On approaches for clustering longitudinal data

On approaches for clustering longitudinal data

WITH EXTENSIONS FOR MODELING THERAPY ADHERENCE OF SLEEP APNEA PATIENTS

Niek Den Teuling

On approaches for clustering longitudinal data
With extensions for modeling therapy adherence of sleep apnea patients

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr. S.K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op woensdag 5 juli 2023 om 13:30 uur

door

Nicolaas Gregorius Petrus Den Teuling

geboren te Brunssum

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

Voorzitter:
prof.dr. M.G.J. van den Brand

Promotoren:
prof.dr. E.R. van den Heuvel
prof.dr. S.C. Pauws (Tilburg University)

Promotiecommissieleden:
prof.dr. J.K. Vermunt (Tilburg University)
prof.dr. M.E. Timmerman (Rijksuniversiteit Groningen)
prof.dr. G. Molenberghs (Universiteit Hasselt)
prof.dr. M. Pechenizkiy

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

On approaches for clustering longitudinal data

With extensions for modeling therapy adherence of sleep apnea patients

Niek Den Teuling

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-5786-8

The research presented in this thesis was supported by Philips Research, Eindhoven, the Netherlands.

Cover design: Niek Den Teuling
Printed by: ADC Dereumaux

Contents

Chapter 1

General introduction

1.1 Longitudinal data

The use of longitudinal data is key in the fields of psychology, sociology, medicine, and others. Through measuring subjects repeatedly at different moments in time, longitudinal studies enable researchers to assess changes or developments in subjects. The change on a response variable of interest can be assessed over any time scale, limited only by the frequency of measurement. Here, subjects may refer to an individual person being a patient or study participant, but also more higher-level groupings such as a college classes, schools, cities, or even countries. Consider, for example, a longitudinal study for investigating the daily hours of positive airway pressure (PAP) therapy usage of patients suffering from sleep apnea. Patients are recommended to use this therapy during sleep for at least four hours per night. Every patient is different and can be expected to have slight differences in their average usage over time, referred to as population heterogeneity. Heterogeneity may occur in the mean level, the change over time, the variability, or other longitudinal characteristics of interest. Researchers may be interested in learning about the general trend of daily therapy usage, but also how patients deviate from the trend or how the hours of usage vary on a day-to-day basis.

Throughout the past two centuries, longitudinal analyses have been explored and approached in different ways. During the 19th century, researchers sought to describe populations using general laws. The focus was on describing change over time in terms of a general trajectory that holds for all individuals of the population, ignoring variability among patients (Gompertz, 1820; Verhulst, 1845). This methodology continued into the early 20th century, during which change was described through increasingly complex trajectory models. In later research, the notion that each subject is different and therefore can be expected to exhibit a different level of change over time, was taken into account. In early examples of such analyses, differences between subjects were addressed by fitting individual curves (Wishart, 1938; Bollen and Curran, 2006).

The analysis of longitudinal datasets progressed substantially halfway into the 20th century. In many longitudinal studies, subjects are hypothesized to approximately follow a general trend, with random deviations from that trend (Hamaker, 2012). Starting in the sixties, various statistical models have been developed for modeling such heterogeneity. Notable examples are multilevel modeling, as developed under different names across fields

(hierarchical linear modeling, mixed modeling, random coefficient modeling) (Bryk and Raudenbush, 1987), and latent curve modeling (Bollen and Curran, 2006). Statistical software for estimating these models became more commonly available in the nineties (Bryk and Raudenbush, 1987).

1.2 Longitudinal cluster analysis

Instead of identifying one general trend from the data, researchers began extending longitudinal methods to identify multiple common trends from the data (Nagin and Land, 1993; Muthén and Shedden, 1999). This is more broadly referred to as clustering longitudinal data. The rapid advancement in computational power throughout the past decades has enabled researchers to estimate increasingly extensive temporal models on larger datasets. Instead of identifying one general trend from the data, researchers began extending temporal methods to identify groups of subjects with a similar temporal pattern. The development of models and algorithms for automatically clustering temporal data started to take shape during the late eighties and nineties across many research domains. An early example of time series clustering is the work of Košmelj and Batagelj on applying cross-sectional cluster algorithms to time series data (Košmelj, 1986; Košmelj and Batagelj, 1990). Nagin and Land (1993) proposed to model the heterogeneity using a mixture of linear regression models. Not long thereafter, Muthén and Shedden (1999) proposed a multilevel mixture model comprising heterogeneous clusters.

Clustering longitudinal data involves the automatic discovery of groups of subjects who follow a similar longitudinal pattern over time. The population is then represented by several common trends instead of a single general trend, with each trend representing a proportion of the population heterogeneity. In the case of PAP therapy adherence, researchers are interested in identifying various groups of patients who used the therapy in a similar way over time. The discovery of distinct groups of patients could help in devising more effective interventions tailored to a specific group of patients.

As is often the case, the population under investigation might not be composed of distinct groups. Instead, differences between subjects may be the result of a complex interaction of several possibly unobserved factors or even unknown factors. In these cases, clustering is a valuable and pragmatic tool, as it provides a flexible approach to representing the population through a finite number of clusters. This yields a more detailed and meaningful description of the data than one would obtain from a single general trend. When it comes to PAP therapy adherence, the population heterogeneity is indeed attributable to a multitude of underlying factors. For example, the adherence to therapy differs considerably between patients over time due to a large number of behavioral, therapy-related, support and environmental factors (e.g., motivation, perceived importance, and support of healthcare professionals and family (Cayanan *et al.*, 2019; Shapiro and Shapiro, 2010)). Several studies have identified patient clusters with different patterns of therapy adherence (Aloia *et al.*, 2008; Babbin *et al.*, 2015; Wohlgemuth *et al.*, 2015).

1.3 Challenges

We see the following challenges in longitudinal clustering: the cross-domain disconnect, the comparison of methods, computational effort, and jointly accounting for other distributional parameters over time. The literature on the topic of longitudinal clustering is rather disconnected across different areas of research such as latent class analysis originating from structural equation modeling, multilevel mixture modeling, and more traditional cluster methods originating from the field of machine learning. This presents a challenge to researchers who are searching the literature for applicable methods for their case study, as it requires a familiarity with the different terminology being used between the fields. Unifying the terminology would help with this, and so would the increased availability of review papers and comparative studies that bridge the gap between the fields. Unfortunately, there are relatively few cross-disciplinary comparisons of methods being done, with most comparison papers evaluating the state-of-the-art methods within the respective field. Currently, researchers would need to learn and combine different software packages to conduct a cross-disciplinary comparison, which can require considerable effort. A unified statistical software package incorporating several longitudinal cluster approaches could be an instrument to bring together methods across domains. This would make it easier to conduct comparison studies, as well as enable researchers to easily compare and evaluate methods for their domain or specific case study.

The automatic identification of unobserved clusters brings about computational challenges. With each additional cluster that needs to be estimated, the number of possible permutations by which subjects can be partitioned goes up drastically. Similarly, for model-based approaches that estimate a parametric representation for each cluster, the number of parameters and number of subject evaluations goes up with the number of clusters, resulting in quadratic scaling. It is generally computationally infeasible to identify the optimal solution during the estimation procedure for a dataset of any meaningful size. Instead, cluster methods may be estimated repeatedly from different starting points in an attempt to discover a better solution. Depending on the sample size, the computational scaling may place a practical limitation on the number of clusters than can be estimated within a reasonable amount of time. Moreover, the increasing complexity of such models may result in a reduced convergence rate, therefore requiring an increasing number of repeated estimations, further increasing the total time needed to identify a good solution.

In most applications of longitudinal clustering, there is an emphasis on only modeling the mean response trajectories. However, under the expectation of subject heterogeneity, accounting for heterogeneity in the variance or other aspects may prove equally important to the identification of longitudinal clusters. Modeling the within-subject variance is of particular interest, as it is often assumed to be constant over time and equal for subjects. In the case of repeated measures data involving counts or time durations, there may be an excessive number of zeros which should be modeled separately. Here, there may be heterogeneity in the occurrence of zeros over time or between subjects, which could be addressed through clustering.

1.4 Aim of the thesis

In this thesis, we aim to make the area of longitudinal clustering more accessible to applied researchers by reducing the disconnect between the different areas of research on clustering

longitudinal data. We achieve this by providing a cross-disciplinary review, an extensive comparison of different approaches to longitudinal clustering, and implementing statistical software that facilitates a variety of longitudinal clustering approaches in a standardized way. Secondly, we investigate how to jointly account for heterogeneity on other longitudinal aspects in addition to the mean response. The proposed model extensions are applied to two real-world datasets described below. We pay attention to the practical aspects of the application of the different methods, in terms of flexibility, robustness, and computational intensiveness.

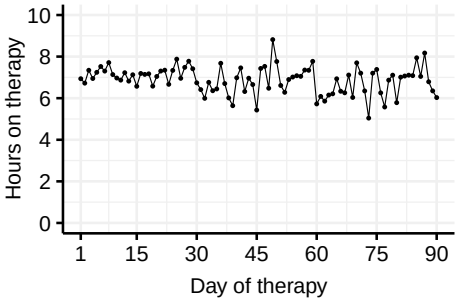
1.4.1 Exploring heterogeneity in PAP therapy adherence

The topic of this thesis originates from a real-life case study involving patients that are on therapy to treat their obstructive sleep apnea. Sleep apnea is a serious and common chronic sleep disorder that is estimated to affect 1 in 5 people over the age of 65 years. People suffering from sleep apnea experience frequent pauses in breathing during their sleep, referred to as apneas. In case of obstructive sleep apnea (OSA), the apneas are caused by an obstruction of the airway due to tissue collapse arising from the relaxation of the airway during sleep. Untreated or undiagnosed sleep apnea is associated with excessive daytime fatigue, increased risk for hypertension and heart failure, stroke and myocardial infarction, and negative health outcomes like increased cardiovascular mortality (Knauert *et al.*, 2015). Untreated sleep apnea is costly for the healthcare system; spending for treating undiagnosed sleep apnea patients is estimated to be $1,950 to $3,899 more per patient per year than treating those that are diagnosed (Knauert *et al.*, 2015). This additional cost is the result of comorbidities being exacerbated by the sleep apnea condition. Fortunately, there are treatment options available for sleep apnea. The gold standard for obstructive sleep apnea treatment is positive airway pressure (PAP) therapy. Through the application of positive airway pressure, the collapse of the airway is prevented, suppressing the occurrence of apnea events. Treating OSA by PAP therapy, even when factoring in treatment costs, can result in a 37% to 48% overall annual healthcare cost reduction, already one year after initiating PAP therapy (Knauert *et al.*, 2015). PAP therapy is a highly effective treatment, but also presents some challenges to patients. To benefit from the associated health improvements, patients should ideally use the therapy every time they sleep, and for at least four hours. In addition, the therapy does not cure the patient of the underlying cause for sleep apnea, therefore patients must continue to use the therapy over time for as long as their OSA condition persists. In the United States of America, patients are reimbursed for the continued use of the therapy if they can demonstrate therapy adherence by day 90. However, reaching therapy adherence within 90 days is by no means a guarantee that a patient will be successful in adhering to the therapy on the long term.

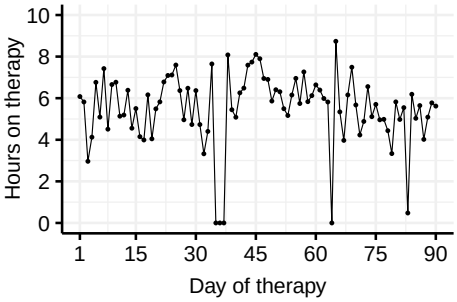
Patients follow the PAP therapy in different ways. Whereas the majority of patients successfully adopt the therapy, many patients struggle with adopting the therapy. Examples of the different adherence trajectories exhibited by patients are shown in Figure 1.1. With many patients abandoning the therapy eventually, managing patients through effective interventions is key in improving adherence to therapy. Patients on PAP therapy represent a highly heterogeneous group, with different behaviors and motivations for using the therapy. Identifying groups of patients with different adherence patterns enables tailored interventions, thereby improving the effectiveness of adherence management solutions.

Figure 1.1: Examples of daily PAP therapy usage of four different patients over the first 90 days.

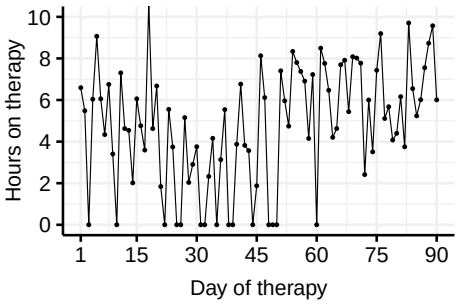
(a) Example patient with stable usage.



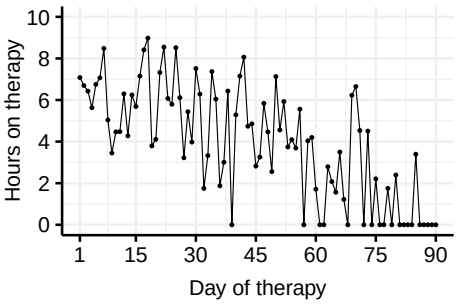
(b) Example patient with variable usage.



(c) Example patient exhibiting improvement over time.



(d) Example patient declining in usage over time.



Our goal is to model patient adherence over time, and to decompose the heterogeneity using an approach for clustering longitudinal data. We model adherence not just on the number of hours of usage per day, but we also model the skipped days of therapy as a separate decision process. The patients, grouped by their pattern of adherence, can then be analyzed and understood in more detail than the population level. Furthermore, the specific adherence patterns can be linked to tailored interventions to optimize the intervention effectiveness. An early example of an adherence cluster analysis is seen in the work of Aloia *et al.* (2008), who explored patterns of adherence during the first year of therapy among 71 OSA patients. They described the heterogeneity across patients in terms of seven adherence patterns. Moreover, they performed an independent individual time series analysis of all patients, providing further insights in the variation within each of the identified groups. In this dissertation, we analyze several datasets comprising sleep apnea patients on PAP therapy that are new to the therapy.

1.4.2 Understanding weekly new regional cases of COVID-19

In the second case study, we address the recent COVID-19 pandemic caused by the SARS-CoV-2 virus. The World Health Organization declared COVID-19 a pandemic on March 11, 2020. The impact of the virus on the health of people and the impact on the

health care system cannot be understated. The ease with which the virus spreads has posed serious challenges to countries across the world. The hospitalization rate and length of stay of patients who are severely impacted by the virus can overload local or even national health care systems. Local outbreaks can quickly expand to neighboring regions if timely and rigorous measures are not taken to halt the spread.

Due to the exponential growth in the early stage, small differences in the reproduction number (R_0) can lead to large differences in the number of cases between regions over time. As an example, consider the differences in the number of weekly new cases between counties of the state of New York in the USA in Figure 1.2 (Dong *et al.*, 2020) since the start of the pandemic. The degree of measures needed may differ between regions, depending on the number of current cases, and on regional factors that affect the effective reproduction rate of virus. Examples of such relevant factors are population size, population density, demographics, and behavioral and cultural norms (e.g., compliance to policy). As the virus transcends regional (e.g., city, county, or provincial) borders, decision makers could consider addressing similar regions at once.

In this dissertation, we explore the weekly number of new COVID-19 cases to see if there are differences in heterogeneity between regions. We analyze data from over 3,100 counties of the USA across all 50 states from June 1 to September 13, 2020. The forecast of the number of cases enables regulators and health system to take appropriate action. Note that the number of new cases and the uncertainty thereof largely depend on the current number of cases. We therefore take a particular interest in modeling the variance as a function of the number of cases to provide more reliable predictive intervals.

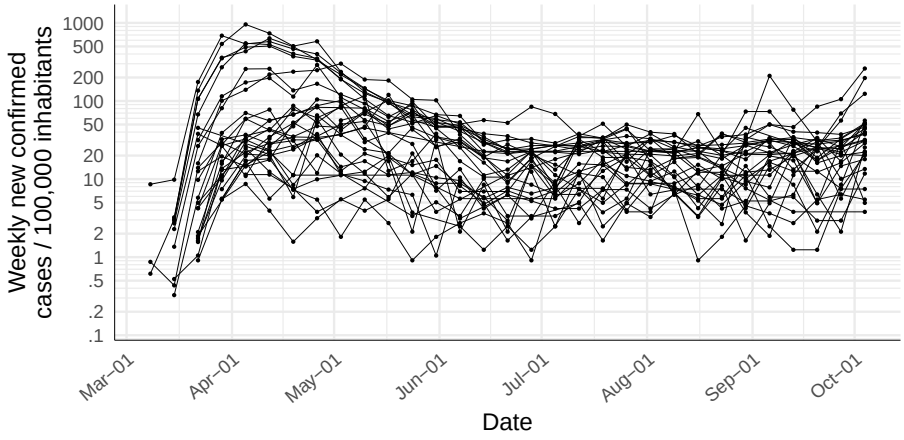
With the aim of identifying regions with similar developments in the number of cases over time, and considering the many possible factors at play, longitudinal clustering is a valuable and pragmatic tool for this purpose. An added benefit of clustering is that the approximate longitudinal county-specific developments can be easily visualized on a map through showing the cluster assignment of each county. The approach has been used in the analysis of the number of COVID-19 related deaths in the work of Maleki *et al.* (2020), for example. Donnat and Holmes (2021) modeled the heterogeneity in R_0. Lastly, Alvarez *et al.* (2020) have applied time series clustering to compare the spread of the virus across countries.

1.5 Outline

In Chapter 2, we provide a comprehensive introduction to longitudinal clustering. We have grouped the methods into classes of similar approaches, and we describe the methods using a unified terminology. The chapter summarizes the strengths, limitations, and extensions of the most commonly used methods. We demonstrate each of the methods and assess their performance on a synthetic dataset inspired by a case study exploring PAP therapy adherence in sleep apnea patients.

In the third chapter, we evaluate and compare a selection of the methods presented in Chapter 2 in detail. We investigate the differences between methods on synthetic data generated under various scenarios. The data scenarios vary in the number of trajectories and observations, within-cluster and within-trajectory variability, and the random effects distribution. The methods are assessed both on trajectory classification accuracy and the ability to recover the true shape of the cluster-representative trajectories. Moreover, we compare the solutions identified by the methods in a real-life setting based on PAP

Figure 1.2: Weekly new cases of 30 counties of the state of New York, USA, from March to October 2020. Intermittent weeks with zero cases are not shown to improve visual clarity.



therapy usage data.

