclusion

The present study proposed a new tree-based algorithm for learning moderated relations in longitudinal (ordinal) regression analysis, by building on MGLMMs and the MOB algorithm of Zeileis et al. (2008). The main innovations relative to MOB are (i) similar

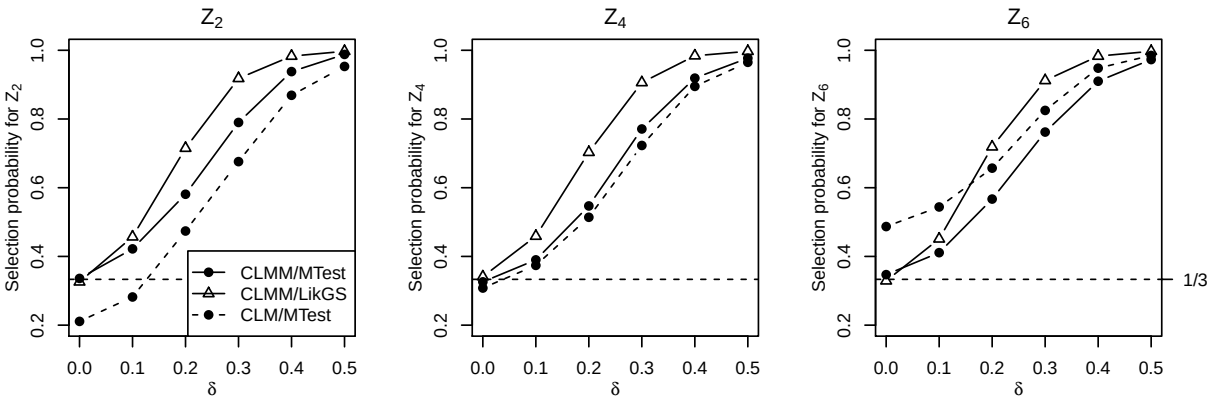

Figure 2.7: Relative frequencies for selecting the true moderator among Z_2, Z_4, and Z_6 for varying moderation strengths δ. This selection is based on: *solid line, circle*, our test implementation; *solid line, triangle*, exhaustive search; and *dotted line, circle*, M-fluctuation tests with a model without random effects.

to the approaches of Hajjem et al. (2011) and Sela and Simonoff (2012), the proposed algorithm can maintain random coefficients across nodes, meaning that observations of the same individual falling into different nodes are not treated as independent, and (ii) the coefficient constancy tests used to select the moderators and tree size are extended for testing based on observation scores. In addition, our algorithm extends the scope of longitudinal regression trees based on mixed models, which include the algorithms of Hajjem et al. (2011) and Sela and Simonoff (2012), to general longitudinal varying coefficient regression. As exemplified by examining the varying effect of unemployment, the resulting models are simple to read and therefore easily accessible to practitioners.

Although this study focused on CLMMs, the algorithm can be implemented more or less straightforwardly for other models of the MGLMM family. Further research could be directed towards improving the numerically challenging components of the algorithm. For example, alternative ways to direct marginal maximum likelihood estimation combined with Gauss-Hermite quadrature could be considered (e.g. Tutz, 2012, Chap. 14.3). Moreover, optimization by using Newton's method for the pre-decorrelation matrix has a tendency to fail for high dimensions and therefore this could be improved. Finally, the statistical power of the coefficient constancy tests could be enhanced by deriving the distribution of the partial sum processes of the raw rather than the pre-decorrelated scores. At present, we investigate the extension of building for each varying coefficient an individual tree. Such an extension allows to deduce which variable moderates which coefficient from the fitted tree structures, instead of from comparing the nodewise coefficients.

To overcome the instability and inaccuracy problems of tree-based algorithms (cf. Section 2.1), we may consider ensemble techniques such as boosting (Freund, 1995) or random forests (Breiman, 2001). Appendix B.2 discusses and evaluates an implementation of random forest, which is available with the `fvcolmm` function of the R **vcrpart** package. In particular, it is shown that this extension improves the predictive performace of our algorithm for the happiness data of Section 2.3.1. A disadvantage of random forest is the complexity of the results. While coefficient functions from tree-based algorithms are fully traceable and easily readable by means of the decision tree representation, those resulting from random forest can often only be approximately understood, e.g., with the help of partial dependency plots (cf. Hastie et al., 2001, Chap. 10) or variable importance measures (e.g. Breiman, 2001; Strobl et al., 2008).

The tree-based algorithm proposed in this article was implemented in the R (R Core Team, 2014) package **vcrpart**. The function `tvcolmm` fits tree-based varying coefficient CLMMs, with the presented methodology and corresponding methods, such as `plot` or `predict`, thereby allowing the diagnosis of the fitted model.

Bibliography

Abdolell, M., M. LeBlanc, D. Stephens, and R. V. Harrison (2002). Binary Partitioning for Continuous Longitudinal Data: Categorizing a Prognostic Variable. *Statistics in Medicine 21*(22), 3395–3409.

Alexander, W. P., S. D. Grimshaw, and P. William (1996). Treed Regression. *Journal of Computational and Graphical Statistics 5*(2), 156–175.

Andrews, D. W. K. (1993). Tests for Parameter Instability and Structural Change with Unknown Change Point. *Econometrica 61*(4), 821–56.

Andrews, D. W. K. and J. C. Monahan (1992). An Improved Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimator. *Econometrica 60*(4), 953–966.

Breiman, L. (2001). Random Forests. *Machine Learning 45*(1), 5–32.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. New York, USA: Wadsworth.

Bürgin, R. (2015). *vcrpart: Tree-Based Varying Coefficient Regression for Generalized Linear and Ordinal Mixed Models*. R package version 0.3-3, URL http://cran.r-project.org/web/packages/vcrpart/.

Chan, F., L. L. Pauwels, and J. Wongsosaputro (2013). The Impact of serial Correlation on Testing for Structural Change in Binary Choice Model: Monte Carlo Evidence. *Mathematics and Computers in Simulation 93*, 175—189.

Eo, S.-H. and H. J. Cho (2014). Tree-Structured Mixed-Effects Regression Modeling for Longitudinal Data. *Journal of Computational and Graphical Statistics 23*(3), 740–760.

Fahrmeir, L. and G. Tutz (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Field, C. A. and A. H. Welsh (2007). Bootstrapping Clustered Data. *Journal of the Royal Statistical Society B 69*(3), 369–390.

Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and computation 121*(2), 1–50.

Hajjem, A. (2010). *Mixed Effect Trees and Forests for Clustered Data*. Ph. D. thesis, HEC Montréal.

Hajjem, A., F. Bellavance, and D. Larocque (2011). Mixed Effects Regression Trees for Clustered Data. *Statistics & Probability Letters 81*(4), 451–459.

Hastie, T. and R. Tibshirani (1993). Varying-Coefficient Models. *Journal of the Royal Statistical Society B 55*(4), 757–796.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Hedeker, D. and R. D. Gibbons (1994). A Random-Effects Ordinal Regression Model for Multilevel Analysis. *Biometrics 50*(4), 933–944.

Hjort, N. L. and A. Koning (2002). Tests for Constancy of Model Parameters Over Time. *Journal of Nonparametric Statistics 14*(1-2), 113–132.

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

Kauermann, G. (2000). Modeling Longitudinal Data with Ordinal Response by Varying Coefficients. *Biometrics 56*(3), 692–698.

Kosmidis, I. (2014). Improved Estimation in Cumulative Link Models. *Journal of the Royal Statistical Society B 76*(1), 169–196.

Liang, K.-Y. and S. Zeger (1986). Longitudinal Data Analysis Using Generalized Linear Models. *Biometrika 73*(1), 13–22.

Loh, W.-Y. (2002). Regression Trees With Unbiased Variable Selection and Interaction Detection. *Statistica Sinica 12*(2), 361–386.

McCullagh, P. (1980). Regression Models for Ordinal Data. *Journal of the Royal Statistical Society B 42*(2), 109–142.

Nyblom, J. (1989). Testing for the Constancy of Parameters Over Time. *Journal of the American Statistical Association 84*(405), 223–230.

Oesch, D. and O. Lipps (2013). Does Unemployment Hurt Less if There is More of it Around? A Panel Analysis of Life Satisfaction in Germany and Switzerland. *European Sociological Review 29*(5), 955–967.

Quinlan, J. R. (1992). Learning with Continuous Classes. In *5th Australian Joint Conference on Artificial Intelligence*, Singapore, pp. 343–348. World Scientific.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rusch, T. and A. Zeileis (2012). Gaining Insight with Recursive Partitioning of Generalized Linear Models. *Journal of Statistical Computation and Simulation 83*(7), 1–15.

Russell, S. J. and P. Norvig (2003). *Artificial Intelligence: A Modern Approach* (3 ed.). New Jersey, USA: Pearson Education Inc.

Sela, R. and J. S. Simonoff (2012). RE-EM trees: A Data Mining Approach for Longitudinal and Clustered Data. *Machine Learning 86*(2), 169–207.

Siddall, P. J., J. M. McClelland, S. B. Rutkowski, and M. J. Cousins (2003). A Longitudinal Study of the Prevalence and Characteristics of Pain in the First 5 Years Following Spinal Cord Injury. *Pain 103*(3), 249–257.

Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). Conditional Variable Importance for Random Forests. *BMC Bioinformatics 9*(307), 1471–2105.

Strobl, C., J. Kopf, and A. Zeileis (2013). Rasch Trees: A New Method for Detecting Differential Item Functioning in the Rasch Model. *Psychometrika*, 1–28. Forthcoming.

Su, X., K. Meneses, P. McNees, and W. Johnson (2011). Interaction Trees: Exploring the Differential Effects of an Intervention Programme for Breast Cancer Survivors. *Journal of the Royal Statistical Society C 60*(3), 457–474.

Taylor, M. F., N. B. John Brice, and E. Prentice-Lane (2010). *British Household Panel Survey User Manual Volume A: Introduction, Technical Report and Appendices*. Colchester, UK: University of Essex.

Tutz, G. (2012). *Regression for Categorical Data*. Cambridge Series in Statistical and Probabilistic Mathematics. New York, USA: Cambridge Series in Statistical and Probablilistic Mathematics.

Tutz, G. and W. Hennevogl (1996). Random Effects in Ordinal Regression Models. *Computational Statistics & Data Analysis 22*(5), 537–557.

Tutz, G. and G. Kauermann (2003). Generalized Linear Random Effects Models with Varying Coefficients. *Computational Statistics & Data Analysis 43*(1), 13–28.

Wang, J. C. and T. Hastie (2014). Boosted Varying-Coefficient Regression Models for Product Demand Prediction. *Journal of Computational and Graphical Statistics 23*(2), 361–382.

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

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

Zhang, D. (2004). Generalized Linear Mixed Models with Varying Coefficients for Longitudinal Data. *Biometrics 60*(1), 8–15.

Chapter 3

Coefficient-wise tree-based varying coefficient regression with vcrpart

Abstract The tree-based TVCM algorithm and its implementation in the R package **vcrpart** is introduced for generalized linear models. The purpose of TVCM is to learn whether and how the coefficients of a regression model vary by moderating variables. A separate partition is built for each varying coefficient, allowing moderators to be selected individually by coefficient and coefficient-specific sets of moderators to be specified. In addition to describing the algorithm, the TVCM is evaluated using a benchmark comparison and the R commands are demonstrated by means of empirical applications.[1]

3.1 Introduction

When carrying out a regression analysis, researchers often wish to know whether and how moderating variables affect the coefficients of predictor variables. For example, medical scientists may be interested in how age or past illnesses moderate the effect of a clinical trial (e.g. Yusuf et al., 1991), and social scientists may examine the wage gap between genders separately for different labor sectors and countries (e.g. Arulampalam et al., 2007).

Varying coefficient models (e.g. Hastie and Tibshirani, 1993) provide a semi-parametric approach for such moderated relations. Consider a response variable Y, where $g(E(Y|\cdot)) = \eta$, with g a known link function and η a predictor function of form:

$$\mathcal{M}_{\text{vc}} : \eta = X_1\beta_1(\mathbf{Z}_1) + \ldots + X_P\beta_P(\mathbf{Z}_P) \ , \tag{3.1}$$

where X_p, $p = 1, \ldots, P$, are predictor variables and $\mathbf{Z}_p$ are the corresponding $L_p \times 1$ vectors of moderator variables, sometimes called *effect modifiers*. Model $\mathcal{M}_{\text{vc}}$ defines the coefficients $\beta_1, \ldots, \beta_P$ as multivariate, nonparameterized functions of the associated moderators. For example, if X_p is an indicator for some treatment and Z_p indicates age, the term $\beta_p(Z_p)$ states that the treatment effect changes as a function of age. In principle, the moderator vectors $\mathbf{Z}_1, \ldots, \mathbf{Z}_P$, can be intersected or can include some of $X_1, \ldots, X_P$. Model $\mathcal{M}_{\text{vc}}$ also covers two simplifications. First, defining $\mathbf{Z}_p \equiv 1$ yields a *non-varying* coefficient for predictor X_p. Second, terms with $X_p \equiv 1$ provide a nonparametric estimate of the direct effects of $\mathbf{Z}_p$ on $E(Y|\cdot)$, henceforth referred to as *varying intercept*.

Various approaches have been considered to fit varying coefficient models, in particular with spline or kernel regression methods. See Fan and Zhang (2008) for an overview and

[1] A supplementary simulation study, details on approximate models, descriptive statistics of the used data sets and R-codes are available in Appendix C.

the R (R Core Team, 2014) packages **mgcv** (Wood, 2006), **svcm** (Heim, 2007), **mboost** (Hothorn et al., 2015), and **np** (Hayfield and Racine, 2008) for software implementations. The tree-based approach considered here is a combination of linear models and recursive partitioning (e.g. Quinlan, 1992; Alexander et al., 1996; Loh, 2002), where Zeileis et al. (2008) and Wang and Hastie (2014) refer explicitly to the use of recursive partitioning to fit models of the form $\mathcal{M}_{\text{vc}}$ (Eq. 3.1). Thus, it approximates the unknown varying coefficients with piecewise constant functions using recursive partitioning. The tree-based approach has certain drawbacks, particularly being a heuristic, and can be unstable for small changes in the data. However, it does have several advantages for statistical learning. Among others, the approach can handle many moderators, interactions between moderators, nonlinearities, treats moderators of different scales uniformly, and yields easily readable outcomes in the form of decision trees.

Both Zeileis et al. (2008) and Wang and Hastie (2014) propose approximating $\mathcal{M}_{\text{vc}}$ as follows: let $\mathbf{X} = (X_1, \ldots, X_P)^\top$, $\mathbf{Z} = \{\mathbf{Z}_1 \cup \ldots \cup \mathbf{Z}_P\}$, and $\{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$ be a partition of the value space of the $\mathbf{Z}$ into M *strata*. Then, their piecewise constant approximation has the form

$$\widehat{\mathcal{M}}_{\text{tree}} : \eta = \sum_{m=1}^{M} 1\left(\mathbf{Z} \in \mathcal{B}_m\right) \mathbf{X}^\top \boldsymbol{\beta}_m \ . \tag{3.2}$$

Model $\widehat{\mathcal{M}}_{\text{tree}}$ (Eq. 3.2) is linear and, consequently, standard estimation methods apply. The nonparametric task is to find a partition such that the varying coefficients $\beta_1(\mathbf{Z}), \ldots, \beta_P(\mathbf{Z})$ vary between the strata $\{\mathcal{B}_1, \ldots, \mathcal{B}_M\}$, but are relatively constant within the strata. Since global partitioning is computationally too challenging, forward-stepwise algorithms are used that, in each iteration, split one of the current strata into two. The resulting partition can be visualized as a decision tree and, therefore, the strata $\mathcal{B}_m$ are referred to as terminal nodes, or simply to as nodes.

Here, we introduce the *tree-based varying coefficient model* (TVCM) algorithm of the R package **vcrpart** (Bürgin, 2015). The TVCM algorithm allows us to approximate $\mathcal{M}_{\text{vc}}$ in a *coefficient-wise* manner. First, we let $\mathbf{X}_0$ be the vector of the $P - K$ predictors that correspond to moderators $\mathbf{Z}_p \equiv 1$, and $X_1, \ldots, X_k$ denote the remaining predictors with corresponding moderator vectors $\mathbf{Z}_1, \ldots, \mathbf{Z}_K$. Further, denote the value space of $\mathbf{Z}_k$ as $\mathcal{Z}_k = \mathcal{Z}_{k1} \times \ldots \times \mathcal{Z}_{kL_k}$ and denote a partition of $\mathcal{Z}_k$ into M_k nodes as $\{\mathcal{B}_{k1}, \ldots, \mathcal{B}_{kM_k}\}$. Then, the proposed approximation is:

$$\widehat{\mathcal{M}}_{\text{tvcm}} : \eta = \mathbf{X}_0^\top \boldsymbol{\beta}_0 + \sum_{k=1}^{K} \sum_{m=1}^{M_k} 1\left(\mathbf{Z}_k \in \mathcal{B}_{km}\right) X_k \beta_{km} \ . \tag{3.3}$$

Compared with $\widehat{\mathcal{M}}_{\text{tree}}$, the TVCM approximation $\widehat{\mathcal{M}}_{\text{tvcm}}$ assigns each varying coefficient a partition and includes non-varying coefficients. This allows us to specify parametrically known relations (the first term) and coefficient-specific sets of moderators (the second term). In addition, $\widehat{\mathcal{M}}_{\text{tvcm}}$ allows us to select moderators individually by varying coefficient. Furthermore, empirical evidence suggests (Sec. 3.4.1) that $\widehat{\mathcal{M}}_{\text{tvcm}}$ can build more accurate and more parsimonious fits than $\widehat{\mathcal{M}}_{\text{tree}}$ is able to do. A technical difference between the two approximations $\widehat{\mathcal{M}}_{\text{tree}}$ and $\widehat{\mathcal{M}}_{\text{tvcm}}$ is that the coefficients of $\widehat{\mathcal{M}}_{\text{tree}}$ are commonly estimated by means of M unconnected models, while the approximation $\widehat{\mathcal{M}}_{\text{tvcm}}$ must be fitted as a closed model.

The remainder of this paper is organized as follows. In Section 3.2, we describe the basic algorithm that we apply to generalized linear models. In Section 3.3, we provide more detail and extend the basic algorithm. Then, in Section 3.4, we present three applications, including a performance comparison with competing algorithms. Finally, Section 3.5 concludes the paper, and includes a discussion on issues for the further development.

3.2 The TVCM algorithm

Similar to *classification and regression trees* (CART, Breiman et al., 1984), TVCM involves two stages: the first stage (Sec. 3.2.2) builds K overly fine partitions; the second stage (Sec. 3.2.3) selects the final partitions by pruning.

To provide a consistent formulation, we restrict our consideration of TVCM to generalized linear models (GLMs). Therefore, Section 3.2.1 summarizes GLMs and introduces an illustrative example. Extensions to other model families are discussed in Section 3.3.3.

3.2.1 Generalized linear models

GLMs cover regression models for various types of responses, such as continuous data (the Gaussian model), count data (the Poisson model), and binary data (the logistic model). Denote the ith response of the training data $\mathcal{D}$ as y_i, with observations $i = 1, \ldots, N$, and the ith $P\times 1$ predictor vector as $\mathbf{x}_i$. Simple GLMs have densities of the form

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right\} \quad , \tag{3.4}$$

where θ_i is the natural parameter of the family, ϕ is the dispersion parameter, and $b(\cdot)$ and $c(\cdot)$ are family-specific functions. For example, the Poisson distribution has density $f(y_i) = \lambda_i^{y_i} e^{-\lambda_i}/k!$ and it can be derived that $\theta_i = \log\lambda_i$, $b(\theta_i) = e^{\theta_i} = \lambda_i$, $\phi = 1$, and $c(y_i, \phi) = \log y_i$. The predictor vector $\mathbf{x}_i$ is incorporated by defining the linear predictor

$$\mathcal{M}_{\text{glm}} : \eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} \quad , \tag{3.5}$$

where $\boldsymbol{\beta}$ is the vector of unknown coefficients. This linear predictor η_i is linked with the conditional mean $\mu_i = E(y_i|\mathbf{x}_i)$ via $g(\mu_i) = \eta_i = \mathbf{x}_i^\top\boldsymbol{\beta}$. The choice of $g(\cdot)$ depends on the specific model. A mathematically motivated choice is to specify $g(\cdot)$, such that $\theta_i = \eta_i$, usually called *canonical link*. For example, for the Poisson model, the canonical is $\log(\mu_i) = \eta_i$. Further details on GLMs can be found, for instance, in McCullagh and Nelder (1989).

Generalized linear models are generally fitted using maximum likelihood estimation (MLE), in other words, by maximizing the total log-likelihood of the training data w.r.t. $\boldsymbol{\beta}$ and ϕ:

$$\ell(\boldsymbol{\beta}, \phi) = \sum_{i=1}^{N} w_i \log f(y_i|\boldsymbol{\beta}, \phi) = \sum_{i=1}^{N} w_i \left(\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi)\right) \quad , \tag{3.6}$$

where w_i is the weight for observation i. The coefficients $\boldsymbol{\beta}$ enter into (Eq. 3.6) via $\theta_i = d(\mu_i) = d(g^{-1}(\mathbf{x}_i^\top\boldsymbol{\beta}))$, with $d(\cdot)$ a known function. To fit GLMs, we use the `glm` function of the **stats** package (see Chambers and Hastie, 1992).

Gender gap in university admissions To illustrate R syntax and explanations, we consider the admission data of the UC Berkeley from 1973. The data consist of 4,526 observations on the response variable `Admit` (0 = rejected, 1 = admitted) and the covariates `Female` (0 = male, 1 = female) and `Dept` (departments A to F). The training data `UCBA` are prepared by

```
R> UCBA <- as.data.frame(UCBAdmissions)
R> UCBA$Admit <- 1 * (UCBA$Admit == "Admitted")
R> UCBA$Female <- 1 * (UCBA$Gender == "Female")
R> head(UCBA, 3)
```

```
  Admit Gender Dept Freq Female
1     1   Male    A  512      0
2     0   Male    A  313      0
3     1 Female    A   89      1
```

Each row of the data `UCBA` represents one combination of values in `Admit`, `Female`, and `Dept`. The column `Freq` gives the frequency for the combinations.[2]

The UCB admission data are a popular application to illustrate Simpson's paradox (see Bickel et al., 1975). The primary interest is the gender gap in the chance to be admitted. Let us first study this gap using the logistic regression model:

```
R> glmS.UCBA <- glm(formula = Admit ~ Female, data = UCBA,
+                   family = binomial(), weights = UCBA$Freq)
```

The estimated coefficients,

```
             Estimate Std. Error z value
(Intercept) -0.220134   0.038788 -5.6753
Female      -0.610352   0.063891 -9.5530
```

suggest that being female decreases (z value > 2) the logit to be admitted significantly. Now, let us extend the basis model `glmS.UCBA` with the `Dept` covariate by defining department-specific intercepts and department-specific gender gaps (without a global intercept):

```
R> glmL.UCBA <- glm(formula = Admit ~ -1 + Dept + Dept:Female,
+                   data = UCBA, family = binomial(),
+                   weights = UCBA$Freq)
```

```
              Estimate Std. Error  z value
DeptA         0.492121   0.071750   6.8589
DeptB         0.533749   0.087543   6.0970
DeptC        -0.535518   0.114941  -4.6591
DeptD        -0.703958   0.104070  -6.7643
DeptE        -0.956962   0.161599  -5.9218
DeptF        -2.769744   0.219781 -12.6023
DeptA:Female  1.052076   0.262708   4.0047
DeptB:Female  0.220023   0.437593   0.5028
```

[2]Descriptive statistics of this data set can be found in Table C.2.

```
DeptC:Female -0.124922   0.143942  -0.8679
DeptD:Female  0.081987   0.150208   0.5458
DeptE:Female -0.200187   0.200243  -0.9997
DeptF:Female  0.188896   0.305163   0.6190
```

In this second fit, the disadvantage for females disappears, and, in the case of department A, the gender gap is significantly positive (`DeptA:Female`: Estimate = 1.05, z value > 2). The apparent disadvantage for females in `glmS.UCBA` arises, as the reader may know, from the tendency of females to apply to departments where the chances to be admitted are low.

The model `glmL.UCBA`, which uncovers the problem, can be seen as a full parametric varying coefficient model that defines the intercept and the gender gap as functions of the department. We will return to this example to investigate whether and how TVCM solves this problem.

3.2.2 Partitioning

The first stage to fit the approximate varying coefficient model $\widehat{\mathcal{M}}_{\text{tvcm}}$ (Eq. 3.3) involves building a partition for each of the value spaces $\mathcal{Z}_k$, $k = 1, \ldots, K$ corresponding to the K varying coefficients. The resulting K partitions should be overly fine so that the best-sized partitions can be found in the subsequent pruning stage.

To partition the value spaces $\mathcal{Z}_1, \ldots, \mathcal{Z}_K$, TVCM uses a breadth-first search (e.g. Russell and Norvig, 2003) that in each iteration fits the current model and splits one of the current terminal nodes into two. Splitting requires four selections in each iteration: the partition k; the node m; the moderator variable l; and the cutpoint j in the selected moderator. Following CART, we employ an exhaustive search over all candidate splits and select the split that reduces the total $-2 \cdot$log-likelihood training error the most.[3] The algorithm iterates until (i) no candidate split provides daughter nodes with more than N_0 observations or (ii) the best split increases the $-2 \cdot$ log-likelihood training error by less than D_{min}. Algorithm 2 provides a more formal summary of the partitioning algorithm.

When searching for a split, there can be differences in the number of candidate splits between partitions, nodes, and moderators. The $-2 \cdot$ log-likelihood reduction statistic is not "standardized" to such differences and, therefore, Algorithm 2 tends to select partitions, nodes, and variables with many candidate splits (cf. Hothorn et al., 2006). As the main consequence, the order in which variables appear in the trees should be interpreted carefully. Reducing this bias is desirable and, therefore, is a potential focus for further investigations.

The `tvcglm` function The `tvcglm` function implements Algorithm 2. For illustration, we fit a logistic TVCM to the UCB admission data. The following command specifies that both the intercept and the gender gap vary across departments.

```
R> library("vcrpart")
R> vcmL.UCBA <-
+    tvcglm(formula = Admit ~ -1 + vc(Dept) + vc(Dept, by = Female),
+           data = UCBA, family = binomial(), weights = UCBA$Freq,
+           control = tvcglm_control(minsize = 30, mindev = 0.0, cv = FALSE))
```

[3] In other words, we maximize the likelihood-ratio statistic compared to the current model.

Algorithm 2: The TVCM partitioning algorithm for generalized linear models.

Parameters: N_0 minimum node size, e.g., $N_0 = 30$
D_{min} minimum $-2 \cdot$ log-likelihood reduction, e.g., $D_{min} = 2$

Initialize $\mathcal{B}_{k1} \leftarrow \mathcal{Z}_{k1} \times \ldots \times \mathcal{Z}_{kL_k}$ and $M_k \leftarrow 1$ for all $k = 1, \ldots, K$.

repeat

1 Compute $\hat{\ell}_{\widehat{\mathcal{M}}} = \max_{\{\boldsymbol{\beta}, \phi\}} \ell_{\widehat{\mathcal{M}}}(\boldsymbol{\beta}, \phi)$ of the current model

$$\widehat{\mathcal{M}} : \eta_i = \mathbf{x}_{i0}^\top \boldsymbol{\beta}_0 + \sum_{k=1}^{K} \sum_{m=1}^{M_k} 1\left(\mathbf{z}_{ik} \in \mathcal{B}_{km}\right) x_{ik} \beta_{km} \ . \tag{3.7}$$

for *partitions* $k = 1$ *to* K **do**
  for *nodes* $m = 1$ *to* M_k *and moderator variables* $l = 1$ *to* L_k **do**
    foreach *unique candidate split* $\triangle_{kmlj}$ *in* $\{z_{kli} : \mathbf{z}_{ik} \in \mathcal{B}_{km}\}$ *that divides* $\mathcal{B}_{km}$ *into two nodes* $\mathcal{B}_{kmlj1}$ *and* $\mathcal{B}_{kmlj1}$ *with* $\min_s \sum_i w_i 1(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}) \geq N_0$ **do**

2 Compute $\hat{\ell}_{\widehat{\mathcal{M}}_{kmlj}} = \max_{\{\beta_1, \beta_2, \phi\}} \ell_{\widehat{\mathcal{M}}_{kmlj}}(\beta_1, \beta_2, \phi)$ of the search model

$$\widehat{\mathcal{M}}_{kmlj} : \eta_i^{(s)} = \hat{\eta}_i + \sum_{s=1}^{2} 1\left(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}\right) x_{ik} \beta_s \ , \tag{3.8}$$

and compute the training error reduction $D_{kmlj} = -2\hat{\ell}_{\widehat{\mathcal{M}}} + 2\hat{\ell}_{\widehat{\mathcal{M}}_{kmlj}}$.

    end
  end
end

3 Split node $\mathcal{B}_{k'm'}$ by $\triangle_{k'm'l'j'}$ where $D_{k'm'l'j'} = \max D_{kmlj}$ and increase $M_{k'} \leftarrow M_{k'} + 1$.

until *no candidate split satisfies* N_0 *or* $D_{k'm'l'j'} < D_{min}$

The syntax for `tvcglm` is quite similar to that of `glm`. The **v**arying **c**oefficient terms "`vc`" in the model formula are new. The `vc` terms treat unnamed arguments as moderators and the `by` argument specifies the predictor. Correspondingly, `vc` terms without a `by` argument are interpreted as varying intercepts. The predictors assigned to `by` must be numeric in the current implementation. This is why we have defined (pg. 54) the `Female` variable for the `UCBA` data as `UCBA$Female <- 1 * (UCBA$Gender == "Female")`. The control parameters are set by the `tvcglm_control()` function. Here, `minsize = 30` specifies $N_0 = 30$ and `mindev = 0` specifies $D_{min} = 0$. We set $D_{min} = 0$ to obtain the largest possible tree and `cv = FALSE` to disable cross-validation (Sec. 3.2.3).