Chapter 5 describes the design and functionality of the code framework that we have developed for clustering longitudinal data in a standardized way. The framework is implemented in the R package `latrend` and enables researchers to compare different methods or implementations with minimal coding effort. We have applied the framework in Chapter 6.

We also propose models for jointly assessing heterogeneity in the mean and other aspects of interest. Here, we use a model-based approach using growth mixture modeling. In Chapter 4, we propose a model which accounts for heterogeneity in adherence not only in the mean, but also in the day-to-day variability, and the probability of patients applying the therapy on a given day through hurdle modeling. This model provides a more detailed description of adherence behavior over time compared to previous longitudinal cluster analyses. In Chapter 6, the feasibility of recovering the clusters and heteroskedastic relationship under the presence of a heterogeneous mean-variance relationship is assessed. The models in this chapter are estimated using Bayesian inference. We evaluated the proposed models on the COVID-19 case study.

Finally, we summarize the current state, our proposed approaches, and future work in Chapter 7.

Chapter 2

Clustering of longitudinal data: a tutorial on a variety of approaches

N.G.P. Den Teuling, S.C. Pauws, E.R. van den Heuvel
Submitted.

Abstract
During the past two decades, methods for identifying groups with different trends in longitudinal data have become of increasing interest across many areas of research. To support researchers, we summarize the guidance from the literature regarding longitudinal clustering. Moreover, we present a selection of methods for longitudinal clustering, including group-based trajectory modeling (GBTM), growth mixture modeling (GMM), and longitudinal k-means (KML). The methods are introduced at a basic level, and strengths, limitations, and model extensions are listed. Following the recent developments in data collection, attention is given to the applicability of these methods to intensive longitudinal data (ILD). We demonstrate the application of the methods on a synthetic dataset using packages available in R.

2.1 Introduction

The analysis of longitudinal data is prominent in correlational studies that look for correspondence between observations of the same variables over extended period of time, such as substance use or mental health in psychology, recidivism behavior in sociology, and relapse or medication adherence in medicine. Longitudinal studies enable researchers to assess and study changes over time of the variables of interest. With the increasing capabilities of data collection and storage, more and more longitudinal studies are designed to involve a large number of repeated measurements of the same variable per subject over time. When a considerable number of observations are available, the data is commonly referred to as intensive longitudinal data (ILD) (Walls and Schafer, 2006). ILD has the advantage of allowing for a more granular assessment of change over time, especially at the subject level.

Analyzing longitudinal data requires models that take the structure of the data into account. The assessment of variability is key, as no two subjects are identical. In addition to the presence of measurement variability within each subject, models should account for differences (i.e., heterogeneity) between subjects. For example, in the analysis of therapy adherence, subjects may exhibit considerably different levels of adherence over time. An example of such a modeling approach is multilevel modeling. Here, the model describes the mean trend (i.e., longitudinal pattern), and captures the differences between subjects by modeling the subject-specific deviations from the trend.

In studies with considerable between-subject variability or non-normal deviations from the trend, subjects may exhibit large deviations, to the point that the mean trend may not be representative of the longitudinal patterns of the subjects (Hamaker, 2012). An intuitive alternative approach is to represent the differences across subjects in terms of a set of common trends. This way, the subject-specific deviations are reduced to the nearest trend. This approach is generally referred to as longitudinal clustering and involves the automatic discovery of groups of subjects with similar longitudinal characteristics. Longitudinal clustering is of interest, for example, in behavioral studies, where subjects can exhibit a range of behaviors that are due to various unobserved factors, resulting in structural deviations. We shall use the level of adherence of sleep apnea patients to positive airway pressure (PAP) therapy as the running example in this work. Factors such as perceived importance, self-efficacy, personality traits, claustrophobia, and many more have been shown to affect the level of adherence to PAP therapy (Cayanan *et al.*, 2019), resulting in a spectrum of longitudinal patterns across patients.

In this tutorial, we present a review of the literature on methods for clustering longitudinal data. While there are several aspects to modeling longitudinal data, we focus on the discovery of subgroups with different forms of longitudinal variations. Moreover, we summarize the guidance from literature on how to conduct such a longitudinal cluster analysis. Several types of methods have been proposed over the past two decades for clustering longitudinal data. Our intent is to assist the reader in making an informed decision on which method to apply in their cluster analysis, and to acquaint the reader with the available methodologies for longitudinal clustering. We describe each method along with its assumptions, advantages, and practical limitations. Secondly, we cover the topic of model specification, with a focus on the number of clusters needed to best represent the data. We survey the commonly used metrics and approaches to identify the most appropriate number of subgroups. Lastly, the methods are demonstrated on a

synthetic dataset inspired by a real-world example in the context of daily PAP therapy adherence of patients with sleep apnea (Aloia *et al.*, 2008). We will use this dataset to highlight differences in the assumptions of the methods and the estimation, as well as to show how to specify and apply each method.

The selection of methods has been based on prevalence and with the aim of creating a varied selection with different strengths and limitations. The variety of methods enables readers to select the most appropriate method for their case study. Moreover, we only considered methods that are applicable for identifying univariate longitudinal patterns of change and have a publicly available implementation in R. Relevant papers were identified via keyword searches and the snowball method. While we present the methods for the purpose of analyzing ILD, each of the methods are applicable to repeated measures data to some degree. The application of longitudinal clustering is becoming more commonplace. Based on a conservative keyword search[1], we observe a considerable increase in the number of publications concerning longitudinal clustering in different fields over time, from 37 publications in the nineties, to 273 publications between 2000–2009, and 1,257 publications between 2010–2019.

Terminology As the scope of this review is intended to be interdisciplinary, we describe the key terms used in this chapter, and list the commonly used alternative terms. We explain the topic of longitudinal cluster analysis in the context of clustering subjects over time, but the methodology applies equally well to any application involving repeated measures data, e.g., modeling devices, animal growth, or accident rates.

At the subject level, the sequence of longitudinal observations is commonly referred to as a trajectory, a time series, a temporal pattern, a curve, a trend, or a dynamic. Due to the frequency of measurement in the case of ILD, measurements are not necessarily equidistant in time. Moreover, subjects can have different non-corresponding times of measurement, and the number of measurements may vary. With this in mind, we define the trajectory of a subject i as a sequence of n_i observations by

$$\boldsymbol{y}_i = \{y_{i,1}, y_{i,2}, ..., y_{i,n_i}\}, \tag{2.1}$$

where the observation $y_{i,j}$ is taken at time $t_{i,j} \in \mathbb{R}$.

By clustering, we refer to the definition of a cluster analysis from the field of machine learning, specifically that of unsupervised learning, where data is grouped (i.e., clustered) based on similarity to the other data points, and the group definitions and assignments are not known in advance. In the field of statistics, a distinction is made between known clusters and unobserved clusters. In the former case, subjects are stratified based on a known nominal factor, for example, by assigning subjects to subgroups based on age or sex. Unknown clusters are commonly referred to as latent (i.e., hidden) classes, groups, profiles, or clusters. In this thesis, we use the term cluster as referring to the unobserved type.

Longitudinal clustering can be regarded as a specific area of time series clustering that is specifically concerned with the identification of common patterns of change or state changes throughout a longitudinal study. Whereas the scope of time series clustering extends to the modeling and assessment of any temporal similarity for any type of time

[1]A systematic search was performed per decade using Web of Science. Articles must contain the keyword "longitudinal", and one of the keywords "mixture", "latent-class", "clustering", or "group-based".

series data (Aghabozorgi *et al.*, 2015). Moreover, it includes the identification of temporal subsequences within time series.

Overview The chapter is organized as follows. We begin by elaborating on the case study in Section 2.2. In Section 2.3, we first summarize the concept of multilevel modeling as a precursor to modeling subgroups. Furthermore, we explain the concept of clustering; both philosophically and practically. The selected methods are described and demonstrated in Section 2.4. We outline the recommended steps involving a longitudinal cluster analysis in Section 2.5. Lastly, Section 2.6 discusses the findings from the case study in addition to the general challenges, limitations, and future work of longitudinal clustering.

2.2 Case study

We use the case study to illustrate the longitudinal cluster methods, and to contrast the strengths and limitations of the methods in a practical setting. The longitudinal methods are applied to a synthetic dataset, which facilitates a more detailed comparison between methods, and enables a fully reproducible and transparent demonstration. The data is generated from the population characteristics and groups as reported by Aloia *et al.* (2008), who investigated patterns of daily time on therapy among 71 obstructive sleep apnea patients in their first year of therapy. The synthetic dataset and analysis code are provided in the supplementary materials.

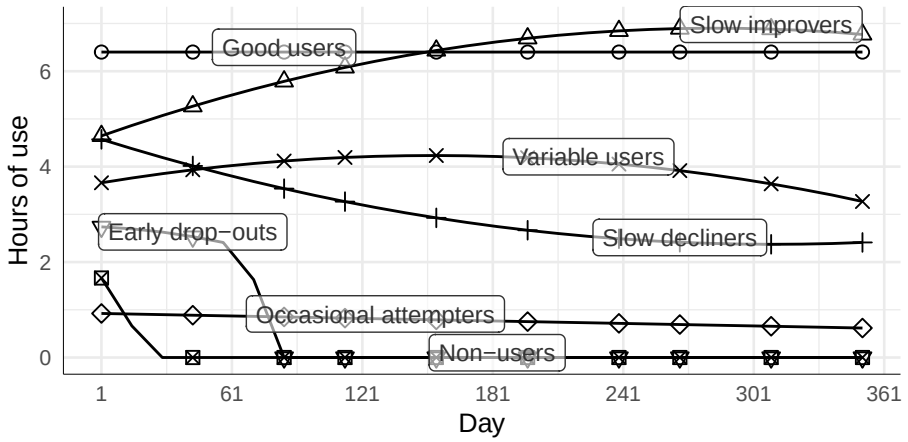
Sleep apnea is a common chronic disorder. Patients suffer from frequent paused or diminished breathing during their sleep, resulting in fragmented sleep and overall poor sleep quality. Sleep apnea is commonly treated using positive-airway pressure (PAP) therapy. This involves a device that assists the patient in breathing during sleep by supplying positive air pressure through a mask worn by the patient. Patients are required to use the device every time they sleep. Considering the inconveniences and difficulties patients can face with the therapy, some patients struggle to comply with the therapy for longer periods of time, whereas others do well. The continuation of the therapy is determined by many factors, e.g., the initial perception patients have of the therapy, the coping ability of the patient, and social support (Weaver and Grunstein, 2008; Cayanan *et al.*, 2019). An effective treatment can only be ensured when patients are compliant to the therapy, where the threshold for therapy compliance is usually set at 4 hours of therapy per day, but PAP use for longer than 6 hours has been shown to have positive effects (Weaver and Grunstein, 2008). The patterns of change in usage hours are therefore of interest. Most past studies have treated the patient population as being homogeneous, whereas others have attempted to stratify the population to address the differences in therapy adherence over time between subjects (Aloia *et al.*, 2008; Babbin *et al.*, 2015).

Aloia *et al.* (2008) modeled the trajectories of daily hours on therapy of each patient in terms of an intercept, slope, variance, autocorrelation, and number of attempted days. Seven clusters were manually identified using two expert raters. The cluster of Good users (24%) have a high number of therapy days and a high average hours of usage (6.6 hours). Slow improvers (13%) have an initially low number of hours early in therapy but increased over time, whereas the Slow decliners (14%) exhibit the opposite pattern. Variable users (17%) have a lower average usage (5 hours) and showed fluctuations in adherence over time. Occasional attempters (8%) have low attempt probability and low hours of use (3.2 hours),

Table 2.1: Group coefficients for generating the trajectories. Values enclosed in parentheses denote the standard deviation of the random effects. The attempt probability is conditional on the patients still being on therapy. The early drop-outs and non-users are modeled to stop prematurely at day 80 (30) and day 20 (10), respectively.

Cluster	π	β_0	$\beta_1 \times 10^2$	$\beta_2 \times 10^4$	σ^2	p_{attempt}
Good users	24%	6.6 (.54)	0.0 (.16)	0.0	2.0 (.82)	97%
Slow improvers	13%	4.8 (1.0)	1.7 (.16)	-0.30	3.6 (1.3)	94%
Slow decliners	14%	6.1 (.63)	-1.9 (.14)	0.30	3.2 (.85)	77%
Variable users	17%	4.4 (.87)	0.96 (0.0)	-0.30	3.4 (1.2)	82%
Occasional attempters	8%	3.2 (1.1)	-0.30 (.91)	0.0	3.6 (1.8)	29%
Early drop-outs	13%	4.0 (1.1)	-0.14 (1.0)	-1.0	5.0 (2.6)	69%
Non-users	11%	2.5 (.93)	-1.5 (1.0)	-1.0	3.0 (1.7)	70%

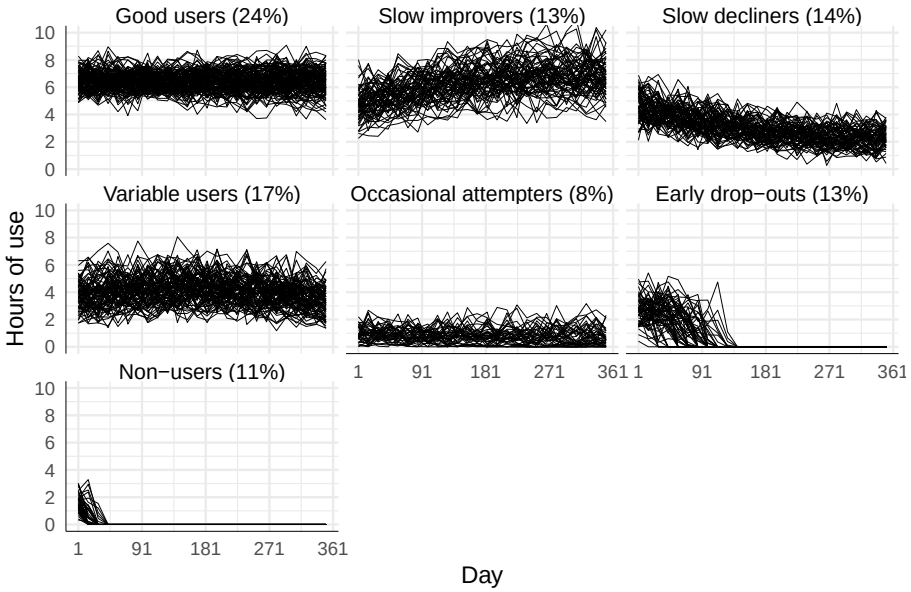
Figure 2.1: The mean cluster trajectories of the generated dataset involving 500 patients. Non-attempted days are included as zero-hour usage.



but the patients did continue therapy. Lastly, a sizable proportion of patients prematurely stopped with therapy, as represented by the Early drop-outs (13%) and Non-users (11%).

We utilize the reported patient and cluster statistics to generate 500 patient trajectories, with each patient comprising at most 361 observations. The trajectories are generated according to the original cluster proportions, and each trajectory is assigned a random deviation in intercept and slope from its respective cluster. Considering the scope on identifying patterns of change, we introduce a second-order term in the cluster trajectory shapes to evaluate whether the methods can recover these shapes. The cluster coefficients used to generate the trajectories are reported in Table 2.1. Due to the considerable computation time of the mixture methods, we downsampled the generated data to a biweekly average, resulting in 26 observations per patient, with 13,000 observations in total. The cluster trajectories and downsampled individual trajectories are visualized in Figure 2.1 and 2.2, respectively. Overall, 21% of biweekly observations are zero, and the mean non-zero usage is 4.6 hours ($\sigma = 2.1$ hours).

Figure 2.2: The generated 14-day averaged patient trajectories.

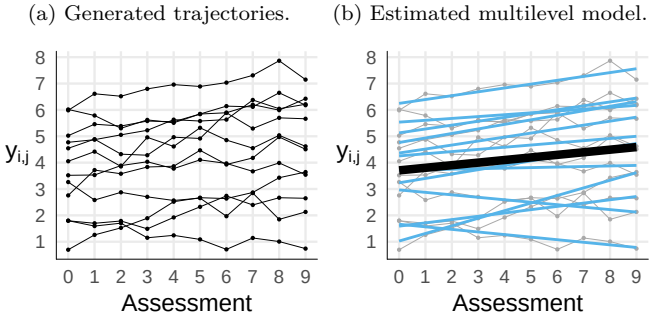


2.2.1 Evaluation

All methods are evaluated in R 3.6.3 using freely available packages (R Core Team, 2022). Each method is evaluated for 1 to 8 clusters to assess the number of clusters that correspond to the most representative solution of each method. If available for the respective method, we use the Bayesian information criterion (BIC) to guide the identification of the most appropriate number of clusters per method. It is defined by $\text{BIC} = p \log n - 2 \log \hat{L}$, where $\hat{L}$ denotes the likelihood of the candidate model, p is the number of model parameters, and n are the number of observations. The BIC is one of the most widely used metrics in longitudinal clustering (Van de Schoot *et al.*, 2017). A lower BIC indicates a more representative model for the data. The BIC includes a penalty factor for model complexity, resulting in a higher score for a model with more parameters. If the BIC values are similar between adjacent solutions, we base the final choice for the number of clusters on a subjective analysis of the variety in patterns identified by the methods (Nagin and Odgers, 2010a).

In case the BIC is not available for the respective longitudinal cluster method (i.e., there is no model likelihood), we apply the average silhouette width (ASW) (Rousseeuw, 1987). The ASW is a data-based measure of class separation. The silhouette value measures the similarity of an object to the objects in its assigned cluster, relative to the similarity of the other clusters. It is expressed as a score between -1 and 1, where a value towards 1 indicates a greater similarity to the assigned cluster. The ASW is obtained by averaging the silhouette values of all trajectories. The topic of selecting the number of clusters is discussed in more detail in Section 2.4.5.

Figure 2.3: A linear mixed model derived from 12 trajectories with random intercept $\sim N(3, 5)$, random slope $\sim N(1/10, 1/100)$ and measurement error $\sim N(0, 1/10)$.



2.3 Background

Prior to performing an exploratory cluster analysis, it is worthwhile to consider the case where the data comprises no clusters, i.e., the case where there is only a single cluster (Greenberg, 2016; Bauer, 2007; Sher *et al.*, 2011). We therefore begin by describing regression modeling, and how regression models can capture heterogeneity without the need for clusters.

Subjects can be considered as independent sources of variation, with each subject having, for example, their own mean response level, change over time, or measurement variability. Under this assumption, and given that subjects have a sufficient number of observations (as is typically the case with ILD), subjects can be modeled independently using established methods from the field of time series analysis (Liu, 2017). This is referred to as a two-step or bottom-up approach. The individual time series are commonly represented using linear regression or autoregressive models.

Aside from between-subject variability, there may be other sources of variation in the data. One can think of the measurement device used by the subject having a certain measurement error, which is shared across subjects using the same device. Another common source of variability are the different sites at which measurements are collected. Mixed modeling (Hartley and Rao, 1967; Laird, 1978) enables researchers to assess subject-specific effects, and to decompose the variability in the data. It is also commonly referred to as hierarchical modeling, random effects modeling, random coefficient modeling, and variance component modeling. The model describes the population-level effects, referred to as fixed effects, and describes a part of the subject-specific observations in terms of a structural deviation from the fixed effects. The subject-specific deviations are a source of variation as the deviations cannot be fully explained in terms of covariates, and therefore are treated as random variables, also referred to as a random effects or latent variables.

A possible way of modeling longitudinal change is to incorporate time as a covariate into the model. First- or second-order polynomials are commonly used to describe change as a function of time. If more flexible curves are required, cubic splines or fractional polynomials can be used. Figure 2.3 illustrates a first-order linear mixed model describing the outcome of each individual over time by an intercept and slope, where the assessment represents time.

In linear mixed-effects modeling, the response is assumed to be normally distributed (although extensions exist (Fitzmaurice *et al.*, 2011)), and the fixed and random effects are assumed to be a linear combination of covariates, giving

$$\begin{aligned} y_{i,j} &= \boldsymbol{x}_{i,j}\boldsymbol{\beta} + \boldsymbol{z}_{i,j}\boldsymbol{u}_i + \varepsilon_{i,j} \\ \boldsymbol{u}_i &\sim N(0, \Sigma) \\ \varepsilon_{i,j} &\sim N(0, \sigma_\varepsilon^2). \end{aligned} \tag{2.2}$$

Here, $\boldsymbol{x}_{i,j}$ denotes the patient-specific covariates at time $t_{i,j}$, and $\boldsymbol{\beta}$ are the respective coefficients. The random effects design vector is denoted by $\boldsymbol{z}_{i,j}$, where the random effects $\boldsymbol{u}_i$ are jointly normally distributed with zero mean and variance-covariance matrix Σ, and uncorrelated with $\varepsilon_{i,j}$. The measurement error is denoted by $\varepsilon_{i,j}$ and is assumed to be independently normally distributed with zero mean, a common variance σ_ε^2, and uncorrelated. Alternatively, the residuals can be modeled to be serially correlated (autocorrelated), but more complex correlation structures are often not possible with the inclusion of random effects due to identifiability problems.

Mixed modeling is advantageous over fitting individual regression models especially for datasets with a small number of measurements per trajectory, as the estimates of the subject-specific trajectory coefficients are more reliable due to the partial pooling of information across subjects. Mixed effects models can be estimated with maximum likelihood estimation. Alternatively, a Bayesian sampling approach can be taken. This has the advantage that researchers can incorporate domain knowledge in each model parameter, improving model estimation especially under small sample size due to the ability to include prior knowledge (Spiegelhalter *et al.*, 1994).

2.3.1 Meaning of clusters

A cluster analysis is generally exploratory in nature, meaning that the definitions of the clusters, or even the number of clusters, are unknown and need to be estimated from the data. There are two possible objectives to clustering, which determines how the resulting clusters are interpreted. In most cases, the motivation for clustering comes from the knowledge or expectation of considerable heterogeneity. In an indirect application of clustering, clustering is used as a tool for approximating a heterogeneous population in terms of a finite number of groups without any distributional assumption on the heterogeneity. The identified subgroups may help in accounting for correlations between longitudinal characteristics (e.g., the association between intercept and change over time). This is applicable when the population heterogeneity cannot be adequately modeled using a parametric approach such as multilevel modeling. Even in the case where a parametric model can represent the heterogeneity, clustering may be preferred as this representation of the heterogeneity can be easier to interpret (Sterba *et al.*, 2012; Rights and Sterba, 2016).

An alternative reason for clustering is to test or develop theories on subgroups (Moffitt, 2003), referred to as a direct application of clustering. Here, the resulting clusters are regarded as representing distinct population groups. Throughout the years however, the approach has been criticized for the lack of a formal test or validation of results (Bauer and Curran, 2003; Bauer, 2007). Overall, a direct application is only advisable under strong

theoretic assumptions or highly distinct (i.e., separated) subgroups. Ideally, the clusters are defined from theory, where clustering is applied as a confirmatory analysis, serving as an empirical validation (Sher *et al.*, 2011). In all other cases, an indirect application is a more practical and lenient interpretation, and is therefore generally preferred (Nagin and Odgers, 2010a; Sher *et al.*, 2011; Skardhamar, 2010).

The challenge of accounting for heterogeneity also applies to the cluster models. An intuitive approach to clustering involves describing the heterogeneity in terms of a number of homogeneous subgroups. In contrast, modeling heterogeneity within clusters allows for a more flexible representation of the overall heterogeneity. We illustrate the concept visually in Figure 2.4, depicting a heterogeneous population in which each subject is represented by a random variable in Figure 2.4a. The parametric approach, assuming a normal distribution, is shown in Figure 2.4b.

Applying a cluster algorithm that models homogeneous clusters produces non-overlapping bins (i.e., the clusters), represent a part of the heterogeneity, without any assumption on the variability within the cluster. This is illustrated in Figure 2.4c, where seven bins are used to represent the population density over the different values. Due to the lack of overlap between classes, this segmentation arbitrarily improves the approximation of the true distribution for an increasing number of bins.

Alternatively, a parametric model can be assumed for the heterogeneity within each cluster. Such a model represents a mixture of distributions, referred to as a finite mixture model (McLachlan and Peel, 2000). Figure 2.4d shows the density of the three normal distributions that make up the mixture model. In this example, this was the true model from which the data was generated. This approach has the advantage of requiring fewer classes due to the ability to model outliers, but these models are more challenging to specify and identify. Moreover, the overlap between classes increases as the number of classes increases.

2.4 Methods

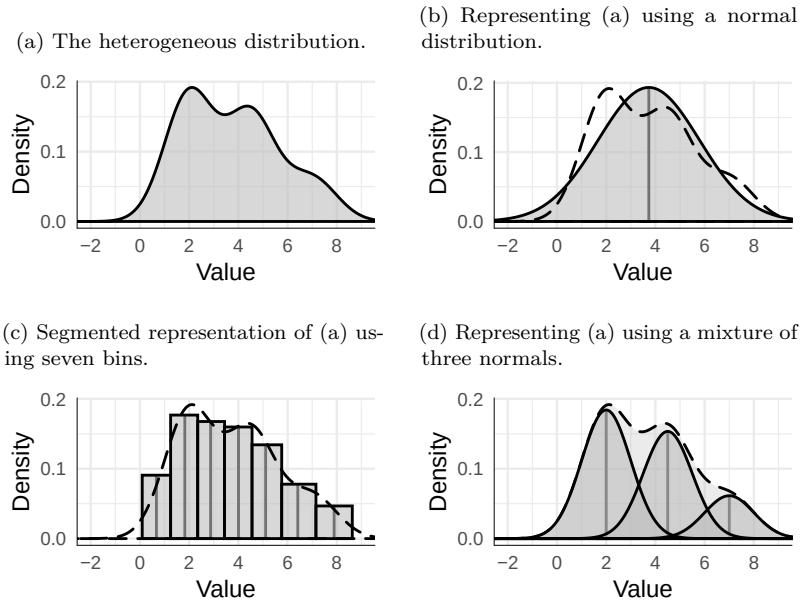
We have organized the methods for longitudinal clustering into three approaches, with increasing model complexity. In the first approach, a cross-sectional cluster algorithm is directly applied to the observations. The second approach comprises feature-based estimation methods which model the trajectories independently, and cluster the trajectories by the model representation. The third approach involves the use of a mixture model to perform clustering using parametric group models. The methods are described under the assumption that the number of clusters is part of the model specification.

2.4.1 Cross-sectional clustering

In a cross-sectional clustering approach, cluster algorithms or mixture methods that are ordinarily applied to cross-sectional data with different variables are applied directly to the longitudinal observations. Here, the trajectories (or objects, in a cross-sectional context) are represented by a sequence of observations measured at fixed times $t_1, \ldots, t_n$, with $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,n})$. Thus, each assessment moment t_j represents a separate (random) variable. This representation requires individuals to be measured at (almost) identical assessment times across subjects, although the assessment times need not be equidistant.

Considering that cross-sectional methods do not model dependence between parameters,

Figure 2.4: Representation of a heterogeneous distribution using different approaches. The vertical gray lines in (b), (c) and (d) denote the class centers.

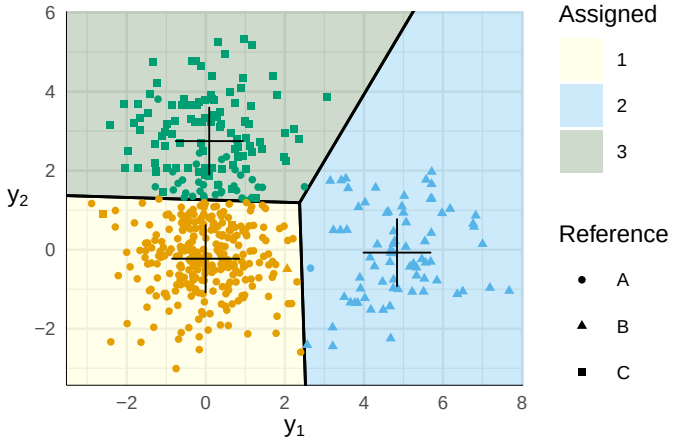


applying cross-sectional methods to longitudinal data carries the assumption that the observations are locally independent (i.e., the temporal ordering of the observations can be ignored). Although this assumption does not hold in a longitudinal setting, it results in a nonparametric trajectory model that can model sudden changes over time. The approach is therefore especially useful in an exploratory analysis in case where the shape of the cluster trajectories is unknown. Another reason why these methods are favorable for an initial exploration is that they are orders of magnitude faster to compute compared to more complex longitudinal models. The approach is also referred to as raw data-based approach (Liao, 2005). While the approach is versatile, its applicability is limited to complete data with identical assessment times across subjects, which are challenging requirements in case of ILD. We describe two commonly used methods for cross-sectional clustering of longitudinal data below. The methods are available in most statistical software packages (e.g., in SPSS, SAS, Stata, and R), and have been used in practice.

2.4.1.1 k-means clustering

The aim of k-means clustering is to represent a set of objects in terms of a predefined number of representative objects (MacQueen, 1967). It is essentially a quantization method, and it is used in many fields, including machine learning, image processing, and signal coding. In the analysis of longitudinal data, the methodology is referred to as k-means for longitudinal data (KML), or longitudinal k-means analysis (LKMA). An early example of this type of analysis can be found in the work of Gude and Odd (2000), who performed a thorough longitudinal cluster analysis on patients with personality disorders receiving

Figure 2.5: Example of k-means applied to 2D data comprised of three Gaussian clusters with means for (y_1, y_2) of (0,0) for group A, (5,0) for group B, and (0,3) for group C, with unit variance. The crosses denote the cluster centroids.



treatment to identify groups of patients with different symptom distress over time. The trajectories comprised three assessments of global symptom distress. They assessed the cluster agreements between different random starting positions and performed post-hoc analyses on the clusters which revealed correlations on other aspects of the patients. Their work has been replicated recently by Jensen *et al.* (2014), with similar results. KML has been used to identify adherence patterns in obstructive sleep apnea patients undergoing nasal CPAP therapy (Wang *et al.*, 2015). Furthermore, they investigated the early prediction of the (ordered) adherence clusters using a cumulative logit model. ANOVA F-tests were used to identify predictor variables that could aid in predicting the adherence pattern. The KML methodology is implemented in the R package `kml`[2], created by Genolini *et al.* (2015).

In k-means, clusters are assumed to be homogeneous, as each representative object defines the center of a cluster, referred to as the centroid. The cluster membership of objects is determined by their nearest centroid. An example of k-means on synthetic 2D data is given in Figure 2.5.

Assuming that the total variance consists of a within-cluster and between-cluster variance component, minimizing the within-cluster variance ensures maximal separation of the clusters. Thus, the k-means algorithm searches for the clustering $\{I_1, I_2, ..., I_G\}$ that minimizes the within-cluster sum of squares, with each cluster I_g having one or more objects. The objective function is described by

$$\underset{I_1,\ldots,I_G}{\arg\min} \sum_{g=1}^{G} \sum_{i \in I_g} ||\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_g||^2, \tag{2.3}$$

where $\hat{\boldsymbol{\mu}}_g$ denotes the centroid of cluster g. Finding the optimal cluster assignments is computationally infeasible as it requires iterating over all possible assignment combinations

[2]https://CRAN.R-project.org/package=kml

for all objects. Instead, the algorithm uses a heuristic iterative approach. The solution is sensitive to the choice of the initial centroids. The centroids can be selected, for example, by selecting k objects at random as the centroids (MacQueen, 1967) or using the output from a cluster algorithm such as agglomerative hierarchical clustering (as seen in the analysis by Axén *et al.* (2011)). A method proposed by Arthur and Vassilvitskii (2007), named k-means++, generally provides better starting conditions by selecting a disperse set of centroids at random.

The k-means method assumes that the within-cluster variance is equal across clusters. When the subgroups in the data have different variation, the estimated cluster boundaries will likely be wrong, even when the centroids are estimated correctly. The challenge is that cluster assignments can be problematic if many objects are relatively distant from the respective cluster centroid (i.e. outliers), or being close to the cluster boundary in-between clusters. An adaptation of k-means, named fuzzy c-means, addresses this concern by using probabilistic cluster assignment based on the distance to the centroids (Dunn, 1973; Bezdek, 1981). Another challenge is the presence of subject outliers, as these are not represented by the cluster centroids. An example of this can be seen in Figure 2.4c on page 18, where the tails of the distribution fall outside any of the bins. In some cases, these outliers can affect the resulting cluster centers. This can be prevented by excluding these subjects from the data (referred to as trimmed k-means).

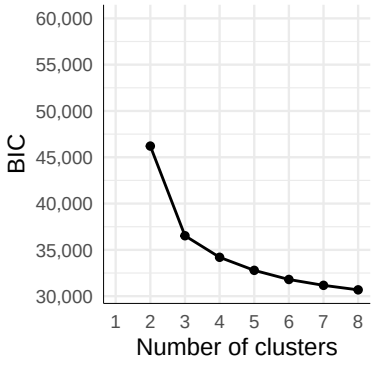
An advantage of k-means is that the algorithm scales well and converges to a solution relatively quickly. In some studies, the trajectories are stratified prior to clustering as a way to guide the clustering process. An example of this approach is seen in the work of Chen *et al.* (2007), where the authors evaluated patterns of change in self-reported back pain over one year of time. The change in pain intensity over time, as computed using linear regression, is used to stratify trajectories in three strata (decreasing, increasing, and constant pain intensity), and clustering is performed within the strata.

Case study We use the R package `kml` (version 2.4.1) to cluster the trajectories (Genolini *et al.*, 2015). For each number of clusters, we run the estimation procedure 20 times, and select the best solution from the repeated runs based on the BIC. The successive solutions for an increasing number of clusters consistently improve the model fit, suggesting a solution with many clusters. The package computes the BIC corresponding to the best solutions for 2 to 8 clusters, as depicted in Figure 2.6a. There is a balance to be found between the practical aspect of the number of clusters and the improvement in model fit. Arguably, the three-cluster solution may be preferred as the latter solutions add relatively little improvement. However, with the objective of identifying patterns of adherence and the improved model fit, we visually assessed the remaining solutions.

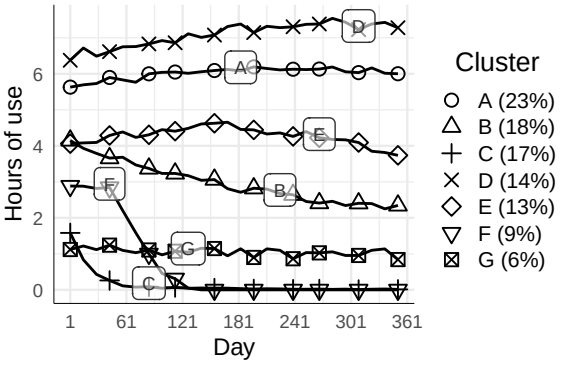
We selected the seven-cluster solution because from this solution onward, the occasional attempters were distinguished from the non-users. The identified cluster trajectories are shown in Figure 2.6b. Although the number matches the true number of clusters, this is only incidental, as KML failed to identify two cluster trajectory shapes correctly, and this does not improve by introducing more clusters. Overall, the solution recovered most of the cluster trajectories, demonstrating the benefit of a nonparametric approach in an exploratory setting.

Figure 2.6: KML case study analysis.

(a) BIC per solution (lower is better).



(b) The identified cluster trajectories.



2.4.1.2 Latent profile analysis

Latent profile analysis (LPA) is a statistical approach in which subjects are modeled to belong to one of several unknown clusters (i.e., profiles) (Lazarsfeld *et al.*, 1968; Vermunt and Magidson, 2002). Furthermore, the measurement error is taken into account into the probability of belonging to a certain class. LPA describes a mixture of profiles represented by multivariate normal distributions, an approach also referred to as Gaussian mixture modeling, or model-based clustering (Aghabozorgi *et al.*, 2015). The method is also commonly referred to as latent class analysis (LCA), although in some fields this name specifically refers to a model involving categorical rather than continuous observations.