The two fitted partitions are shown in Figure 3.1, along with the nodewise coefficients and the corresponding 95% confidence intervals. These plots were produced by the following commands:

```
R> plot(vcmL.UCBA, type = "coef", part = "A")
R> plot(vcmL.UCBA, type = "coef", part = "B")
```

The shown confidence intervals are extracted from the underlying `glm` object and do not account for the model selection procedure. Both partitions separate the departments

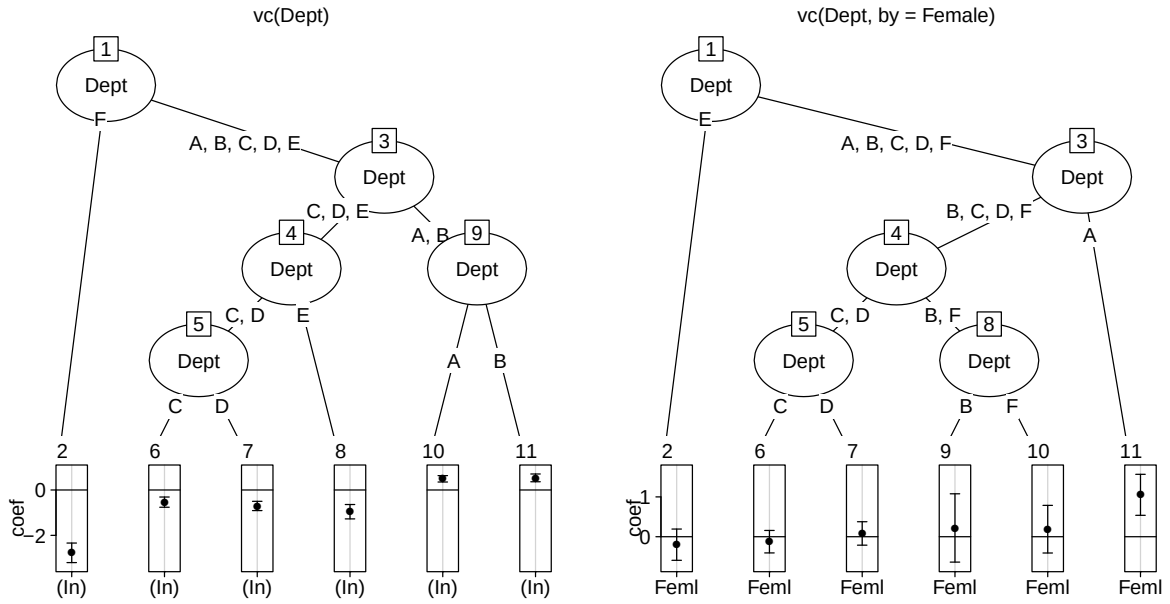

Figure 3.1: `vcmL.UCBA`: fitted tree structures, nodewise coefficients and corresponding 95% confidence intervals. Left panel, the varying intercept; right panel, the varying gender gap.

fully and, therefore, the fits (fitted models) for `vcmL.UCBA` and `glmL.UCBA` of page 54 are equivalent. The partitioning process can be backtracked using the `splitpath` function. The following command summarizes the first iteration.

```
R> splitpath(vcmL.UCBA, steps = 1, details = TRUE)

Step: 1

Selected Split:
Partition: A
Node: 1
Variable: Dept
Cutpoint: {F}, {A, B, C, D, E}

Loss Reduction Statistics:
Partition: A Node: 1 Variable: Dept
  A B C D E F      dev npar
1 0 0 0 0 0 1 443.5131    1
2 0 0 0 0 1 1 409.2954    1
3 0 0 0 1 1 1 384.1886    1
4 0 0 1 1 1 1 416.2448    1
5 0 1 1 1 1 1 226.4871    1

Partition: B Node: 1 Variable: Dept
  A B C D E F       dev npar
1 0 0 0 0 0 1 130.59742    1
2 0 0 0 0 1 1 124.65130    1
3 0 0 1 0 1 1  76.78492    1
```

```
4 0 0 1 1 1 1  99.26478    1
5 0 1 1 1 1 1 120.99035    1
```

Based on the training error reduction statistic D_{kmlj} (column `dev`), the algorithm selects the split $\{F\}$ vs. $\{A, B, C, D, E\}$ for the varying intercept (partition A). The evaluated splits, listed in the lower part, show that only a subset of possible splits was evaluated. For example, the split $\{A, F\}$ vs. $\{B, C, D\}$ was excluded from the search. This relates to the implemented acceleration technique that orders the six categories A to F and treats the `Dept` as ordinal (details follow).

3.2.2.1 Computational details

A breadth-first search can be computationally burdensome because it cycles in each iteration through all current nodes. Even so, we do not consider a depth-first search, which is more common for recursive partitioning and which evaluates only one node in each iteration, because it seems unclear whether the search sequence has consequences on the resulting partitions. To speed up the search, we use the approximate search model $\widehat{\mathcal{M}}_{kmlj}$ (Eq. 3.8) to compute the training error reduction of split $\triangle_{kmlj}$, instead of using the following accurate search model

$$\widehat{\mathcal{M}}^*_{kmlj} : \eta_i^{(s)} = \mathbf{x}_{i0}^\top \boldsymbol{\gamma}_0 + \sum_{(k',m')\neq(k,m)} 1\left(\mathbf{z}_{k'i} \in \mathcal{B}_{k'm'}\right) x_{k'i}\gamma_{k'm'} + \sum_{s=1,2} 1\left(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}\right) x_{ik}\gamma_s \ . \tag{3.9}$$

In particular, the approximate search model $\widehat{\mathcal{M}}_{kmlj}$ incorporates as offsets the fitted values $\hat{\eta}_i$ of the current model $\widehat{\mathcal{M}}$ (Eq. 3.7). Therewith, as derived in Appendix C.2.1, $\widehat{\mathcal{M}}_{kmlj}$ estimates the coefficients γ_s, $s = 1, 2$ of $\widehat{\mathcal{M}}^*_{kmlj}$ by $\hat{\gamma}_s = \hat{\beta}_{km} + \hat{\beta}_s$. The approximation reduces the optimization to the three unknown parameters β_1, β_2 and ϕ. In our experience, the approximation is reliable, although it does not necessarily result in the same partitions that the accurate search would produce. More specifically, the approximation will tend to neglect splits that improve the fit through interplays with the temporarily fixed coefficients.

Eliminating split candidates and cleverly choosing the stopping parameters are further efficient acceleration techniques. We describe these techniques in more detail here.

Splits for ordered scales In Algorithm 2, the splits $\triangle_{kmlj}$ for continuous and ordinal moderators are defined as rules of the form $\{\text{Is } z_{kli} \leq \zeta_{kmlj}?\}$. The candidate cutpoints, $\{\zeta_{kml1}, \ldots\}$, are the unique values in set $\{z_{kli} : \mathbf{z}_{ik} \in \mathcal{B}_{km}\}$. Note that splits at boundaries may be omitted to respect the minimum node size criterion. To reduce the computational burden, we allow the set of candidate cutpoints to shrink to a prespecified cardinality N_S, which is $N_S = 9$ by default.[4] Specifically, the unique values of the (non-interpolated) quantiles of $\{z_{kli} : \mathbf{z}_{ik} \in \mathcal{B}_{km}\}$ are extracted at the N_S equidistant probabilities $(1, \ldots, N_S)/(N_S + 1)$. In cases of tied data, where this procedure potentially yields fewer than N_S splits, the number of equidistant probabilities is increased until the set of candidate splits has the desired size.

[4] See the `maxnumsplit` and `maxordsplit` arguments in `tvcglm_control`.

Splits for nominal scales The splits $\triangle_{kmlj}$ for nominal moderators are rules of the form $\{\text{Is } z_{kli} \in \zeta_{kmlj}?\}$, where ζ_{kmlj} are merged categories from the set $\{z_{kli} : \mathbf{z}_{ik} \in \mathcal{B}_{km}\}$. The number of unique candidate merges for C categories is 2^{C-1}, which increases exponentially with C. An approximation that restricts the number of splits to be linearly increasing with C deduces a category order and then treats the moderator as ordinal. For CART, Breiman et al. (1984) propose using the category-wise averages in the current node to deduce such an order. Following this idea, we propose ordering the categories by the category-wise estimated coefficients. This reduces the computational expenses to fitting the model that provides the category-wise coefficients, and fitting the (maximally) $C-1$ models that evaluate the ordinal splits. By default, the approximation is applied for $C \geq 5$.[5]

On page 57, we referred to the category ordering technique when demonstrating the `splitpath` function for the first iteration of partitioning. For instance, for partition B (the gender gap), we used the order $F < E < C < D < B < A$. The rank of a category can be deduced from the row where first "1" appears. The category-wise coefficients can be estimated by using the model:

```
R> glmCW.UCBA <-
+    glm(formula = Admit ~ 1 + Dept:Female, family = binomial(),
+        data = UCBA, weights = UCBA$Freq)
```

The model `glmCW.UCBA` substitutes the effect of `Female` of the current model, which is just the model `glmS.UCBA` (pg. 54), by an interaction term with `Dept` and `Female`. The category ordering is then obtained by ordering the estimated department-specific gender effects.

```
R> round(sort(coef(glmCW.UCBA)[-1]), 2)

DeptF:Female DeptE:Female DeptC:Female DeptD:Female DeptB:Female
       -2.36        -0.94        -0.44        -0.40         0.97
DeptA:Female
        1.76
```

Internally, our implementation uses an approximation technique to estimate category-wise coefficients, which is analogous to the technique used for approximating the search model $\widehat{\mathcal{M}}^*_{kmlj}$ (Eq. 3.9). See Appendix C.2.2 for the details.

Stopping criteria Algorithm 2 applies two stopping criteria. First, to have sufficient observations to estimate the coefficients nodewise, we require a minimum node size N_0. Here, $N_0 = 30$ seems a reasonable rule of thumb value, but can be modified according to the model. Second, to reduce the computational burden, we stop partitioning as soon as the maximal training error reduction falls below D_{min}. Large values of D_{min} yield rougher partitions and require less computation, and vice versa. Therefore, it is crucial to choose D_{min} to be small enough so that the best-sized partitions are not overlooked. The default $D_{min} = 2$ was selected based on the forward-stepwise AIC algorithm (e.g. Venables and Ripley, 2002), which also requires the total $-2 \cdot$ log-likelihood training error to decrease by at least 2 to continue. In our experience, $D_{min} = 2$ is small enough to capture the

[5] See the `maxnomsplit` argument in `tvcglm_control`. After the transformation to the ordinal scale, the argument `maxordsplit` controls the effective number of evaluated splits.

best-sized partition, yet reduces the computational burden considerably. In Section 3.4.1, we evaluate the impact of N_0 and D_{min} on a real data application.

The `tvcglm_control` function also allows us to control classic tree growth parameters. These parameters, which can include the maximum number of terminal nodes and the maximal depth of the trees, can restrict the complexity of the final model.

3.2.3 Pruning

The pruning stage selects the final model by collapsing the inner nodes of the overly fine partitions produced by Algorithm 2. In other words, it cuts branches stemming from the same node. Here, we broadly follow the minimal cost-complexity pruning approach of Breiman et al. (1984, Chap. 8). Let $\widehat{\mathcal{M}}$ be a fitted model of form (Eq. 3.3), where the nodes $\mathcal{B}_{km}$ result from Algorithm 2. Define the cost-complexity error criterion by

$$\text{err}_\lambda(\widehat{\mathcal{M}}) := -2\hat{\ell}_{\widehat{\mathcal{M}}} + \lambda \sum_{k=1}^{K} (M_k^{(\widehat{\mathcal{M}})} - 1), \quad \lambda \geq 0 \ . \tag{3.10}$$

In other words, we define the criterion as the total $-2 \cdot$ log-likelihood training error plus a tuning constant λ multiplied by the total number of splits. Here, λ trades off the in-sample performance and the complexity (i.e., the number of splits) of the model. When minimizing $\text{err}_\lambda(\widehat{\mathcal{M}})$, small choices of λ yield models with many splits, and vice versa. In general, λ is unknown and must be chosen adaptively from the data.

Pruning algorithm Pruning hierarchically collapses inner nodes of the initially overly fine partition to find the model that minimizes $\text{err}_\lambda(\widehat{\mathcal{M}})$, given λ. A global search that collapses multiple inner nodes simultaneously would be too computationally expensive and, therefore, we adopt the weakest link pruning algorithm of (Breiman et al., 1984). Algorithm 3 summarizes the implemented algorithm.

Algorithm 3: The TVCM weakest-link pruning algorithm for generalized linear models.

Input: A fitted model $\widehat{\mathcal{M}}$ from Algorithm 2
Parameters: λ: the cost-complexity penalty, $\lambda \geq 0$
repeat
  forall the *inner nodes* $\mathcal{B}^*_{kj}$ *of* $\widehat{\mathcal{M}}$, $k = 1, \ldots, K$ *and* $j = 1, \ldots, M_k - 1$ **do**
    Fit the model $\widehat{\mathcal{M}}_{kj}$ that collapses the inner node $\mathcal{B}^*_{kj}$ of $\widehat{\mathcal{M}}$.
    Compute the per-split increase of the training error
    $\bar{D}_{kj} = \frac{-2\hat{\ell}_{\widehat{\mathcal{M}}_{kj}} + 2\hat{\ell}_{\widehat{\mathcal{M}}}}{\sum_k M_k^{(\widehat{\mathcal{M}})} - \sum_k M_k^{(\widehat{\mathcal{M}}_{kj})}}$.
  if *any* $\bar{D}_{kj} \leq \lambda$ **then**
    Set $\widehat{\mathcal{M}} \leftarrow \widehat{\mathcal{M}}_{k'j'}$ with $\{k', j'\} = \arg\min_{k,j} \bar{D}_{kj}$
until *all* $\bar{D}_{kj} > \lambda$

Each iteration in Algorithm 3 collapses the inner node that yields the smallest per-split increase in the total $-2 \cdot$ log-likelihood training error. The procedure starts with the model from the partitioning stage and continues until the smallest per-split increase is

larger than λ (i.e., all remaining collapses would increase $\mathrm{err}_\lambda(\widehat{\mathcal{M}})$). The `prune` function implements Algorithm 3. For example, the fit for `vcmL.UCBA` on page 56 is pruned with $\lambda = 6$, as follows.

```
R> vcm.UCBA <- prune(vcmL.UCBA, cp = 6)
```

The pruning algorithm can be backtracked with the `prunepath` function.

```
R> prunepath(vcm.UCBA, steps = 1)
```

```
Step: 1
       part node loss     npar nsplit dev
<none>           5167.284 12   10
1      A     1   5682.041  7    5     102.9512893
2      A     3   5364.078  8    6      49.1984990
3      A     4   5171.914 10    8       2.3149860
4      A     5   5168.463 11    9       1.1789649
5      A     9   5167.420 11    9       0.1353552
6      B     1   5187.488  7    5       4.0408551
7      B     3   5184.859  8    6       4.3938080
8      B     4   5168.969  9    7       0.5614854
9      B     5   5168.272 11    9       0.9878986
10     B     8   5167.288 11    9       0.0034093
```

The above R output provides various information about the first iteration of Algorithm 3, applied on the fit for `vcmL.UCBA`. The columns `part` and `node` identify the collapsed inner node, and `dev` shows the per-split increase of the training error. In the first iteration, the inner node 8 of partition B (the gender gap) yields the smallest $\bar{D}_{kj}$ and is therefore collapsed.

Choosing λ The per-split penalty λ is generally unknown and, hence, must be chosen adaptively from the data. To do so, the validation-set or cross-validation methods are suitable. The validation-set method works as follows. First, divide the training data $\mathcal{D}$ randomly into a subtraining set $\mathcal{D}_1$ and a validation set $\mathcal{D}_2$, e.g., with a ratio of 3 : 1. Second, replicate the fit with Algorithm 2 based on $\mathcal{D}_1$. Third, repeatedly prune the new fit with increasing λ values and compute the validation error each time an inner node is collapsed. This yields two sequences, $\{\lambda_1 = 0, \ldots, \lambda_S, \lambda_{S+1} = \infty\}$ and $\{\overline{\mathrm{err}}_1^{\mathcal{D}_2}, \ldots, \overline{\mathrm{err}}_S^{\mathcal{D}_2}\}$, where $\overline{\mathrm{err}}_s^{\mathcal{D}_2} = \frac{-2}{\sum_{i\in\mathcal{D}_2} w_i} \sum_{i\in\mathcal{D}_2} w_i \log f_{\widehat{\mathcal{M}}}(y_i|\mathbf{x}_i, \mathbf{z}_i)$ is the average[6] prediction error on $\mathcal{D}_2$ of the new model pruned by λ values in interval $[\lambda_s, \lambda_{s+1})$. We retain the estimate for λ,

$$\hat{\lambda} = \frac{\lambda_{s'} + \lambda_{s'+1}}{2} \qquad \text{with} \qquad s' = \arg\min_{s\in\{1,\ldots,S\}} \overline{\mathrm{err}}_s^{\mathcal{D}_2} \ . \tag{3.11}$$

This is the center of the interval $[\lambda_{s'}, \lambda_{s'+1})$ that minimizes the validation error $\overline{\mathrm{err}}^{\mathcal{D}_2}$. The estimation potentially yields $\hat{\lambda} = \infty$, in which case no split is necessary. Cross-validation methods repeat the validation-set method to include the entire data. In particular, cross-validation combines the obtained sequences $\{\lambda_1, \ldots, \lambda_{S+1}\}$ to a finer grid and averages the errors $\overline{\mathrm{err}}_s^{\mathcal{D}_2}$ accordingly.

[6] We use the average to avoid having the validation error depend on the number of observations in $\mathcal{D}_2$.

The `cvloss` function implements the validation-set and the cross-validation methods to estimate $\hat{\lambda}$. By default, 5-fold cross-validation is used. To estimate λ for the `UCBA` data, we use the commands:

```
R> cv.UCBA <- cvloss(vcmL.UCBA,
+                    folds = folds_control(weights = "freq", seed = 13))
```

The argument "`weights = "freq"`" indicates that the weights of `vcmL.UCBA` represent counts rather than unit-specific weights (default). The `seed` argument is used to control the random generator when creating the cross-validation folds, which allows the results to be replicated. If available, the `cvloss` function processes the validation sets parallelized.

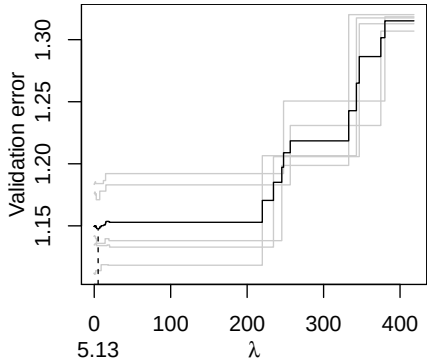

Figure 3.2: `cv.UCBA`: validation errors in function of λ from 5-fold cross-validating fits for `vcmL.UCBA`. Black solid line, the cross-validated error; gray solid lines, the validation errors on individual validation sets; vertical dotted line, the estimated value for λ.

The black solid line in Figure 3.2 shows the per-split penalty λ against the cross-validated error of fits for `vcmL.UCBA`, which is minimal with $\overline{\text{err}}^{\text{cv}}_{28} = 1.148$ at $\hat{\lambda} = 5.1$. The original fit for `vcm.UCBA` can be pruned by $\hat{\lambda} = 5.1$ with the command:

```
R> vcm.UCBA <- prune(vcmL.UCBA, cp = cv.UCBA$cp.hat)
```

The varying coefficients of the model obtained from pruning with $\hat{\lambda} = 5.1$ are shown in Figure 3.3. Both the varying intercept and the varying gender gap are split into three strata. The final model collapses several departments. For example, in the right panel, we see that the departments B, C, D, and F share the same gender gap. In contrast, the large negative intercept in department F and the large gender gap in department A remain detached.

Alternatives to $\hat{\lambda}$ (Eq. 3.11) could be considered. For example, Breiman et al. (1984, Chap. 3) propose the 1-SE rule to decrease the variance of $\hat{\lambda}$. We prefer $\hat{\lambda}$ for its simple form, but with `cvloss` and `prune`, we provide the tools to use these alternative rules.

3.3 Details and extensions

In Section 3.2, we explained the basic parts of the TVCM algorithm. This section describes the algorithm in more detail and explains how TVCM can be extended to other model classes.

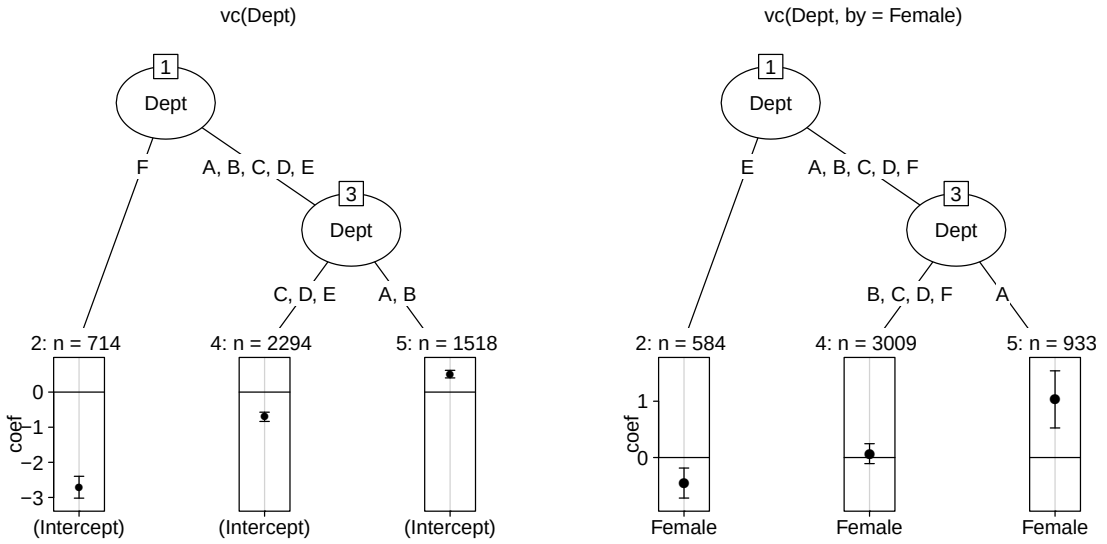

Figure 3.3: `vcm.UCBA`: pruned tree structures and nodewise coefficient plots. Left panel, the varying intercept; right panel, the varying gender gap

3.3.1 Mean-centering the predictors of the search model

A useful technique to improve the split selection with the search model $\widehat{\mathcal{M}}_{kmlj}$ (Eq. 3.8) is to mean-center its predictors. That is, we substitute the values x_{ik} in (Eq. 3.8) with the values $\tilde{x}_{ik} = x_{ik} - 1/N \sum_{i=1}^{N} x_{ik}$. Consider Figure 3.4. Both panels show the same scenario where the slope of a predictor x varies between two groups, A and B. In the left panel, the TVCM partitioning algorithm tries to uncover this moderation when x is not centered and the current model specifies a global intercept and a global slope for x. The search model uses the fitted values of the current model (solid line) as offsets and incorporates separate slopes for each group. This restricts the slopes to pass through the origin, and hence the fit (dotted and dashed lines) do not really identify the moderation pattern. The right panel shows that, in this scenario, the moderation pattern is perfectly identified by using the same search procedure, but when x is mean-centered.

The centering trick is applied by default, but can be disabled with the control argument `center`. Note that the output model is not affected by the mean-centering technique, because it is applied only to the search model $\widehat{\mathcal{M}}_{kmlj}$ (Eq. 3.8).

3.3.2 Additive expansion of multivariate varying coefficients

So far, we have implicitly assumed that the predictors $X_1, \ldots, X_P$ of model $\mathcal{M}_{\text{vc}}$ (Eq. 3.1) differ from one another. Here, we expand the multivariate varying coefficients into additive, moderator-wise components, in which case predictors appear repeatedly and identification issues arise. First, consider a multivariate varying coefficient term $x_{ip}\beta_p(\mathbf{z}_{ip}) = x_{ip}\beta_p(z_{ip1}, \ldots, z_{ipL_p})$, possibly $x_{ip} = 1$ for all i. The additive expansion is

$$x_{ip}\beta(\mathbf{z}_{ip}) \quad \longrightarrow \quad x_{ip}\beta_{p0} + x_{ip}\beta_{p1}(z_{ip1}) + \ldots + x_{ip}\beta_{ipL_p}(z_{ipL_p}) \ . \tag{3.12}$$

Here, we decompose $x_{ip}\beta(\mathbf{z}_p)$ into the "isolated" contributions of the individual moderators, including a global term $x_{ip}\beta_{p0}$. In this expansion, the individual varying coef-

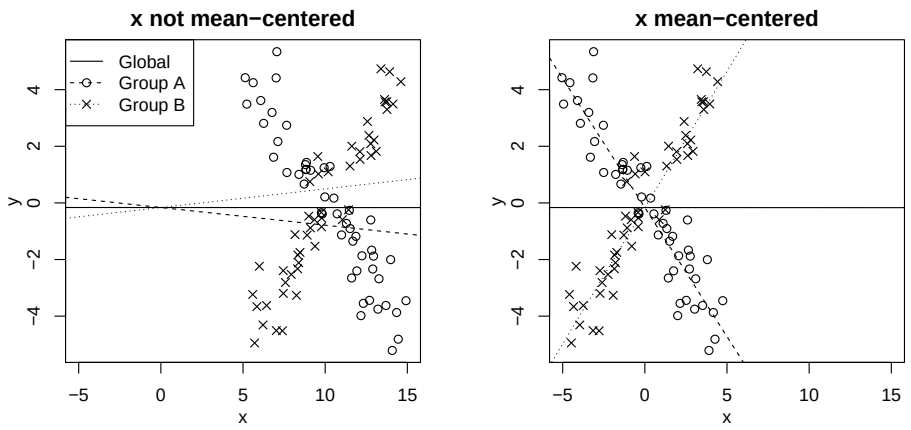

Figure 3.4: (Artificial) scenario where the slope of a predictor x with $\bar{x} = 10$ varies between two groups: circles, group A; crosses, group B; left panel, x in original scale; right panel, x mean-centered; solid line, the slope of the global model; dashed and dotted lines, the group specific slopes where the intercept is fixated on that of the global model.

ficients $\beta_{pl}(z_{ipl})$, $l = 1, \ldots, L_p$ act as local contributions to the global coefficient β_{p0}. To identify the additive expansion (Eq. 3.12), we mean-center the approximations for $\beta_{p1}(\cdot), \ldots, \beta_{pL_p}(\cdot)$ using node-weighted sum contrasts that restrict the sample-average of the coefficient functions to zero. That is, we approximate $\beta_{pl}(\cdot)$ with the piecewise constant function $\sum_{m=1}^{M_{pl}} 1(z_{ipl} \in \mathcal{B}_{plm})\beta_{plm}$, and estimate the coefficients $\beta_{pl1}, \ldots, \beta_{plM_{pl}}$ subject to

$$\sum_{i=1}^{N} \sum_{m=1}^{M_{pl}} 1(z_{ipl} \in \mathcal{B}_{plm}) w_i \beta_{plm} = 0 \ . \tag{3.13}$$

The nodewise-weighted sum contrasts are computed with the `contr.wsum` function of **vcrpart**. We also considered extending the additive expansion with second- and higher-order interactions between moderators. However, such an extension likely needs further considerations for the partitioning algorithm.

3.3.3 Extension to other model classes

To extend the scope of the algorithm, the TVCM requires functions to extract the training and validation errors from fitted models of the considered model class. The training error is required for partitioning (Algorithm 2) and pruning (Algorithm 3), and the need for extracting validation errors arises from cross-validating λ. Both errors refer to loss functions, which can (but must not) be the same as the one for estimating the coefficients. For GLMs, we use as the training error the total $-2 \cdot$log-likelihood loss on the training sample, which can be extracted from the coefficient estimation. Then, as the validation error, we use the average $-2 \cdot$ log-likelihood loss of predictions on validation sets, which can be extracted using the `predict` and `family` functions. Using the $-2 \cdot$ log-likelihood loss for both the training and validation errors synchronizes the criteria for estimating the coefficients, selecting the split, pruning, and choosing λ. The same, or similar implementation could be considered for other likelihood-based regression models.

The **vcrpart** package also provides implementations for the baseline-category and the cumulative-link models to allow for regression with nominal and ordinal responses. Both

these models are multivariate extensions of GLMs (cf. Fahrmeir and Tutz, 2001, Chap. 3). Therefore, we can adopt the definitions for the training and validation errors for GLMs.

3.4 Applications

In this section, we investigate three real data applications. Section 3.4.1 evaluates the TVCM on performance and sensitivities to changed stopping parameters, and Sections 3.4.2 and 3.4.3 illustrate moderated regression problems in social science research. All three applications use the default control parameters and a fixed seed for creating cross-validation folds.

```
R> control <- tvcglm_control(folds = folds_control(seed = 13))
```

For the presentation, we use the multivariate varying coefficient specification in Section 3.4.2 and the additive expansion in Section 3.4.3. A performance comparison between the two specifications is provided in Section 3.4.1.