Similar to KML, LPA can be applied for the identification of longitudinal patterns without any assumption on the shape by modeling the observations as locally independent variables at the subject level (Feldman *et al.*, 2009; Twisk and Hoekstra, 2012). This type of application of LPA is sometimes specifically referred to as a longitudinal latent-profile analysis (LLPA). The expected value of an observation at time t_j depends on the cluster membership. Given the cluster membership g, we have

$$y_{i,j} = \mu_{g,j} + \varepsilon_{g,i,j}, \quad i \in I_g, \tag{2.4}$$

where $\mu_{g,j}$ represents the cluster-specific mean at time t_j, $\varepsilon_{g,i,j} \sim N(0, \sigma_{g,j})$, and $\sigma_{g,j}$ is the cluster-specific standard deviation at time t_j. The probability density of the observations of subject i is computed by marginalizing over all G clusters, giving

$$f\left(\boldsymbol{y}_i\right) = \sum_{g=1}^{G} \pi_g \prod_{j=1}^{n} \phi\left(y_{i,j} | \mu_{g,j}, \sigma_{g,j}\right), \tag{2.5}$$

where $\phi(\cdot)$ denotes the probability density function of the normal distribution, and π_g denotes the cluster proportion with $\pi_g > 0$ and $\sum_{g=1}^{G} \pi_g = 1$. To reduce the number of parameters, the variance is commonly assumed to be equal across measurements over time, i.e., $\sigma_{g,j} = \sigma_g$ (Peugh and Fan, 2013).

The model is usually estimated through maximum likelihood estimation using the EM algorithm (McLachlan and Peel, 2000). Here, the data is sought to be explained in terms of the unknown observation model $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_{G-1}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_G)$, and the unknown cluster membership matrix z, where $z_{i,g}$ is the probability of patient i belonging to cluster g conditional on $\boldsymbol{\theta}$. The algorithm takes an iterative approach, involving an alternating estimation of z and $\boldsymbol{\theta}$, conditional on the other. In the E-step, the cluster assignment probabilities are estimated from the given parameters $\boldsymbol{\theta}$ and $\boldsymbol{y}$. In the M-step, the parameters $\boldsymbol{\theta}$ are estimated given z. The iterations are repeated until the improvement in log-likelihood is sufficiently low. The estimation must be initialized with some values for $\boldsymbol{\theta}$. Here, random values can be used, or preferably, the output of a cluster model with fewer parameters.

While LPA is more computationally expensive than KML, it allows for greater flexibility in fitting the data due to the ability to account for cluster-specific variances, and even time-specific variances (Magidson and Vermunt, 2002). LPA is available in many software packages, including in Mplus[3] (Muthén and Muthén, 1998–2012), Latent GOLD[4] (Vermunt and Magidson, 2016), and the R package `mclust`[5] (Scrucca *et al.*, 2016).

Case study We estimate the LPA models using the `mclust` package (version 5.4.5) in R (Scrucca *et al.*, 2016) with cluster-specific diagonal covariance matrices. Ten repeated runs were found to be sufficient in identifying the best solution per number of clusters (determined by the BIC). The BIC per number of clusters is visualized in Figure 2.7a, showing a considerable improvement up to four clusters. While the eight-cluster solution compares favorably, it comprises a small clusters of only ten patients, which would limit the power of a post-hoc analysis. Based on the BIC, one would ordinarily select the four-cluster solution, as it provides a trade-off between a good fit and a lower number of clusters. However, upon inspection of the successive solutions, the five-cluster solution distinguishes the early drop-outs from the non-users, which would be of added value in an exploratory analysis for patterns of adherence. Moreover, the solutions involving more than five clusters comprise spurious empty clusters, or clusters which are too small to be of practical use.

The cluster trajectories of the preferred five-cluster solution are shown in Figure 2.7b, showing an emphasis on representing trajectories with lower usage due to the modeling of cluster-specific and time-varying variances, because these decline over time for the early drop-out and non-user groups.

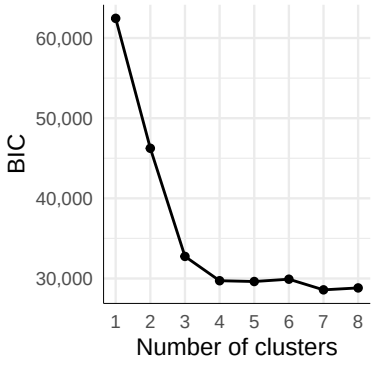
2.4.2 Distance-based clustering

In a distance-based cluster approach, trajectories are clustered based on their pairwise similarity, as measured by a user-specified dissimilarity metric, i.e., distance measure. This approach comprises cluster methods for which the user can specify an arbitrary distance metric. This enables fast experimentation with different measures of similarity suitable to the application at hand. The distance between two trajectories $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is defined by a distance measure $d(\boldsymbol{y}_1, \boldsymbol{y}_2)$. A commonly used measure is the Euclidean distance

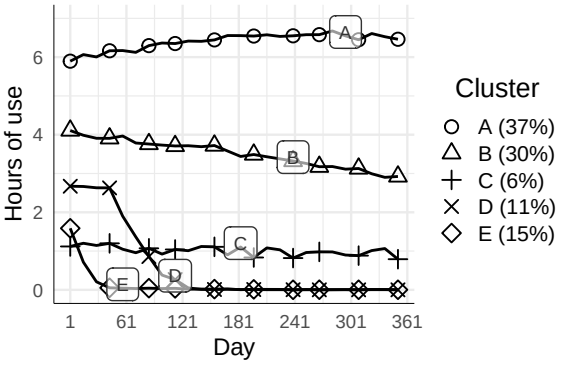
[3] https://www.statmodel.com/
[4] http://www.statisticalinnovations.com/latent-gold-5-1/
[5] https://CRAN.R-project.org/package=mclust

Figure 2.7: LLPA case study analysis.

(a) BIC per solution (lower is better).



(b) The identified cluster trajectories.



$$d(\boldsymbol{y}_1, \boldsymbol{y}_2) = \sqrt{\sum_j (y_{1,j} - y_{2,j})^2}, \tag{2.6}$$

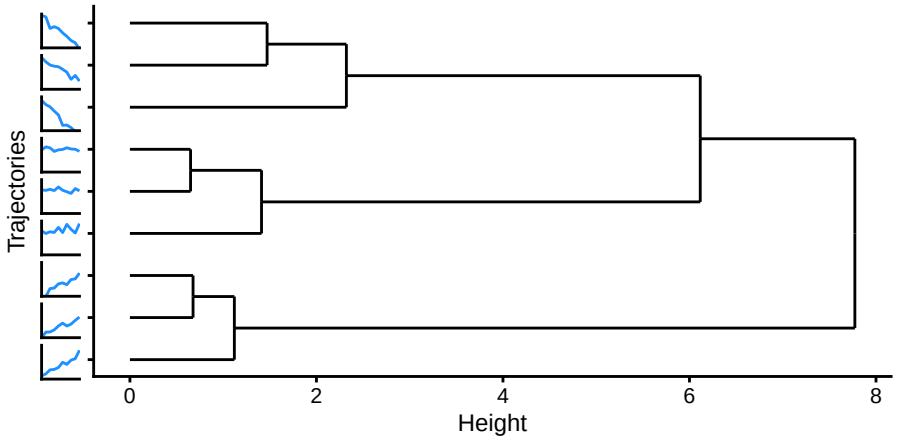
which essentially yields a raw-data based approach. However, the advantage of a distance-based approach is that domain knowledge can be taken into account in specifying the distance measure to capture the relevant properties of the trajectories. The Euclidean distance has been shown to be applicable to longitudinal data (Genolini and Falissard, 2010), resulting in cluster trajectories with arbitrary shapes, but conversely the measure is sensitive to temporal offsets between subjects, and noise. Many alternative distance measures have been suggested, including measures that account for temporal offsets (e.g., dynamic time warping), or reduce the complexity of the trajectory (e.g., piecewise-constant approximation) (Aghabozorgi *et al.*, 2015; Wang *et al.*, 2013). Another advantage is that the pairwise distances between trajectories yields a hierarchy which provides additional information on the heterogeneity.

2.4.2.1 Agglomerative hierarchical clustering

Hierarchical clustering is a type of cluster method that identifies a hierarchy in a set of objects based on a distance measure. The resulting hierarchy provides an ordering of the objects based on their similarities, which can be a useful tool in visualizing a spectrum of trajectories with different shapes. The number of clusters can be estimated from the distance between hierarchical clusters (Islam *et al.*, 2015).

Agglomerative hierarchical clustering (AHC) uses a bottom-up approach to identify the hierarchical structure of the objects. Each of the objects start out as separate clusters. The AHC approach is commonly used in combination with a post-hoc analysis to identify factors that may differ between clusters. Babbin *et al.* (2015) investigated the daily time on CPAP therapy of patients with obstructive sleep apnea to identify clusters of patients with similar adherence trends. Other examples include the investigation of Hoeppner *et al.* (2008) of daily smoking patterns after patients went through a reduction program, and patterns of alcohol use (Harrington *et al.*, 2014).

Figure 2.8: Example of a dendrogram computed from longitudinal data comprising three groups, each having three trajectories. The cluster trajectories are described by an intercept and slope, with coefficients $\boldsymbol{\beta}_A = (3, -0.3)$, $\boldsymbol{\beta}_B = (2, 0)$, and $\boldsymbol{\beta}_C = (0, 0.2)$, respectively.



In AHC, the hierarchy is identified using a greedy approach, where at each step the two most similar clusters are combined into a new cluster. This is repeated until a single cluster remains containing all objects. The resulting hierarchy can be visualized in a dendrogram. To illustrate, Figure 2.8 depicts the hierarchy of nine trajectories generated from three different linear models.

Combining objects and clusters involves two distance measures. Firstly, a distance measure $d\left(\boldsymbol{y}_i, \boldsymbol{y}_j\right)$ is needed for determining the similarity of the trajectory of individual i and individual j, with $i \neq j$. Secondly, a distance measure between clusters I_r and I_s is needed, where a cluster I_g is the subset of individuals (out of all individuals I) that belong to cluster g. The distance $d\left(I_r, I_s\right)$ is referred to as the linkage criterion, with $r \neq s$.

An intuitive approach to measuring the distance between clusters is to measure the average pairwise distances between the clusters. This is referred to as the unweighted average linkage, and is computed by

$$d(I_r, I_s) = \frac{1}{|I_r| \cdot |I_s|} \sum_{i \in I_r} \sum_{j \in I_s} d(\boldsymbol{y}_i, \boldsymbol{y}_j).$$

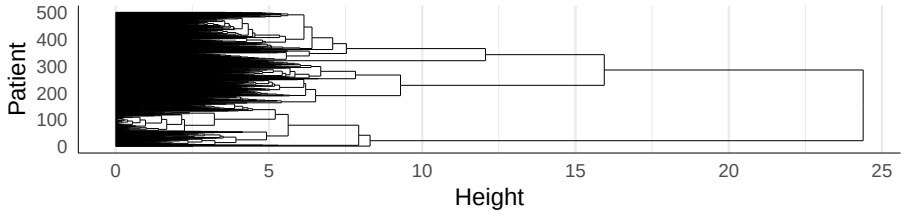
Alternative linkage criteria which are commonly used include the minimum linkage $\min\{d(\boldsymbol{y}_i, \boldsymbol{y}_j) : i \in I_r, j \in I_s\}$, centroid linkage, and Ward's minimum variance method.

AHC provides a good trade-off between finding a reasonable hierarchy quickly, and identifying the optimal hierarchy (i.e., the hierarchy that minimizes the overall distance). However, the computation time quickly grows with the number of trajectories, due to the pairwise distances that must be computed between all subjects.

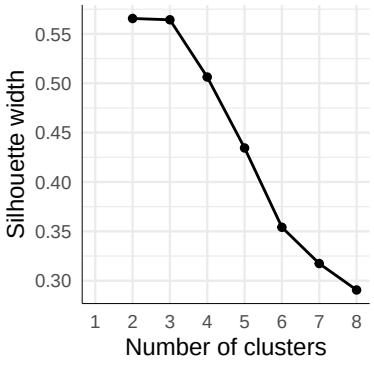
Case study As all measurements across patients are aligned in the case study, we can apply the Euclidean distance to compute the pairwise similarity between patient trajectories. We then apply the agglomerative hierarchical cluster algorithm that is available in R using the average linkage. For each number of clusters, the trajectory

Figure 2.9: Case study analysis using AHC.

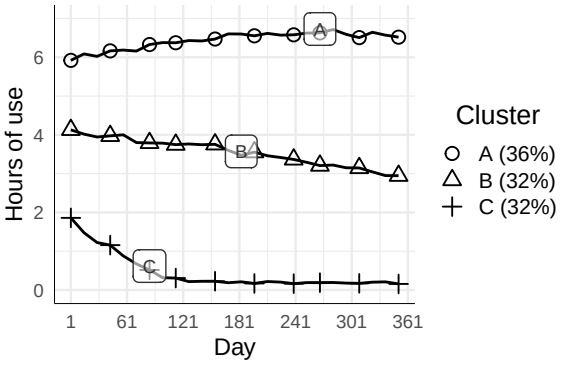
(a) The identified trajectory hierarchy.



(b) Average silhouette width per solution (higher is better).



(c) The identified cluster trajectories.



assignments are obtained based on the identified hierarchy visualized in Figure 2.9a. The solutions are compared using the ASW. As can be seen in Figure 2.9b, the ASW is considerably lower for solutions with more than three clusters. A cluster solution with an ASW above 0.5 is generally considered to have some consistent structure. In an exploratory setting it may be worthwhile to forfeit this rule of thumb in favor of identifying additional meaningful temporal patterns, given that the clusters are of sufficient size. In this case however, the solutions with a larger number of clusters include clusters of outliers comprising only a single trajectory (as can be seen from the hierarchy), so the three-cluster solution is preferred.

The solution comprising three clusters is shown in Figure 2.9c. Here, the cluster trajectories are computed by averaging across all trajectories that are assigned to it. The solution provides a balanced representation of the seven groups, combined based on the respective mean level.

2.4.3 Feature-based clustering

In a feature-based clustering approach, individual trajectories are described in terms of a parametric model that captures the relevant characteristics. Here, each trajectory $\boldsymbol{y}_i$ is reduced to a set of model parameters $\boldsymbol{b}_i = (b_{i,1}, b_{i,2,}, ..., b_{i,p})$ which can be regarded as the p features of the trajectory. Clusters of trajectories with similar characteristics can then

be identified by applying a cross-sectional cluster algorithm to the model parameters. The appeal of a feature-based clustering approach is that researchers can incorporate domain knowledge in defining the similarity between trajectories by using an appropriate model, or by computing several independent characteristics. The characteristics may better capture the differences between trajectories than a cross-sectional approach based on the shape alone, resulting in more well-separated clusters (Wang *et al.*, 2006). The approach is also commonly referred to as a feature-based or model-based approach[6] (Aghabozorgi *et al.*, 2015).

The second step of clustering is easy to implement and available in most statistical software packages through the widespread availability of clustering algorithms such as k-means. There are several strengths to the approach: especially within the context of ILD. Firstly, the parametric representation of trajectories is naturally more robust to missing observations, as the computed characteristics tend to be based on multiple observations. Secondly, the approach can handle trajectories of varying lengths between individuals or measured at different intervals (Wang *et al.*, 2006). Lastly, the method scales well with the amount of data, as the representation is of constant size independent of the number of observations. Moreover, the trajectory representations only need to be computed once, after which a cluster algorithm can be fitted to the feature data for varying settings as part of the model selection.

2.4.3.1 Individual time series representations

Trajectories can be represented in many ways. An intuitive approach to describing each trajectory is in terms of a linear model dependent on time (e.g., an intercept and slope), as seen in multilevel modeling, where the individual trajectory representations can be obtained from the estimated random effects and are then clustered using a cluster algorithm (e.g., k-means (Twisk and Hoekstra, 2012)). While this is a useful approach when there are relatively few observations per trajectory, independently estimating the representation of each trajectory frees researchers of any assumptions on the population heterogeneity, yielding a more detailed description of the heterogeneity (Liu, 2017). This approach is referred to as an individual time series (ITS) analysis (Bushway *et al.*, 2009; Greenberg, 2016; Liu, 2017).

An example of the ITS approach to clustering is seen in a method named anchored k-medoids, created by Adepeju *et al.* (2019), where the trajectories are represented by time-dependent linear regression models. The trajectories are then clustered based on the coefficients using k-medoids. The k-medoids cluster algorithm is similar to k-means, but uses one of the observations (i.e., objects) as the cluster center instead of an average across observations. This is especially useful for ITS representations because the algorithm can handle arbitrary distance metrics and does not require the computation of an average cluster representation, which may not be sensible for some model coefficients or distance metrics.

There are two advantages to the ITS approach. Firstly, the trajectory models can be estimated independently, allowing for a trivial parallelization of the estimation process. Secondly, the independent models do not need to account for any variability between trajectories and are therefore easier to estimate than a multilevel approach. A disadvantage

[6]The term "model-based clustering" appears to be used for both feature-based clustering of model parameters and mixture modeling.

of modeling each trajectory independently is that there may be trajectories for which the model fit is poor, resulting in clusters primarily containing poorly fitted models of which the original trajectories may not be similar. A poor fit can be the result of a trajectory not meeting the model assumptions, or the number of data points being insufficient for the model. The possibility to combine multiple representations into a single model vector provides additional challenges like those seen in cross-sectional clustering involving high-dimensional data: The coefficients may need to be normalized to ensure that the distance function is not disproportionately affected by coefficients of a higher magnitude. On the other hand, a subset of coefficients may be deemed more important (Fulcher and Jones, 2014).

In its simplest form, trajectories can be represented by summary statistics such as the mean, standard deviation, skewness, range, degree of stationarity, periodicity, autocorrelation, or entropy (Fulcher *et al.*, 2013; Aghabozorgi *et al.*, 2015). Another practical approach is to categorize the response variable into a finite number of values (i.e., states). Kiwuwa-Muyingo *et al.* (2011) modeled the adherence behavior of patients undergoing medical treatment for HIV infection using a first-order Markov chain, modeling the transitional probabilities of the non-adherent and adherence states. They used AHC using Ward's minimum variance method to cluster the transitional probability vectors. A limitation of many of the statistical measures proposed is that they are under the assumption that the statistical properties do not change over time. This can be resolved by correcting for longitudinal changes, by estimating the properties over multiple segments of the trajectory, or by fitting a linear model that represents the change over time.

In other cases, an abrupt change in the observations may be expected from domain knowledge. Change points are for example modeled in the work of Axén *et al.* (2011), who investigated patients diagnosed with non-specific low back pain. In this work, the trajectories were modeled using two linear models which describe the early and late trajectory, respectively, fitted using spline regression. The linear model coefficients, as well as the estimated intersection point of the two lines, were used as inputs for the second step clustering.

Wang *et al.* (2006) propose a set of nine statistical features for describing a trajectory: Firstly, a trajectory can be decomposed into several components (Kendall *et al.*, 1983); a trend T_t (the long-term average level), a seasonal effect S_t, and a cyclic effect C_t (also referred to as periodicity or frequency). Assuming that the components are not proportional, a time series can be described using an additive model

$$y_t = T_t + C_t + S_t + \varepsilon_t, \tag{2.7}$$

where ε_t denotes the irregular component (i.e., the residual). Secondly, Wang *et al.* (2006) suggest describing aspects of the measurement distribution in terms of the skewness and kurtosis. Thirdly, the temporal structure of the data is expressed in terms of the autocorrelation and a test for non-linearity, e.g., through a nonparametric kernel test or neural network test. Lastly, the trajectory complexity is assessed using self-similarity (a measure of long-term dependence) and other methods commonly used in describing chaotic systems (e.g., the Lyapunov exponent, which is a measure of divergence in response to small perturbations). Especially, the latter metrics require a sizable number of observations per trajectory to be estimated reliably, so these are only suitable for ILD.

The irregular component ε_t describes the local changes of a trajectory. A straightforward way to describe the component is through a white noise process of zero mean and

variance σ^2, assuming independent and identically distributed observations. When the local changes are assumed to correlate with past values, an autoregressive (AR) model is typically used. This model regresses past values using a p^{th}-order polynomial. Alternatively, the model error may depend on previous errors, which can be described using a moving average (MA) model of the past q error terms. Combining these two models, we obtain an ARMA(p, q) model describing a stochastic process

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t, \tag{2.8}$$

with c describing the model intercept, and $\phi_{1,\dots,p}$ and $\theta_{1,\dots,q}$ describing the parameters of the AR and MA models, respectively. The model residuals are denoted by ϵ_t, and are assumed independent and follow a normal distribution with zero mean and variance σ_ϵ^2. ARMA can be applied to non-stationary data by first applying one or more differencing steps $y'_t = y_t - y_{t-1}$ to the data, in which case the approach is referred to as ARIMA (where the I stands for integrated). ARIMA is mostly used in the economic and financial domain for predicting future values, but it is also of use for process modeling (e.g., adaptive control), and descriptive modeling (Jebb *et al.*, 2015; Aloia *et al.*, 2008). Kalpakis *et al.* (2001) have proposed a distance measure for comparing ARIMA models, which are then clustered using k-medoids. This approach could be regarded as a hybrid of the feature-based and distance-based approaches to longitudinal clustering. Other useful methods for describing stochastic processes are autoregressive conditional heteroskedasticity (ARCH), Gaussian processes, and state space models (Fulcher *et al.*, 2013).

Case study We model each patient independently on several aspects. Each trajectory is represented in terms of an intercept $\beta_{i,0}$, an orthogonal polynomial of degree two with coefficients $(\beta_{i,1}, \beta_{i,2})$, a residual error $\sigma_{\varepsilon,i}$, and the log-number of attempted days $\log N_i$. This yields the patient representation $\boldsymbol{b}_i = (b_{i,1} = \beta_{i,0}, b_{i,2} = \beta_{i,1}, b_{i,3} = \beta_{i,2}, b_{i,3} = \sigma_{\varepsilon,i}, b_{i,4} = \log N_i)$. The patient representation vectors $\boldsymbol{b}_i$ are scaled to ensure zero mean and unit variance across the features. We compute a distance matrix using the Euclidean distance, and then apply k-medoids using the `cluster` package[7] (version 2.1.0) (Maechler *et al.*, 2019) to obtain clusters which are represented by one of the computed representation vectors. As with the AHC analysis, we evaluate cluster solutions using the ASW.

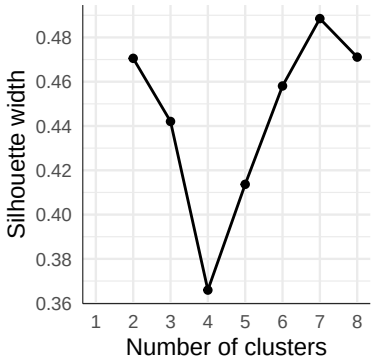
The inclusion of irrelevant features can negatively affect the cluster separation. It is therefore important to select the relevant aspects. Moreover, the approach is sensitive to spurious estimates of the patient-specific coefficients, as only a limited number of observations are available. These aspects reduce the separation between clusters, resulting in a lower ASW. We investigated different subsets of the patient representation vector by assessing the highest observed ASW. This revealed that the inclusion of $b_{i,3} = \sigma_{\varepsilon,i}$ negatively affected cluster separation, and should be excluded. This is despite the fact that the data was generated with some degree of group-specific variance. It was found that the residual error is often underestimated, likely resulting from overfitting of the polynomial trajectory of some of the patients.

The ASW per number of clusters for the final model $\boldsymbol{b}_i = (b_{i,1} = \beta_{i,0}, b_{i,2} = \beta_{i,1}, b_{i,3} = \beta_{i,2}, b_{i,3} = \log N_i)$ is displayed in Figure 2.10a. The highest ASW of 0.49 is obtained for

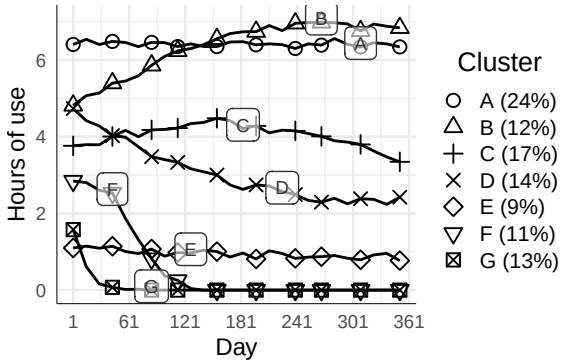
[7] https://CRAN.R-project.org/package=flexmix

Figure 2.10: Case study analysis using feature-based clustering.

(a) Average silhouette width per solution (higher is better).



(b) The identified cluster trajectories.



the seven-cluster solution. The cluster trajectories visualized in Figure 2.10b were obtained by averaging across the respective trajectories. The solution matches the ground truth, demonstrating the ability to recover the underlying clusters when the relevant longitudinal aspects are used.

2.4.4 Mixture modeling

Mixture models describe a distribution in terms of a set of underlying distributions, under the assumption that the distribution comprises different data-generating processes or random variables. Usually, the submodels assume the same parametric distribution, but with different coefficients. An example of a mixture distribution comprising normals was shown in Figure 2.4d on page 18. In a statistical analysis setting, mixture models comprise a set of regression models. Here too, the submodels tend to have an identical specification.

The basic idea behind a longitudinal mixture model is to fit a mixture distribution to the longitudinal observations $\boldsymbol{y}_i$. Thus the mixture model density $f\left(\boldsymbol{y}_i|\boldsymbol{\theta}\right)$ with model parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G)$ is defined by

$$f\left(\boldsymbol{y}_i|\boldsymbol{\theta}\right) = \sum_{g=1}^{G} \pi_g f\left(\boldsymbol{y}_i|\boldsymbol{\theta}_g\right). \tag{2.9}$$

Here, $f\left(\boldsymbol{y}_i|\boldsymbol{\theta}_g\right)$ denotes the conditional density of $\boldsymbol{y}_i$ given that i belongs to cluster g. The cluster membership of individual trajectories is unknown and therefore treated as being probabilistic. The probability of a random subject belonging to cluster g is denoted by π_g, where $0 \leq \pi_g \leq 1$ and $\sum_g \pi_g = 1$. This can also be interpreted as the cluster proportion. The posterior probability of a subject belonging to a given cluster is computed by normalizing the respective cluster density over all clusters by

$$\Pr(\boldsymbol{y}_i|i \in I_g, \boldsymbol{\theta}) = \frac{\pi_g f\left(\boldsymbol{y}_i|\boldsymbol{\theta}_g\right)}{\sum_{g'=1}^{G} \pi_{g'} f\left(\boldsymbol{y}_i|\boldsymbol{\theta}_{g'}\right)}. \tag{2.10}$$

While the cluster assignments are probabilistic, meaning that a subject can belong to any cluster with a certain probability, the subject is usually assumed to belong to the cluster with the highest posterior probability.

The effect of baseline covariates on the cluster membership can be explored by including a multinomial logistic regression component for π_g. For a vector of covariates $\boldsymbol{x}_i$ of subject i, the cluster membership probability is computed by

$$\pi_g(\boldsymbol{x}_i) = \frac{\exp(\boldsymbol{\eta}_g \boldsymbol{x}_i)}{\sum_{g'=1}^{G} \exp(\boldsymbol{\eta}_{g'} \boldsymbol{x}_i)}, \tag{2.11}$$

where $\boldsymbol{\eta}_g$ denotes the multinomial regression coefficients for cluster g. For the purpose of model identifiability, the last cluster is used as the reference, with $\boldsymbol{\eta}_G = 0$. In the remainder of the overview, we assume a model without cluster membership covariates for brevity.

The estimation of these mixture models follow the same approach as the EM algorithm described in subsection 2.4.1.2 on LPA, as this is a type of mixture model as well. The important distinction is that the mixture models presented in this section allow for an arbitrary number of measurements per trajectory, and at arbitrary moments in time.

2.4.4.1 Group-based trajectory modeling

Similar to the concept of methods such as k-means or LPA, group-based trajectory modeling (GBTM[8]) describes the population heterogeneity via a set of homogeneous clusters, where subjects are only represented by their respective cluster trajectory (Nagin and Odgers, 2010a; Nagin and Tremblay, 2005). In contrast, GBTM represents the trajectories using a parametric model. It can be regarded as a multilevel model with nonparametric random effects (i.e., a finite number of random effect values, representing the clusters), which is especially useful when random effects are non-normal or correlated (Rights and Sterba, 2016). The model is easy to interpret due to its distinct cluster trajectories. The method is also referred to as latent-class growth analysis or modeling (LCGA, LCGM), semi-parametric group-based modeling (SGBM), TRAJ[9], and sometimes as nonparametric multilevel mixture modeling (NPMM).

The method originates from the field of criminology. Two decades ago, Nagin and Land (1993) suggested a model for describing developmental trajectories in individuals for whom the number of yearly crimes was measured in relation to age. They proposed a longitudinal Poisson mixture model for separating the trajectories, comprising count data, into distinct clusters. In a later paper, Nagin (1999) presented GBTM as a flexible method for identifying distinct trajectories in a set of longitudinal measurements. Furthermore, models were proposed that assume (censored) normal data or binary data for the observations. Its applications extend further than the domain it was originally created for. GBTM has been applied in the field of psychology, medicine (Nagin and Odgers, 2010a; Franklin *et al.*, 2013), and ecology (Matthews, 2015), among others.

A GBTM describes the trajectories using a linear model. The design vector at time $t_{i,j}$ is denoted by $\boldsymbol{x}_{i,j}$. The cluster trajectories are often modeled using polynomials. As an example, $\boldsymbol{x}_{i,j} = \left(1, t_{i,j}, t_{i,j}^2\right)$ describes a second-order polynomial time trajectory. The

[8] Abbreviated as GTM in some articles.

[9] Named after the macro in SAS, PROC TRAJ.

trajectories as modeled by a GBTM, given membership to a specific cluster g, are described by

$$y_{i,j} = \boldsymbol{x}_{i,j}\boldsymbol{\beta}_g + \varepsilon_{g,i,j}, \quad i \in I_g, \tag{2.12}$$

where $\boldsymbol{\beta}_g$ denotes the cluster-specific regression coefficients, and the residual error $\varepsilon_{g,i,j}$ is assumed to be independently normally distributed with zero mean and variance σ_g^2. The marginal mean is computed by

$$\mathbb{E}(y_{i,j}) = \sum_{g=1}^{G} \pi_g \boldsymbol{x}_{i,j}\boldsymbol{\beta}_g. \tag{2.13}$$

The GBTM parameters and clusters are estimated by maximizing the likelihood of the model for a given number of clusters G using the EM algorithm. Missing observations tend to be assumed missing at random and are therefore ignored. The model can be adapted to fit a wide variety of response distributions. It has also been used to model data under a censored normal, zero-inflated Poisson, logistic, or beta distribution (Jones and Nagin, 2007; Elmer *et al.*, 2018).

Jones and Nagin (2007) proposed the estimation of confidence intervals on cluster membership probabilities and trajectories using Taylor-series expansion. Nielsen *et al.* (2014) proposed an alternative to model estimation and selection using a cross-validation error methodology. Nagin *et al.* (2018) extended the GBTM to account for multiple outcome trajectories, in which the outcomes are assumed to be conditionally independent. This is found to be favorable to the alternative of clustering each outcome separately and then combining the results.

There are a couple of disadvantages to modeling trajectories through polynomials. Firstly, the possible shapes a polynomial may represent is limited, so the model may not be able to fit the longitudinal shape. Secondly, higher-order polynomials tend to overfit the data or produce spurious shapes. Researchers should therefore be careful in interpreting the shapes in detail. As a more reliable alternative, Francis *et al.* (2016) proposed smoothing the cluster trajectories using a cubic B-spline. They observed an improved model fit while allowing for more flexible cluster trajectories.

Overall, GBTM is applicable to ILD in many aspects. The model can handle missing data, observations measured at different times, and estimation is relatively fast due to the low number of parameters involved. In addition, the probabilistic nature of the model has been shown to make it suitable for real-time prediction, where cluster membership and the expected trajectory can be computed as new observations become available (Elmer *et al.*, 2019).

Implementations of GBTM are available in SAS[10] (Jones *et al.*, 2001), Stata[10](Jones and Nagin, 2013), Mplus[11] (Muthén and Muthén, 1998–2012) and OpenMx[12] (Boker *et al.*, 2011), and in R via the `lcmm`[13] (Proust-Lima *et al.*, 2017), `crimCV`[14], `flexmix`[15] (Grün and Leisch, 2008), or `mixtools`[16] (Benaglia *et al.*, 2009) package, among others.

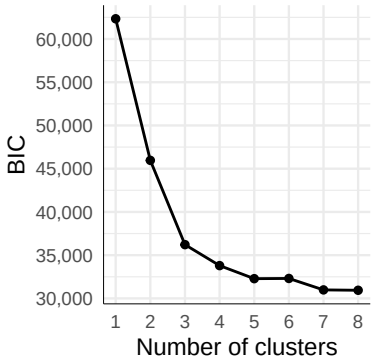
[10]The plugin is available at http://www.andrew.cmu.edu/user/bjones
[11]https://www.statmodel.com
[12]http://openmx.psyc.virginia.edu
[13]https://CRAN.R-project.org/package=lcmm
[14]https://CRAN.R-project.org/package=crimCV
[15]https://CRAN.R-project.org/package=flexmix
[16]https://CRAN.R-project.org/package=mixtools

Table 2.2: Single-cluster analysis using mixed modeling with a normalized time covariate. Here, $\sigma_0, \ldots, \sigma_3$ represent the square root of the diagonal of the covariance matrix Σ.

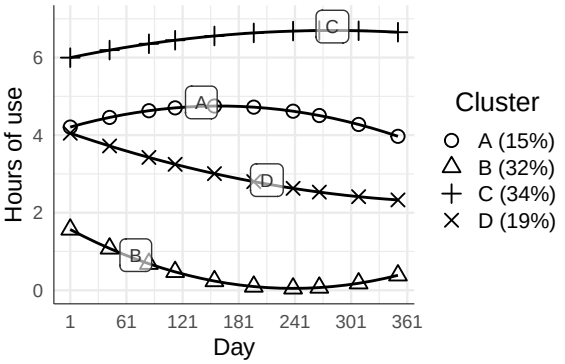
Model degree	σ_0	σ_1	σ_2	σ_3	σ_ε	BIC
0	2.5	-	-	-	0.79	33,433
1	2.2	1.4	-	-	0.63	29,205
2	2.1	4.4	3.2	-	0.58	27,375
3	2.0	8.7	6.3	4.1	0.57	27,146

Figure 2.11: GBTM case study analysis.

(a) BIC per solution (lower is better).



(b) The identified cluster trajectories.



Case study Prior to the GBTM analysis, we investigate the appropriate trajectory representation by evaluating mixed models with different polynomial degrees in the fixed and random effects. We normalize the 26 measurement times by scaling the range from $[1, 351]$ to $[0, 1]$ for numeric stability. The mixed models and GBTMs are estimated with the R package `lcmm` (version 1.7.8), developed by Proust-Lima *et al.* (2017). The model fit and variance components are reported in Table 2.2. The residual standard error and BIC indicate an improved fit with a higher order polynomial. While the model with polynomial representation of degree 3 achieves the best fit, the improvement over the second-degree model is relatively small. In consideration of the linear increase in the number of model parameters with an increasing number of clusters, we settle for a quadratic representation.

We estimate the GBTM solutions using a grid search with 20 random starts to identify a good starting position during model optimization. As depicted in Figure 2.11a, the model fit improves with an increasing number of clusters. Judging from the change in improvement from one solution to the next, a three- or four-cluster solution is preferred. Upon visual inspection of both solutions, we opt for the four-cluster solution due to the addition of the cluster trajectory similar to the Variable users group in the ground truth.

The four cluster trajectories are shown in Figure 2.11b. Overall, this solution adequately captures the heterogeneity of the data. Cluster B (32%) represents the non-users, early drop-outs, and occasional attempters. Cluster C (34%) comprises the slow improvers and good users. The remaining clusters match the ground truth.