3.4.1 Benchmark application: Pima Indians diabetes data

To evaluate the TVCM algorithm, we consider the pima indians diabetes data of Smith et al. (1988). These data are available from the UC Irvine machine learning repository (Bache and Lichman, 2013) and record diabetes tests of 768 Pima Indian women, along with eight covariates. Here, we use the `PimaIndiansDiabetes2` data of the R package **mlbench** (Leisch and Dimitriadou, 2010) containing a version of the original data corrected for physical impossibilities, such as zero values for blood pressure. We exclude the two variables `tricepts` and `insulin` and omit cases with missing values of the remaining data to avoid expanding the discussion to the missing value problem. The `Pima` data, prepared by the following commands, include 724 observations on the seven variables listed in Table 3.1.[7]

```
R> library("mlbench")
R> data("PimaIndiansDiabetes2")
R> Pima <- na.omit(PimaIndiansDiabetes2[, -c(4, 5)])
```

Table 3.1: Variables of the `Pima` data.

	Variable	Label	Scale (Unit)	Range
1	Diabetes	`diabetes`	Binary	Negative, Positive
2	Plasma glucose concentration	`glucose`	Continuous	$[44, 199]$
3	Number of times pregnant	`pregnant`	Continuous	$[0, 17]$
4	Diastolic blood pressure	`pressure`	Cont. $(mmHg)$	$[24, 122]$
5	Body mass index	`mass`	Cont. (kg/m^2)	$[18.2, 67.1]$
6	Diabetes pedigree function	`pedigree`	Continuous	$[0.08, 2.42]$
7	Age	`age`	Cont. (years)	$[21, 81]$

For this illustration, we follow Zeileis et al. (2006) and model the response variable `diabetes` with a logistic model with a varying intercept and a varying slope for `glucose` in the predictor function. The remaining covariates 3–7 of Table 3.1 are used as moderators for both varying coefficients. The described model can be fitted with the command

[7] Descriptive statistics of these variables can be found in the Tables C.3 and C.4.

```
R> vcm.Pima.1 <-
+    tvcglm(diabetes ~ -1 + vc(pregnant, pressure, mass, pedigree, age) +
+           vc(pregnant, pressure, mass, pedigree, age, by = glucose),
+           data = Pima, family = binomial(), control = control)
```

where the first `vc` term specifies the varying intercept and the second term specifies the varying slope for `glucose`. We use "`-1`" to remove the global intercept so that the fitted varying intercepts represent local intercepts. Keeping the global intercept would produce the same fit. However, the fitted varying intercepts would represent local contributions to the global intercept. The alternative additive expansion introduced in Section 3.3.2 is fitted using the command:

```
R> vcm.Pima.2 <-
+    tvcglm(diabetes ~ 1 + glucose +
+           vc(pregnant) + vc(pregnant, by = glucose) +
+           vc(pressure) + vc(pressure, by = glucose) +
+           vc(mass) + vc(mass, by = glucose) +
+           vc(pedigree) + vc(pedigree, by = glucose) +
+           vc(age) + vc(age, by = glucose),
+           data = Pima, family = binomial(), control = control)
```

The additive expansion includes a global intercept and a global slope for `glucose`, which implies that the remaining varying coefficients, which consist of moderator-wise varying intercepts and varying slopes for `glucose`, represent local contributions.

Zeileis et al. (2006) fit the same varying coefficient model using the *model-based recursive partitioning* algorithm (MOB, Zeileis et al., 2008), which is based on the single-tree approximation $\mathcal{M}_{\text{tree}}$ (Eq. 3.2). First, we compare the fit for `vcm.Pima.1` with the fit based on MOB to discuss the structural differences between the two approximations $\mathcal{M}_{\text{tree}}$ and $\mathcal{M}_{\text{tvcm}}$.

On the left, Figure 3.5 shows the fit for `vcm.Pima.1`, and on the right, the fit based on the MOB algorithm. The structural difference between the approaches is that the TVCM fits separate partitions for the varying intercept and the varying slope for `glucose`, while MOB algorithm fits a common partition for the two varying coefficients. Interestingly, the tree of the varying intercept from the TVCM is identical to the tree from the MOB algorithm. In contrast, the TVCM does not retain splits for the slope of `glucose`. This illustrates the flexibility of the TVCM in adapting to situations in which coefficient functions differ. If a single partition for all varying coefficients is accurate, then the TVCM can fit the same partition multiple times. Otherwise, it can tailor the partition individually for each varying coefficient. As a result, the TVCM potentially produces more parsimonious and/or more accurate fits than does the $\mathcal{M}_{\text{tree}}$ approximation.

To evaluate the performance of the TVCM, we extend the benchmark study of Zeileis et al. (2006) for the `Pima` data, comparing MOB with the *conditional inference tree* (CTree, Hothorn et al., 2006), CART (Breiman et al., 1984), *logistic model tree* (LMT, Landwehr et al., 2005), and C4.5 (Quinlan, 1993) algorithms.[8] The MOB and CTree algorithms are implemented in the **Rpartykit** package (Hothorn and Zeileis, 2014) (and **party**), CART in **rpart** (Therneau et al., 2014), and LMT and C4.5 in **RWeka** (Hornik et al., 2009). We

[8] Zeileis et al. (2006) also include the *quick, unbiased, efficient, statistical tree algorithm* (QUEST Loh and Shih, 1997).

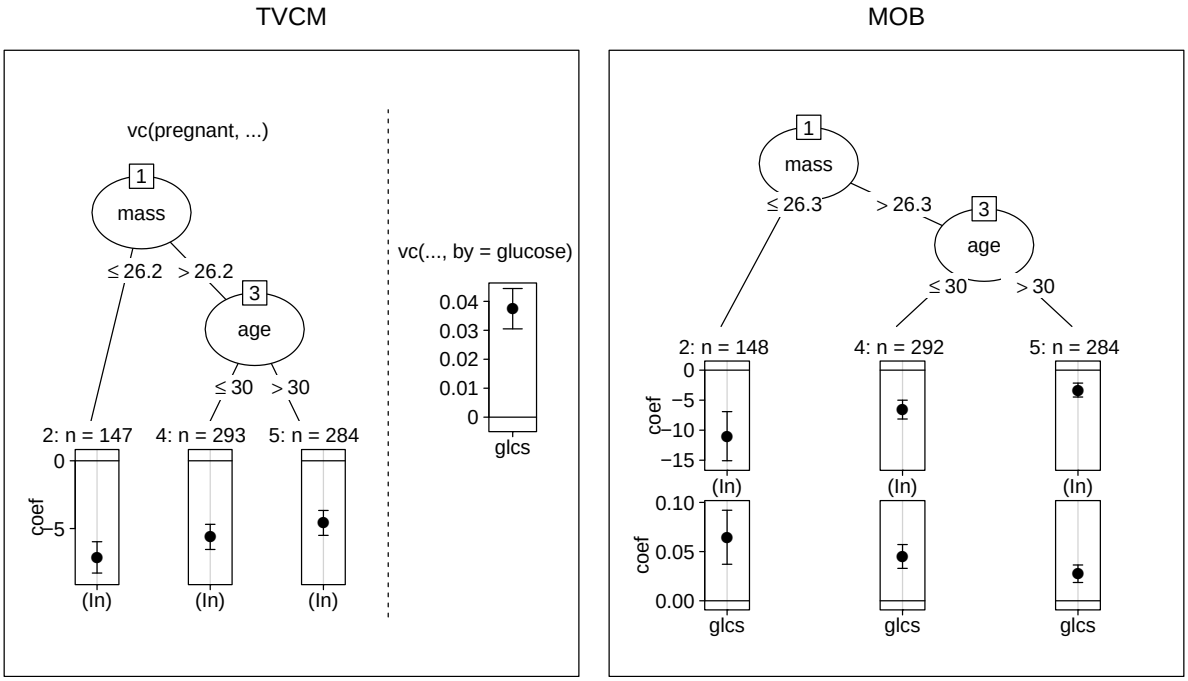

Figure 3.5: Fitted tree structures and nodewise coefficient plots. Left panel, the varying intercept (left) and the varying slope for `glucose` (right, no split) of the fit for `vcm.Pima.1`; right panel, the fit based on MOB (cf. Zeileis et al., 2006).

denote CART as RPART and C4.5 as J4.8 because the corresponding R implementations are slightly modified.

The performance comparison relies on 250 bootstrap samples (with replacement) using the `Pima` data. For each bootstrap sample, we fit a model with each algorithm to predict the excluded observations. In the case of the TVCM, we fit five models on each bootstrap sample to compare the fits for `tvcm.Pima.1` and `tvcm.Pima.2` and to evaluate the sensitivity of fits for `tvcm.Pima.1` to changes from the defaults for N_0 (the minimum node size), D_{min} (the minimum training error reduction), and N_S (the maximum number of splits). For the competitors, we employ the default control parameters. Three comparison measures are considered: *misclassification*, the median 0-1 loss on excluded data; *complexity*, the median of the number of coefficients plus the number of splits; and *time*, the median computation time. Furthermore, with each algorithm, we fit a model on the original data. To run the simulation, we use a computer with an Intel Xeon 3.50GHz processor.

Table 3.2 shows that the TVCM outperforms the competitors in terms of performance and complexity. That is, it builds smaller models with better predictive performance than the other algorithms. In contrast, the TVCM performs worst in terms of computational time because it evaluates far more candidate models than do the competitors. Increasing N_0 and D_{min} accelerates the burden significantly, with surprisingly little effect on the performance. Apparently, in this application, it is not necessary to grow very large trees in the partitioning stage to produce an accurate fit. Furthermore, the difference between the multivariate varying coefficient specification and the additive expansion is negligibly small in this application.

Figure 3.6 shows averages of 250 pairwise differences between the competitors and the TVCM. The confidence intervals for the averages are based on the Student's t-distribution.

Table 3.2: Performances for the `Pima` data: Boot, results from fits on 250 bootstrap samples; Orig, results on the original data; Misclassification, misclassification error; Complexity, the number of coefficients plus the number of splits; Time, computation time in seconds.

	Misclassification		Complexity		Time	
	Boot	Orig	Boot	Orig	Boot	Orig
TVCM	0.245	0.232	10	6	120.00	96.39
TVCM (additive)	0.245	0.231	14	6	159.14	52.75
TVCM ($N_0 = 50$)	0.242	0.232	12	6	42.26	33.58
TVCM ($D_{min} = 50$)	0.250	0.250	2	2	1.84	1.06
TVCM ($N_S = 19$)	0.245	0.232	10	6	143.79	126.19
MOB	0.254	0.238	23	8	2.57	1.67
CTree	0.256	0.222	19	15	0.07	0.03
RPART	0.258	0.211	27	11	0.02	0.02
LMT	0.279	0.222	63	1	0.22	1.10
J4.8	0.279	0.213	89	11	0.07	0.09

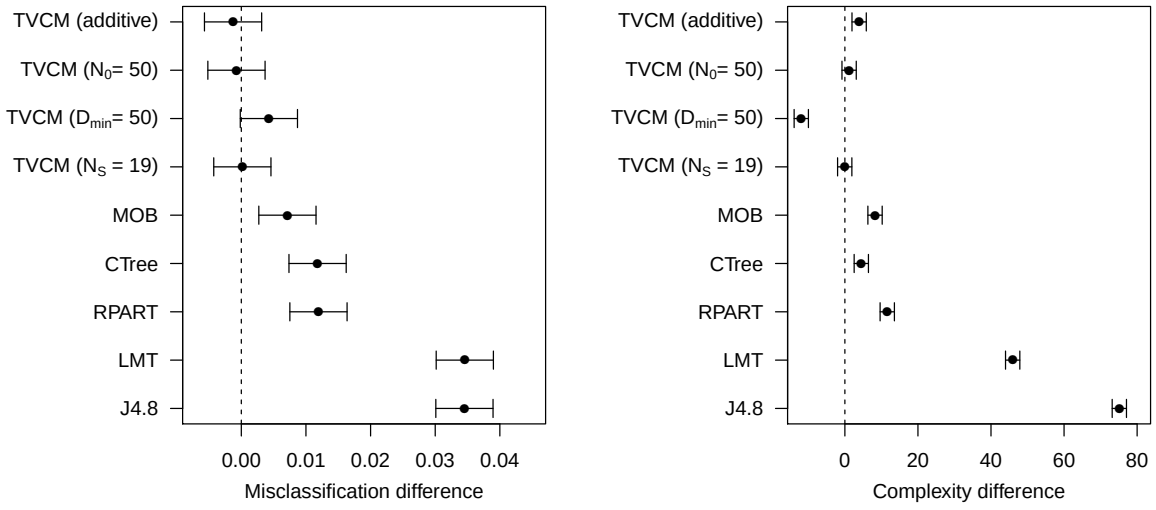

Figure 3.6: Performances for the `Pima` data relative to TVCM with default control parameters. Left, average difference in misclassification errors; right, average difference in complexity.

Several average differences are significant in favor of the TVCM. It may be that the CTree, RPART, LMT, and J4.8 algorithms perform worse because they merely use piecewise constant regression functions, whereas the TVCM and the MOB algorithm include a (prespecified) slope for `glucose`.

3.4.2 The racial wage gap

As the second application, we examine the racial wage gap and, specifically, whether the wage gap varies across strata. A suitable data set to examine the issue are the `Schooling` data of the R package **Ecdat** (Croissant, 2014). The `Schooling` data are a cross-section of 3,010 men prepared by Card (1993) from the 1976 wave of the US National Longitudinal Survey of Young Men (NLSYM).[9] Table 3.3 describes the variables of the `Schooling` data, where `lwage76` represents the response variable and the dummy `black` represents the predictor of interest.[10] The data were prepared as follows.

```
R> library("Ecdat")
R> data("Schooling")
R> Schooling <- Schooling[c(19, 21, 7, 28, 9, 14, 17, 18, 20, 23, 2, 4)]
R> Schooling$black <- 1 * (Schooling$black == "yes")
```

Table 3.3: The subset of used variables of the `Schooling` data.

	Variable	Label	Scale (Unit)	Values
1	Logarithm wage per hour 1976	`lwage76`	Cont. (¢/h)	[4.6, 9]
2	Is person black?	`black`	Binary	0=No, 1=Yes
3	Education in 1976	`ed76`	Continuous	[1, 18]
4	Working experience in 1976	`exp76`	Continuous	[0, 23]
5	Age in 1976	`age76`	Cont. (years)	[24, 34]
6	Lived with mom/ dad at age 14?	`momdad14`	Binary	No, Yes
7	Lived in south in 1966?	`south66`	Binary	No, Yes
8	Lived in south in 1976?	`south76`	Binary	No, Yes
9	Mother-father education class	`famed`	Continuous	[1, 9]
10	Enrolled in 1976?	`enroll76`	Binary	No, Yes
11	Lived in smsa in 1976?	`smsa76`	Binary	No, Yes
12	Grew up near 4-yr college?	`nearc4`	Binary	No, Yes

A standard model for wage is provided by the Mincer equation (Mincer, 1974), stating that schooling and working experience are the principal predictors for wage. Therefore, a (Gaussian) linear model that predicts `lwage76` by `ed76`, `exp76` (linear and squared), and `black` seems a suitable basis model for the examination of the racial wage gap.

Since the literature (e.g. Ashenfelter and Card, 1999) has widely discussed the endogeneity problem in regressing wages on schooling, we implement an instrumental variable (IV) approach using *college proximity* (`nearc4`) as the instrument for *schooling* (`ed76`). This instrument, which we computed with

```
R> Schooling$ed76.IV <- fitted(lm(ed76 ~ nearc4, Schooling))
```

[9] See http://davidcard.berkeley.edu/data_sets.html.

[10] Descriptive statistics of these variables can be found in the Tables C.5 and C.6.

has been proposed and evaluated by Card (1993). We rely on their evaluation and do not go into detail, because the endogeneity problem is not the point of this illustration.

With the instrument `ed76.IV` for `ed76`, the intended basis model, including the Mincer equation and the interesting `black` dummy in the predictor function, is fitted by

```
R> lm.School <- lm(lwage76 ~ ed76.IV + exp76 + I(exp76^2) + black,
+                  data = Schooling)

               Estimate  Std. Error  t value
(Intercept)  3.90434300  0.26330621  14.8281
ed76.IV      0.16421990  0.01964518   8.3593
exp76        0.05483130  0.00718382   7.6326
I(exp76^2)  -0.00243190  0.00035145  -6.9197
black       -0.31594171  0.01806455 -17.4896
```

The fit reveals that `black` has a significantly (t value > 2) negative impact on `lwage76`.

The aim of this application is to illustrate how the TVCM could be used to study moderations on the effect of `black`. To do this, we consider the covariates of 3–11 of Table 3.3 as potential moderators. Furthermore, we want to account for the direct effects of the covariates 5–11 on wage, which are those covariates not included in `lm.School`. To integrate these two extensions, we replace the constant coefficient of `black` with a varying coefficient and we replace the global intercept with a varying intercept. However, we continue to assume the Mincer equation and, therefore, use the same specification for the direct effects of `ed76.IV` and `exp76` as in `lm.School`. To fit the described extended model, we use the following formula.

```
R> f.School <- lwage76 ~ -1 + ed76.IV + exp76 + I(exp76^2) +
+    vc(age76, momdad14, south66, south76, famed, enroll76, smsa76) +
+    vc(ed76.IV, exp76, age76, momdad14, south66,
+       south76, famed, enroll76, smsa76, by = black)
```

Then, we fit the varying coefficient model using

```
R> vcm.School <- tvcglm(formula = f.School, data = Schooling,
+                        family = gaussian(), control = control)
```

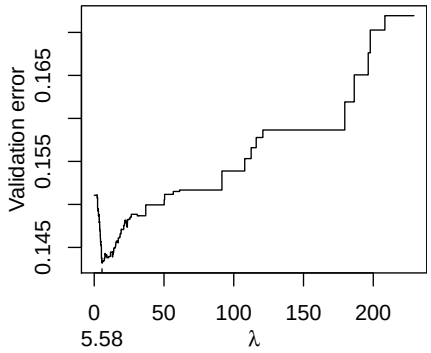

Figure 3.7: `vcm.School`: 5-fold cross-validated error in function of λ.

Figure 3.7 shows the 5-fold cross-validated error as a function of λ. The estimated $\hat{\lambda} = 5.58$ is situated in a clear dump. Hence, its selection is unanimous.

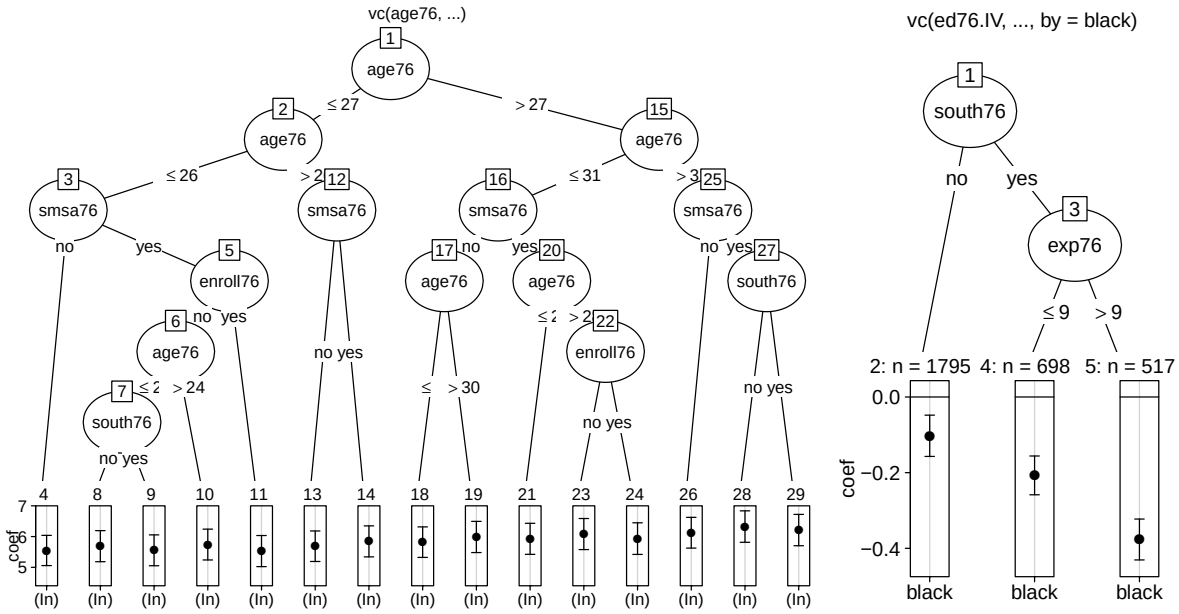

Figure 3.8: `vcm.School`: fitted tree structures and nodewise coefficient plots. Left panel, the varying intercept; right panel, the varying race gap.

The fitted varying intercept and varying wage gap are shown in Figure 3.8. The varying intercept on the left consists of 15 terminal nodes. The tree structure suggests that, in particular, `age76` and `smsa76` (**s**tandard **m**etropolitan **s**tatistical **a**rea) have strong direct effects on wage. We do not study the varying intercept in detail because it was mainly implemented to allow the study of the racial wage gap while controlling for the direct effects of the considered variables.

The interesting varying racial wage gap, shown in the right panel of Figure 3.8, includes three strata. It turns out that the gap is particularly negative for people that live in a southern state and have high working experience. For people that live in the north or that live in the south and have low working experience (equal or less than 9 years), the gap is smaller. However, the negative gap remains.

The estimated non-varying coefficients for `ed76.IV` and `exp76` (linear, squared) can be extracted using the `summary` or the `coef` functions, for example, with

```
R> coef(vcm.School)$fe

     ed76.IV        exp76   I(exp76^2)
 0.042089677  0.006525149 -0.001878967
```

3.4.3 The effect of parental leave duration on return to work

As the last application, we consider an example from the literature on the effects of welfare reforms. Specifically, we investigate the effect of the 1990 reform of the Austrian parental-leave (PL) system. Before 1990, working women had the right to stay off work after childbirth up to 12 months and, thereafter, return to the same (or similar) job at the same employer. The 1990 reform extended the duration of this leave from 12 months to 24 months. Lalive and Zweimüller (2009) investigated the effect of the 1990 PL reform on various fertility and labor marked outcomes, based on linear regression models and

using the Austrian Social Security Database (ASSD). They provide a background to the Austrian PL system and describe the data subset.[11] Here, using the same data, we reanalyze the effect of the reform on whether women returned to work at least once in the 10 years after childbirth.

The subset of the ASSD data includes 6,180 women who gave birth in June or July 1990 and were eligible to access the Austrian PL system. With **vcrpart**, the data are loaded by

```
R> data("PL")
```

The interesting PL reform dummy is `july`. A "0" in `july` means childbirth in June 1990, which is the last month under the old PL system, and a "1" indicates a birth in July 1990, which is the first month under the new PL system. The response variable `uncj10` is binary, where "1" refers to women who returned to work at least once in the 10 years after the considered childbirth. Both `july` and `uncj10` are summarized in Table 3.4, along with eight further covariates used as moderators.[12]

Table 3.4: The subset of used variables of the `PL` data.

	Variable	Label	Scale	Range
1	Whether returned to work 0-120 months after childbirth	`uncj10`	Binary	0 = No, 1 = Yes
2	Whether childbirth was in July	`july`	Binary	0 = June, 1 = July
3	Years employed before birth	`workExp`	Continuous	[0, 17.5]
4	Years unemployed before birth	`unEmpl`	Continuous	[0, 5.8]
5	Daily earnings at birth	`laborEarnings`	Cont. (€/d)	[0, 1510.7]
6	Whether white collar worker	`whiteCollar`	Binary	no, yes
7	Daily earnings 1989	`wage`	Cont. (€/d)	[0, 98.6]
8	Age	`age`	Ordinal	1, [15−19]; ...; 5, [35−44]
9	Industry	`industry.SL`	Nominal	20 categories
10	Region	`region.SL`	Nominal	9 categories

First, we consider a basis logistic model for `uncj10` with only `july` in the predictor function.

```
R> glm.PL <- glm(uncj10 ~ july, data = PL, family = binomial)
```

```
             Estimate Std. Error   z value
(Intercept)  1.8399616 0.05349103 34.397575
july        -0.2338688 0.07133637 -3.278394
```

The estimated effect of `july` is −0.23 (corresponding to an odds ratio of $e^{-0.23} = 0.79$) and is significant (z value > 2). This means that the 1990 PL reform decreases the logit for returning to work.

The aim of this application is to investigate whether and how the effect of the PL reform is moderated by covariates 3–10 of Table 3.4, which include for example age and region.

[11] The data subset is available from https://sites.google.com/site/rafaellalive/research.

[12] Descriptive statistics of these variables can be found in the Tables C.7 and C.8.

Furthermore, we want to study such moderation by considering the direct effects of the moderators. To implement this, we use the additive expansion for multivariate varying coefficients introduced in Section 3.3.2. The additive expansion is restrictive because it ignores interactions between moderators. However, in applied regression analysis it is common to limit the scope on first-order interactions between the predictor of interest and further covariates (e.g. Cox, 1984). For each considered moderator, the intended model adds varying intercepts and varying coefficients for `july` to the basis model `glm.PL`, and is specified by the formula

```
R> f.PL <- uncj10  ~  1 + july +
+       vc(age) + vc(age, by = july) +
+       vc(workExp) + vc(workExp, by = july) +
+       vc(unEmpl) + vc(unEmpl, by = july) +
+       vc(laborEarnings) + vc(laborEarnings, by = july) +
+       vc(whiteCollar) + vc(whiteCollar, by = july) +
+       vc(wage) + vc(wage, by = july) +
+       vc(industry.SL) + vc(industry.SL, by = july) +
+       vc(region.SL) + vc(region.SL, by = july)
```

Note that we keep the global intercept and the global effect of `july` as global references for the individual varying coefficients. The model is fitted with

```
R> vcm.PL <- tvcglm(formula = f.PL, family = binomial(),
+                       data = PL, control = control)
```

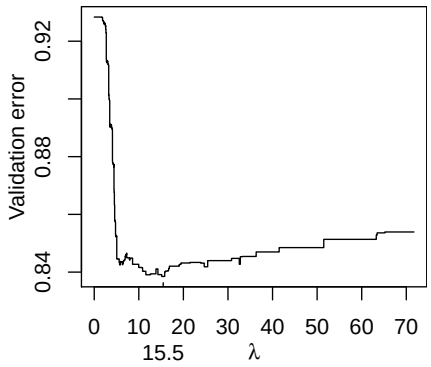

Figure 3.9: `vcm.PL`: 5-fold cross-validated error in function of λ.

Figure 3.9 shows that the 5-fold cross-validated error is smallest at $\hat{\lambda} = 15.47$. The error curve is relatively flat on the right of the minimum. Hence, the selection of λ is a little less evident compared to the selection based on Figure 3.7.

Figure 3.10 renders the fitted varying coefficients with at least one split. The top row shows the varying intercepts, which are contributions to the global intercept $\hat{\beta}^{(\texttt{Intercept})} = 1.93$, and are interpreted as direct effects on the logits for return to work. The result suggests that women with low working experience ($\leq$ 2.2 years) and a high wage ($>$ 45.8€/d) have increased logits to return to work, and that there are also differences between industries. Specifically, working in an industry corresponding to Node 3, which includes service industries such as social insurance, education and bank industries, has a positive direct effect on return to work.

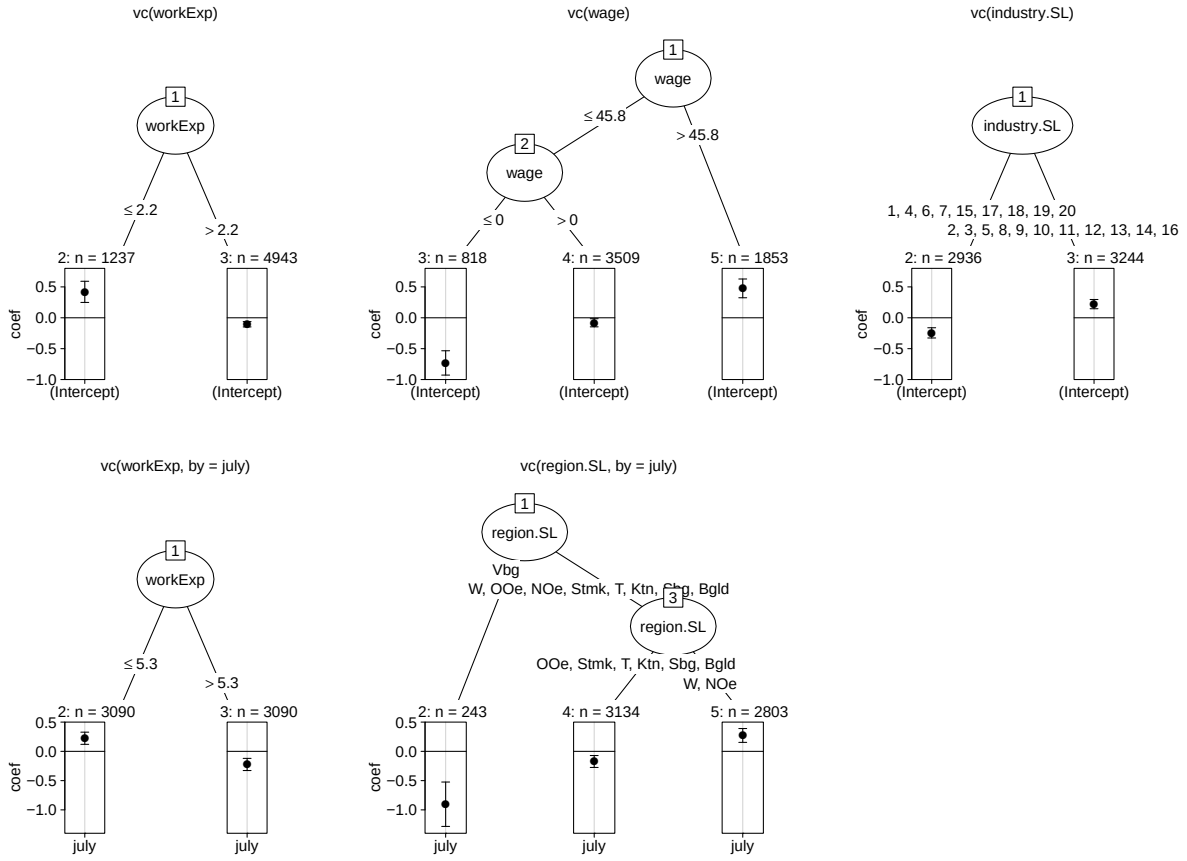

Figure 3.10: `vcm.PL`: fitted varying coefficients with at least one split. Top row, the varying intercepts; bottom row, the varying PL reform effects. The coefficients are contributions to the global intercept $\hat{\beta}^{\texttt{(Intercept)}} = 1.93$ resp. the global PL reform effect $\hat{\beta}^{\texttt{july}} = -0.23$.