2.4.4.2 Growth mixture modeling

Growth mixture modeling (GMM) extends GBTM with the inclusion of parametric random effects, enabling a better fit to the data under the assumption of within-cluster variability (Verbeke and Lesaffre, 1996; Muthén and Shedden, 1999; Muthén *et al.*, 2002; Muthén, 2004). The method is also described as a longitudinal latent-class mixed model, a multilevel mixture model, or a finite mixture of mixed models. GMM has been applied across many domains in the past decade. Although GMM is commonly applied and described in the structural equation modeling framework, we present it in a mixed-modeling approach to be consistent with the notation of the previous sections. The longitudinal observations, conditional on belonging to cluster g, can be described by the linear mixed model specified in Equation 2.2 on page 16 with

$$\begin{aligned} y_{i,j} &= \boldsymbol{x}_{i,j}\boldsymbol{\beta}_g + \boldsymbol{z}_{i,j}\boldsymbol{u}_{g,i} + \varepsilon_{g,i,j}, \quad i \in I_g \\ \boldsymbol{u}_{g,i} &\sim N(0, \Sigma_g) \\ \varepsilon_{g,i,j} &\sim N(0, \sigma^2_{\varepsilon,g}). \end{aligned} \tag{2.14}$$

where the symbols have the same meaning as defined for the mixed model, but are cluster-specific. The marginal mean of a GMM is thus given by

$$\mathbb{E}(y_{i,j}|\boldsymbol{u}_{g,i}) = \sum_{g=1}^{G} \pi_g \left[\boldsymbol{x}_{i,j}\boldsymbol{\beta}_g + \boldsymbol{z}_{i,j}\boldsymbol{u}_{g,i}\right]. \tag{2.15}$$

The model parameters are commonly estimated using maximum likelihood estimation (MLE) via the expectation-maximization (EM) algorithm. Due to the large degrees of freedom, the iterative EM procedure is unlikely to find the optimal solution, and instead tends to converge towards suboptimal solutions. A better solution can be found by fitting the model many times from random starting points and selecting the best fit from these candidates. Alternatively, the solution of simpler models is used as a starting point, e.g., using a GBTM (Jung and Wickrama, 2008).

Although GMM is suitable for ILD much like GBTM, it is considerably slower to compute due to the number of model parameters growing drastically with the number of clusters. Consider that each cluster has a new set of parameters $\boldsymbol{\beta}_g$, Σ_g, and $\sigma^2_{\varepsilon,g}$, in addition to the cluster-specific random variables $\boldsymbol{u}_{g,i}$ (Twisk and Hoekstra, 2012). The model complexity is typically reduced to speed-up estimation by assuming that certain parameters are identical across the different clusters (e.g., the residuals and variances). These challenges also inspired a different approach to performing a covariate analysis. While these could be included into the GMM, for larger datasets it is more practical to estimate an unconditional GMM, followed by a multinomial logistic regression of the covariates based on the subject cluster membership, referred to as a three-step approach (Asparouhov and Muthén, 2014). We address the three-step approach in a more general context in Section 2.5.

Bayesian estimation In a Bayesian approach, the model parameters $\boldsymbol{\theta}$ of the GMM are treated as a random variable, whereas in MLE a point estimate is obtained (Gelman *et al.*, 2013). The posterior distribution of the model parameters given the data can be computed using Bayes' rule

$$\Pr(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{\Pr(\boldsymbol{Y}|\boldsymbol{\theta})\Pr(\boldsymbol{\theta})}{\Pr(\boldsymbol{Y})}, \tag{2.16}$$

where $\boldsymbol{Y}$ denotes the dataset, $\Pr(\boldsymbol{Y}|\boldsymbol{\theta})$ denotes the likelihood of observing the data under the given model, $\Pr(\boldsymbol{\theta})$ denotes the prior information about the model parameters, and $\Pr(\boldsymbol{Y})$ denotes the evidence for the model. As $\Pr(\boldsymbol{Y})$ is constant over $\boldsymbol{\theta}$, it suffices to consider $\Pr(\boldsymbol{\theta}|\boldsymbol{Y}) \propto \Pr(\boldsymbol{Y}|\boldsymbol{\theta}) \cdot \Pr(\boldsymbol{\theta})$ for statistical inference. Bayesian inference is most beneficial when informative priors can be provided, as the ability to incorporate domain knowledge into the parameter estimation through priors improves model estimation under low sample sizes (Gelman *et al.*, 2013). However, specifying informative priors could be challenging in an exploratory cluster analysis setting, especially when a large number of clusters is sought out.

Compared to MLE, Bayesian inference allows for the estimation of more complex models involving a large number of parameters, for which numerical integration is infeasible (Ansari *et al.*, 2000). In a comparison between MLE and Bayesian estimation, Depaoli (2013) found that the Bayesian approach resulted in an improved recovery of the model parameters. Serang *et al.* (2015) demonstrated the improved parameter recovery and smaller standard errors for estimating nonlinear trajectories, applied to reading development trajectories of children, although they noted an increase in convergence problems.

Due to the identical definition of the clusters in the mixture, the cluster ordering (i.e., labels) can change freely during sampling, referred to as the label switching problem. This presents a problem when attempting to interpret the cluster-specific posterior distribution samples. A possible solution to label switching is to add constraints to the model to ensure identifiability, such as enforcing an ordering on the cluster intercepts $\beta_{1,0} < \beta_{2,0} < ... < \beta_{G,0}$ (Sperrin *et al.*, 2010).

Advanced applications Growth mixture modeling is a powerful and flexible statistical approach for exploratory analyses, which is likely why the method has been applied extensively by researchers throughout the past two decades. Moreover, the model is applicable to ILD for the same reasons as GBTM. An example of an ILD application is seen in the work of Shiyko *et al.* (2012), who proposed an approach to applying a Poisson-GMM to ILD to investigate patient's daily number of smoked cigarettes. Many researchers have adapted GMM to meet their analysis needs. We highlight some of the proposed extensions here.

Type of response The method can be applied to different types of data such as binary, categorical, ordinal, count, and zero-inflated data, requiring different distributions for the response (Muthén and Asparouhov, 2009). Proust-Lima *et al.* (2009) demonstrated a joint application of GMM in modeling multiple longitudinal outcomes with time-to-event data. While the response distribution can be determined from the data, the distribution of the random effects is more difficult to establish, as wrongly modeling the within-cluster heterogeneity simply results in additional clusters (Bauer, 2007). The assumption of normally distributed subgroups has been reconsidered in recent years. Alternative distributions such as a skewed-normal or skewed-t have been proposed to account for the skewness and thickness of the tails of the random effects distribution, resulting in a GMM which is more robust to non-normal groups and group outliers (Lu and Huang, 2014; Muthén and Asparouhov, 2015; Wei *et al.*, 2017). A disadvantage of support for

thicker tails is that it results in an even larger overlap between clusters than is already the case for a mixture of normals.

Trajectory representations Many researchers have explored different temporal shapes and structures. Grimm *et al.* (2010) investigated nonlinear trajectories in the reading development of children using specific functions. Nonlinear trajectories have also been estimated using regularized polynomials (Shedden and Zucker, 2008), fractional polynomials (Ryoo *et al.*, 2017), and splines (Marcoulides and Khojasteh, 2018; Ding, 2019). Researchers have also accounted for sudden changes over time using piecewise trajectory representations, referred to as a piecewise GMM (PGMM) (Li *et al.*, 2001). PGMMs have also been proposed to handle multiple change points, change points determined by the model (Liu *et al.*, 2018; Ning and Luo, 2018), and subject-specific change points (Lock *et al.*, 2018). The intervals between change points can also be regarded as a possible state change. In a multiphase or sequential-process GMM (Kim and Kim, 2012; Reinecke *et al.*, 2015), the latent class membership is estimated per interval. State change patterns can then be assessed using latent transition models (Collins and Lanza, 2010).

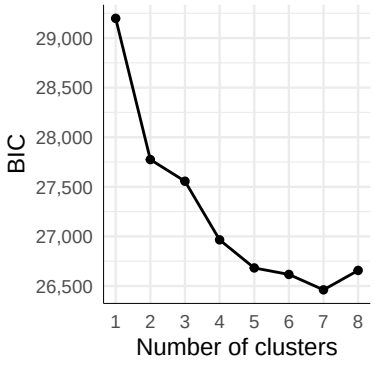
Missing data Longitudinal datasets often comprise missing observations. In most analyses, missing data is assumed to be missing completely at random. However, it is not uncommon for the missing-data mechanism to affect the longitudinal data process, resulting in biased estimates if unaccounted for (Little, 1995). This can happen, for example, in case of loss to follow-up or due to subject-specific factors. Over the past decade, adaptions to GMM have been proposed to account for different missing data mechanisms. For example, Lu *et al.* (2011) applied a Bayesian approach to modeling a GMM where the missing data mechanism is conditional on the cluster membership and observed covariates. A detailed overview of adaptations to model missing-data mechanisms is provided by Enders (2011) and Muthén *et al.* (2011). As the assumptions for the type of missing data are inherently untestable, it is desirable to conduct a sensitivity analysis. Bruckers *et al.* (2018) provide an overview of models for handling non-ignorable subject dropout, and show how the estimation of different missing-data models can be used to determine the reliability of the cluster results in relation to different assumptions.

Software GMM is available through several modeling programs, e.g., Mplus (Muthén and Muthén, 1998–2012), Latent GOLD (Vermunt and Magidson, 2016), and the R packages `OpenMx` (Boker *et al.*, 2011), `lcmm` (Proust-Lima *et al.*, 2017), `mixAK`[17](Komárek and Komárková, 2014), `flexmix` (Grün and Leisch, 2008), and `mixtools` (Benaglia *et al.*, 2009). GMM can be estimated using a Bayesian approach in several software packages, including OpenBUGS[18] (Lunn *et al.*, 2009), Jags[19] (Depaoli *et al.*, 2016), and Stan[20] (Carpenter *et al.*, 2017), all of which have interfaces to R. In R, specific Bayesian models are implemented, for example, in `mixAK` (Komárek and Komárková, 2014), and `brms`[21] (Bürkner, 2017).

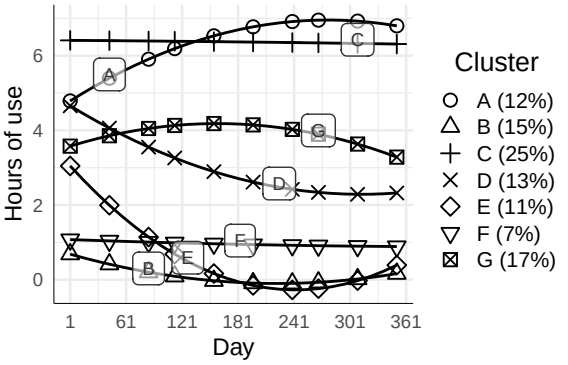
[17]https://CRAN.R-project.org/package=mixAK
[18]http://openbugs.net
[19]http://mcmc-jags.sourceforge.net
[20]http://mc-stan.org
[21]https://CRAN.R-project.org/package=brms

Figure 2.12: GMM case study analysis.

(a) BIC per solution (lower is better).



(b) The identified cluster trajectories.



Case study The GMM analysis follows the same steps as the GBTM analysis, and the same software is used to estimate the model. We therefore refer to Table 2.2 for the exploration of the trajectory shape using a single-cluster mixed model. We employ a quadratic GMM with cluster-specific random patient intercepts, and cluster-independent structured covariance matrices. We found that a grid search with 20 random starts was sufficient for consistently arriving at the best solution. The resulting model BICs are shown in Figure 2.12a, indicating that the best model fit is obtained at the seven-cluster solution. The cluster trajectories thereof are shown in Figure 2.12b, showing a close match with the ground truth insofar the cluster trajectories can be represented using second-order polynomials.

2.4.4.3 Time-varying effect mixture modeling

In regression analysis, the associations between the covariates and outcome are typically modeled as being constant over time. In practice however, associations may change over time, resulting in a lack of understanding of the true temporal association if this change is not accounted for. In a varying-coefficient model (VCM), the dynamic association between covariates is modeled using smooth functions (Hastie and Tibshirani, 1993). VCM has been applied in longitudinal studies, in which covariates are modeled with one or more time-varying coefficient functions denoted by $\beta(\cdot)$. In this form, the model is referred to as a time-varying coefficient model (TVCM), time-varying effect model (TVEM), or dynamic generalized linear model. The individual trajectory is described by

$$y_{i,j} = \sum_{q=0}^{Q} x_{q,i,j}\beta_q(t_{i,j}) + \varepsilon_{i,j}, \tag{2.17}$$

where $x_{0,i,j} = 1$, β_0 denotes the time-varying intercept over time, and β_q denotes the temporal association between the covariate $x_{q,i,j}$ and time. Furthermore, the residuals $\varepsilon_{i,j}$ are assumed to be normally and independently distributed with zero mean and variance σ^2. The coefficient functions are described through smooth continuous functions (i.e.,

the first-order derivative is continuous), and can capture nonlinear longitudinal relations between the covariates and time. Note that in the absence of covariates, the model comprises a single coefficient function that captures the longitudinal trajectory. TVEM is a promising approach for ILD, as the large volume of data enables the identification of more complex dynamic associations (Tan *et al.*, 2012).

Over the years, several approaches have been suggested for the estimation of coefficients for the smoothing functions for the coefficients. Spline regression is used to describe the function by a piecewise polynomial (typically of order 2 or 3) over a given series of intervals (Liang *et al.*, 2003; Hoover *et al.*, 1998). The interval boundaries, referred to as knots, need to be selected carefully based on the data. An alternative approach named spline smoothing does not require selection of intervals, but is much more computationally intensive (Hoover *et al.*, 1998; Hastie and Tibshirani, 1993). A more recent approach involving P-splines takes the middle ground, using a penalty factor to prevent overfitting while ensuring a smooth fit (Song and Lu, 2010; Tan *et al.*, 2012). Splines are described through a linear model, and consequently, a TVEM describes a linear model of which the model parameters can be estimated using ordinary least-squares (OLS).

Mixtures of VCMs or TVEMs have been proposed for handling heterogeneity, where the different groups are represented through clusters-specific coefficient functions (Lu and Song, 2012; Dziak *et al.*, 2015; Huang *et al.*, 2018; Ye *et al.*, 2019). Lu and Song (2012) used an approach similar to GMM, where a random intercept and slope are included to model within-cluster heterogeneity. However, the use of linear random effects in combination with nonlinear cluster trajectories may be limiting, as the nonlinear changes remain homogeneous within cluster. Dziak *et al.* (2015) proposed an alternative model which they named MixTVEM, given by

$$y_{i,j} = \sum_{q=0}^{Q} x_{q,i,j}\beta_{g,q}(t_{i,j}) + \varepsilon_{g,i,j}, \quad i \in I_g. \tag{2.18}$$

The measured outcome $y_{i,j}$ is assumed to be normally distributed when conditioned on the cluster variable. The model is similar to GBTM, but accounts for cluster heterogeneity using an AR-1 model with measurement error. Huang *et al.* (2018) proposed a mixture of VCMs with flexible mixing proportions and dispersion, enabling the modeling of these aspects over a covariate, e.g., time.

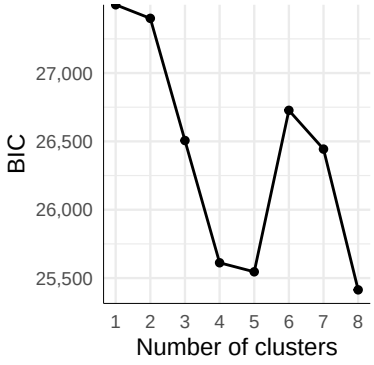
The parameters of the model can be estimated using the EM algorithm (Dziak *et al.*, 2015) or a Bayesian approach (Lu and Song, 2012). The optimization procedure for MixTVEM is initialized by assigning random posterior probabilities to the classes. Dziak *et al.* (2015) recommend to run the procedure for at least 50 random starts as the optimization may converge on different solutions, or fail to converge altogether. Due to the needed repeated runs, the tuning of the penalty factor, and the relative complexity of the model, the method is highly computationally intensive to estimate, as noted by Yang *et al.* (2019).

Case study The MixTVEM models are estimated using the R code provided by Dziak *et al.* (2015)[22] (version 1.2). P-splines of a third degree polynomial order are used with six interior knots, spaced equally over time. The model is fitted from 20 random starts

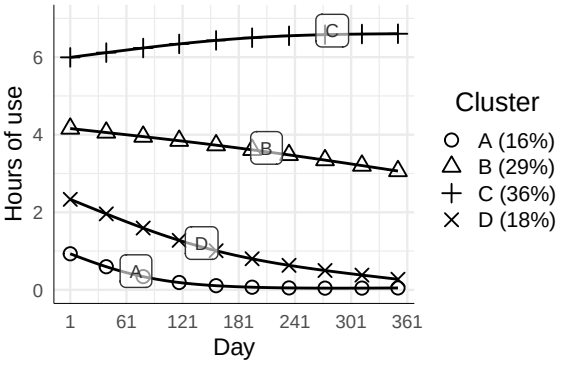
[22] https://github.com/dziakj1/MixTVEM

Figure 2.13: MixTVEM case study analysis.

(a) BIC per solution (lower is better).



(b) The identified cluster trajectories.



to obtain good starting conditions, although on a few occasions, a rerun was needed due to convergence problems. Moreover, the single-cluster estimation consistently failed due to observations with zero variability from the non-users, which we resolved by adding a negligible amount of perturbation to the measurements with zero hours. The BICs of the selected solutions are depicted in Figure 2.13a, showing different levels of model fit to the data across the number of clusters. We select the solution involving four clusters as it best captures the different patterns of change over time.

The four cluster trajectories are visualized in Figure 2.13b. Cluster C (36%) comprises the good users and the slow improvers. Cluster B (29%) represents the variable users and slow decliners. Cluster A (16%) and D (18%) comprise the non-users, early drop-outs, and occasional attempters. However, the presence of the occasional attempters appears to have affected both cluster trajectories, such that neither matches the ground truth.

2.4.5 Number of clusters

The determination of the number of clusters is a prominent topic in the field of cluster analysis, as the number of clusters is usually part of the model definition and can greatly affect the resulting solution. However, there is no consensus on how to identify the true number of clusters. This is largely attributable to the different types of cluster analyses; each having different purposes, expectations, and applications (Von Luxburg *et al.*, 2012). The identification of the number of clusters is part of a broader search for the appropriate cluster model, which we shall refer to as model selection. Interestingly, considerable attention is given in the literature to the identification of the number of clusters, over the more general topic of ensuring the overall best model specification, referred to as model selection. This is arguably justifiable in a longitudinal context under the assumption that the trajectory models are sufficiently flexible. We summarize the many approaches and metrics used for identifying the number of clusters.

Model metrics Although no tests exist for the number of clusters or the presence of clusters, approximate likelihood ratio tests (LRT) enable researchers to test whether the

model with $G + 1$ clusters describes the data statistically significantly better than the identically specified model with G clusters. Commonly used variants are the Vuong-Lo-Mendell-Rubin (VLMR) LRT (Lo *et al.*, 2001), the adjusted Lo-Mendell-Rubin (aLMR) LRT (Lo *et al.*, 2001), and the bootstrap LRT (BLRT) (McLachlan and Peel, 2000). These approaches are useful on smaller datasets for preventing overfitting. However, the tests tend to result in the identification of too many clusters (i.e., overextraction) on large datasets, where smaller changes between models are statistically but not practically significant (Grimm *et al.*, 2017). A similar concept is seen in difference-like criteria, which measure the relative improvement between successive cluster solutions (Vendramin *et al.*, 2010). Along similar lines, Grimm *et al.* (2017) have applied k-fold cross-validation for model selection based on how well the model represents previously unseen data (in terms of the likelihood).