The global effect of the PL reform on the logits for return to work is estimated to be $\hat{\beta}^{\texttt{july}} = -0.23$. The moderation effects of the two selected variables, working experience and region, are shown in the bottom row of Figure 3.10. From the nodewise coefficient plots, we learn that low working experience ($\leq$ 5.3 years) increases $\hat{\beta}^{\texttt{july}}$ by 0.22, and living in Vienna (W) or Lower Austria (NOe) increases $\hat{\beta}^{\texttt{july}}$ by 0.27. These positive contributions imply that the effect of the PL reform locally surpasses zero, especially for those women who combine the two characteristics *low working experience* and *living in Vienna or Lower Austria*.

3.5 Discussion and outlook

In this study, we showed how to use the TVCM algorithm for varying coefficient regression, as well as its implementation in the R package **vcrpart**. Unlike alternative tree-based algorithms, the TVCM can build a separate partition for each varying coefficient. Thus, it allows us to select moderators individually by varying the coefficient and to specify coefficient-specific sets of moderators. Furthermore, empirical evidence (Sec. 3.4.1) suggests that the TVCM potentially builds more accurate and/or more parsimonious models than competing tree-based algorithms. In addition to the description of the TVCM, we

discussed the model specification, provided R commands, and evaluated the performance by applying the algorithm to real data.

Further research could investigate the theoretical properties of the TVCM in more detail. This could include simulation studies and/or comparisons with smoothing splines and/or kernel regression techniques, in line with the comparison study of Wang and Hastie (2014).

There is also potential for improving the TVCM. This could include: (i) developing an unbiased selection procedure for partitioning; (ii) decreasing the computational time; (iii) refining the pruning criterion; and (iv) stabilizing the algorithm.

Improvement (i) requires finding an alternative criterion that does not tend to select partitions, nodes, and moderators with many split candidates (cf. Sec. 3.2.2). At the outset, we considered implementing the score-based coefficient constancy tests of Zeileis and Hornik (2007), used in the MOB algorithm. We were particularly interested into these tests because they would have allowed to select the partition, the node and the moderator based on the scores of the current model $\widehat{\mathcal{M}}$ (Eq. 3.7), without estimating search models. However, we discarded the idea because the tests work under the condition that the predictors of the model are *stationary and ergodic* (cf. Andrews, 1993) with respect to the tested moderator, which seems difficult to control when partitioning coefficient-wise. Another adjustment would be to derive the distribution of the maximally selected likelihood ratio statistics $D_{k'm'l'j'}$ of Algorithm 2. This would allow us to select the split based on p-values, which eliminates the dependence of the selection on the number of splits in the moderator variable. Andrews (1993) developes the distribution of maximally selected likelihood ratio statistics, however, again under the stationarity assumption. Indeed, the stationarity assumption could be resolved by using bootstrap techniques (e.g. Jouini, 2008), but such techniques are computationally complex. Finally, F- or χ^2-type tests, such as those proposed in Loh and Shih (1997), could be implemented. For example, Brandmaier et al. (2012) implement such tests for building structural equation model trees, and they show that their implementation reduces the variable selection bias substantially.

With regard to point (ii), the TVCM seems more time consuming than the alternative algorithms (cf. Sec. 3.4.1), although we integrated several acceleration techniques and parallelized the cross-validation. This hindrance, which might be relevant for big data applications, could be partly solved by rewriting the bottleneck functions in a low-level programming language. With regard to improvement (iii), we could consider refining the cost-complexity criterion (Eq. 3.10), which assumes that the "optimism" of the training error linearly increases with each split. Ye (1998) showed that this assumption is violated for CART, and the same probably applies to the TVCM. Ye (1998) and Efron (2004) provide more accurate solutions using resampling techniques, though these solutions are highly time consuming. Finally, with regard to improvement (iv), to stabilize the algorithm regarding perturbations to the data and to improve the accuracy, we provide with the `fvcglm` function an implementation of the random forest (Breiman, 2001) ensemble algorithm for the TVCM. However, we have not addressed this implementation so that we could focus on the original parts of our work.

Along with `tvcglm`, `tvcglm_control`, `splitpath`, `prunepath`, and `plot`, this study introduced the main functions for the fitting and diagnosis of coefficient-wise tree-based varying coefficient models. Additional diagnosis functions, such as `summary` and `predict`, are easily found in the manual.

Bibliography

Alexander, W. P., S. D. Grimshaw, and P. William (1996). Treed Regression. *Journal of Computational and Graphical Statistics 5*(2), 156–175.

Andrews, D. W. K. (1993). Tests for Parameter Instability and Structural Change with Unknown Change Point. *Econometrica 61*(4), 821–56.

Arulampalam, W., A. L. Booth, and M. L. Bryan (2007). Is There A Glass Ceiling Over Europe? Exploring the Gender Pay Gap Across the Wage Distribution. *Industrial and Labor Relations Review 60*(2), 163–186.

Ashenfelter, O. and D. Card (Eds.) (1999). *Handbook of Labor Economics*, Volume 3. Amsterdam, Netherlands: Elsevier.

Bache, K. and M. Lichman (2013). UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Bickel, P. J., E. A. Hammel, and J. W. O'Connell (1975). Sex Bias in Graduate Admissions: Data from Berkeley. *Science 187*(4175), 398–403.

Brandmaier, A. M., T. von Oertzen, J. J. McArdle, and U. Lindenberger (2012). Structural Equation Model Trees. *Psychological Methods 18*(1), 71–86.

Breiman, L. (2001). Random Forests. *Machine Learning 45*(1), 5–32.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. New York, USA: Wadsworth.

Bürgin, R. (2015). *vcrpart: Tree-Based Varying Coefficient Regression for Generalized Linear and Ordinal Mixed Models*. R package version 0.3-3, URL http://cran.r-project.org/web/packages/vcrpart/.

Card, D. (1993). Using Geographic Variation in College Proximity to Estimate the Return to Schooling. Technical report, National Bureau of Economic Research.

Chambers, J. M. and T. Hastie (1992). *Statistical Models in S*. Advanced Books & Software. Pacific Grove, USA: Wadsworth & Brooks/Cole.

Cox, D. R. (1984). Interaction. *International Statistical Review / Revue Internationale de Statistique 52*(1), 1–24.

Croissant, Y. (2014). *Ecdat: Data Sets for Econometrics*. R package version 0.2-7, URL http://CRAN.R-project.org/package=Ecdat.

Efron, B. (2004). The Estimation of Prediction Error; Covariance Penalties and Cross-Validation. *Journal of the American Statistical Association 99*(467), 619–642.

Fahrmeir, L. and G. Tutz (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Fan, J. and W. Zhang (2008). Statistical Methods with Varying Coefficient Models. *Statistics and Its Interface 1*(1), 179–195.

Hastie, T. and R. Tibshirani (1993). Varying-Coefficient Models. *Journal of the Royal Statistical Society B 55*(4), 757–796.

Hayfield, T. and J. S. Racine (2008). Nonparametric Econometrics: The np Package. *Journal of Statistical Software 27*(5), 1–32.

Heim, S. (2007). *svcm: 2D and 3D Space-Varying Coefficient Models in R*. R package version 0.1.2, URL http://cran.r-project.org/package=svcm.

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

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2015). *mboost: Model-Based Boosting*. R package version 2.4-2, URL http://CRAN.R-project.org/package=mboost.

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

Hothorn, T. and A. Zeileis (2014). partykit: A Modular Toolkit for Recursive Partytioning in R. In *Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics*, Number 2014-10. Universität Innsbruck.

Jouini, J. (2008). Bootstrap Methods for Single Structural Change Tests: Power Versus Corrected Size and Empirical Illustration. *Statistical Papers 51*(1), 85–109.

Lalive, R. and J. Zweimüller (2009). Does Parental Leave Affect Fertility and Return-to-Work? Evidence from Two Natural Experiments. *The Quarterly Journal of Economics 124*(3), 1363–1402.

Landwehr, N., M. Hall, and E. Frank (2005). Logistic Model Trees. *Machine Learning 95*(1-2), 161–205.

Leisch, F. and E. Dimitriadou (2010). *mlbench: Machine Learning Benchmark Problems*. R package version 2.1-1, URL http://cran.r-project.org/web/packages/mlbench/.

Loh, W.-Y. (2002). Regression Trees With Unbiased Variable Selection and Interaction Detection. *Statistica Sinica 12*(2), 361–386.

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

McCullagh, P. and J. Nelder (1989). *Generalized Linear Models* (2 ed.). Monographs on Statistics and Applied Probability. London, UK: Chapman and Hall.

Mincer, J. A. (1974). *Schooling, Experience, and Earnings*. NBER Books. National Bureau of Economic Research, Inc.

Quinlan, J. R. (1992). Learning with Continuous Classes. In *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, Singapore, pp. 343–348. World Scientific.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, USA: Morgan Kaufmann Publishers Inc.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL http://www.R-project.org.

Russell, S. J. and P. Norvig (2003). *Artificial Intelligence: A Modern Approach* (3 ed.). New Jersey, USA: Pearson Education Inc.

Smith, J., J. Everhart, W. Dickson, W. Knowler, and R. Johannes (1988). Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 261–265. American Medical Informatics Association.

Therneau, T., B. Atkinson, and B. D. Ripley (2014). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-8, URL http://CRAN.R-project.org/package=rpart.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (4 ed.). Statistics and Computing. New York: Springer-Verlag.

Wang, J. C. and T. Hastie (2014). Boosted Varying-Coefficient Regression Models for Product Demand Prediction. *Journal of Computational and Graphical Statistics 23*(2), 361–382.

Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Texts in Statistical Science. Boca Raton, USA: Chapman and Hall.

Ye, J. (1998). On Measuring and Correcting the Effects of Data Mining and Model Selection. *Journal of the American Statistical Association 93*(441), 120–313.

Yusuf, S., J. Wittes, J. Probstfield, and H. A. Tyroler (1991). Analysis and Interpretation of Treatment Effects in Subgroups of Patients in Randomized Clinical Trials. *Journal of the American Medical Association 266*(1), 93–98.

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

Zeileis, A., T. Hothorn, and K. Hornik (2006). Evaluating Model-Based Trees in Practice. In *Research Report Series*, Number 32. Wirtschaftsuniversität Wien.

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

Conclusion

The present dissertation proposed contributions to graphical longitudinal data analysis and moderated regression analysis, in form of three self-contained articles. Chapter 1 developed the decorated parallel coordinate plot and the Chapters 2 and 3 developed two tree-based algorithms for moderated regression analysis.

The decorated parallel coordinate plot is an exploratory tool for identifying typical categorical longitudinal patterns. Although the article placed the focus on rendering chronological orders of event occurrences, the plot equally applies to other types of categorical longitudinal data, see for example Appendix A.1 for an application of the plot on ordinally scaled repeated measurements data. The main strength of the plot is its ability to highlight typical patterns (e.g., the frequent ones) while rendering at the same time the diversity of the observed longitudinal patterns. While there are no technical limitations on the scalability of the plot, increasing the number and/or the length of the sequences or the number of response categories may impair the plot interest. Hence, further research could be on advanced filtering methods or techniques for merging similar patterns.

The Chapters 2 and 3 developed tree-based algorithms for building moderated regression models, for settings with many potential moderators. The algorithm of Chapter 2 allows fitting tree-structured varying coefficients in mixed models while maintaining the random effect component globally. It combines and extends the technique for splitting and tree size selection of Zeileis et al. (2008) and the technique for incorporating a regression tree into a mixed model of Hajjem et al. (2011) and Sela and Simonoff (2012).

A key development for the algorithm of Chapter 2 is the adjustment of the score-based coefficient constancy tests of Zeileis and Hornik (2007) for mixed models of Section 2.2.3. This adjustment entails an additional pre-decorrelation step that aims to remove intra-individual correlations between observation scores. With regard to the algorithm, the adjustment is required to ensure that the variable selection is unbiased and that the stopping rule is statistically interpretable. Besides their usage in the algorithm, the implemented constancy tests could be used as a diagnosis tool for evaluating coefficient instabilities regarding omitted third variables of a given mixed model. The presentation leaves certain room for rigidifying the proposed adjustment. Therefore, Appendix B.4 provides a supplementary simulation study that demonstrates that the accuracy of the tests is maintained for models with random slopes und for unbalanced data. Further research could include analytic investigations on (a) the validity of the assumptions (v) to (vii) of Section 2.2.3, which state that the variance, the intra-individual covariance and the inter-individual covariance of observation scores are constant under coefficient constancy, and (b) asymptotic aspects of the test implementation. Moreover, extensions of the constancy tests to other longitudinal regression models, such as marginal models (e.g. Bergsma et al., 2009), could be considered. Although such extensions may not be straightforward, the insights gained here on how to deal with intra-individually correlated observation scores are presumably a useful basis.

Chapter 3 introduced a coefficient-wise partitioning algorithm for varying coefficients in generalized linear models, and its implementation in R. Building individual partitions for each varying coefficient allows moderators to be selected individually by coefficient and coefficient-specific sets of moderators to be specified. Furthermore, empirical evidence suggests that the algorithm can build more accurate and/or more parsimonious models than alternative single-tree approaches are able to do. The algorithm could also be applied in a mixed model setting, however, it might be computationally too expensive compared to single partition approaches such as that of Chapter 2. Chapter 3 focused primarily on real data applications. Therefore, I provide in Appendix C.1 a supplementary simulation study. Specifically, I show that the performance of the proposed coefficient-wise algorithm for identifying an underlying data model improves with increasing numbers of observations, and that it is more powerful than the single-partition approach if the coefficient functions differ from each other.

The coefficient-wise partitioning algorithm is entirely based on exhaustive search, in line with the algorithms of Breiman et al. (1984) and Wang and Hastie (2014). The advantage out of this is easy extensibility, but, it can be computationally costly and biased towards selecting variables with many splits. Consequently, the order in which variables are selected should be interpreted carefully. At the outset, I considered implementing the score-based constancy tests of Chapter 2, for simultaneously selecting the partition, the node and the variable. However, as mentioned above, I have discarded this idea because these tests work only under certain conditions, which seem hard to control in coefficient-wise partitioning. Thus, further research could be on a (further) adjustment of the score-based constancy tests or on identifying or developing a suitable alternative.

The two algorithms of the Chapters 2 and 3 are, in a certain respect, opposed to each other. While the algorithm of Chapter 2 groups all varying coefficients to approximate them with a single partition, the algorithm of Chapter 3 treats each varying coefficient individually. In specific situations, it might be desirable to mix the two approaches. For example, grouping the coefficients of dummy variables corresponding to a common categorical predictor is such a case, e.g., see Yuan et al. (2006) for a related discussion in penalized linear regression. Assigning groups of coefficients to a partition is implemented in the provided software. The problem is, due to differences in the degree of freedom, the partitioning algorithm will tend to select the partitions associated with many coefficients.

The two algorithms of the Chapters 2 and 3 also distinguish by their in-built tree size selection criteria. The algorithm of Chapter 2 selects the tree size by continuing partitioning until the p-values of the constancy tests indicate coefficient constancy regarding all moderators in all nodes. By contrast, the algorithm of Chapter 3 attempts minimizing the cost-complexity criterion of (Eq. 3.10) by pruning. It is controversial whether to use one or the other criterion. The constancy tests require less computational efforts and are easier to interpret. However, the algorithm controls merely locally the statistical error for not stopping iterating. Furthermore, by selecting the tree size by stopping, the algorithm remains a simple forward-stepwise procedure that potentially overlooks relevant splits that appear only after poor splits (e.g. Breiman et al., 1984). Pruning is a stepwise-backward procedure that collapses inner nodes to find the globally optimal trade-off between model complexity and in-sample performance. By starting with overly fine partitions that include the relevant splits with high certainty, pruning should avoid overlooking important splits that appear only after poor splits. However, pruning requires the estimation of the cost-complexity parameter by using validation set techniques, which can be time consuming and/or introduce perturbations because of the randomness involved

in selecting validation sets. Moreover, pruning requires a complexity measure for a fitted tree-structured model, which is difficult to derive analytically (cf. Ritschard, 2006). The number of splits, which I defined as the complexity measure in Chapter 3, is indeed a heuristic and may be refined in the future development, e.g., by pursuing the techniques of Ye (1998) and Efron (2004). In the regression tree literature, p-value based tree size selection is standard for algorithms that employ statistical tests for variable selection (e.g. Hothorn et al., 2006; Zeileis et al., 2008). An exception is the algorithm of Loh (2002) that performs tests for variable selection and cost-complexity pruning for tree size selection. Pruning is standard for algorithms that are entirely based on exhaustive search (e.g. Breiman et al., 1984; Wang and Hastie, 2014). Comparisons are usually made between algorithms (e.g. Lim, 2000; Hothorn et al., 2006), so that the performance of the tree size selection criteria cannot be separated from the performance of the partitioning procedure. Hence, although the choice of the criterion is also a philosophical and computational question, an isolated evaluation of the performance of these two tree size criteria – by keeping fixed the partitioning procedure – could be a topic for future investigations.

The Chapters 2 and 3 focused on binary partitioning. Extensions, such as multiway partitioning (e.g. Kim and Loh, 2001) or partitioning by linear combinations of variables (e.g. Breiman et al., 1984), can provide substantial merits and could be implemented in the further development. Furthermore, the handling of missing values could be investigated. Such investigations would have to differentiate between missing values in the response variable, in the predictor variables and in the moderator variables. Missing values in the response or in the predictors may be handled with the corresponding methods for parametric models, which include expectation-maximization algorithms and imputation techniques, see for example Little and Rubin (2002). Missing values in the moderators may be handled with the corresponding methods for recursive partitioning, which include surrogate splitting (e.g. Breiman et al., 1984) or probability splitting, see for example the in-depth discussion of Quinlan (1989) on this topic.

The common thread of the Chapters 2 and 3 is that both proposed algorithms are based on a closed model approach. In particular, while most approaches that combine recursive partitioning and linear regression models fit a multitude of linear models on disjoint subsets of the training data (e.g. Zeileis et al., 2008; Wang and Hastie, 2014), the approaches presented herein fit a single linear model that incorporates one or multiple tree-structures in the predictor function. This idea was initially inspired by Hajjem et al. (2011); Sela and Simonoff (2012), and the motivation for the closed model approach was in Chapter 2 to maintain the random effect component globally, and in Chapter 3 to allow for coefficient-wise partitions. Thus, the contribution of the Chapters 2 and 3 is also to promote the closed model approach for combining recursive partitioning and linear models, including identifying situations where using a closed model approach is necessary.

The dissertation focused on tree-based approaches for incorporating moderated relations into a predictor function. The linear approach, which is more common in applied moderated regression, was considered for the purpose of comparison, see for example the Sections 2.3.1 and 3.2.1. The popularity of the linear approach is certainly due to its availability in most statistical software environments, its well understood statistical properties and its allowance for statistically powerful conclusions. However, the linearity assumption may be inappropriate, in particular if it is neither clear whether nor how the moderators should be incorporated. Therefore, nonparametric alternatives, such as those proposed herein, seem qualified for considerably many situations in applied moderated regression.

For linear moderated regression, there are many techniques for variable selection.

These include, amongst others, stepwise Akaike information criterion search (AIC, e.g. Akaike, 1974; Venables and Ripley, 2002), sparse regression, such as least absolute shrinkage and selection operator regression (LASSO, e.g. Tibshirani, 1996; Bühlmann and Van De Geer, 2011) or smoothly clipped absolute deviation penalty regression (SCAD, e.g. Fan and Li, 2001), and model-based boosting (e.g. Bühlmann and Hothorn, 2007). The basic aim of these techniques is to discover and incorporate the variables that directly affect the response variable. Their application in moderated regression analysis, which also (or principally) aims to discover and incorporate the variables that affect the coefficients of predictors of interest, requires additional considerations. For example, it should be ensured that the relations of interest are not dropped from the predictor function during the selection process. Often, constraints are specified that ensure the interpretability of the finally selected model. For example, dummy coded variables corresponding to a common categorical moderator may be selected as a group (e.g. Meier et al., 2008), or interactions terms may be selected hierarchically, e.g., a first order interaction is only included if both involved variables have an important direct effect (e.g. Bien et al., 2013).

The literature on nonparametric moderated regression has primarily been focused on smoothing spline and kernel regression techniques. Both approaches have their own merits and can be more accurate than the tree-based approach, see for example Wang and Hastie (2014) for an in-depth comparison. The potential of kernel and spline regression techniques has hardly been exploited. For example, Li and Racine (2008) and Li et al. (2013) have elaborated kernel regression varying coefficient techniques for mixed scaled sets of moderators. Wang et al. (2008) and Xue and Qu (2012) combine smoothing spline with sparse regression techniques to fit time-varying coefficients for many predictors. The R package **mboost** (Hothorn et al., 2015) implements a boosted spline approach that allows to include many moderators. Smoothing spline techniques combined with sparse regression or model-based boosting is scalable to many moderators and, therewith, is promising for the moderation problem considered herein. However, such approaches are focused on continuous moderators and require prespecified basis functions and knots. Thus, their application often requires much specification work, while tree-based approaches can generally be applied without excessive tuning.

The principal disadvantages of the developed tree-based algorithms are their instability regarding data perturbations and their inaccuracy in approximating smooth coefficient functions. Ensemble techniques, such as boosting (Freund, 1995) and random forest (Breiman, 2001), have proven to improve the algorithms regarding these two aspects, however, at the expense of the comprehensibility of the built models. Appendix B.2 implements a random forest extension for the algorithm of Chapter 2, which is demonstrated to improve the predictive performance by means of an empirical evaluation. This result motivates a comprehensive implementation of the random forest technique for the provided algorithms as a project for the further development. Fitting routines for the random forest extensions are already available in the R **vcrpart** package, see in the manual for the functions `fvcolmm` (random forest for Algorithm 1) and `fvcglm` (random forest for Algorithm 2). The package also provides partial dependency plots (e.g. Hastie et al., 2001, Chap. 10) for the diagnosis, and variable importance measures (e.g. Breiman, 2001; Strobl et al., 2008) may be implemented in the next revision. Herein, I did not focus on these extensions because I consider the implementation of the two modeling techniques – tree-structured fixed effect components in mixed models and coefficient-wise partitioning – as the principal contributions of the dissertation.

The dissertation also placed special emphasis on practical aspects. The presented

articles include various real data applications that could prove instructive for practitioners to identify the utility of the presented methods for their specific problem. Moreover, the software implementations in the **TraMiner** and **vcrpart** packages allow for easy access to these methods. Beside the main functions (Table 5), the packages provide various diagnosis functions which are easily found in the manual. The R codes for the real data applications are found in the Appendices A.3, B.7 and C.4.

Bibliography

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Bergsma, W., M. A. Croon, and J. A. Hagenaars (2009). *Marginal Models for Dependent, Clustered, and Longitudinal Categorical Data*. Statistics for Social and Behavioral Sciences Series. New York, USA: Springer-Verlag.

Bien, J., J. Taylor, and R. Tibshirani (2013). A Lasso For Hierarchical Interactions. *The Annals of Statistics 41*(3), 1111–1141.

Breiman, L. (2001). Random Forests. *Machine Learning 45*(1), 5–32.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. New York, USA: Wadsworth.

Bühlmann, P. and T. Hothorn (2007). Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science 22*(5), 477–505.

Bühlmann, P. and S. Van De Geer (2011). *Statistics for High-Dimensional Data*. Heidelberg, Germany: Springer-Verlag.

Efron, B. (2004). The Estimation of Prediction Error; Covariance Penalties and Cross-Validation. *Journal of the American Statistical Association 99*(467), 619–642.

Fan, J. and R. Li (2001). Variable Selection via Conconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and computation 121*(2), 1–50.

Hajjem, A., F. Bellavance, and D. Larocque (2011). Mixed Effects Regression Trees For Clustered Data. *Statistics & Probability Letters 81*(4), 451–459.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2015). *mboost: Model-Based Boosting*. R package version 2.4-2, URL http://CRAN.R-project.org/package=mboost.

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

Kim, H. and W.-Y. Loh (2001). Classification Trees With Unbiased Multiway Splits. *Journal of the American Statistical Association 96*(454), 589–604.

Li, Q., D. Ouyang, and J. S. Racine (2013). Categorical Semiparametric Varying-Coefficient Models. *Journal of Applied Econometrics 28*(4), 551–579.

Li, Q. and J. S. Racine (2008). Smooth Varying-Coefficient Estimation and Inference for Qualitative and Quantitative Data. *Econometric Theory 26*(6), 1607–1637.

Lim, Tjen-Sien, L. W.-Y. S. Y.-S. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning 40*(3), 203–228.

Little, R. J. A. and D. B. Rubin (2002). *Statistical Analysis with Missing Data* (2 ed.). Wiley Series in Probability and Statistics. New York, USA: John Wiley & Sons.

Loh, W.-Y. (2002). Regression Trees with Unbiased Variable Selection and Interaction Detection. *Statistica Sinica 12*(2), 361–386.

Meier, L., S. Van De Geer, and P. Bühlmann (2008). The Group Lasso for Logistic Regression. *Journal of the Royal Statistical Society B 70*(1), 53–71.

Quinlan, J. R. (1989). Unknown Attribute Values in Induction. In *Proceedings of the Sixth International Workshop on Machine Learning*, San Francisco, USA, pp. 164–168. Morgan Kaufmann Publishers Inc.

Ritschard, G. (2006). Computing and Using the Deviance with Classification Trees. In A. Rizzi and M. Vichi (Eds.), *COMPSTAT 2006 - Proceedings in Computational Statistics*, Heidelberg, Germany, pp. 55–66. Springer-Verlag.

Sela, R. and J. S. Simonoff (2012). RE-EM trees: A Data Mining Approach for Longitudinal and Clustered Data. *Machine Learning 86*(2), 169–207.

Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). Conditional Variable Importance for Random Forests. *BMC Bioinformatics 9*(307), 1471–2105.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society B 58*(1), 267–288.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (4 ed.). Statistics and Computing. New York: Springer-Verlag.

Wang, J. C. and T. Hastie (2014). Boosted Varying-Coefficient Regression Models for Product Demand Prediction. *Journal of Computational and Graphical Statistics 23*(2), 361–382.

Wang, L., H. Li, and J. Huang (2008). Variable Selection in Nonparametric Varying-Coefficient Models for Analysis of Repeated Measurements. *Journal of the American Statistical Association 103*(484), 1556–1569.

Xue, L. and A. Qu (2012). Variable Selection in High-Dimensional Varying-Coefficient Models with Global Optimality. *Journal of Machine Learning 13*(1), 1973–1998.

Ye, J. (1998). On Measuring and Correcting the Effects of Data Mining and Model Selection. *Journal of the American Statistical Association 93*(441), 120–313.

Yuan, M., M. Yuan, Y. Lin, and Y. Lin (2006). Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society B 68*(1), 49–67.

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

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

List of Figures

List of Tables

Appendix A

Supplementary materials: Chapter 1

A.1 Marijuana use among U.S. teenagers

The aim of this second illustrative application is to demonstrate the potential of our plot for rendering state sequences. The difference with event sequences is that the position in a state sequence conveys time information and that simultaneous states cannot occur. In this application the x-axis reports ages.

We consider data about the use of marijuana taken from Lang et al. (1999) and based on the first five annual waves (1976-1980) of the U.S. National Youth Survey (Elliott et al., 1989). The data concern adolescents aged 13 at the first wave (1976) and report adolescents' marijunana-use state at the successive ages between 13 and 17 years old. The *marijuana-use* is a categorical ordinal variable with three levels ("never", "no more than once a month", "more than once a month") obtained by Lang et al. (1999) by collapsing the nine levels of the original *marijuana-use* scale.[1]

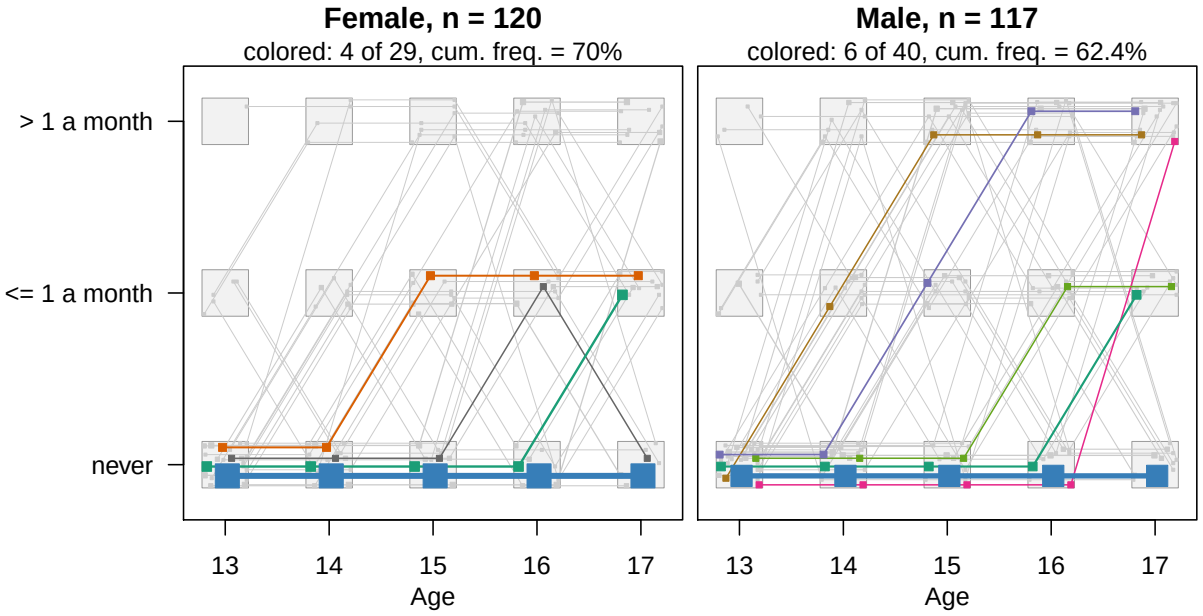

Figure A.1: *Marijuana use* of U.S. teenagers between ages 13 to 17. The trajectories shared by at least three adolescents in the group are highlighted in different colors.

[1]Further descriptive statistics of this data set can be found in Appendix A.2.2.

Figure A.3 exhibits the evolution of *marijuana use* by females and males. Colored patterns are the unique patterns shared by at least three adolescents (3%) in each group. The most frequent trajectory is to never use marijuana between ages 13 to 17 for both genders. Looking at the other patterns including the greyed lines we observe a higher diversity among trajectories followed by males. There are 40 unique trajectories for males versus 29 for females. The plots also reveal, for both groups, a tendency to increasing marijuana with age. Focusing on the colored lines – most frequent patterns, we observe what is the main conclusion found by Lang et al. (1999), i.e., a higher risk for males to use marijuana. More specifically the plots reveal a tendency for males to start with *marijuana use* earlier than females.

In this example, all sequences are complete and, therefore, right- and left-aligned. When all sequences are complete, no unique sequence can be embedded in another unique sequence. Plotting only non embeddable sequences would thus produce the same plot. Shifting sequences of different length in order to left or right align them would result in loss of the time alignment. Thus, the embedding trick is useful for time-aligned sequences of different length only when all of them either start or end at the same time.

A.2 Descriptive statistics of the used data sets

A.2.1 Family life event history data set

The family life event history data set concern 1,372 Scandinavians (Danes, Norwegians and Swedes) born between 1930 and 1959. The data set is available online from the supplementary materials of Bürgin and Ritschard (2014).

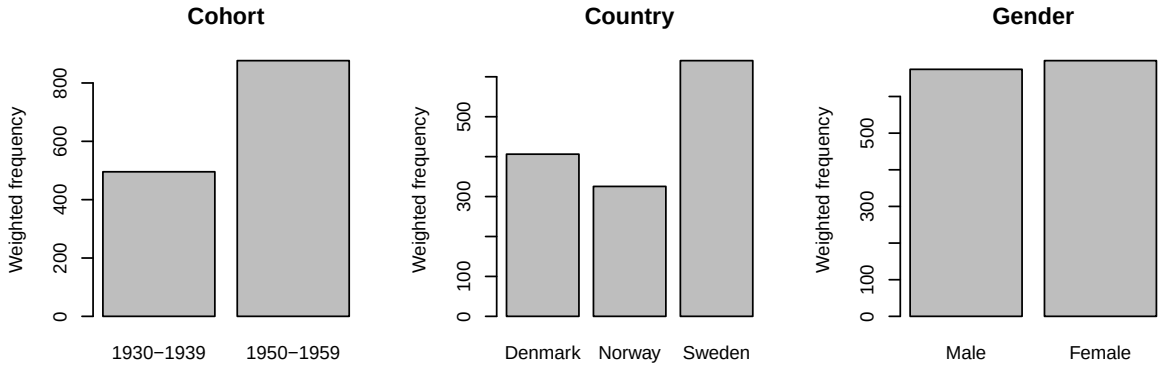

Figure A.2: Weighted univariate distributions of Scandinavians included in the family life event history data set.

Figure A.2 shows the weighted distributions of the included Scandinavians regarding *cohort*, *country* and *gender*. In the far left panel it is striking that the older cohort (1930-1939) includes much less people than the younger cohort (1950-1959).

A.2.2 Marijuana data set

The marijuana data of Section A.1 are available from the R **drm** package (Jokinen, 2012). They record yearly marijuana-use measurements of 237 adolescents between 13 and 17.

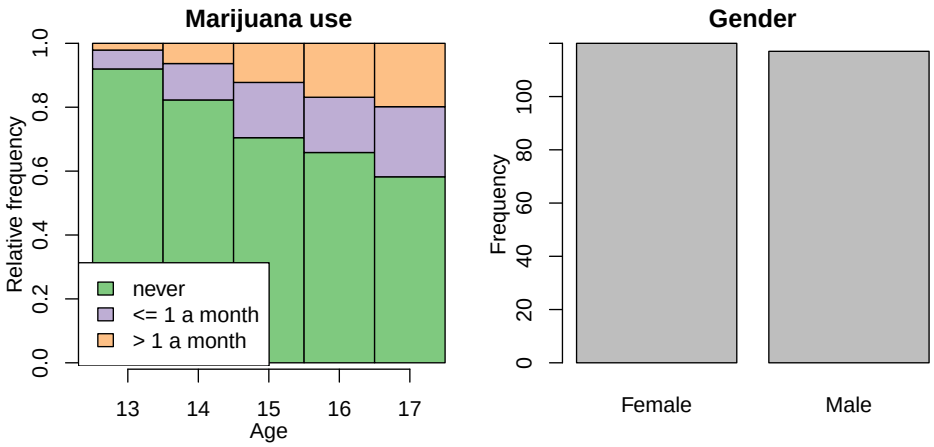

Figure A.3: Descriptive statistics of the marijuana data set. *Left*, cross-sectional distributions of *marijuana-use* across age; *right*, the gender distribution.

The left panel of Figure A.3 shows the cross-sectional distributions of the *marijuana-use* variable across age. The right panel shows the *gender* distribution, which is the only additional covariate in the data set.

Bibliography

Bürgin, R. and G. Ritschard (2014). A Decorated Parallel Coordinate Plot for Categorical Longitudinal Data. *The American Statistician 68*(2), 98–103.

Elliott, D. S., D. Huizinga, and S. Menard (1989). *Multiple Problem Youth: Delinquency, Substance Use, and Mental Health Problems*. New York, USA: Springer-Verlag.

Jokinen, J. (2012). *drm: Regression and Association Models for Repeated Categorical Data*. R package version 0.5-8, URL http://CRAN.R-project.org/package=drm.

Lang, J. B., J. W. McDonald, and P. W. F. Smith (1999). Association-Marginal Modeling of Multivariate Categorical Responses: A Maximum Likelihood Approach. *Journal of the American Statistical Association 94*(448), 1161–1171.

A.3 R-codes

```
## ------------------------------------------------------------ #
## Date:          2015-03-20
## Authors:       Reto Buergin and Gilbert Ritschard
## Institution:   Swiss National Centre of Competence in
##                Research LIVES (http://www.lives-nccr.ch/)
##
## Contents:
## R-code for generating the plots in "A decorated parallel
## coordinate plot for categorical longitudinal data. Code
## requires TraMineR (>= 1.8-7) and drm (>= 0.5-8).
##
## Contents:
## - Family life event histories
## - Marijuana use among U.S. teenagers
##
## We cannot garantuee that the functions work with future
## versions of R and the indicated packages.
##
## Copyright R. Buergin and G. Ritschard, 2015
## distributed under license Creative Commons BY-NC-SA
## http://creativecommons.org/licenses/by-nc-sa/3.0/
## ------------------------------------------------------------ #

## install.packages(c("TraMineR", "drm"))
library("TraMineR")
library("drm")

## ------------------------------------------------------------ #
## Family life event histories
## ------------------------------------------------------------ #

## prepare the data

## family.LONG: one row per event
family.LONG <- read.csv("family-LONG.csv")
levs <- c("Leaving Home", "First Union", "First Marriage",
          "First Child", "_end")
family.LONG$event <- factor(family.LONG$event, levels = levs)

## family.WIDE: one row per case (covariates)
family.WIDE <- read.csv("family-WIDE.csv")

## create a 'seqelist' object with one sequence per row
family.seqe <- seqecreate(family.LONG[, c("time", "id", "event")])

par(mar = c(4,8,3,1))

## Left plot of Figure 1
## --------------------

## select subset
subs <- family.WIDE$cohort == "1930-1939"
w <- family.WIDE$weights[subs]
w <- w / sum(w) * sum(subs)
```

```
## set filter
filter <- list(type="function", value="minfreq", level = 0.05)

## plot
seqpcplot(family.seqe[subs], weights = w, filter = filter)

## Right plot of Figure 1
## ----------------------

## set filter
filter <- list(type="function", value="minfreq", level = 0.01)

## plot
seqpcplot(family.seqe[subs], weights = w, filter = filter,
          grid.scale = 0.5, lwd = 2.5, order.align = "time",
          xlim = c(12,32), title = "")

## Figure 2
## --------

## set weights
w <- family.WIDE$weights / sum(family.WIDE$weights) * nrow(family.WIDE)

## set filter
filter <- list(type="function", value="minfreq", level = 0.05)

## plot
par(mar = c(0,0,3,0), oma = c(4,8,0,1))
seqpcplot(family.seqe, group = family.WIDE$cohort,
          weights = w, filter = filter, main = "")
## note: here the number of observations refers to the sum
##       of weights

## Left plot of Figure 3
## ---------------------

## extract subset
subs <- family.WIDE$cohort == "1950-1959"

par(mfrow = c(1,1))
p <- seqpcplot(family.seqe[subs], grid.scale = 0,
               cpal = "black", plot = FALSE)
p$lns$lwd.1 <- 0.15
plot(p)

## Right plot of Figure 3
## ----------------------

## extract subset
subs <- family.WIDE$cohort == "1950-1959"
w <- family.WIDE$weights[subs]
w <- w / sum(w) * sum(subs)

## set filter
filter <- list(type="function", value="minfreq", level = 0.05)

## plot
seqpcplot(family.seqe[subs], weights = w, filter = filter,
```

```
          ltype = "non-embeddable")


## ------------------------------------------------------------ #
## Marijuana use among U.S. teenagers
## ------------------------------------------------------------ #

## data preparation
data("marijuana")

## convert data from long into wide format
mar <- reshape(marijuana, idvar = "id", timevar = "age",
               direction = "wide", v.names = "y")

## create a 'stslist' object
states <- c("never", "<= 1 a month", "> 1 a month")
mar.seq <- seqdef(mar, var = 3:7, alphabet = 1:3,
                  states = states, cnames = 13:17)

## plot
par(mar = c(0,0,3,1), oma = c(4,6,0,1))
seqpcplot(mar.seq, group = mar$sex,
          filter = list(type="function",value="minfreq",level = 0.03),
          order.align = "time", xlab = "Age", ylim = c(0.75,3.25))
```

Appendix B

Supplementary materials: Chapter 2

B.1 Additional details on coefficient constancy tests

B.1.1 Covariance of $\Psi_N(\cdot)$ (Sec. 2.2.3.1)

The constraint $\sum_{i=1}^{N} \boldsymbol{\psi}_i(\hat{\boldsymbol{\beta}}_1) = \sum_{i=1}^{M} \hat{\boldsymbol{\psi}}_i = \mathbf{0}$, implies that

$$\begin{aligned} \text{Var}\left(\sum_{i=1}^{N} \hat{\boldsymbol{\psi}}_i\right) = \mathbf{0} &= \sum_{i=1}^{N}\sum_{i'=1}^{N} \text{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) \\ &= \sum_{i=1}^{N} \text{Var}(\hat{\boldsymbol{\psi}}_i) + \sum_{i=1}^{N}\sum_{i'=1}^{N} 1_{(i\neq i')}\text{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) \ . \end{aligned} \tag{B.1}$$

Under H_0, we assume that (ii) $\text{Var}(\hat{\boldsymbol{\psi}}_i) = \text{Var}(\hat{\boldsymbol{\psi}}_1)$ and (iii) $\text{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) = \text{Cov}(\hat{\boldsymbol{\psi}}_1, \hat{\boldsymbol{\psi}}_2)$, $\forall\ i \neq i'$. Based on these two assumption and (Eq. B.1),

$$\text{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) = -\frac{1}{N-1}\text{Var}(\hat{\boldsymbol{\psi}}_1),\ \forall i \neq i' \ . \tag{B.2}$$

In consequence, the covariance of the process $\boldsymbol{\Psi}_N(\tau)$ (Eq. 2.8) is

$$\begin{aligned} \text{Cov}(\boldsymbol{\Psi}_N(\tau_1), \boldsymbol{\Psi}_N(\tau_2)) &= \text{Cov}\left(\frac{1}{\sqrt{N}}\sum_{i=1}^{\lfloor N\tau_1\rfloor} \hat{\boldsymbol{\psi}}_{\sigma(v_i)}, \frac{1}{\sqrt{N}}\sum_{i'=1}^{\lfloor N\tau_2\rfloor} \hat{\boldsymbol{\psi}}_{\sigma(v_{i'})}\right) \\ &\overset{\tau_1\leq\tau_2}{=} \frac{1}{N}\sum_{i=1}^{\lfloor N\tau_1\rfloor} \text{Var}(\hat{\boldsymbol{\psi}}_i) + \sum_{i=1}^{\lfloor N\tau_1\rfloor}\sum_{i'=1}^{\lfloor N\tau_1\rfloor} \text{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) + \sum_{i=1}^{\lfloor N\tau_1\rfloor}\sum_{i'=\lfloor N\tau_1\rfloor+1}^{\lfloor N\tau_2\rfloor} \text{Cov}(\hat{\boldsymbol{\psi}}_i, \hat{\boldsymbol{\psi}}_{i'}) \\ &= \frac{1}{N}\left[\lfloor N\tau_1\rfloor\text{Var}(\hat{\boldsymbol{\psi}}_1) - \frac{\lfloor N\tau_1\rfloor[\lfloor N\tau_1\rfloor - 1]}{N-1}\text{Var}(\hat{\boldsymbol{\psi}}_1) - \frac{\lfloor N\tau_1\rfloor[\lfloor N\tau_2\rfloor - \lfloor N\tau_1\rfloor]}{N-1}\text{Var}(\hat{\boldsymbol{\psi}}_1)\right] \\ &= \frac{\lfloor N\tau_1\rfloor(N - \lfloor N\tau_2\rfloor)}{N[N-1]}\text{Var}(\hat{\boldsymbol{\psi}}_1) \ . \end{aligned} \tag{B.3}$$

B.1.2 Pre-decorrelation of scores (Sec. 2.2.3.2)

Here we derive the pre-decorrelated observations scores $\hat{\mathbf{u}}_{it}^{*}$ and the computation of the transformation matrix $\mathbf{T}$ of Section 2.2.3.2. First, we consider balanced data where $N_i =$

$N_1\ \forall\ i$. In these cases, we assume that

$$\begin{aligned}
\mathrm{Var}(\hat{\mathbf{u}}_{it}) = \quad & \mathrm{Var}(\hat{\mathbf{u}}_{11}) \quad && := \boldsymbol{\Delta}, \quad \forall\ i,t\ , && \text{(B.4)}\\
\mathrm{Cov}(\hat{\mathbf{u}}_{it},\hat{\mathbf{u}}_{it'}) = \quad & \mathrm{Cov}(\hat{\mathbf{u}}_{11},\hat{\mathbf{u}}_{12}) \quad && := \boldsymbol{\Omega}, \quad \forall\ i \text{ and } t\neq t' \text{ and} && \text{(B.5)}\\
\mathrm{Cov}(\hat{\mathbf{u}}_{it},\hat{\mathbf{u}}_{i't'}) = \quad & \mathrm{Cov}(\hat{\mathbf{u}}_{11},\hat{\mathbf{u}}_{21}) \quad && := \boldsymbol{\Psi}, \quad \forall\ t \text{ and } i\neq i'\ , && \text{(B.6)}
\end{aligned}$$

where $N_T = \sum_{i=1}^{N} N_i$. Since $\mathrm{E}(\hat{\mathbf{u}}_{it}) = \mathbf{0}$ and $\mathrm{Var}(\sum_{i,t}\hat{\mathbf{u}}_{it}) = \mathbf{0}$, these matrices can be estimated by

$$\begin{aligned}
\hat{\boldsymbol{\Delta}} &= \frac{1}{N_T}\sum_{i=1}^{N}\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it}\hat{\mathbf{u}}_{it}^{\top}\ , && \text{(B.7)}\\
\hat{\boldsymbol{\Omega}} &= \frac{1}{NN_1(N_1-1)}\left[\sum_{i=1}^{N}\left[\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it}\right]\left[\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it}\right]^{\top} - N_T\hat{\boldsymbol{\Delta}}\right] \text{ and} && \text{(B.8)}\\
\hat{\boldsymbol{\Psi}} &= -\frac{1}{N_T^2 - N_T - NN_1(N_1-1)}\sum_{i=1}^{N}\left[\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it}\right]\left[\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it}\right]^{\top}\ . && \text{(B.9)}
\end{aligned}$$

It follows that the *intra-individual* covariance of the $\hat{\mathbf{u}}_{it}^{*}$'s is

$$\begin{aligned}
\mathrm{Cov}(\hat{\mathbf{u}}_{it}^{*},\hat{\mathbf{u}}_{it'}^{*}) &\overset{t\neq t'}{=} \mathrm{Cov}\left(\hat{\mathbf{u}}_{it} + \mathbf{T}\sum_{t''=1,t''\neq t}^{N_1}\hat{\mathbf{u}}_{it''},\quad \hat{\mathbf{u}}_{it'} + \mathbf{T}\sum_{t''=1,t''\neq t'}^{N_1}\hat{\mathbf{u}}_{it''}\right)\\
&= \mathrm{Cov}(\hat{\mathbf{u}}_{it},\hat{\mathbf{u}}_{it'}) + \sum_{t''=1,t''\neq t'}^{N_1}\mathrm{Cov}(\hat{\mathbf{u}}_{it},\hat{\mathbf{u}}_{it''})\mathbf{T}^{\top} + \mathbf{T}\sum_{t''=1,t''\neq t}^{N_1}\mathrm{Cov}(\hat{\mathbf{u}}_{it''},\hat{\mathbf{u}}_{it'})\\
&\quad + \mathbf{T}\sum_{t''=1,t''\neq t}^{N_1}\sum_{t'''=1,t'''\neq t'}^{N_1}\mathrm{Cov}(\hat{\mathbf{u}}_{it''},\hat{\mathbf{u}}_{it'''})\mathbf{T}^{\top}\\
&= \dots\\
&= \boldsymbol{\Delta}\mathbf{T}^{\top} + \mathbf{T}\boldsymbol{\Delta}^{\top} + [N_1-2]\mathbf{T}\boldsymbol{\Delta}\mathbf{T}^{\top} + \boldsymbol{\Omega} + [N_1-2]\boldsymbol{\Omega}\mathbf{T}^{\top} +\\
&\quad [N_1-2]\mathbf{T}\boldsymbol{\Omega}^{\top} + [[N_1-1]^2 - [N_1-2]]\mathbf{T}\boldsymbol{\Omega}\mathbf{T}^{\top}\ , \qquad \text{(B.10)}
\end{aligned}$$

and the *inter-individual* covariance of the $\hat{\mathbf{u}}_{it}^{*}$'s is

$$\begin{aligned}
\mathrm{Cov}(\hat{\mathbf{u}}_{it}^{*},\hat{\mathbf{u}}_{i't'}^{*}) &\overset{i\neq i'}{=} \mathrm{Cov}\left(\hat{\mathbf{u}}_{it} + \mathbf{T}\sum_{t''=1,t''\neq t}^{N_1}\hat{\mathbf{u}}_{it''},\quad \hat{\mathbf{u}}_{i't'} + \mathbf{T}\sum_{t''=1,t''\neq t'}^{N_1}\hat{\mathbf{u}}_{it''}\right)\\
&= \mathrm{Cov}(\hat{\mathbf{u}}_{it},\hat{\mathbf{u}}_{i't'}) + \sum_{t''=1,t''\neq t'}^{N_1}\mathrm{Cov}(\hat{\mathbf{u}}_{it},\hat{\mathbf{u}}_{i't''})\mathbf{T}^{\top} + \mathbf{T}\sum_{t''=1,t''\neq t}^{N_1}\mathrm{Cov}(\hat{\mathbf{u}}_{it''},\hat{\mathbf{u}}_{i't'})\\
&\quad + \mathbf{T}\sum_{t''=1,t''\neq t}^{N_1}\sum_{t'''=1,t'''\neq t'}^{N_1}\mathrm{Cov}(\hat{\mathbf{u}}_{it''},\hat{\mathbf{u}}_{i't'''})\mathbf{T}^{\top}\\
&= \dots\\
&= \boldsymbol{\Psi} + [N_1-1]\boldsymbol{\Psi}\mathbf{T}^{\top} + [N_1-1]\mathbf{T}\boldsymbol{\Psi}^{\top} + [N_1-1][N_1-1]\mathbf{T}\boldsymbol{\Psi}\mathbf{T}^{\top}\ . \qquad \text{(B.11)}
\end{aligned}$$

The goal is to determine the $MP_{\boldsymbol{\beta}} \times MP_{\boldsymbol{\beta}}$ matrix $\mathbf{T}$ such that

$$\mathrm{Cov}(\hat{\mathbf{u}}_{it}^{*},\hat{\mathbf{u}}_{it'}^{*}) = \mathrm{Cov}(\hat{\mathbf{u}}_{it}^{*},\hat{\mathbf{u}}_{i't''}^{*}) = -\frac{1}{\sum_{i=1}^{N}N_1 - 1}\mathrm{Var}(\hat{\mathbf{u}}_{11}^{*})\ , \qquad \text{(B.12)}$$

for all $(i,t) \neq (i,t')$ and $(i,t) \neq (i',t'')$. $\mathbf{T}$ is found by solving

$$\begin{aligned}\mathbf{0} &= \text{Cov}(\hat{\mathbf{u}}^*_{it}, \hat{\mathbf{u}}^*_{it'}) - \text{Cov}(\hat{\mathbf{u}}^*_{it}, \hat{\mathbf{u}}^*_{i't''}) \\ &= \Big[\boldsymbol{\Delta}\mathbf{T}^\top + \mathbf{T}\boldsymbol{\Delta}^\top + [N_1-2]\mathbf{T}\boldsymbol{\Delta}\mathbf{T}^\top + \boldsymbol{\Omega} + [N_1-2]\boldsymbol{\Omega}\mathbf{T}^\top + \\ &\quad [N_1-2]\mathbf{T}\boldsymbol{\Omega}^\top + [[N_1-1]^2 - [N_1-2]]\mathbf{T}\boldsymbol{\Omega}\mathbf{T}^\top\Big] - \Big[\boldsymbol{\Psi} + [N_1-1]\boldsymbol{\Psi}\mathbf{T}^\top + \\ &\quad [N_1-1]\mathbf{T}\boldsymbol{\Psi}^\top + [N_1-1][N_1-1]\mathbf{T}\boldsymbol{\Psi}\mathbf{T}^\top\Big] \ , \end{aligned} \tag{B.13}$$

for $\mathbf{T}$, using $\hat{\boldsymbol{\Delta}}$, $\hat{\boldsymbol{\Omega}}$ and $\hat{\boldsymbol{\Psi}}$. Either, this equation system is solved with respect to all $(MP_\beta)^2$ components, or $\mathbf{T}$ is assumed to be symmetric, which reduces the number of unknowns to $(MP_\beta(MP_\beta+1))/2$. The symmetry assumption is natural because $\mathbf{T}$ is used for a decorrelation transformation. Note that, because of the sum of scores remains zero after the transformation,

$$\sum_{i=1}^{N}\sum_{t=1}^{N_i}\hat{\mathbf{u}}^*_{it} = \sum_{i=1}^{N}\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it} + N_1(N_1-1)\mathbf{T}\sum_{i=1}^{N}\sum_{t=1}^{N_1}\hat{\mathbf{u}}_{it} = \mathbf{0} \ , \tag{B.14}$$

and the variance of the pre-decorrelated scores remains constant,

$$\begin{aligned}\text{Var}(\hat{\mathbf{u}}^*_{it}) &= \text{Var}(\hat{\mathbf{u}}_{it} + \mathbf{T}\sum_{t'=1,t'\neq t}^{N_1}\hat{\mathbf{u}}_{it'}) \\ &= \dots \\ &= \boldsymbol{\Delta} + [N_1-1]\mathbf{T}\boldsymbol{\Omega}^\top + [N_1-1]\boldsymbol{\Omega}\mathbf{T}^\top \\ &\quad + \mathbf{T}\left[[N_1-1]\boldsymbol{\Delta} + \frac{(N_1-1)(N_1-2)}{2}\boldsymbol{\Omega}\right]\mathbf{T}^\top \ \forall \ (i,t) \ , \end{aligned} \tag{B.15}$$

the equality with the third term in (Eq. B.12) holds automatically if the equality between the first two terms holds.

B.1.3 Imputation procedure for unbalanced data (Sec. 2.2.3.2)

The imputation for a missing observation t of individual i in model $\mathcal{M}$ (Eq. 2.1) requires values for the design matrices $\mathbf{X}_{it}$ and $\mathbf{W}_{it}$ and the moderator $\mathbf{z}_{it}$. We propose to randomly draw these data from the N_i sets of observed predictor vectors of individual i. Next, $\mathbf{y}_{it}$ is randomly drawn from the conditional distribution $f(y_{it}|\hat{\mathbf{b}}_i; \mathbf{X}_{it}, \mathbf{W}_{it}, \mathbf{z}_{it})$ of the estimated model under H_0, in order to control the type I error of the test. To estimate the random coefficients $\mathbf{b}_i$ we use the posterior mean estimate, see Tutz (2012, Chap. 14.3.2).

B.1.4 Nodewise tests (Sec. 2.2.3.3)

This appendix specifies the properties of the nodewise, mean centered scores. Let $\hat{\mathbf{U}}^*_m$ be the $N_m \times MP_\beta$ matrix of pre-decorrelated scores (Sec. 2.2.3.2) corresponding to observations $\mathbf{z}_{it} \in \mathcal{B}_m$. Let $\hat{\mathbf{U}}^{**}_m$ be the $\hat{\mathbf{U}}^*_m$ minus its column means. In Section 2.2.3.2, we established that $\text{Cov}(\hat{\mathbf{u}}^*_{it}, \hat{\mathbf{u}}^*_{i't'}) = -\frac{1}{N_T-1}\text{Var}(\hat{\mathbf{u}}^*_{11}) \ \forall \ (i,t) \neq (i',t')$. It follows that the covariance between the rows of $\hat{\mathbf{U}}^{**}_m$,

$$\begin{aligned}\mathrm{Cov}(\hat{\mathbf{u}}^{**}_{mit}, \hat{\mathbf{u}}^{**}_{mi't'}) &= \mathrm{Cov}\left(\hat{\mathbf{u}}^{*}_{mit} - \frac{1}{N_m}\sum_{i'',t''}\hat{\mathbf{u}}^{*}_{mi''t''},\ \hat{\mathbf{u}}^{*}_{mi't'} - \frac{1}{N_m}\sum_{i'',t''}\hat{\mathbf{u}}^{*}_{mi''t''}\right)\\ &= \ldots\\ &= \mathrm{Cov}(\hat{\mathbf{u}}^{*}_{m11}, \hat{\mathbf{u}}^{*}_{m21})\left[1 - 2\frac{N_m - 1}{N_m} + \frac{(N_m-1)^2}{N_m^2}\right] + \mathrm{Var}(\hat{\mathbf{u}}^{*}_{m11})\left[-\frac{2}{N_m} + \frac{N_m}{N_m^2}\right]\\ &= -\frac{1}{N_m}\mathrm{Var}\left(\hat{\mathbf{u}}^{*}_{m11}\right)\ \forall\ (i,t) \neq (i',t')\ ,\end{aligned} \tag{B.16}$$

takes the required covariance structure (cf. Sec. 2.2.3.2). Further, the covariance between mean-centered scores of different nodes, say, $\mathcal{B}_m$ and $\mathcal{B}_{m'}$,

$$\begin{aligned}\mathrm{Cov}(\hat{\mathbf{u}}^{**}_{mit}, \hat{\mathbf{u}}^{**}_{m'i't'}) &= \mathrm{Cov}\left(\hat{\mathbf{u}}^{*}_{mit} - \frac{1}{N_m}\sum_{i'',t''}\hat{\mathbf{u}}^{*}_{mi''t''},\ \hat{\mathbf{u}}^{*}_{m'i't'} - \frac{1}{N_{m'}}\sum_{i'',t''}\hat{\mathbf{u}}^{*}_{m'i''t''}\right)\\ &= \ldots = \mathrm{Cov}(\hat{\mathbf{u}}^{*}_{11}, \hat{\mathbf{u}}^{*}_{21})\left[1 - \frac{N_m}{N_m} - \frac{N_{m'}}{N_{m'}} + \frac{N_m N_{m'}}{N_m N_{m'}}\right]\\ &= \mathbf{0}\ ,\end{aligned} \tag{B.17}$$

which implies that the tests on $\mathcal{B}_m$ are independent from tests on $\mathcal{B}_{m'}$.