The most applied approach involves the estimation of a cluster model for a range of number of clusters. A metric is then used to identify which of the models provides the best fit. Information criteria strike a balance between model fit (the likelihood) and model complexity (the number of model parameters). These model metrics can also be used to compare across model specifications, selecting the model that minimizes the metric.

Metrics for identifying the number of clusters have been studied extensively for GMM and GBTM (Nylund *et al.*, 2007; Feldman *et al.*, 2009; Klijn *et al.*, 2017; Tofighi and Enders, 2008). Overall, the findings are mixed, likely due to the different settings (e.g., sample size, cluster separation, noise) under which these evaluations have been performed (Grimm *et al.*, 2017). Overall, the BIC is commonly used for the class enumeration in GBTM and GMM. The BLRT has been demonstrated to be a reliable alternative (Nylund *et al.*, 2007; McNeish and Harring, 2017).

Malsiner-Walli *et al.* (2016) proposed a metric based on the occurrence of empty clusters in a mixture model with many clusters, where the true number of clusters is determined based on the number of non-empty clusters. This approach has the advantage of only requiring a single model to be fitted. Nasserinejad *et al.* (2017) experimented with this metric for different thresholds for the number of trajectories that constitute a non-empty cluster. Another metric of interest is the entropy of the posterior probability matrix, as a measure of cluster separation (i.e., probabilities should be close to either zero or one).

Bayesian metrics Different criteria have been proposed for models estimated through Bayesian inference. They make use of the posterior distribution of the model coefficients. One of the more commonly used criteria is the deviance information criterion (DIC), introduced by Spiegelhalter *et al.* (2002). Its use has not been without criticism. For example, there exist multiple definitions of the DIC, each with a different interpretation (Celeux *et al.*, 2006). Spiegelhalter *et al.* (2014) have summarized and addressed the concerns. Recent alternative criteria are the widely applicable AIC (WAIC), and Pareto-smoothed importance sampling using leave-one-out cross validation (PSIS-LOO) (Vehtari *et al.*, 2017).

Cluster criteria Criteria for cluster algorithms tend to assess the solution based on the underlying data and assume a hard partitioning of the data. The advantage of such an approach is that it is independent from the method that was used, making it possible to assess the model fit across cluster methods. The criteria tend to contrast the within-cluster

variability against the between-cluster variability to assess the separation between clusters, as seen, e.g., the Calinski-Harabasz (CH) and Davies-Bouldin criteria. As an example, Todo and Usami (2016) found the CH criterion to perform better for model selection than BIC in a latent profile analysis. Another commonly used criterion is the ASW. A comprehensive overview of commonly used cluster criteria is provided by Vendramin *et al.* (2010).

Upper bound The upper bound on the largest number of clusters to be evaluated is based on multiple factors. Firstly, prior knowledge may give researchers reasons to expect the true number of clusters to be below a certain number. Computational factors are also at play (Nasserinejad *et al.*, 2017), as the model computation time scales non-linearly with an increasing number of clusters for complex models, making the evaluation of a larger number of clusters impractical. Along similar lines, the increasing model complexity with an increasing number of clusters tends to result in more frequent occurrences of convergence issues. The largest number of clusters that can be estimated is also limited by the sample size, considering that all submodels must comprise enough trajectories to obtain reliable estimates (Sterba *et al.*, 2012). Studies involving a small sample size are therefore naturally limited to identifying a lower number of clusters. Similarly, having many clusters limits the power of a post-hoc cluster comparison.

Subjective assessment Researchers have argued against the optimization of a sole metric for the identification of the number of clusters, as it is rather mechanical in nature, and disregards the domain-dependent aspect of the analysis (Nagin *et al.*, 2018). Moreover, the commonly used metrics tend to focus on discerning a sufficiently improved model fit, however, a better model may fit aspects of the heterogeneity which are not of interest for the purpose of the analysis (Van Den Bergh and Vermunt, 2019). This issue can occur in datasets with considerable overlap between clusters, where the introduction of additional clusters may consistently improve the model fit, albeit with diminishing returns. Due to the non-linear nature of these diminishing improvements, there tends to be a point or region where the marginal improvement drops. The identification of this turning point, representing the preferred number of clusters, creates room for subjectivity into the decision. This approach, often assessed visually, is commonly referred to as the "elbow method". It is used, for example, by Dziak *et al.* (2015) in their MixTVEM analysis, to assess the relative improvement in terms of the BIC.

Arguably, the choice of metric or metrics involves a domain-dependent decision. As the choice of the best metric may not be clear-cut, taking into consideration multiple metrics can provide a more reliable result (Ram and Grimm, 2009). However, because fit metrics capture different aspects of the model fit, it is inevitable that some of the metrics disagree on the optimal number of clusters.

Hierarchical models There is a practical limit to the number of clusters that can be used to approximate the heterogeneity, for it becomes increasingly difficult to produce unique labels for each of the clusters (Sterba *et al.*, 2012). Instead of identifying an independent set of clusters, one can search for a cluster tree hierarchy using a hierarchical cluster algorithm, where each cluster is further explained in terms of subclusters. In this way, an arbitrary level of granularity can be obtained up to the subject level. This approach can be estimated through a cross-sectional or feature-based approach using an

agglomerative hierarchical cluster algorithm. In recent years, Van Den Bergh and Vermunt (2017) proposed a top-down parametric approach based on GBTM, named latent-class growth trees (LCGT). They identified the root of the hierarchy using a standard GBTM analysis and metric, but subsequent clusters are fitted using a two-cluster GBTM until no more significant improvement is obtained. Another advantage of the approach is that the tree accounts for classification error, as opposed to the hard partitioning used in traditional hierarchical cluster algorithms. Furthermore, covariates can be included for comparison or cluster membership prediction through a three-step estimation approach (Van Den Bergh and Vermunt, 2019).

Nonparametric mixture models A promising alternative to the post-hoc identification of the number of clusters, or model selection in general, is seen in nonparametric modeling, where only a single model is estimated. Here, the model complexity is grown as needed to represent the data in an infinite parameter space. In such a model, the number of clusters G is part of the model parameters to be estimated (Richardson and Green, 1997; Green and Richardson, 2001). This model can be realized using a Bayesian approach by placing a Dirichlet process (DP) (Ferguson, 1973) prior on the number of clusters G. The DP mixture model (DPMM) describes the observations as a function of the model parameters $\boldsymbol{\theta}$ provided by the DP (Lo, 1984). It has been applied to clustering gene expression data (Sun *et al.*, 2017). Heinzl and Tutz (2013) demonstrated that a DPMM could also be estimated with an EM algorithm instead of MCMC, although they did not compare between estimation algorithms. DPMMs can be estimated in R for example via the `DPpackage`[23] by Jara *et al.* (2011) or the `BClustLonG`[24] package by Sun *et al.* (2017).

2.5 Guidelines for conducting a longitudinal cluster analysis

Many decisions are involved in a longitudinal cluster analysis. The need for guidelines comes not only from obtaining reliable results, but also comes from ensuring proper reporting to enable reproducible research. Unfortunately, the exploratory and domain-dependent nature of clustering inevitably means that there is no single formal process or method that covers all applications and purposes (Nagin and Odgers, 2010a). Instead, the analysis should be adapted to the research questions or intended application of the model. Guidelines can still play a role here, as there are common themes to any longitudinal cluster analysis.

We broadly outline the typical aspects and approaches involved in a longitudinal cluster analysis. We focus on the guidance given by researchers on the topic of mixture modeling, as these typically parametric models tend to involve many decisions (Nagin and Odgers, 2010a). We summarize the steps as follows:

1. **Analyze the model variables.** The type of longitudinal response (e.g., categorical, ordinal, continuous) and the distribution thereof (e.g., normal, Poisson, zero-inflated, truncated) should be understood. In addition, the distribution of the covariates should be investigated, as outliers may skew the results.

[23] https://CRAN.R-project.org/package=DPpackage
[24] https://CRAN.R-project.org/package=BClustLonG

2. **Investigate the missing data mechanism.** This step is crucial for ILD, where the varying continuous measurement times may be underlying to patterns of missingness. An advantage of clustering is that for a missing data mechanism related to the longitudinal outcome, this is handled by the clusters. Data is therefore usually assumed to be missing at random (MAR). Missing not at random (MNAR) data has been handled by using pattern mixture models.
3. **Model the data as a single cluster.** Prior to the cluster analysis, it is good practice to understand the performance of the single-cluster case (Ram and Grimm, 2009; Van de Schoot *et al.*, 2017). If the single-cluster model achieves a good fit, there may be little added value from complicating the analysis by introducing additional clusters. Alternatively, the heterogeneity could be assessed by comparing the coefficients obtained from separate models for each trajectory if the sample size allows for it. The single-cluster model may also be of use for identifying the approximate trajectory shape.
4. **Provide a rationale for clustering.** Ideally, the analysis is justified by theory or domain knowledge (e.g., previous studies) that strongly hint at the existence of clusters. This step also pertains to the way the clusters are interpreted, i.e., whether to use direct or indirect clustering.
5. **Identify the best model.** This step is by far the most intricate, both in terms of number of decisions and computation time. In view of exploring the data heterogeneity, it is preferable to start with a model that does not account for covariates other than time (Vermunt, 2010), referred to as the unconditional model. An example of method and model selection is found in the analysis by Feldman *et al.* (2009). The choice of method, model specification, estimation method, and the selected number of clusters all affect the model fit to the data. As such, arriving at the final model may involve several iterations of the following substeps:
   (a) **Choose the cluster method.** The methods have different strengths and limitations in terms of, e.g., flexibility in modeling trajectory shapes, capability to model heterogeneity, sample size requirements, and computational scalability. It is worthwhile to weigh these aspects in deciding on the method to use.
   (b) **Choose the estimation approach and method.** Cluster models can be challenging to estimate, as estimation algorithms may be unable to identify the optimal solution in the vast parameter space. It is therefore recommended to perform repeated runs with different random starting values, and to select the model with the best fit from the candidate models (Jung and Wickrama, 2008; Sher *et al.*, 2011; McNeish and Harring, 2017). Moreover, it is worthwhile to experiment with different estimation methods for improved convergence (e.g., by increasing the number of iterations) and computational efficiency. The estimation algorithm may fail to converge, or the identified solution is invalid due to various reasons (e.g., out-of-bound coefficients, or empty clusters).
   (c) **Specify and select the most appropriate model.** This typically manual process involves many decisions, including the specification of the trajectory shape (e.g., polynomial, or spline), the distribution of the response variable, any covariates, the shared parameters between clusters (e.g., the covariance matrix), and cluster heterogeneity. These decisions can be guided by domain knowledge or by metrics for assessing the improved fit to the data. In particular, the trajectory shapes can be explored using a cluster model with a nonparametric representation of the cluster trajectory (Todo and Usami, 2016). Alternatively, the task of model

specification and selection can be considerably simplified by using regularized or nonparametric models.

(d) **Identify the number of clusters.** There are many approaches to identify the number of clusters, as described in 2.4.5. Typically a forward selection approach is used where a cluster model is fit and evaluated for an increasing number of clusters (Van de Schoot *et al.*, 2017). One or more metrics, possibly taking into account domain knowledge, are used to gauge the best number of clusters.

(e) **Assess the model adequacy.** The model fit can be assessed from the residual observation errors of the model, which may reveal structural deviations (Wang *et al.*, 2005; Feldman *et al.*, 2009; Lennon *et al.*, 2018). Adequacy may also be considered in terms of model parsimony, as similar clusters or clusters representing only a small proportion of the trajectories add little value to the overall model fit. The separation between clusters can be evaluated through the cluster membership probability matrix (Nagin, 2005), or by comparing the cluster trajectories and the variability within clusters (either visually or by the means or coefficients) (Feldman *et al.*, 2009; Nagin and Odgers, 2010a; Lennon *et al.*, 2018). It is also worthwhile to assess the standard errors or confidence interval of the model coefficients for meaningful effects.

(f) **Validate the model.** Longitudinal cluster models can involve many parameters as the number of parameters scales linearly with the number of classes, and thus the models are sensitive to overfitting (i.e., may not generalize well) on small datasets. If a model is estimated on random subsets of the data (e.g., via bootstrapping) and yields the same solution, this is indicative that the estimation of the model is robust. Preferably, the model is evaluated on a holdout (i.e., validation) sample (Frankfurt *et al.*, 2016), or using a k-fold cross-validation approach. Here, the data is split into k folds, where $k-1$ folds are used for training, and the remaining fold is used for testing. It is a useful approach for model evaluation or selection under a more limited sample size (Grimm *et al.*, 2017). Overall, we observe few examples in literature where this step is performed, nevertheless, it is advisable to assess the robustness of the selected model, as an overspecification or overextraction of the number of clusters may result in a model that does not generalize well.

6. **Analyze covariates.** In many analyses, the association of the longitudinal patterns with other variables is of interest. Covariates may be included to explain the cluster membership or the variability within and between clusters. There are different ways to go about analyzing these effects. In a one-step approach, the covariates are included in the model specification in step 5c. The inclusion of covariates into the model results in a more complex model which may be difficult to estimate, leading to convergence issues or long estimation times. Moreover, the interpretation of the identified longitudinal patterns becomes more difficult, as the clusters are based on more than the longitudinal change over time. In a standard three-step approach, the longitudinal cluster model is first estimated without covariates to establish the underlying latent groups. In the second step, individual trajectories are assigned to a cluster. In the third step, the covariates are analyzed. The last step can be approached in several ways. A post-hoc analysis for comparing covariates between clusters is commonly done either by comparing the means of covariates between clusters using ANOVA, or by predicting cluster membership using multinomial logistic regression. However, it is important to correct for the uncertainty in cluster assignments when comparing covariates between

clusters (Vermunt, 2010; Bakk *et al.*, 2013). A more detailed overview of the different estimation approaches is given by Van de Schoot *et al.* (2017).

7. **Interpret the findings.** The implication of the identification of clusters depends on the type of cluster application. A substantial overlap between clusters may still yield meaningful findings in an indirect application yet discredit the existence of truly distinct clusters for a direct application of clustering. Similarly, a predictive application of the model with high accuracy depends on a large separation between clusters. Most importantly, researchers should consider whether the identified clusters or differences between clusters are statistically and practically meaningful.

With so many decisions involved in the analysis, reporting these decisions is of the utmost importance. Van de Schoot *et al.* (2017) developed a comprehensive 21-item checklist based on the consensus of 27 experts, referred to as the guidelines for reporting on latent trajectory studies (GRoLTS), with the aim of improving the transparency and replicability of the analysis. Complementary to the guidelines summarized above, the checklist recommends reporting the software and version that was used to perform the analysis, and to make the analysis source code available. While we will not repeat the other items, we encourage the reader to read the GRoLTS in full.

Van de Schoot *et al.* (2017) conducted a preliminary analysis of the state of reporting in the literature by applying GRoLTS to a selection of studies. They selected 38 papers that used latent-class trajectory modeling for identifying patterns of post-traumatic stress symptoms after a traumatic event. On average, the papers only met nine of the requirements, with the most complete paper meeting fifteen requirements. We believe these findings help to quantify the broader problem across domains of a lack of sufficient reporting. Guidelines such as GRoLTS are therefore valuable and practical tools towards achieving greater transparency, with more interpretable and reproducible findings.

2.6 Discussion

The case study highlights the differences and similarities between the evaluated approaches to longitudinal clustering. The most apparent contrast is the different number of clusters of the best solutions (either determined by a cluster metric or manual assessment). The discrepancy is largely attributable to the different trajectory representations and within- and between-cluster assumptions of the methods. All methods converged on a solution for each of the requested number of clusters. Moreover, the solutions for four clusters or less were highly similar across methods.

The synthetic case study data comprised considerable between- and within-patient variability. Despite this, the relatively straightforward KML and LLPA approaches yielded useful solutions. While LLPA uses the same nonparametric representational approach as KML, the identified cluster trajectories were different. LLPA does account for variability at each day allowing it to distinguish trend on the basis. In the case study that resulted into detecting drop-outs from attempters. Both methods are fast to compute and involve a minimal number of modeling decisions and are therefore practical approaches for quickly obtaining a sense of the variability in trajectory shapes in a heterogeneous dataset. There are similarities to the solutions of KML and GBTM, where KML is preferably for non-linear trajectories (Genolini and Falissard, 2010). However, the ability to incorporate domain knowledge into a GBTM analysis makes it suitable to assess heterogeneity even under small sample size (Feldman *et al.*, 2009; Twisk and Hoekstra, 2012).

The solutions found by GBTM and MixTVEM were similar. However, MixTVEM is more flexible yet conservative in the trend shapes due to its regularization, which is generally preferable. The differences observed between the GBTM and GMM solutions demonstrate the importance of the model specification. Because of the large variation in intercept between patients, GBTM needs more clusters to represent the many different patient trajectory intercepts, whereas GMM can accommodate larger variability in intercepts into a single cluster, leaving more clusters to model other temporal differences (e.g., slope). However, this advantage comes at the cost of a more complex model, resulting in significantly longer computation times, and possibly convergence problems, as evident from the considerably longer computation time of GMM over GBTM (Feldman *et al.*, 2009; Twisk and Hoekstra, 2012). In a comparison between KML and GMM, it was found that GMM is preferred (Twisk and Hoekstra, 2012); this was the case even for a small sample size (Martin and von Oertzen, 2015). In contrast, under the presence of homogeneous subgroups, Verboon and Pat-El (2022) found that KML performed better than GMM for datasets with few observations per trajectory, and marginally better under more sufficient data conditions. Overall, in the case study, the feature-based approach and GMM most closely approximated the true group trajectories from which the data was generated.

With the relatively recent attention for ILD, the number of studies evaluating the methods on this type of data is limited, however. This is unfortunate as ILD presents new challenges with respect to the volume of data, missing data, model complexity, and higher computational demands. Many methods scale poorly with an increasing number of clusters, placing practical limitations on the model complexity and volume of the data. In case of large sample size or large number of observations, this provides a serious practical limit on the maximum number of clusters that can be estimated. An almost inevitable problem associated with ILD is the missingness of data. Patterns for missingness. We only briefly touched upon this topic.