B.2 Random forest extension

Tree-based algorithms have become popular in particular because they can capture complex interactions and nonlinearities in high-dimensional settings and thus build models with low bias. By contrast, they involve hard decisions with indicators (Bühlmann and Yu, 2002) and, therefore, they can be "instable" regarding small data perturbations. Moreover, their piecewise constant approximation can be inaccurate if relationships are smooth.

Ensemble techniques have proved quite successful to improve the stability and the predictive performance of tree-based algorithms. To date, the standard ensemble techniques may be random forest (Breiman, 2001) and boosting (Freund, 1995). Which technique to use is controversial, empirical comparisons (e.g. Hastie et al., 2001, Chap. 15) suggest that the performance of boosting and random forest is data dependent. For tree-based varying coefficient regression, boosting has been considered by Wang and Hastie (2014).

This supplementary section introduces an implementation of random forest for Algorithm 1. The goal is to demonstrate the potential of the extension to improve the predictive performance of Algorithm 1, which would motivate its comprehensive development.

The section is organized as follows. First, we explain the implementation of random forest for Algorithm 1 and in particular its modifications relative to the implementation of Breiman (2001) for regression trees. Afterwards, we compare the predictive performance of the extension with that of its integrated algorithm, by using the happiness data of Section 2.3.1. The extension is available with the `fvcolmm` function of the R **vcrpart** package (Bürgin, 2015), and the R-code for the empirical evaluation is shown in Appendix B.7.

B.2.1 Method

Random forest for tree-based algorithms combines two techniques: Bootstrap aggregation (or variants of) and random split selection. Bootstrap aggregation (bagging, Breiman,

1996) is an aggregation technique that first fits a collection of models by employing the algorithm on random subsets of the original data, and then aggregates the prediction functions of these fitted models. The aggregated prediction function will stabilize from a sufficiently large number of models, thus, it will have lower variance than individual prediction functions. Moreover, the aggregation will smooth the prediction function without introducing a bias, which often improves their accuracy. For an in-depth study of bagging, see for example Bühlmann and Yu (2002). The technique can be used for various procedures, including linear models, but its variance reduction effect seems particularly strong for instable procedures such as tree-based algorithms.

The second technique, random split selection, modifies the splitting procedure of the tree algorithm in the following way: In each iteration, the set of candidate splits is reduced to the splits corresponding to a randomly selected subset of the partitioning variables. The motivation is the following (cf. Hastie et al., 2001, Chap. 15): The variance of the aggregated prediction function depends on the correlation between the individual models. In cases where the individual models (and their tree structures) are very similar, this correlation is large and, therefore, the variance reduction effect of the aggregation is limited. Random split selection is a proven technique to decrease the correlation between models, which in turn decreases variance of the aggregated prediction functions.

The implementation of random forest for Algorithm 1 requires modifications relative to the implementation for regression trees, which we overview in the following.

Resampling scheme for data subsets Random forest for regression trees generally uses bootstrapping, i.e., observations are randomly selected with replication so that the subsets have the same size as the original data. More recent studies (e.g. Bühlmann and Yu, 2002; Strobl et al., 2007) use subsampling, i.e., a prespecified number of observations is selected by chance and without replication. Bühlmann and Yu (2002) argument that subsampling is computationally cheaper but maintains the accuracy. Our software implementation uses by default subsampling with a selection probability of 0.632, which is the expected percentage of non-replicated observations in bootstrap samples.

Resampling for longitudinal data requires additional considerations. Here, we assume that the data were collected with a simple random sampling scheme on the individual level. Hence, our default scheme randomly selects individuals and then includes all the observations corresponding to the selected individuals. For practical applications, one may has to modify the resampling scheme according to the data collection.

Random split selection Random forest for regression trees performs random split selection by retaining splits corresponding to a randomly selected subset of the partitioning variables. Because Algorithm 1 additionally requires to select a node in each iteration, the random split selection is slightly modified. Specifically, in each iteration, we first investigate for each node the partitioning variables (here moderators) with at least one split that satisfies the minimum node size criterion, and then retain the splits corresponding to a randomly selected subset of the found combinations of nodes and moderators. For example, suppose the current tree structure has three terminal nodes and each terminal node could be splitted by two moderators. In this case, there are six combinations from which to choose, namely moderator one in node one, moderator two in node one, moderator one in node two, and so on. Then, we randomly select a subset of these combinations, e.g., we select moderator two in node one and moderator two in node three.

Aggregating coefficient functions Random forest for regression trees uses the individual models to predict the response, and then takes the average of these predictions. In varying coefficient regression, however, the focus is set on the shape of coefficient functions. To study such shapes, we aggregate the fixed coefficients of the individual models by (Eq. B.18). The variance coefficients of the random effect component, which does not depend on the moderators, are aggregated by using (Eq. B.19). The two aggregations provide estimates for the unknown coefficients of the original target model $\mathcal{M}$ (Eq. 2.1).

Algorithm 4 summarizes the proposed random forest extension for Algorithm 1. The technique first applies a modified version of Algorithm 1 on B random subsets of the data, and then aggregates the coefficient functions to build the varying coefficient model $\mathcal{M}$ (Eq. 2.1). The extension requires three tuning parameters: B , the number of trees (i.e., the number of data subsets); N_0 the minimum node size and; m_{try}, the number of randomly selected combinations of nodes and moderators.

Algorithm 4: Fitting random forest varying coefficients in MGLMMs.

Parameters:

B	number of trees, e.g., $B = 100$
N_0	minimum node size, e.g., $N_0 = 50$
m_{try}	number of randomly selected combinations of nodes and moderators

for $b \leftarrow 1$ ***to*** B **do**

Draw a random data subset $\mathcal{D}_b$ from the total data $\mathcal{D}$ by using cluster-wise subsampling

Fit a tree-based varying coefficient MGLMM of form $\widehat{\mathcal{M}}$ (Eq. 2.4) on $\mathcal{D}_b$ by using Algorithm 1 and the following adjustments:

- Set the stopping parameter α of Algorithm 1 to $\alpha \leftarrow 1$.
- In each iteration, evaluate only the splits corresponding to m_{try} randomly selected combinations of nodes and moderators, $\{\mathcal{B}_m, Z_l\}$, that include at least one split that satisfies N_0.
- Stop if no candidate splits remain.

end

Aggregate the coefficients of the B fitted MGLMMs to form a model by

$$\hat{\boldsymbol{\beta}}(\mathbf{z}_{it}) = \frac{1}{B}\sum_{b=1}^{B}\sum_{m=1}^{M_b}\hat{\boldsymbol{\beta}}_{bm}1(\mathbf{z}_{it} \in \mathcal{B}_{bm}) \text{ and} \tag{B.18}$$

$$\hat{\mathbf{Q}}_b = \frac{1}{B}\sum_{b=1}^{B}\hat{\mathbf{Q}}_b \ . \tag{B.19}$$

Number of trees B The number of trees should be chosen such that the aggregated predictor function is approximately stable, a value which is generally unknown. Oshiro et al. (2012) evaluated this number on random forest for regression trees, and conclude that $B = 128$ is often sufficient. Such a rule of thumb should however be verified for our extension. Our software implementation uses a default of $B = 100$.

Minimum node size N_0 To have sufficient observations to fit the fixed coefficients nodewise, we require a minimum number of observations. For multivariate generalized linear mixed models, $N_0 = 50$ may be a reasonable rule of thumb. Note that, a common default for random forest is $N_0 = 5$. The reason for our larger default value is that the estimation of nodewise fixed coefficients is much more complex than the computation of averages. Moreover, our experiments showed that fitting models with many terminal nodes is often challenging. Hence, for practical reasons, N_0 may be set even larger than $N_0 = 50$. Alternatively, a maximal number of terminal nodes may be set to control the complexity of the resulting models.

Number of randomly selected combinations of nodes and moderators m_{try} The number of randomly selected combinations of nodes and moderators also requires rule of thumbs. For regression trees, such a rule of thumb is the integer number of a third of the number of partitioning variables. This rule could also be applied to our algorithm. The default value of our software implementation is arbitrarily set to $m_{try} = 5$.

B.2.2 Performance study

We evaluated the implemented random forest extension by comparing its predictive performance with that of the integrated Algorithm 1. Specifically, we fitted with both algorithms 250 times the model $\mathcal{M}_1$ (Eq. 2.17) on the happiness data, following the bootstrap scheme of Section 2.3.1. These computations resulted in 250 pairs of negative-likelihood prediction errors for comparison.

For the random forest extension we used the default tuning parameters, that is, $B = 100$, $N_0 = 50$ and $m_{try} = 5$. Additionally we restricted the maximal number of terminal nodes to ten because the estimation often failed when the models became complex. This practical issue may also appear in other applications, and should be discussed in more detail in a comprehensive description of the extension. Thus, the chosen tuning parameters may not be optimal. Even so, the random forest extension outperforms its integrated tree-based algorithm.

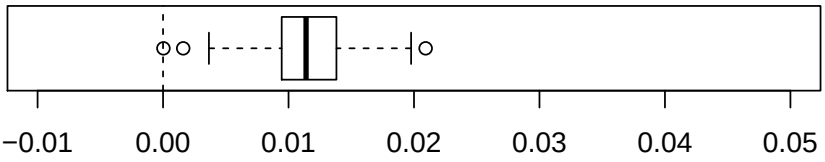

Figure B.1: Boxplot for 250 pairwise differences $\mathrm{err}(\widehat{M}^*_{1k}) - \mathrm{err}(\widehat{M}^*_{1k,rf})$ comparing the prediction error of fits for $\mathcal{M}_1$ (Eq. 2.17) with the tree-based algorithm (Algorithm 1, $\widehat{M}_{1k}$) and its random forest extension (Algorithm 4, $\widehat{M}_{1k,rf}$).

Figure B.1 shows the boxplot of the differences between the 250 computed pairs of prediction errors. It can be seen that fits from Algorithm 1 have, without exception, higher prediction errors than its random forest extension. The median difference of 0.011 is however rather small, a possible reason being the arbitrarily chosen tuning parameters.

B.3 Comparison with MOB

Since our algorithm is a redesign of the *model-based recursive partitioning* algorithm (MOB, Zeileis et al., 2008), it is interesting to compare results of the two algorithms. For this purpose, we compare the tree structure from our Algorithm 1 for the happiness data with the corresponding tree structure from MOB.

The fitting with Algorithm 1 is described in Section 2.3.1, and the resulting tree structure is shown in Figure 2.2 and below in Figure B.2. For the fitting with MOB, we specify as target model the varying coefficient cumulative logit model

$$\mathcal{M}_4 : \text{logit}(P(Y_{it} \leq q)) = \beta_q(\mathbf{z}_{it}) + \text{UE}_{it}\beta_4(\mathbf{z}_{it}) \ . \tag{B.20}$$

By comparison, the target model deployed for Algorithm 1 ($\mathcal{M}_1$, Eq. 2.17) has the same varying coefficient specification as $\mathcal{M}_4$, but it does not include random intercepts. For fitting $\mathcal{M}_4$ with MOB we used the `mob` function of the R **partykit** package (Hothorn and Zeileis, 2014) combined with the `polr` function of the **MASS** package (Venables and Ripley, 2002). The R-code for the fitting is provided in Appendix B.7. We have also considered to include random intercepts by combining `mob` with the `olmm` function of the **vcrpart** package. However, we dropped this idea because the use of mixed models with `mob` seems experimental and has not yet been documented.

Figure B.2 shows in the left panel the tree structure from Algorithm 1 for model $\mathcal{M}_1$ (Eq. 2.17), and in the right panel the tree structure from MOB for model $\mathcal{M}_4$ (Eq. B.20). The principal difference between the two tree structures are the splits for node 6. With our algorithm, node 6 is splitted by *gender*, while MOB uses the moderators *age*, *financial situation* (FISIT), *gender*, *regional unemployment* (UEREG), *household income* (HHINC) and *marital status* (MASTAT) to split node 6 into 7 terminal nodes.

Possible reasons for the differences include the following two: First, we used different target models (model $\mathcal{M}_1$ vs $\mathcal{M}_4$) because of the technical issue explained above. Second, Algorithm 1 integrates the pre-decorrelation transformation of Section 2.2.3.2, and therefore the moderator selection of the two algorithms is technically different.

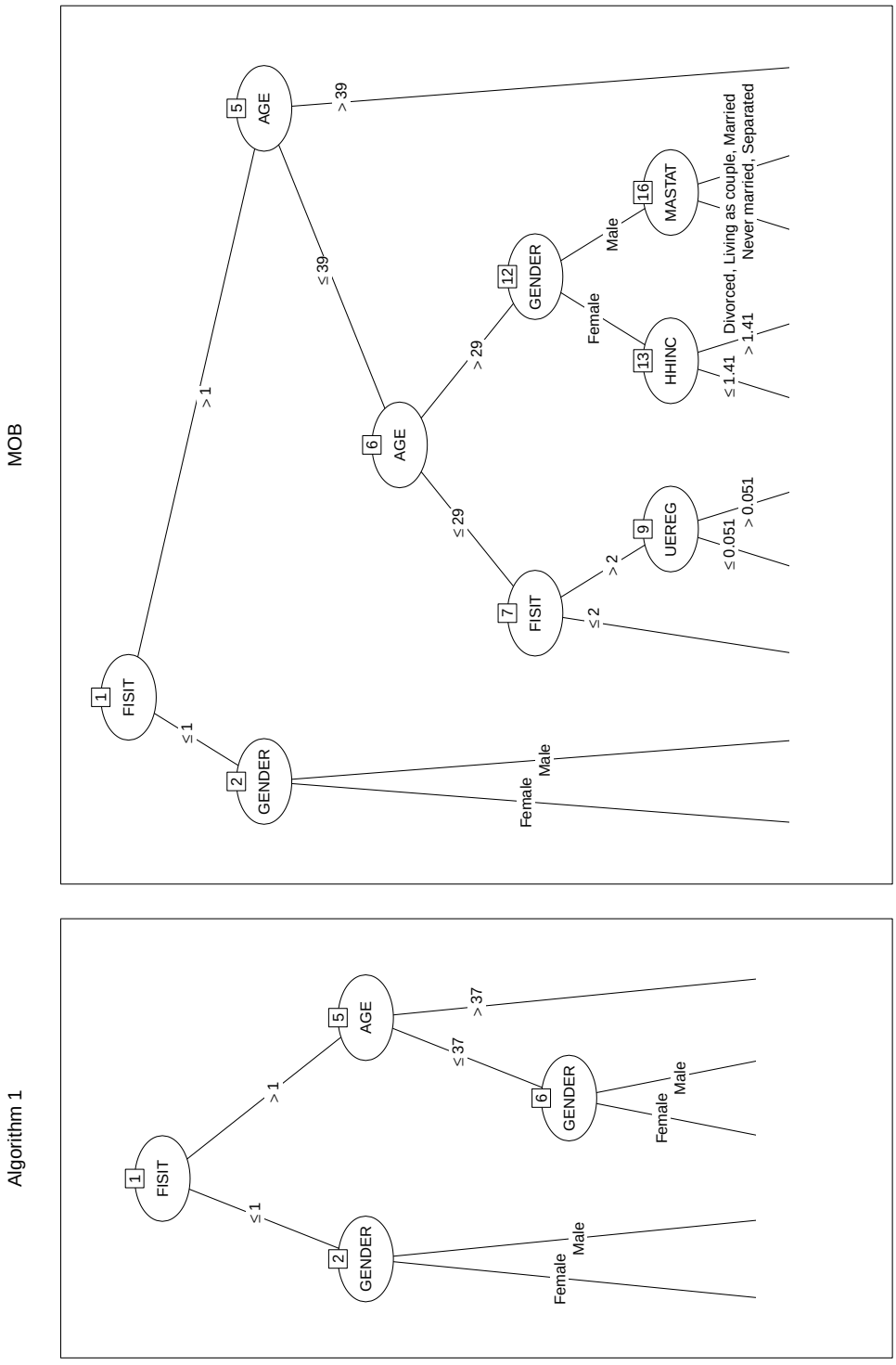

Figure B.2: Comparison of fitted tree structures from Algorithm 1 and MOB for the happiness data. *Left*, the result from Algorithm 1 (pg. 32); *right*, the result from MOB.

B.4 Supplementary simulations

The following considerations extend the simulation studies of the Sections 2.2.3.2 and 2.3.2.1. The most important conclusions are as follows:

- Under coefficient constancy, the variance of standardized cumulative score processes $\breve{\Psi}$ from raw scores depends on the intra-individual correlations in the tested variable. In particular, the deviation from a Brownian bridge increases with the degree of intra-individual correlation. By contrast, the variances of the $\breve{\Psi}$'s from pre-decorrelated scores are fairly close to that of a Brownian bridge.
- Under coefficient constancy, the implemented tests applied on data from random slope models and unbalanced data achieve fairly accurate type I errors.

B.4.1 Variance of standardized cumulative score process

Section 2.2.3.2 motivates the pre-decorrelation adjustment for the score-based coefficient constancy tests for mixed models by considering the variance of computed processes $\breve{\Psi}$ of (Eq. 2.9) for a case example. Here, I extend this study in order to show that the variance of processes from raw scores depends on intra-individual correlations in the tested variable.

In line with Section 2.2.3.2, I repeatedly (5,000 times) generated responses $\mathbf{y}_{it}$ with $i = 1, \ldots, 50$ and $t = 1, \ldots, 10$, from the logistic mixed model:

$$\mathcal{M}_{ex,1} : \text{logit}(P(Y_{it} = 1)) = \beta_0 + b_i, \qquad b_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1),$$

with $\beta_0 = 0$; fitted the true model $\mathcal{M}_{ex,1}$ on these data; and computed $\breve{\Psi}$ from the raw scores $u_{it}(\hat{\beta}_0)$ and from the pre-decorrelated scores $u^*_{it}(\hat{\beta}_0)$. In each iteration, I computed $\breve{\Psi}$ regarding the three variables Z_1, Z_3 and Z_5, which are described in Section 2.3.2. The three variables are continuous and they can be distinguished by their degree of intra-individual correlations (uncorrelated vs correlated vs time-invariant).

Figure B.3 compares the variance of a Brownian bridge with variances of computed processes $\breve{\Psi}(\tau)$. The red solid lines (those closer to the Brownian bridge) correspond to processes from pre-decorrelated scores, and the blue lines to processes from raw scores. The difference between the raw scores and the Brownian bridge increases with the degree of the intra-individual correlations in the variable over which the scores were cumulated. Specifically, for the uncorrelated variable Z_1 (top left), the variance from raw scores and pre-decorrelated scores cover each other. For the time-invariant variable Z_5 (bottom left), the variance is smallest. By looking closely at the plot for Z_5, it can be seen that the variance from raw scores comprises 50 zigzags corresponding to the 50 individuals.

I repeated the simulation study above under the inclusion of random slopes. Specifically, to generate the responses and as the model under H_0, I used the model

$$\mathcal{M}_{ex,2} : \text{logit}(P(Y_{it} = 1)) = \beta_0 + b_{1i} + w_{it} b_{2i}, \qquad \mathbf{b}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_2),$$

with $\beta_0 = 0$. The random slope predictor W consists of intra-individual equidistant sequences between -0.5 and 0.5 of length 10 (the used number of observations per individual). Figure B.4 shows the variances of the computed $\breve{\Psi}(\tau)$. No irregularities appear.

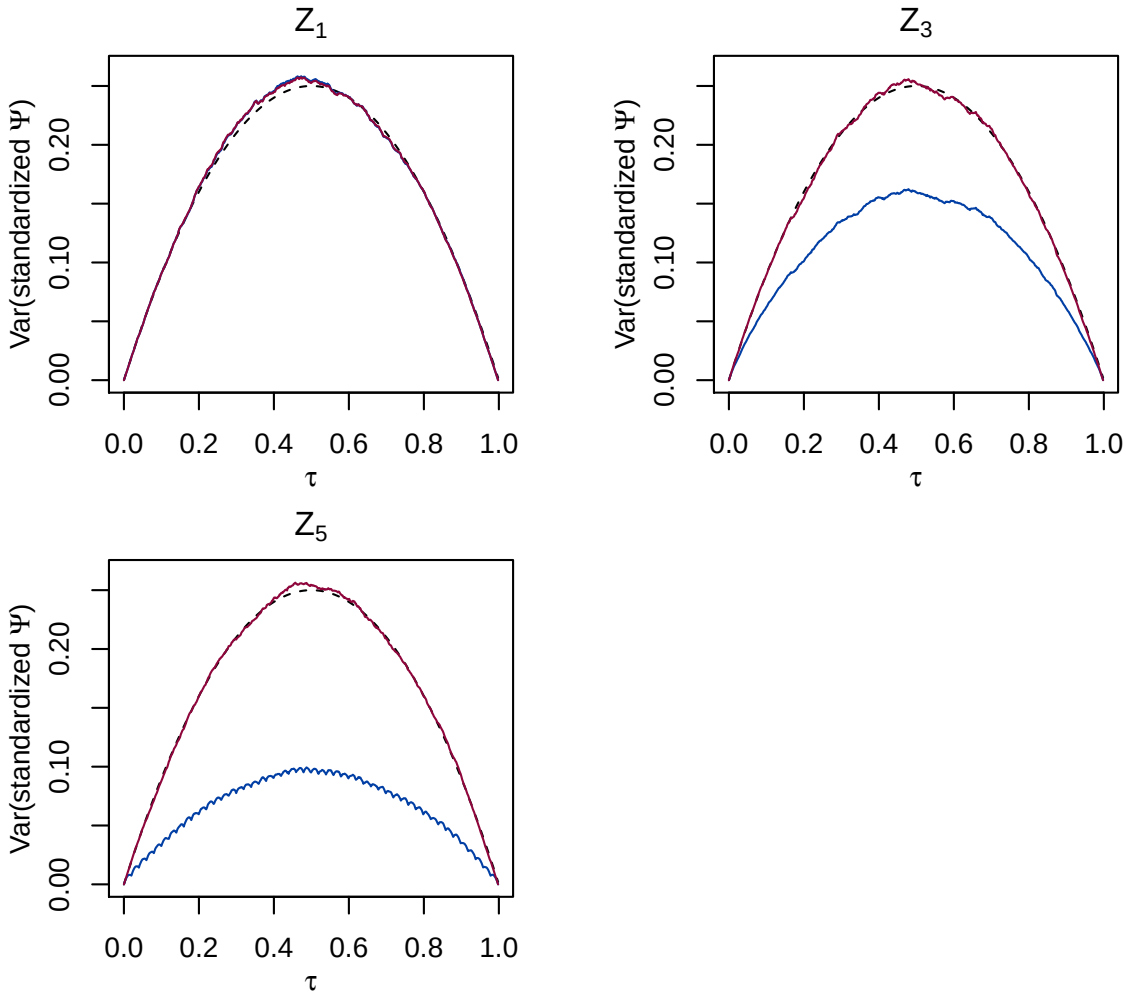

Figure B.3: Variance of standardized cumulative score processes $\breve{\Psi}$ from a logistic random intercept model for the variables Z_1, Z_3 and Z_5 of Section 2.3.2. *Solid lines*, the variance of simulated processes based on the raw scores (blue) and based on the pre-decorrelated scores (red); *dashed lines*, the variance of a Brownian bridge.

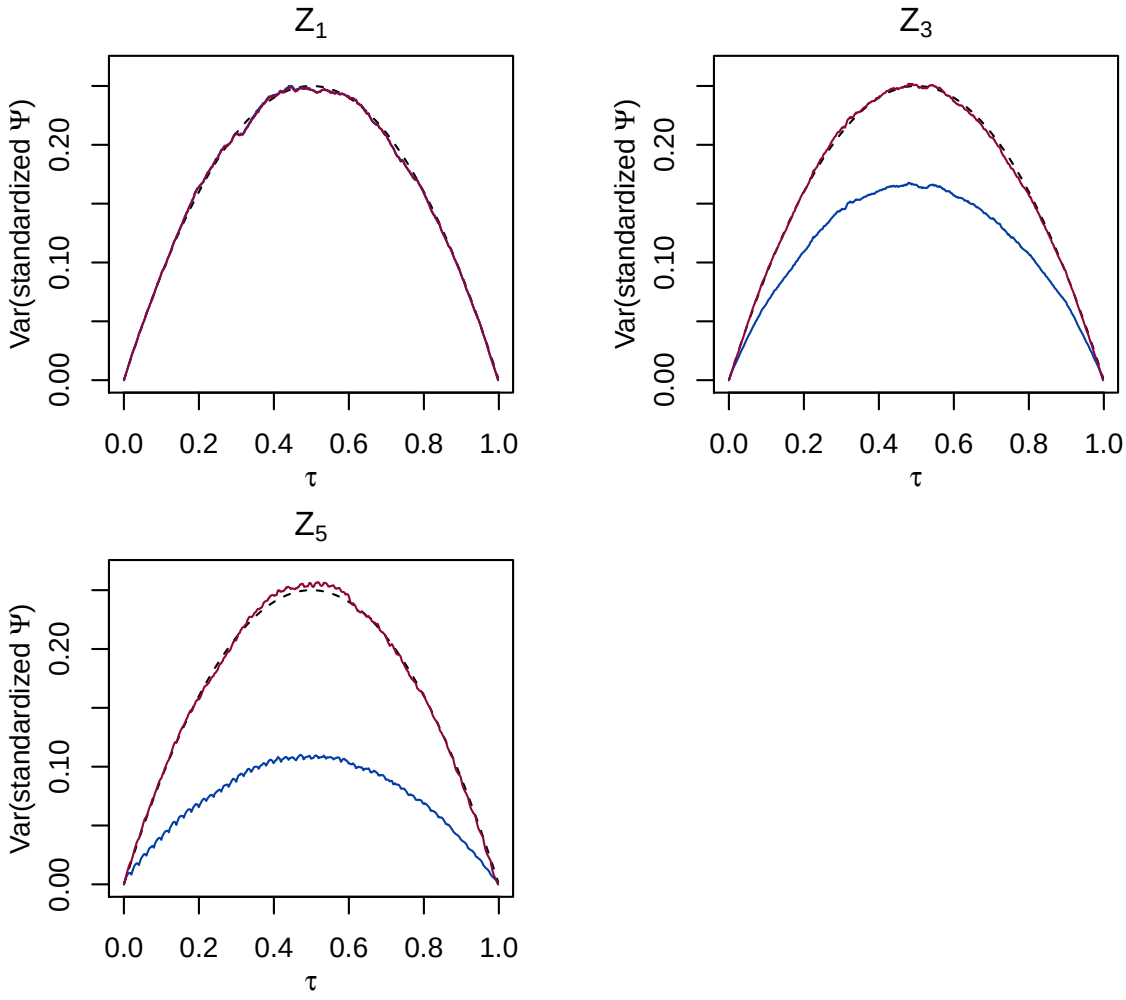

Figure B.4: Variance of standardized cumulative score processes $\breve{\Psi}$ from a logistic random slope model for the variables Z_1, Z_3 and Z_5 of Section 2.3.2. *Solid lines*, the variance of simulated processes based on the raw scores (blue) and based on the pre-decorrelated scores (red); *dashed lines*, the variance of a Brownian bridge.

B.4.2 Type I errors on random slope models

An aspect that was left unconsidered in Section 2.3.2.1 is whether the accuracy of the constancy tests carries over to models with random slopes. Therefore, I replicated the simulation study with responses from the random slope cumulative logit mixed model

$$\mathcal{M}_{\text{sim}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}[\delta \cdot 1_{(z_{lit} \in \mathcal{B}_l)}] + b_{1i} + w_{it} b_{2i}, \ \mathbf{b}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_2).$$

As in Section B.4.1, the predictor variable W consists of intra-individual equidistant sequences between -0.5 and 0.5 of N_i elements. Each scenario was repeated $5,000$ times. Table B.1 reports the resulting type I errors for a nominal level of 5% when using the correctly specified model $\mathcal{M}_{\text{root,1}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}\delta + b_{i1} + w_{it} b_{i2}$ under H_0 (i.e., $\delta = 0$). The results are similar to those in Table 2.2.

Table B.1: Evaluation on random slope models. Relative frequencies of Type I errors in coefficient constancy tests for a nominal level of 5%. Values in brackets correspond to tests without pre-decorrelating the scores.

N/N_i	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
ii-cor	0	0	$\sim 2/3$	$\sim 2/3$	1	1
Scale	cont	cat	cont	cat	cont	cat
50/5	.046 (.048)	.052 (.052)	.039 (.027)	.044 (.034)	.033 (.023)	.039 (.019)
50/10	.054 (.051)	.047 (.048)	.045 (.026)	.045 (.027)	.037 (.018)	.040 (.012)
100/5	.048 (.048)	.051 (.052)	.051 (.032)	.055 (.038)	.043 (.022)	.044 (.019)
100/10	.056 (.057)	.052 (.053)	.052 (.028)	.046 (.028)	.049 (.018)	.041 (.012)
500/5	.052 (.056)	.051 (.050)	.061 (.038)	.045 (.033)	.048 (.022)	.052 (.019)
500/10	.054 (.053)	.049 (.048)	.059 (.031)	.051 (.030)	.054 (.021)	.051 (.013)

I also computed the tests using the misspecified random intercept model $\mathcal{M}_{\text{root,2}} : \text{logit}(P(Y_{it} \leq q)) = \beta_q + x_{it}\delta + b_{i1}$ under H_0. Table B.2 reports the resulting type I errors. The results are fairly comparable with those in Table B.1. This insight does however not mean that a misspecified random coefficient component has no impact on the tests. In particular, here, the random coefficient predictor W is uncorrelated with the tested variables Z_1, Z_3 and Z_5, which may not be realistic. Further research would be necessary to provide deeper insights.

Table B.2: Evaluation on misspecified random intercept models. Relative frequencies of Type I errors in coefficient constancy tests for a nominal level of 5%. Values in brackets correspond to tests without pre-decorrelating the scores.

N/N_i	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
ii-cor	0	0	$\sim 2/3$	$\sim 2/3$	1	1
Scale	cont	cat	cont	cat	cont	cat
50/5	.045 (.045)	.048 (.051)	.037 (.027)	.050 (.035)	.034 (.021)	.041 (.017)
50/10	.049 (.050)	.051 (.050)	.045 (.026)	.046 (.029)	.039 (.019)	.038 (.014)
100/5	.047 (.046)	.049 (.048)	.045 (.031)	.049 (.037)	.050 (.028)	.044 (.018)
100/10	.049 (.049)	.051 (.050)	.051 (.029)	.049 (.033)	.037 (.015)	.046 (.010)
500/5	.054 (.051)	.042 (.045)	.053 (.034)	.049 (.034)	.051 (.024)	.048 (.018)
500/10	.057 (.057)	.048 (.048)	.056 (.027)	.048 (.030)	.054 (.020)	.049 (.012)

B.4.3 Type I errors on unbalanced data

Another aspect that deserves to be studied is the accuracy of the constancy tests applied to unbalanced data. I replicated the simulation study of Section 2.3.2.1 by generating data in which 50% of the individuals have twice as many observations as have the other 50%. Table B.3 reports the resulting type I errors for a nominal level of 5%, when using the correct model under H_0. The values in brackets corresponds to tests without imputation, but with pre-decorrelation. The numbers of observations per individual are indicated in the first column. For example, the results in the first row are based on 25 individuals with 3 observations and further 25 individuals with 6 observations.

The resulting type I errors are fairly close to 5% in most scenarios. For larger numbers of observations the errors tend to exceed the 5% by about 1%. By comparison, the errors with imputation are more accurate than those without.

Table B.3: Evaluation on unbalanced data. Relative frequencies of Type I errors in coefficient constancy tests for a nominal level of 5%. Values in brackets correspond to tests without imputation.

N/N_i	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
ii-cor	0	0	$\sim 2/3$	$\sim 2/3$	1	1
Scale	cont	cat	cont	cat	cont	cat
50/(3,6)	.044 (.044)	.047 (.047)	.042 (.045)	.047 (.050)	.036 (.040)	.055 (.062)
50/(6,12)	.051 (.050)	.047 (.049)	.054 (.055)	.044 (.045)	.044 (.050)	.050 (.054)
100/(3,6)	.056 (.058)	.053 (.053)	.050 (.053)	.046 (.050)	.060 (.063)	.053 (.061)
100(6,12)	.048 (.048)	.051 (.053)	.051 (.041)	.058 (.047)	.045 (.037)	.046 (.040)
500/(3,6)	.053 (.054)	.051 (.050)	.066 (.057)	.051 (.043)	.050 (.046)	.059 (.056)
500/(6,12)	.051 (.051)	.051 (.052)	.061 (.064)	.051 (.051)	.061 (.074)	.065 (.071)

B.5 Q-Q plots of p-values (Sec. 2.3.2.1)

The simulation studies in Section 2.3.2.1 focused on the accuracy of the implemented coefficient constancy tests for the (practically important) nominal level of 5%. For completion, Figure B.5 shows the quantile-quantile (Q-Q) plots for the resulting p-values from simulations with $N_i = 5$ (the number of observations per individual) and on the continuous moderators Z_1, Z_3 and Z_5. The theoretical distribution of the p-values is the uniform distribution for the range zero to one. The figure shows in blue the p-values from the raw scores, and in red the p-values from the pre-decorrelated scores.

It can be seen that the p-values from raw scores are generally to conservative, in particular for the highly intra-individually correlated variable Z_5. The p-values from the pre-decorrelated scores are fairly accurate. The tests are slightly too liberal, which can be seen from the tendency of the red lines to run below the dashed black lines.

The corresponding Q-Q plots for the remaining scenarios, which include results from $N_i = 10$, the variables Z_2, Z_4 and Z_6, the nodewise tests, and the tests in Sections B.4.2 and B.4.3, do not substantially differ from Figure B.5 and are therefore omitted.

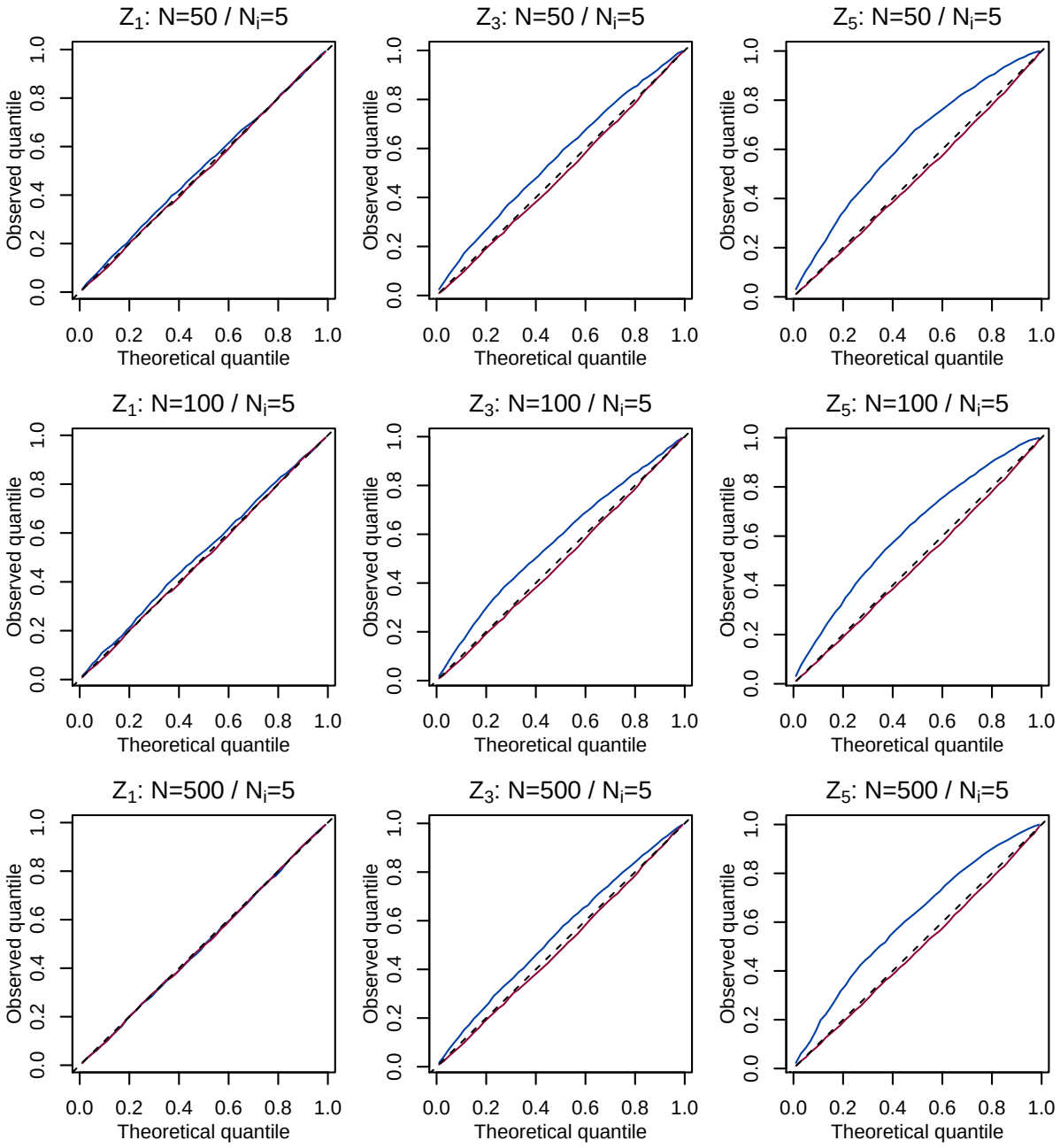

Figure B.5: Q-Q plots for the p-values from the first part of the simulation study of Section 2.3.2.1. *Black, dashed lines*, the theoretical uniform distribution; *blue, solid lines*, the Q-Q lines for p-values from raw scores; *red, solid lines*, the Q-Q lines for p-values from pre-decorrelated scores.

B.6 Details on the happiness data set

B.6.1 Data subset and *happiness* variable

The happiness data set considered in Section 2.3.1 is a subset of the British Household Panel Survey (Taylor et al., 2010) and it is available online from the supplementary materials of Bürgin and Ritschard (2015). The subset includes those respondents who experienced at least one switch from (self-) employment to unemployment between two consecutive waves. More specifically, for each included respondent, we retained a single trajectory formed by the up-to-three-year employment period before the unemployment spell and the up-to-three-year unemployment spell that followed employment.

Figure B.6: *Happiness* trajectories of three respondents. x-axis, the number of years elapsed since the transition to unemployment; y-axis, the *happiness* level.

Figure B.6 shows the *happiness* trajectories of three of the 1,487 included respondents. The trajectories are aligned with the transition to unemployment. They may include gaps. For example, for Respondent 1 there is no observation at one and two years after the transition to unemployment. Either, this respondent has changed the employment status (e.g., found a job), or he or she did not respond to the survey in these years.

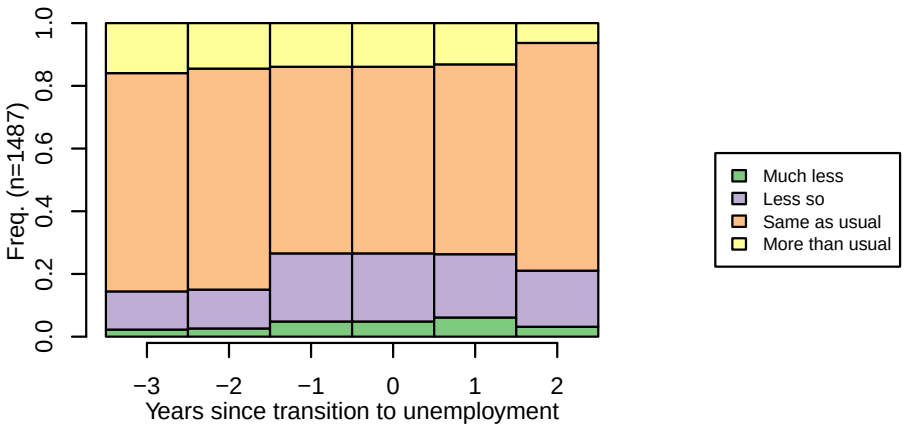

Figure B.7: Cross-sectional distributions of *happiness* across *years since unemployed*.

Figure B.7 shows the cross-sectional distributions of *happiness* (missing values are

ignored). It can be seen that the frequencies of lower happiness levels ("Much less", "Less so") increase already in the year before the transition to unemployment.

B.6.2 Univariate descriptive statistics

The Tables B.4 and B.5 give univariate summary statistics of the variables of the happiness data set. The variables refer to the first year in unemployment.

Table B.4: Descriptive statistics of the nominal and ordinal variables of the happiness data. The variables refer to the first year of the respondents unemployment period.

Variable	**Levels**	**n**	%	∑%
Happiness	Much less	72	4.8	4.8
	Less so	323	21.7	26.6
	Same as usual	885	59.5	86.1
	More than usual	207	13.9	100.0
	all	1487	100.0	
Gender	Female	884	59.5	59.5
	Male	603	40.5	100.0
	all	1487	100.0	
Education	Lower Secondary	417	28.0	28.0
	Upper Secondary	690	46.4	74.4
	Tertiary	380	25.6	100.0
	all	1487	100.0	
Lives with spouse	No	497	33.4	33.4
	Yes	990	66.6	100.0
	all	1487	100.0	
Financial situation	Finding it very difficult	200	13.4	13.4
	Finding it quite difficult	250	16.8	30.3
	Just abt getting by	497	33.4	63.7
	Doing alright	362	24.3	88.0
	Living comfortably	178	12.0	100.0
	all	1487	100.0	
Spouse has job	No	329	22.1	22.1
	No spouse/partner	497	33.4	55.5
	Yes	661	44.5	100.0
	all	1487	100.0	
Marital status	Divorced	75	5.0	5.0
	Living as couple	275	18.5	23.5
	Married	723	48.6	72.1
	Never married	356	23.9	96.1
	Separated	47	3.2	99.2
	Widowed	11	0.7	100.0
	all	1487	100.0	
Head of household	Head of household	751	50.5	50.5
	Not head	736	49.5	100.0

	all	1487	100.0	
Resp. for child < 16	No	1262	84.9	84.9
	Yes	225	15.1	100.0
	all	1487	100.0	

Table B.5: Descriptive statistics of the continuous variables of the happiness data. The variables refer to the first year of the respondents' unemployment period.

Variable	**n**	**Min**	Q_1	**Med**	**Mean**	Q_3	**Max**	**SD**	**IQR**
Age	1487	17.000	26.000	37.000	37.335	48.000	64.000	12.605	22.000
Household income	1487	0.550	1.000	1.230	1.284	1.560	4.650	0.488	0.560
Regional unemployment	1487	0.010	0.031	0.040	0.043	0.051	0.102	0.016	0.020
Sectoral unemployment	1487	0.007	0.027	0.031	0.040	0.052	0.136	0.020	0.025
Number of children	1487	0.000	0.000	0.000	0.643	1.000	6.000	1.034	1.000

Bibliography

Breiman, L. (1996). Bagging Predictors. *Machine Learning 45*(1), 123–140.

Breiman, L. (2001). Random Forests. *Machine Learning 45*(1), 5–32.

Bühlmann, P. and B. Yu (2002). Analyzing Bagging. *The Annals of Statistics 30*(4), 927–961.

Bürgin, R. (2015). *vcrpart: Tree-Based Varying Coefficient Regression for Generalized Linear and Ordinal Mixed Models*. R package version 0.3-3, URL http://cran.r-project.org/web/packages/vcrpart/.

Bürgin, R. and G. Ritschard (2015). Tree-Based Varying-Coefficient Regression for Longitudinal Ordinal Responses. *Computational Statistics & Data Analysis 86*, 65–80. Forthcoming.

Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and computation 121*(2), 1–50.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning* (2 ed.). Springer Series in Statistics. New York, USA: Springer-Verlag.

Hothorn, T. and A. Zeileis (2014). partykit: A Modular Toolkit for Recursive Partytioning in R. In *Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics*, Number 2014-10. Universität Innsbruck.

Oshiro, T. M., P. S. Perez, and J. A. Baranauskas (2012). How Many Trees in a Random Forest? In *Proceedings of the 8th International Conference on Machine Learning and Data Mining in Pattern Recognition*, MLDM'12, Berlin, Germany, pp. 154–168. Springer-Verlag.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn (2007). Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution. *BMC Bioinformatics 8*(25), 1471–2105.

Taylor, M. F., N. B. John Brice, and E. Prentice-Lane (2010). *British Household Panel Survey User Manual Volume A: Introduction, Technical Report and Appendices*. Colchester, UK: University of Essex.

Tutz, G. (2012). *Regression for Categorical Data*. Cambridge Series in Statistical and Probabilistic Mathematics. New York, USA: Cambridge Series in Statistical and Probablilistic Mathematics.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (4 ed.). Statistics and Computing. New York: Springer-Verlag.

Wang, J. C. and T. Hastie (2014). Boosted Varying-Coefficient Regression Models for Product Demand Prediction. *Journal of Computational and Graphical Statistics 23*(2), 361–382.

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

B.7 R-codes

```
## ------------------------------------------------------------ #
## Date:          2015-03-20
## Authors:       Reto Buergin and Gilbert Ritschard
## Institution:   Swiss National Centre of Competence in
##                Research LIVES (http://www.lives-nccr.ch/)
##
## R-code for the real data application in "Tree-based
## varying coefficient regression for ordinal longitudinal
## responses". Code requires vcrpart (>= 0.2-3).
##
## Contents:
## - Data preparation
## - Fitting
## - Bootstrap evaluation
##
## We cannot garantuee that the functions work with future
## versions of R and the indicated packages.
##
## Copyright R. Buergin and G. Ritschard, 2015
## distributed under license Creative Commons BY-NC-SA
## http://creativecommons.org/licenses/by-nc-sa/3.0/
## ------------------------------------------------------------ #

## install.packages("vcrpart")
library("vcrpart")
library("partykit")
library("MASS")
## load("varcoef-applications.RData") # load pre-runned results

## ------------------------------------------------------------ #
## Data preparation
## ------------------------------------------------------------ #

## read the *.csv file
bhps <- read.csv("bhps.csv")

## codings for nominal and ordinal variabls
bhps$PID <- factor(bhps$PID)
levs <-
  c("Much less", "Less so", "Same as usual", "More than usual")
bhps$GHQL <- ordered(bhps$GHQL, levels = levs)
levs <-
  c("Finding it very difficult", "Finding it quite difficult",
    "Just abt getting by", "Doing alright", "Living comfortably")
bhps$FISIT <- ordered(bhps$FISIT, levels = levs)
levs <- c("LowerSecondary", "UpperSecondary", "Tertiary")
bhps$EDU <- ordered(bhps$EDU, levels = levs)

## code ordinal variables with numbers for the labeling
levels(bhps$GHQL) <- 1:nlevels(bhps$GHQL)
levels(bhps$EDU) <- 0:(nlevels(bhps$EDU) - 1)
levels(bhps$FISIT) <- 0:(nlevels(bhps$FISIT) - 1)

## ------------------------------------------------------------ #
## Fitting
```

```
## ------------------------------------------------------------ #

## estimate model M1
## -----------------

## define the vector moderator variables
z <- c("GENDER", "AGE", "EDU", "SPINHH", "HHINC",
       "UEREG", "UESEC", "FISIT", "SPJB", "MASTAT", "HOH",
       "NCHILD", "RACH16")

## set the formula using a 'vc' term
ff.M1 <- GHQL ~ -1 + vc(z, by = UE, intercept = TRUE) + re(1|PID)

## set control argument (here we just enable verbose output)
ctrl.M1 <- tvcolmm_control(seed = 13)

## fit the model
t.M1 <- system.time(
          M1 <- tvcolmm(formula = ff.M1, data = bhps,
                        control = ctrl.M1))[3]


## estimate model M2
## -----------------

## set the formula
ff.M2 <- GHQL ~ UE + re(1|PID)

## fit the model
M2 <- olmm(formula = ff.M2, data = bhps)


## estimate model M3
## -----------------

## data preparation
bhps$AGES <- (bhps$AGE - mean(bhps$AGE)) / sd(bhps$AGE)
bhps$AGES.SQ <- bhps$AGES^2
bhps$BUE <- 1 * (bhps$TUE == -1)
bhps$HHINC.LOG <- log(bhps$HHINC)

## set the formula
ff.M3 <- GHQL ~ GENDER +
    GENDER:(AGES + AGES.SQ + EDU + SPINHH +
    HHINC.LOG + UEREG + UESEC +
    BUE + UE + UE:UEREG + UE:UESEC) + re(1|PID)
con <- list(EDU = contr.treatment(levels(bhps$EDU)))

## fit the model
M3 <- olmm(formula = ff.M3, data = bhps, contrasts = con)


## estimate model M4
## -----------------

bhps$FISIT.C <- factor(bhps$FISIT, ordered = FALSE)

## set the formula
```

```
ff.M4 <- GHQL ~ ce(FISIT.C) + GENDER +
    GENDER:(AGES + AGES.SQ + EDU + SPINHH +
    HHINC.LOG + UEREG + UESEC +
    BUE + UE + UE:UEREG + UE:UESEC) + re(1|PID)
con <- list(EDU = contr.treatment(levels(bhps$EDU)))

## fit the model
M4 <- olmm(formula = ff.M4, data = bhps, contrasts = con)


## estimate model M5 (M1 without random effects, by using MOB)
## ------------------------------------------------------------

## set the formula
ff.M5 <- GHQL ~ UE | GENDER + AGE + EDU + SPINHH + HHINC +
       UEREG + UESEC + FISIT + SPJB + MASTAT + HOH + NCHILD + RACH16

## set control argument (here we just enable verbose output)
ctrl.M5 <- mob_control(verbose = FALSE)

## 'fit' function for 'mob' of partykit package
polr2 <- function(y, x, start, weights, offset, ...) {
  xNames <- colnames(x)
  y <- as.ordered(y)
  xNames <- xNames[xNames != "(Intercept)"]
  formula <-
    as.formula(paste(paste("y ~", paste(xNames, collapse = " + "))))
  data <- data.frame(x[, xNames, drop = FALSE])
  data$y <- y
  call <- list(as.name("polr"),
               formula = formula,
               data = data,
               weights = weights)
  if (!is.null(start)) call$start <- start
  if (!is.null(offset)) call$offset <- offset
  if (length(list(...)) > 0) call <- append(call, list(...))
  mode(call) <- "call"
  rval <- eval(call)
  return(rval)
}

## fit the model
M5 <- mob(formula = ff.M5, data = bhps, control = ctrl.M5, fit = polr2)


## estimate model M6 (random forests)
## ----------------------------------

## set the formula using a 'vc' term
ff.M6 <- GHQL ~ -1 + vc(z, by = UE, intercept = TRUE) + re(1|PID)

## set control argument (here we just enable verbose output)
ctrl.M6 <- fvcolmm_control(vtry = 5, maxwidth = 10, minsize = 50,
                           folds = folds_control("subsampling",
                             K = 100, prob = 0.632),
                           papply.args = list(mc.cores = 12))

## fit the model
```

```
M6 <- fvcolmm(formula = ff.M6, data = bhps,
              control = ctrl.M6)


## ------------------------------------------------------------ #
## Bootstrap evaluation
## ------------------------------------------------------------ #

## functions
## ---------

## function to extract the negative likelihood prediction error
mce <- function(object, newdata) {
  if (inherits(object, "try-error")) return(NA) # error handling
  ids <- levels(droplevels(newdata$PID))
  ranef <- matrix(0, length(ids), 1)
  rownames(ranef) <- ids
  colnames(ranef) <- "(Intercept)"
  if (inherits(object, "tvcm") |
      inherits(object, "olmm") |
      inherits(object, "fvcm")) {
    pred <- predict(object, newdata, ranef = ranef, type = "response")
  }
  return(-sum(log(t(pred)[t(model.matrix(~ -1 + GHQL, newdata)) > 0])))
}

## function to compute the complexity of the model
npar <- function(object) {
  if (inherits(object, "try-error")) return(NA) # error handling
  if (inherits(object, "tvcm")) {

    ## tree-model: number of coefficients + number of splits
    return(extractAIC(extract(object, "model"))[1] +
           sum(sapply(object$info$node, width) - 1))

  }  else {

    ## linear models: number of coefficients
    return(extractAIC(object)[1])
  }
}

## function to define repeatedly selected individuals as different
rel <- function(PID, folds) {
  PID <- as.character(PID)
  PID <-
    paste(PID,
          unlist(sapply(folds, function(x) seq(1, x, length.out = x))),
          sep = ".")
  PID <- factor(PID)
  return(PID)
}

## evaluation
## ----------

nsim <- 250
```

```
## create individual-wise bootstrap folds
folds <- folds_control(type = "bootstrap", K = nsim, seed = 11)
folds <- vcrpart:::tvcm_folds(M1, folds)

e.boot <- c.boot <- matrix(, nsim, 5)

for (i in 1:ncol(folds)) {
  cat("\n\tfold", i, "...")

  ## create training sample
  training <- bhps[rep(1:nrow(bhps), folds[, i]), ]
  training$PID <- rel(training$PID, folds[, i])

  ## create validation sample
  validation <- bhps[folds[, i] == 0, ]

  ctrl.M1$seed <- i

  ## fit the models
  M1boot <- try(tvcolmm(ff.M1, bhps, control = ctrl.M1))
  M2boot <- try(olmm(formula = ff.M2, data = bhps))
  M3boot <- try(olmm(formula = ff.M3, data = bhps, contrasts = con))
  M4boot <- try(olmm(formula = ff.M4, data = bhps, contrasts = con))
  M6boot <- try(fvcolmm(formula = ff.M6, data = bhps,
                        control = ctrl.M6))


  ## negative log-likelihood prediction error
  e.boot[i, ] <- c(mce(M1boot, validation),
                   mce(M2boot, validation),
                   mce(M3boot, validation),
                   mce(M4boot, validation),
                   mce(M6boot, validation))

  ## complexity of model
  c.boot[i, ] <- c(npar(M1boot),
                   npar(M2boot),
                   npar(M3boot),
                   npar(M4boot),
                   NA)

  cat(" OK")

}
```

Appendix C

Supplementary materials: Chapter 3

C.1 Simulation study

The purpose of this supplementary simulation study is to evaluate the coefficient-wise partitioning algorithm on its ability to identify an underlying, data generating varying coefficient model. Two simplifications are made. First, the data generating models are identifiable by recursive partitioning, that is, the considered varying coefficients are all tree-structured. Second, only binary moderators are included so that the model building problem reduces to a variable selection problem. For the evaluation, the coefficient-wise partitioning approach $\widehat{\mathcal{M}}_{\text{tvcm}}$ (Eq. 3.3) is compared with the single partition approach $\widehat{\mathcal{M}}_{\text{tree}}$ (Eq. 3.2), using for both approaches the in Chapter 3 proposed partitioning and tree size selection criteria. The main conclusions of the studies are as follows:

- The performance of coefficient-wise partitioning improves with increasing numbers of observations. Specifically, the chances of identifying the underlying model increase and the chances of selecting "noisy" moderators decrease.
- The results indicate a certain tendency towards overfitting. For sufficiently large numbers of observations, the fitted models practically always include the underlying model as a nested model, but, in about one of five cases, they incorporate at least one noisy moderator.
- Coefficient-wise partitioning outperforms single partitioning if the coefficient functions differ one from another, and (vice versa) the single partition approach outperforms coefficient-wise partitioning if all coefficient functions have the equivalent tree structure.

Generating the simulation data The responses are generated from Gaussian varying coefficient models of form

$$\mathcal{M} : y_i = \beta_0(z_{1i}, \ldots, z_{5i}) + x_i\beta_1(z_{0i}, \ldots, z_{5i}) + \varepsilon_i, \quad \varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1),$$

for $i = 1, \ldots, N$, and (tree-structured) coefficient functions $\beta_0(\cdot)$ and $\beta_1(\cdot)$, as specified below. The predictor, X, is drawn from a standard normal distribution and the six moderators, $Z_0, \ldots, Z_5$, are drawn from a binomial distribution with success probability 0.5. The number of observations, N, is varied between 50 and 500, and each scenario was repeated $2,000$ times.

Specification of the algorithm Two models are fitted on each generated data set, a first that fits separate partitions for each of $\beta_0(\cdot)$ and $\beta_1(\cdot)$, and a second that incorporates a single partition for the two varying coefficients. Models with coefficient-wise partitions are fitted with the command:

```
R> z <- c("z0", "z1", "z2", "z3", "z4", "z5")
R> M1 <- tvcglm(y ~ - 1 + vc(z) + vc(z, by = x),
+               family = gaussian(), data = dat,
+               control = control)
```

with "`z`" a character vector that includes the names of Z_0 to Z_5 in the generated data "`dat`". The single partition approach is fitted with the command:

```
R> M2 <- tvcglm(y ~ - 1 + vc(z, by = x, intercept = TRUE),
+               family = gaussian(), data = dat,
+               control = control)
```

The `control` argument is adjusted according to the sample size. For scenarios including more than 300 observations, the default parameters are used. In scenarios with less than 300 observations, the default parameters would stop the partitioning stage too early. Stopping too early is a disadvantage for finding the true model, but an advantage for not selecting noisy variables. Thus, the principal partitioning parameters (cf. Algorithm 2) N_0 (minimum node size) and D_{min} (minimum training error reduction) are decreased. Specifically, for $N \leq 100$, I used $N_0 = 5$ and $D_{min} = 0.5$, for $100 < N \leq 200$, I used $N_0 = 10$ and $D_{min} = 1$ and for $200 < N \leq 300$, I used $N_0 = 20$ and $D_{min} = 2$.

Performance measures Four measures are considered: (i) *identified underlying model*, the relative frequency of identifying the underlying model exactly; (ii) *model is nested*, the relative frequency of including the underlying model as a nested model, possibly (not necessarily) with additional noisy variables; (iii) *selected the true moderators*, the relative frequency of identifying all moderators that determine the coefficient functions; and (iv) *false variable selections*, the average number of selected noisy moderators.

C.1.1 Coefficient-wise different moderation

In this first scenario, the varying intercept, $\beta_0(\cdot)$, is an indicator function of Z_0; and the varying coefficient of X, $\beta_1(\cdot)$, is an indicator function of Z_1. $Z_2, \ldots, Z_5$ are noisy variables. The data generating model is

$$\mathcal{M}_1 : y_i = -1 + 2 \cdot 1(z_{0i} = 1) + x_i - x_i \cdot 1(z_{1i} = 1) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 1).$$

The scenario is tailored for the coefficient-wise partitioning approach. The single partition approach is not able to identify the structure exactly. Therefore, single partition fits are defined to identify the underlying model exactly if the terminal nodes divide the data into the four strata $\mathcal{B}_1 = \{Z_1 = 0 \cap Z_2 = 0\}$, $\mathcal{B}_2 = \{Z_1 = 0 \cap Z_2 = 1\}$, $\mathcal{B}_3 = \{Z_1 = 1 \cap Z_2 = 0\}$ and $\mathcal{B}_4 = \{Z_1 = 1 \cap Z_2 = 1\}$.

Figure C.1 shows the results on the four performance measures. It can be seen that the relative frequency of identifying the true model increases with larger N's, but stagnates

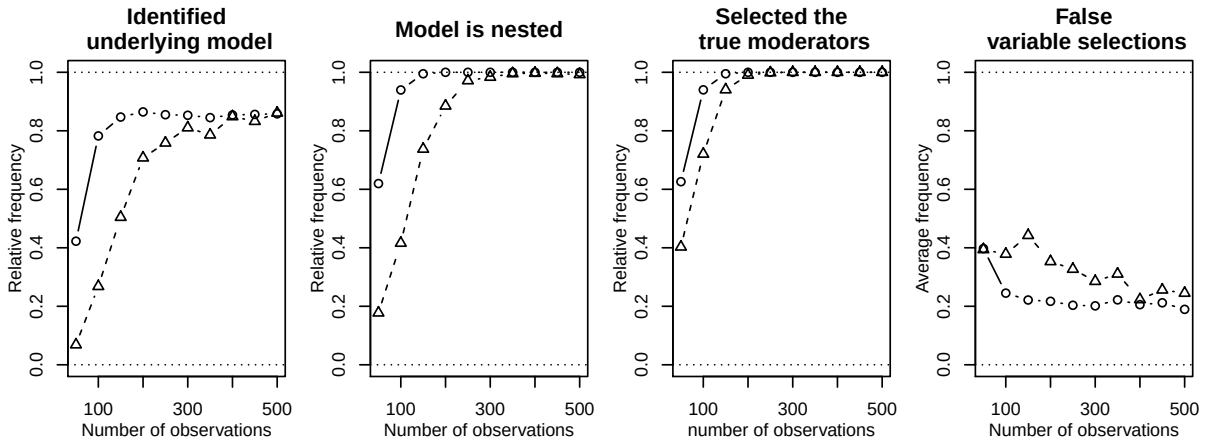

Figure C.1: Coefficient-wise different moderation: *Solid lines*, results for coefficient-wise partitioning; *dashed lines* results for single partitioning.

at about 85%. By contrast, the relative frequencies of including the underlying model as a nested model and of identifying all moderators that determine the coefficient functions reach 100% from about 300 resp. 200 observations. Furthermore, the average number of selected noisy moderators decreases by the number of observations, but stagnates at about 0.2 variables. By comparison, the coefficient-wise partitioning approach (solid lines) outperforms the single partition approach for small N's, but is very similar for large N's.

C.1.2 Single partition moderation

The second scenario is tailored for the single partitioning approach. The varying intercept, $\beta_0(\cdot)$, and the varying coefficient of X, $\beta_1(\cdot)$, are based on a common tree structure. Specifically, I consider the three strata $\mathcal{B}_1 = \{Z_0 = 0\}$ with $(\beta_0, \beta_1) = (-1, -1)$; $\mathcal{B}_2 = \{Z_0 = 1 \cap Z_1 = 0\}$ with $(\beta_0, \beta_1) = (0, 0)$; and $\mathcal{B}_3 = \{Z_0 = 1 \cap Z_1 = 1\}$ with $(\beta_0, \beta_1) = (1, 1)$. Again, $Z_2, \ldots, Z_5$ are noisy variables. The model can be written as

$$\mathcal{M}_2 : y_i = -1(z_{0i} = 0) - 1(z_{0i} = 1 \cap z_{1i} = 1) + \\ x_i(1(z_{1i} = 0) - 1(z_{0i} = 1 \cap z_{1i} = 1)) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 1)$$

The resulting performances are shown in Figure C.2. Like in the first scenario, the performance of both partitioning approaches improves with increasing N's. By contrast, here, the single partition approach outperforms the coefficient-wise partitioning approach. The reason may be the following. The coefficient-wise partitioning approach must find two times the identical tree-structure, which is apparently more challenging than finding it only once as must the single partition approach. A consequence is also that, for small N's, the false selection rate of the coefficient-wise approach is with 0.74 quite high.

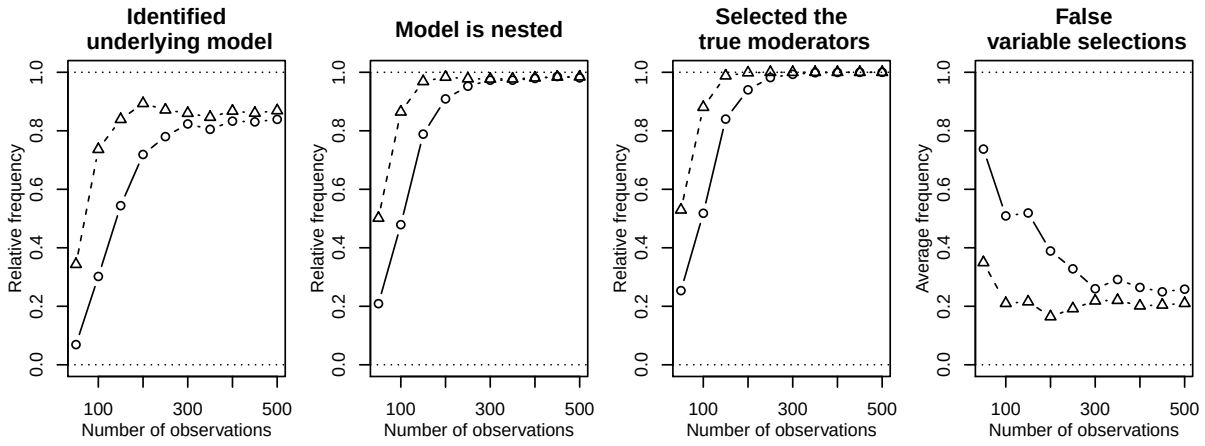

Figure C.2: Single-partition moderation: *Solid lines*, results for coefficient-wise partitioning; *dashed lines* results for single partitioning.

C.2 Details to approximate models of Algorithm 2

C.2.1 Relation between the accurate and the approximate search model

This supplementary section derives and discusses the relation between the accurate search model $\widehat{\mathcal{M}}^*_{kmlj}$ (Eq. 3.9) and the corresponding approximate model $\widehat{\mathcal{M}}_{kmlj}$ (Eq. 3.8). It is shown that $\widehat{\mathcal{M}}^*_{kmlj}$ and $\widehat{\mathcal{M}}_{kmlj}$ have an equivalent structure and with it how the coefficients of the two models relate to each other.

By way of reminder, the "current" model $\widehat{\mathcal{M}}$ (Eq. 3.7) of Algorithm 2 has the form

$$\widehat{\mathcal{M}} : \eta_i = \mathbf{x}_{i0}^\top \boldsymbol{\beta}_0 + \sum_{k=1}^{K} \sum_{m=1}^{M_k} 1(\mathbf{z}_{ik} \in \mathcal{B}_{km}) x_{ik} \beta_{km} \ . \tag{C.1}$$

Now, consider that we search for a binary split for some node $\mathcal{B}_{km}$, by using Z_l as the partitioning variable. We defined in (Eq. 3.9) the accurate search model for the jth split in Z_l as

$$\widehat{\mathcal{M}}^*_{kmlj} : \eta_i^{(s)} = \mathbf{x}_{i0}^\top \boldsymbol{\gamma}_0 + \sum_{(k',m') \neq (k,m)} 1\left(\mathbf{z}_{k'i} \in \mathcal{B}_{k'm'}\right) x_{k'i} \gamma_{k'm'} + \sum_{s=1,2} 1\left(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}\right) x_{ik} \gamma_s \ . \tag{C.2}$$

The model $\widehat{\mathcal{M}}^*_{kmlj}$ simply augments the current model $\widehat{\mathcal{M}}$ by replacing the terms for node $\mathcal{B}_{km}$ by terms for its subnodes $\mathcal{B}_{kmlj1}$ and $\mathcal{B}_{kmlj2}$. To simplify the estimation, Algorithm 2 substitutes the accurate model $\widehat{\mathcal{M}}^*_{kmlj}$ by the approximate model

$$\widehat{\mathcal{M}}_{kmlj} : \eta_i^{(s)} = \hat{\eta}_i + \sum_{s=1}^{2} 1(\mathbf{z}_{ik'} \in \mathcal{B}_{kmljs}) x_{ik'} \beta_s \ . \tag{C.3}$$

The approximate model $\widehat{\mathcal{M}}_{kmlj}$ incorporates as offsets the fitted values $\hat{\eta}_i$ of the current model $\widehat{\mathcal{M}}$. This reduces the complexity of the estimation to the three unknown coefficients β_1, β_2 and ϕ (the dispersion parameter).

From the following decomposition of $\widehat{\mathcal{M}}_{kmlj}$,

$$\begin{aligned}
\eta_i^{(s)} &= \hat{\eta}_i + \sum_{s=1}^{2} 1(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}) x_{ik}\beta_s & \text{(C.4)}\\
&= \mathbf{x}_{i0}^\top \hat{\boldsymbol{\beta}}_0 + \sum_{k'=1}^{K}\sum_{m'=1}^{M_k} 1(\mathbf{z}_{ik'} \in \mathcal{B}_{k'm'}) x_{ik'}\hat{\beta}_{k'm'} + \sum_{s=1}^{2} 1(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}) x_{ik}\beta_s & \text{(C.5)}\\
&= \mathbf{x}_{i0}^\top \hat{\boldsymbol{\beta}}_0 + \sum_{(k',m')\neq(k,m)} 1(\mathbf{z}_{ik'} \in \mathcal{B}_{k'm'}) x_{ik'}\hat{\beta}_{k'm'} + \\
&\quad \sum_{s=1}^{2} 1(\mathbf{z}_{ik} \in \mathcal{B}_{kmljs}) x_{ik}(\hat{\beta}_{km} + \beta_s) \ , & \text{(C.6)}
\end{aligned}$$

it can be seen that the accurate model $\widehat{\mathcal{M}}^*_{kmlj}$ and the approximate model $\widehat{\mathcal{M}}_{kmlj}$ have an equivalent structure. It becomes also apparent that the only differences between the two models are the contrasts for the coefficients of the predictor variable X_k, which relate to each other as

$$\gamma_s = \hat{\beta}_{km} + \beta_s \quad \text{for } s = 1, 2 \ . \tag{C.7}$$

Note that, generally, $\hat{\gamma}_s \neq \hat{\beta}_{km} + \hat{\beta}_s$ because $\widehat{\mathcal{M}}^*_{kmlj}$ includes with the coefficients $\boldsymbol{\gamma}_0$ and $\gamma_{k'm'}$ for $(k', m') \neq (k, m)$ more free parameters.

C.2.2 Approximate model for ordering nominal categories

The technique for ordering the categories of nominal moderators of Section 3.2.2.1 requires us to estimate a model with category-specific coefficients. To simplify the estimation of this model, we use an analogous approximation technique to that used for the search model $\widehat{\mathcal{M}}^*_{kmlj}$ (Eq. 3.9) in Algorithm 2. The goal of this supplementary section is to explain the details of this approximation technique.

Consider that we dispose of the model $\widehat{\mathcal{M}}$ (Eq. 3.7),

$$\widehat{\mathcal{M}} : \eta_i = \mathbf{x}_{i0}^\top \boldsymbol{\beta}_0 + \sum_{k=1}^{K}\sum_{m=1}^{M_k} 1(\mathbf{z}_{ik} \in \mathcal{B}_{km}) x_{ik}\beta_{km} \ . \tag{C.8}$$

The aim is to estimate the category-specific coefficients of a nominal moderator Z_l with categories $1, \ldots, C$ in some node $\mathcal{B}_{km}$. The accurate model would be

$$\begin{aligned}
\widehat{\mathcal{M}}^*_{kml} : \eta_i^{(c)} = \mathbf{x}_{i0}^\top \boldsymbol{\gamma}_0 + \sum_{(k',m')\neq(k,m)} 1\left(\mathbf{z}_{k'i} \in \mathcal{B}_{k'm'}\right) x_{k'i}\gamma_{k'm'} + \\
\sum_{c=1}^{C} 1\left(\mathbf{z}_{ik} \in \mathcal{B}_{km}\right) 1\left(\mathbf{z}_{ikl} = c\right) x_{ik}\gamma_c \ , \quad \text{(C.9)}
\end{aligned}$$

where $\gamma_1, \ldots, \gamma_C$ are the category-specific coefficients. Now, analogous to the approximation of model $\widehat{\mathcal{M}}^*_{kmlj}$ (Eq. 3.9) by the model $\widehat{\mathcal{M}}_{kmlj}$ (Eq. 3.8), we approximate the model $\widehat{\mathcal{M}}^*_{kml}$ by the model

$$\widehat{\mathcal{M}}_{kml} : \eta_i^{(c)} = \hat{\eta}_i + \sum_{c=1}^{C} 1(\mathbf{z}_{ik} \in \mathcal{B}_{km}) 1(z_{ikl} = c) x_{ik'}\beta_c \ . \tag{C.10}$$

In model $\widehat{\mathcal{M}}_{kml}$, $\hat{\eta}_i$ are the fitted values of the current model $\widehat{\mathcal{M}}$. The category-specific coefficients γ_c, $c = 1, \ldots, C$, can then approximately be estimated by

$$\hat{\gamma}_c = \hat{\beta}_{km} + \hat{\beta}_c, \quad \text{for } c = 1, \ldots, C \ . \tag{C.11}$$

This can easily be seen from the derivation shown in (Eq. C.4)–(Eq. C.6).

C.3 Descriptive statistics of the used data sets

The following Tables C.2–C.8 provide univariate descriptive statistics of the variables of the data sets used in Chapter 3. Table C.1 overviews these tables and links them with the corresponding sections.

Table C.1: Overview of tables with descriptive statistics of the used data sets.

Table	Name	Section	Types of variables
C.2	`UCBA` data set	3.2.1	Binary, nominal
C.3	`Pima` data set	3.4.1	Binary
C.4	`Pima` data set	3.4.1	Continuous
C.5	`Schooling` data set	3.4.2	Binary
C.6	`Schooling` data set	3.4.2	Continuous
C.7	`PL` (parental-leave) data	3.4.3	Binary, nominal
C.8	`PL` (parental-leave) data	3.4.3	Continuous

Table C.2: Descriptive statistics of variables of the `UCBA` data set.

Variable	**Levels**	**n**	%	$\sum$ %
Admitted	No	2771	61.2	61.2
	Yes	1755	38.8	100.0
	all	4526	100.0	
Gender	Male	2691	59.5	59.5
	Female	1835	40.5	100.0
	all	4526	100.0	
Department	A	933	20.6	20.6
	B	585	12.9	33.5
	C	918	20.3	53.8
	D	792	17.5	71.3
	E	584	12.9	84.2
	F	714	15.8	100.0
	all	4526	100.0	

Table C.3: Descriptive statistics of the binary variables of the `Pima` data set.

Variable	**Levels**	**n**	%	$\sum$%
Diabetes	Negative	475	65.6	65.6
	Positive	249	34.4	100.0
	all	724	100.0	

Table C.4: Descriptive statistics of the continuous variables of the `Pima` data set.

Variable	**n**	**Min**	Q_1	**Med**	**Mean**	Q_3	**Max**	**SD**	**IQR**
Plasma glucose concentration	724	44.0	99.8	117.0	121.9	142.0	199.0	30.8	42.2
Number of times pregnant	724	0.0	1.0	3.0	3.9	6.0	17.0	3.4	5.0
Diastolic blood pressure	724	24.0	64.0	72.0	72.4	80.0	122.0	12.4	16.0
Body mass index	724	18.2	27.5	32.4	32.5	36.6	67.1	6.9	9.1
Diabetes pedigree function	724	0.1	0.2	0.4	0.5	0.6	2.4	0.3	0.4
Age	724	21.0	24.0	29.0	33.4	41.0	81.0	11.8	17.0

Table C.5: Descriptive statistics of the binary variables of the `Schooling` data set.

Variable	**Levels**	**n**	%	$\sum$ %
Is person black?	No	2307	76.6	76.6
	Yes	703	23.4	100.0
	all	3010	100.0	
Lived with mom/ dad at age 14?	No	634	21.1	21.1
	Yes	2376	78.9	100.0
	all	3010	100.0	
Lived in south in 1966?	No	1763	58.6	58.6
	Yes	1247	41.4	100.0
	all	3010	100.0	
Lived in south in 1976?	No	1795	59.6	59.6
	Yes	1215	40.4	100.0
	all	3010	100.0	
Enrolled in 1976	No	2732	90.8	90.8
	Yes	278	9.2	100.0
	all	3010	100.0	
Lived in smsa in 1976?	No	864	28.7	28.7
	Yes	2146	71.3	100.0
	all	3010	100.0	
Grew up near 4-yr college?	No	957	31.8	31.8
	Yes	2053	68.2	100.0
	all	3010	100.0	

Table C.6: Descriptive statistics of the continuous variables of the `Schooling` data set.

Variable	**n**	**Min**	Q_1	**Med**	**Mean**	Q_3	**Max**	**SD**	**IQR**
Logarithm of wage per hour 1976	3010	4.6	6.0	6.3	6.3	6.6	7.8	0.4	0.6
Education in 1976	3010	1.0	12.0	13.0	13.3	16.0	18.0	2.7	4.0
Working experience in 1976	3010	0.0	6.0	8.0	8.9	11.0	23.0	4.1	5.0
Age in 1976	3010	24.0	25.0	28.0	28.1	31.0	34.0	3.1	6.0
Mother-father education class	3010	1.0	3.0	6.0	5.9	8.0	9.0	2.6	5.0

Table C.7: Descriptive statistics of the binary and nominal variables of the PL data.

Variable	**Levels**	**n**	%	∑%
Returned to work	No	944	15.3	15.3
	Yes	5236	84.7	100.0
	all	6180	100.0	
Whether childbirth was in July	June	2955	47.8	47.8
	July	3225	52.2	100.0
	all	6180	100.0	
Whether white collar worker	No	3475	56.2	56.2
	Yes	2705	43.8	100.0
	all	6180	100.0	
Age	15-19	601	9.7	9.7
	20-24	2675	43.3	53.0
	25-29	2130	34.5	87.5
	30-34	607	9.8	97.3
	35-44	167	2.7	100.0
	all	6180	100.0	
Industry	Other	1329	21.5	21.5
	Retail	777	12.6	34.1
	Social Insurance	743	12.0	46.1
	Hotels	547	8.8	54.9
	Health	395	6.4	61.3
	Wholesale	385	6.2	67.6
	Clothes	248	4.0	71.6
	Hygiene	244	4.0	75.5
	Food	213	3.5	79.0
	Banks	189	3.1	82.0
	Law	175	2.8	84.9
	Education	153	2.5	87.3
	Electrical	139	2.2	89.6
	Construction	114	1.8	91.4
	Metal	106	1.7	93.2
	Text	102	1.6	94.8
	PaperPrint	87	1.4	96.2
	Wood	85	1.4	97.6
	Chemicals	80	1.3	98.9
	Leather	69	1.1	100.0
	all	6180	100.0	
Region	Vienna	2001	32.4	32.4
	Upper Austria	977	15.8	48.2
	Lower Austria	802	13.0	61.2
	Styria	727	11.8	72.9
	Tyrol	478	7.7	80.7
	Carinthia	428	6.9	87.6
	Salzburg	406	6.6	94.2
	Voralberg	243	3.9	98.1
	Burgenland	118	1.9	100.0
	all	6180	100.0	

Table C.8: Descriptive statistics of the continuous variables of the PL data set.

Variable	**n**	**Min**	Q_1	**Med**	**Mean**	Q_3	**Max**	**SD**	**IQR**
Years employed before birth	6180	0	2.9	5.3	5.8	8.2	17.6	3.9	5.3
Years unemployed before birth	6180	0	0.0	0.0	0.3	0.3	5.8	0.5	0.3
Daily earnings at birth	6180	0	23.2	30.5	33.9	40.2	1510.7	40.5	17.0
Daily earnings 1989	6180	0	26.7	37.1	36.5	48.9	98.6	20.7	22.1

C.4 R-codes

```
## ------------------------------------------------------------ #
## Date:          2015-03-20
## Authors:       Reto Buergin and Gilbert Ritschard
## Institution:   Swiss National Centre of Competence in
##                Research LIVES (http://www.lives-nccr.ch/)
##
## R-code real data applications in "Coefficient-wise
## tree-based varying coefficient regression with
## R-package 'vcrpart'. Code requires vcrpart (>= 0.2-3),
## mlbench (>= 2.1-1) and Ecdat (>= 0.2-7). Running the
## entire code will take about two hours.
##
## Contents:
## - Load required packages
## - UCBA data
## - Pima data
## - Schooling data
## - Parental-leave data
##
## We cannot garantuee that the functions work with future
## versions of R and the indicated packages.
##
## Copyright R. Buergin and G. Ritschard, 2015
## distributed under license Creative Commons BY-NC-SA
## http://creativecommons.org/licenses/by-nc-sa/3.0/
## ------------------------------------------------------------ #

## ------------------------------------------------------------ #
## Load required packages
## ------------------------------------------------------------ #

## install.packages(c("vcrpart", "mlbench", "Ecdat"))
library("vcrpart") # fitting functions
library("mlbench") # Pima data
library("Ecdat") # Schooling data
## load("vcrpart-applications.RData") # load pre-runned results

## ------------------------------------------------------------ #
## UCBA data
## ------------------------------------------------------------ #

## load the data
data("UCBAdmissions")
UCBA <- as.data.frame(UCBAdmissions)
UCBA$Admit <- 1 * (UCBA$Admit == "Admitted")
UCBA$Female <- 1 * (UCBA$Gender == "Female")
head(UCBA, 3)

## fit the two linear models
## -------------------------

## fit the basis model with only 'Female' in the predictor
glmS.UCBA <- glm(formula = Admit ~ Female, data = UCBA,
                 family = binomial(), weights = UCBA$Freq)
summary(glmS.UCBA)
```

```
## fit the extended model with Department-wise effects
glmL.UCBA <- glm(formula = Admit ~ -1 + Dept + Dept:Female,
                 data = UCBA, family = binomial(),
                 weights = UCBA$Freq)
summary(glmL.UCBA)

## partitioning
## ------------

vcmL.UCBA <- tvcglm(Admit ~ -1 + vc(Dept) + vc(Dept, by = Female),
                    data = UCBA, family = binomial(),
                    weights = UCBA$Freq,
                    control = tvcglm_control(minsize = 30,
                      mindev = 0.0, cv = FALSE))

## decision tree plots
plot(vcmL.UCBA, type = "coef", part = "A", tnex = 3)
plot(vcmL.UCBA, type = "coef", part = "B", tnex = 3)

## split path (first iteration)
splitpath(vcmL.UCBA, steps = 1, details = TRUE)

## computational details: splits in nominal moderators
glmCW.UCBA <- glm(formula = Admit ~ - 1 + Dept:Female,
                  family = binomial(),
                  data = UCBA,
                  weights = UCBA$Freq,
                  offset = predict(glmS.UCBA))
coef(glmCW.UCBA)

## pruning
## -------

## prune with lambda = 6 (fixed)
vcm.UCBA <- prune(vcmL.UCBA, cp = 6)

## show the prune path (first iteration)
prunepath(vcm.UCBA, steps = 1)

## cross-validation
cv.UCBA <- cvloss(vcmL.UCBA,
                  folds = folds_control(weights = "freq",
                    seed = 13))
plot(cv.UCBA)

## prune with 'lambda' from cross-validation
vcm.UCBA <- prune(vcmL.UCBA, cp = cv.UCBA$cp.hat)

## decision tree plots
plot(vcm.UCBA, type = "coef", part = "A")
plot(vcm.UCBA, type = "coef", part = "B")


## ------------------------------------------------------- #
## Pima data
## ------------------------------------------------------- #
```

```
## the general control parameter object
control <- tvcglm_control(folds = folds_control(seed = 13))

## load and prepare the data
data("PimaIndiansDiabetes2")
Pima <- na.omit(PimaIndiansDiabetes2[, -c(4, 5)])

## model with multivariate coefficient functions (slow)
vcm.Pima.1 <-
  tvcglm(diabetes ~ -1 + vc(pregnant, pressure, mass, pedigree, age) +
         vc(pregnant, pressure, mass, pedigree, age, by = glucose),
         data = Pima, family = binomial(), control = control)

summary(vcm.Pima.1)
plot(vcm.Pima.1, type = "coef", part = "A")
plot(vcm.Pima.1, type = "coef", part = "B")

## model with additive coefficient functions (slow)
vcm.Pima.2 <-
  tvcglm(diabetes ~ 1 + glucose +
         vc(pregnant) + vc(pregnant, by = glucose) +
         vc(pressure) + vc(pressure, by = glucose) +
         vc(mass) + vc(mass, by = glucose) +
         vc(pedigree) + vc(pedigree, by = glucose) +
         vc(age) + vc(age, by = glucose),
         data = Pima, family = binomial(), control = control)
summary(vcm.Pima.2)

## model from MOB algorithm
mob.Pima.1 <-
  glmtree(diabetes ~ glucose | pregnant + pressure + mass +
           pedigree + age, data = Pima, family = binomial())

summary(mob.Pima.1)
plot(mob.Pima.1)
## note: the plot in the article was constructed manually

## ------------------------------------------------------------ #
## Schooling data
## ------------------------------------------------------------ #

## load and prepare the data
data("Schooling")
Schooling <- Schooling[c(19, 21, 7, 28, 9, 14, 17, 18, 20, 23, 2, 4)]
Schooling$black <- 1 * (Schooling$black == "yes")

## construct the instrumental variable
Schooling$ed76.IV <- fitted(lm(ed76 ~ nearc4, Schooling))

## fit the basis model
lm.School <- lm(lwage76 ~ ed76.IV + exp76 + I(exp76^2) + black,
                data = Schooling)

## fit the TVCM model (slow)
f.School <- lwage76 ~ -1 + ed76.IV + exp76 + I(exp76^2) +
  vc(age76, momdad14, south66, south76, famed, enroll76, smsa76) +
  vc(ed76.IV, exp76, age76, momdad14, south66,
     south76, famed, enroll76, smsa76, by = black)
```

```
vcm.School <- tvcglm(formula = f.School, data = Schooling,
                     family = gaussian(), control = control)
summary(vcm.School)

## plot the cross-validated error
plot(vcm.School, "cv")

## plot the tree-structure
plot(vcm.School, "coef", part = "A", tnex = 3)
plot(vcm.School, "coef", part = "B", tnex = 3)

## show all coefficients
coef(vcm.School)


## ------------------------------------------------------------ #
## Parental-leave data
## ------------------------------------------------------------ #

data("PL")

## fit the basic model
glm.PL <- glm(uncj10 ~ july, data = PL, family = binomial)
summary(glm.PL)

## specify the formula
f.PL <- uncj10  ~  1 + july +
    vc(age) + vc(age, by = july) +
    vc(workExp) + vc(workExp, by = july) +
    vc(unEmpl) + vc(unEmpl, by = july) +
    vc(laborEarnings) + vc(laborEarnings, by = july) +
    vc(whiteCollar) + vc(whiteCollar, by = july) +
    vc(wage) + vc(wage, by = july) +
    vc(industry.SL) + vc(industry.SL, by = july) +
    vc(region.SL) + vc(region.SL, by = july)

## fit the TVCM model (slow)
vcm.PL <- tvcglm(formula = f.PL, family = binomial(),
                 data = PL, control = control)

summary(vcm.PL)
plot(vcm.PL, "cv")

## plot the tree-structures with at least one split
par(ask = TRUE)
which <- which(sapply(vcm.PL$info$node, width) > 1)
for (i in which) plot(vcm.PL, "coef", tnex = 2, part = i)
```

Appendix D

Publications

List of publications with peer review during the dissertation, in their chronological order of publication:

- A. Bariska and R. Bürgin. Case Study of Likelihood and Bayes Approaches for Measurement Based on Nonlinear Regression. In F. Pavese, M. Bär, J.-R. Filtz, A. B. Forbes, L. Pendrill, and K. Shirono, editors, *Advanced Mathematical And Computational Tools In Metrology And Testing IX*, volume 84 of *Series on Advances in Mathematics for Applied Sciences*, chapter 2, pages 27–34. World Scientific, 2012

- R. Bürgin, G. Ritschard, and E. Rousseaux. Visualisation de Séquences d'Événements. *Revue des Nouvelles Technologies de l'Information, Extraction et gestion des connaissances (RNTI-E)*, 23:559–560, 2012

- G. Ritschard, R. Bürgin, and M. Studer. Exploratory Mining of Life Event Histories. In J. J. McArdle and G. Ritschard, editors, *Contemporary Issues in Exploratory Data Mining in Behavioral Sciences*, chapter 9, pages 221–253. Routledge, 2013

- R. Bürgin and G. Ritschard. Rendering the Order of Life Events. In *LIVES Working Papers*, pages 1–16. NCCR LIVES, 2013. doi: 10.12682/lives.2296-1658.2013.29

- D. Baumberger, R. Bürgin, and S. Bartholomeyczik. Streuung des Pflegeaufwandes in SwissDRG-Fallgruppen. *PFLEGE*, 27(2):105–115, 2014

- R. Bürgin and G. Ritschard. A Decorated Parallel Coordinate Plot for Categorical Longitudinal Data. *The American Statistician*, 68(2):98–103, 2014

- M.-M. Jeitziner, S. Zwakhalen, R. Bürgin, V. Hantikainen, and J. Hamers. Long-Term Consequences of Pain, Anxiety and Agitation for Critically Ill Older Patients After an Intensive Care Unit Stay. *Journal of Clinical Nursing*, 2015. Forthcoming

- R. Bürgin and G. Ritschard. Tree-Based Varying-Coefficient Regression for Longitudinal Ordinal Responses. *Computational Statistics & Data Analysis*, 86:65–80, 2015