Due to the broad scope of this tutorial, we cannot possibly cover all areas of research on methods for longitudinal clustering. Nevertheless, we do wish to mention some of these unaddressed areas. We restricted the scope to a single outcome, whereas for example, KML, GBTM and GMM have extensions that support multivariable longitudinal outcomes, also referred to as joint trajectories. Furthermore, with the aim of presenting the commonly used approaches to longitudinal clustering, we may have omitted several alternative approaches. For example, we only briefly touched upon the field of functional data analysis. This is a class of methods that attempt to represent the data in terms of smooth functions, a method to which TVEM is related. There has been an increasing interest in further modeling sources of variation in the data by modeling subject-specific variability in addition the mean level, referred to as joint mean-variance modeling. As seen in the LLPA case study demonstration, this can have an impact on the identified clusters. In other applications, trajectories may be expected to change cluster membership over time. Here, the clusters represent different unobserved states in which the subject resides over time. Here, a latent transition analysis can be used to model the transitions between clusters (Collins and Lanza, 2010).

2.7 Summary

The area of longitudinal clustering has gained much traction over the past two decades. We have attempted to present a comprehensive guide on how longitudinal cluster analyses

can be conducted, with an emphasis on the different methods which are available for this purpose. Clustering is a powerful tool for exploratory purposes, but such analyses should be performed thoughtfully. We encourage researchers to experiment with different methods and model specifications to identify the most appropriate model for the data, and to report the steps and decisions that were part of the analysis to ensure interpretable results.

Supplementary materials

The dataset and R code used in each of the examples is available online at https://github.com/niekdt/demo-clustering-longitudinal-data.

Appendix

2.A Strengths and limitations per approach

Table 2.3: High-level comparison between approaches.

Approach	Strengths	Limitations
Cross-sectional clustering	• Fast to compute • Algorithm implementations are widely available • Nonparametric cluster trajectory representation	• Observation moments must be aligned across trajectories • Requires complete data • Sensitive to measurement noise (Green, 2014)
Distance-based clustering	• Versatile; many available distance metrics, which could also be combined • The distance matrix only needs to be computed once • Fast to evaluate for a large number of clusters	• Only practical up to a limited number of trajectories, as the number of pairwise distances to compute grows quadratically with the number of trajectories • No robust cluster trajectory representation (centroid trajectory may not be insightful) • Some distance metrics require aligned observations (e.g., Euclidean)
Feature-based clustering	• Versatile; longitudinal features can be arbitrarily combined into a trajectory model • Fast to compute • Can incorporate domain knowledge • Compact trajectory representation	• Generally requires ILD to ensure a reliable estimation of the features per trajectory • Feature estimates may be unreliable for trajectories that cannot be represented
Mixture modeling	• Parametric cluster trajectory representation • Versatile; choice of latent-class model, trajectory model, latent-class membership model • Compact trajectory representation • Relatively low sample size requirement, both in number of trajectories, and number of observations per trajectory (Martin and von Oertzen, 2015) • Domain knowledge can be incorporated • Can assess the association of external variables or distant outcomes	• Computationally intensive • Number of parameters typically scales linearly with the number of clusters • The estimation procedure may not converge to a good solution; many random starts are needed

2.B Strengths and limitations of mixture models

Table 2.4: High-level comparison between the described mixture models.

Approach	Relative strengths	Relative limitations
GBTM	• Fast to compute • Few parameters • Easy to interpret	• Sensitive to outliers • Poor fit, as individual trajectories are not modeled • Tends to overestimate the number of clusters (Twisk and Hoekstra, 2012)
GMM	• Within-cluster heterogeneity • Fewer clusters needed to represent heterogeneity (Muthén and Asparouhov, 2015) • Random effects allow for cluster trajectories with a lower emphasis on, e.g., intercept. • Forecast individual trajectories	• Slow to compute (Twisk and Hoekstra, 2012) • Requires many random starts (McNeish and Harring, 2017) • Convergence issues (Twisk and Hoekstra, 2012) • Clusters can overlap considerably (Feldman *et al.*, 2009) • Sensitive to the specified distribution of the random effects
MixTVEM	• Easy to interpret • Assess time-dependent association of external variables • Penalized splines result in less spurious temporal patterns	• Slow to compute (Yang *et al.*, 2019) • Requires tuning of penalization factor • Convergence issues (Yang *et al.*, 2019)

Chapter 3

A comparison of methods for clustering longitudinal data with slowly changing trends

N.G.P. Den Teuling, S.C. Pauws, E.R. van den Heuvel
Communications in Statistics - Simulation and Computation. 2021.
DOI: 10.1080/03610918.2020.1861464

Abstract
Longitudinal clustering provides a detailed yet comprehensible description of time profiles among subjects. With several approaches that are commonly used for this purpose, it remains unclear under which conditions a method is preferred over another method. We investigated the performance of five methods using Monte Carlo simulations on synthetic datasets, representing various scenarios involving polynomial time profiles. The performance was evaluated on two aspects: The agreement of the group assignment to the simulated reference, as measured by the split-join distance, and the trend estimation error, as measured by a weighted minimum of the mean squared error (WMMSE). Growth mixture modeling (GMM) was found to achieve the best overall performance, followed closely by a two-step approach using growth curve modeling and k-means (GCKM). Considering the model similarities between GMM and GCKM, the latter is preferred for large datasets for its computational efficiency. Longitudinal k-means (KML) and group-based trajectory modeling were found to have practically identical solutions in the case that the group trajectory model of the latter method is correctly specified. Both methods performed less than GMM and GCKM in most settings.

3.1 Introduction

The connectivity, storage, and sensor solutions of today enable researchers to collect many data points from subjects over any period of time. The larger volume of data collected presents both new opportunities and challenges for longitudinal data analysis when it comes to understanding the data. Notably, a higher number of subjects allows for a data-driven exploration under the assumption of heterogeneity for subgroups of subjects with different trends (i.e., group trajectories). It is then important to profile subjects into different subgroups. While longitudinal cluster analyses have typically comprised only a small number of repeated measurements over time per subject (e.g., less than ten), there is a growing availability of high-frequent longitudinal datasets, referred to as intensive longitudinal data (ILD) (Walls and Schafer, 2006). This type of data enables the estimation of subject-specific trajectories, especially under the presence of high within-subject or between-subject variability. Moreover, the increased number of observations allow for the estimation of more time-sensitive changes.The application of longitudinal clustering spans many domains, including criminology, sociology, medicine, and ecology. Recent examples of applications include the identification of subgroups with different cigarette smoking patterns with the aim of predicting health outcome (Lee *et al.*, 2016), and describing adolescent substance use trajectories and its association to leisure experience (Weybright *et al.*, 2016).

Longitudinal data often comprises trajectories with different observation times, or a different number of observations. ILD comes with additional challenges over repeated measurements data, such as the high volume of data, modeling the dynamics or volatility of trajectories, and accounting for strong correlations due to measurements being close in time.

As demonstrated by past ILD applications, traditional methods are generally applicable to ILD despite the increased volume of data, although they may not address all challenges. For example, Shiyko *et al.* (2012) applied growth mixture modeling to ILD for the flexible identification of patterns of smoking cessation behavior of up to 29 days to account for the correlation between observations. Babbin *et al.* (2015) explored patterns of ILD comprising daily therapy usage among patients undergoing sleep apnea treatment during the first six months of therapy. They used a non-parametric trajectory representation, allowing for high flexibility in the shape of the group trajectories. Lastly, Ernst *et al.* (2019) analyzed ecological momentary assessments of subjects, assessing their emotional state three times per day over a period of 30 days. Subgroups with different emotion dynamics were discovered by clustering subjects based on individual vector autoregressive model coefficients.

The methods that have been introduced over the past two decades for the purpose of longitudinal clustering can be divided into three categories: Firstly, the naive approach clusters on the observations, in which the temporal relation between the measurements is not modeled. Secondly, a two-step approach that first describes subject trajectories in terms of a statistical model or other metrics (which can be regarded as dimensionality reduction), and then clusters on the model parameters. Lastly, the mixture model approach describes the clusters using a mixture of statistical models (Muthén and Shedden, 1999).

Considering the different methods for longitudinal clustering that are available, the question of which method is preferred for a given context arises naturally. In most published applications, the rationale for selecting a particular longitudinal cluster method

is not provided. This could be because alternative methods were not considered, or because the existing body of work on comparisons between methods did not address the relevant context.

Some combinations of methods have been compared, with contradicting findings, suggesting that the optimal method depends on the scenario being considered. Martin and von Oertzen (2015) compared growth mixture modeling (GMM) against naive approaches such as longitudinal k-means (KML) on synthetic data comprising five repeated measurements, and two or three groups. They found that GMM outperforms the other methods even for small sample size. Feldman *et al.* (2009) preferred the group-based trajectory model (GBTM) over GMM due to the complexity of the latter, and the convergence problems that arise from it (Frankfurt *et al.*, 2016). They noted similarities in performance between longitudinal latent class analysis (LLCA) and GBTM, in contrast to Twisk and Hoekstra (2012), who found LLCA to be more similar to KML and a two-step approach involving clustering the random effects of a mixed model. Overall, there seems to be a preference for mixture-based methods, but considering the different approaches to mixture models, the results are not conclusive.

Most of the comparison studies investigate longitudinal data involving 4-6 repeated measurements, with few studies exceeding 10 measurements. It is questionable whether these findings generalize to ILD. For example, having an increased number of observations per trajectory enables a more reliable estimation of longitudinal change under a higher degree of variability, which some methods will benefit more from than others. Moreover, with a growing number of observations per trajectory, a faster but less data-efficient method may become preferable over a better but considerably more computationally demanding method.

In a recent study, Verboon and Pat-El (2022) compared the performance of KML, GMM, and a three-step approach referred to as *traj* in terms of the recovery of the latent classes and the membership of the trajectories. They simulated stable, linear, and quadratic group trajectories, with the trajectories deviating at random at each measurement moment from the respective group trajectory. Under this scenario of homogeneous groups, KML was found to perform the best, with GMM performing nearly equally well under enough observations (10). The performance of the *traj* method was found to be consistently lower than the other two methods.

We contribute to the existing body of work by evaluating the performance of five longitudinal clustering methods in an exploratory setting, applied to many scenarios comprising group trajectories that smoothly and slowly change over time. The methods are longitudinal k-means, a mixed-effects model combined with k-means, group-based trajectory modeling, growth mixture modeling, and time-varying effect mixture modeling. These methods are chosen because they take different approaches to clustering, are commonly used in applications, or are applicable in an exploratory setting without prior knowledge on the clusters. We investigate how well the methods are able to identify the underlying groups (in terms of subject assignments) and group trajectories in each of these scenarios. In addition, we assess the sensitivity of the methods to different forms of heterogeneity. Specifically, we simulate different forms of within-group heterogeneity, using normal and log-normal distributions for the random effects. The scenarios involve many permutations on the different levels of within-group variability, sample sizes, number of observations, and levels of heteroskedasticity of the residual variance. Lastly, we study the effect of a proportional measurement error on the model estimation and assess the

reliability of selecting the correct number of groups per method. The simulated datasets comprise heterogeneous subgroups with varying degrees of overlap, described by quadratic group trajectories. This establishes a ground truth against which the output of the methods can be evaluated. The comparison also serves as a benchmark for the computation time with respect to the number of trajectories, number of observations, and number of groups. In view of the exploratory nature, the mixture methods are all estimated using maximum likelihood estimation instead of a Bayesian approach. In addition to the simulation study, a case study involving therapy compliance of sleep apnea patients is included to relate the findings from the simulations to a real-life setting.

The rest of the chapter is organized as follows. Section 3.2 briefly describes the selected methods. The simulation scenarios are described in Section 3.3. In Section 3.4, the simulation results are reported, along with the description and results of the case study. The resulting findings and recommendations are discussed in Section 3.5, and conclusions are given in Section 3.6.

3.2 Methods

We denote a trajectory of subject $i \in I$ of the available set of subjects I with T measurements by $\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,T})$, where the measurement $y_{i,j}$ is recorded at time $t_{i,j}$. We denote the ILD $\psi_i(t)$ for individual $i \in I$ by $\mathbb{E}y_{i,j} = \psi_i(t_{i,j})$ with $y_{i,j}$ the measurement at time $t_{i,j}$, with $j = 1, 2, ..., T$, and with ψ_i a continuous function $\psi_i : \mathbb{R}_+ \to \mathbb{R}$. We will focus on polynomial time profiles: $\psi_i(t) = \sum_{r=0}^{p} \beta_{i,r} t^r$. Alternatively, we may also study piecewise linear profiles when we apply naive clustering methods, but these profiles would be developed over groups of participants. At a subject level the piece-wise linear model is defined by

$$\psi_i(t) = \sum_{r=1}^{T} (\alpha_{i,r} + \beta_{i,r} t) 1_{(t_{i,r-1}, t_{i,r}]}(t) \tag{3.1}$$

with restrictions $\alpha_{i,r} = \alpha_{i,r-1} + \beta_{i,r-1} t_{i,r}$. The two-step approach and the mixture methods allow for the inclusion of covariates, but this is not evaluated in this work.

Longitudinal k-means Longitudinal k-means (KML) is a commonly used naive approach (Genolini and Falissard, 2010). The vectors of observations are assumed to be of equal length and aligned, i.e., $t_{a,j} = t_{b,j}$ for $a, b \in I,\ j = 1, \ldots, T$. The vectors are passed as observations to the k-means clustering algorithm (MacQueen, 1967; Genolini and Falissard, 2010). The k-means algorithm aims to find the partitioning $I_1, I_2, \ldots, I_G$ with $\bigcup_{g=1}^{G} I_g = I$ and $I_g \cap I_h = \emptyset$ when $g \neq h$, that minimizes the within-cluster variance, which in term maximizes the between-cluster variance. The objective function is given by

$$\underset{I_1, I_2, \ldots, I_G}{\arg\min} \sum_{g=1}^{G} \sum_{i \in I_g} ||\boldsymbol{y}_i - \hat{\boldsymbol{\mu}}_g||^2, \tag{3.2}$$

with $\hat{\boldsymbol{\mu}}_g$ the mean vector of the group elements, i.e., $\hat{\boldsymbol{\mu}}_g = |I_g|^{-1} \sum_{i \in I_g} \boldsymbol{y}_i$, where summation is performed element-wise. The algorithm uses an iterative approach to arrive at a solution. Starting from a random partitioning, the algorithm refines the partitioning at each iteration until the solution cannot be further improved, i.e., converges. In the case of KML, the

resulting cluster centers represent the group trajectory of each cluster. The resulting groups are assumed to be homogeneous, i.e., subjects belonging to a given group are assumed to follow the group trajectory $\hat{\boldsymbol{\mu}}_g$.

Two-step clustering We represent the two-step clustering approach by modeling the trajectories using a growth curve model (GCM), and clustering the subject parameter estimates (i.e., the random effects) using k-means (MacQueen, 1967). We will refer to this method as GCKM. This method is also described in the comparison of Twisk and Hoekstra (2012). The GCM is estimated in a mixed model framework (Laird and Ware, 1982). The model represents the longitudinal dataset in terms of a single group trajectory (i.e., the fixed effects), and for each subject, their deviation from this trajectory (the random effects). The trajectories are typically described by a polynomial of order 1 or 2 (Nagin and Odgers, 2010a). A trajectory described in terms of a polynomial of order K and random effects in all terms is given by

$$\begin{aligned} y_{i,j} &= \sum_{k=0}^{K} \beta_{k,i} t_{i,j}^{k} + \varepsilon_{i,j}, \\ \beta_{k,i} &= \alpha_k + \zeta_{k,i}. \end{aligned} \tag{3.3}$$

Here, α_k represents the kth order coefficient of the polynomial trajectory, $\zeta_{k,i}$ denotes the random effect of subject i for the kth coefficient (i.e., the between-subject variability), and $\varepsilon_{i,j}$ denotes the measurement error (within-subject variability). The random effects are assumed to be multivariate normally distributed with zero mean, possibly with an unstructured variance-covariance matrix, and uncorrelated with the measurement error ε. The measurement error is assumed to be independently normally distributed with zero mean and common variance, although an autoregressive correlation structure would be possible too. The model is estimated using maximum likelihood (ML) estimation (Verbeke and Molenberghs, 2000). Alternatively, the model parameters can be inferred using a Bayesian approach (Gelman *et al.*, 2013).

The random effects $\zeta_{k,i}$ of each trajectory can be predicted using the best linear unbiased predictors (BLUPs), and they are passed to the k-means algorithm as input vectors $\boldsymbol{y}_i = \left(\hat{\zeta}_{0,i}, \hat{\zeta}_{1,i}, \ldots, \hat{\zeta}_{K,i}\right)$. The estimation of the input vectors is independent of the number of groups to be identified in the second step and therefore needs to be performed only once. Scaling or standardization may be required to ensure equal weights across the BLUPs, depending on the difference in size of the variance components of $\zeta_{k,i}$. Similarly, covariates could be included into the model as additional random effects to account for other factors and can be clustered accordingly.

Group-based trajectory modeling A group-based trajectory model (GBTM) describes a longitudinal dataset in terms of a mixture of group trajectories, without regard of within-group variability (Nagin and Land, 1993; Nagin and Odgers, 2010a). This draws similarities to k-means in the sense that the subjects in a group are assumed to follow the group profile, but in the case of GBTM these profiles can be smooth (Nagin and Tremblay, 2005). GBTM is also commonly referred to as latent class growth analysis (LCGA), and semi-parametric group-based modeling (SGBM) (Nagin, 1999).

For a given trajectory $\boldsymbol{y}_i$, its observations are described by the group trajectory of group g as follows:

$$y_{i,j}^{(g)} = \sum_{k=0}^{K} \alpha_k^{(g)} t_{i,j}^k + \varepsilon_{i,j}, \tag{3.4}$$

where $\alpha_k^{(g)}$ denotes the kth coefficient for the polynomial of group g, and $\varepsilon_{i,j}$ describes the residual at time $t_{i,j}$. In this setting the subject trajectories $\psi_i(t_{i,j})$ are all the same to $\sum_{k=0}^{K} \alpha_k^{(g)} t_{i,j}^k$ in (3.4) when subject i belongs to group I_g. The residual is assumed to be independently normally distributed with zero mean. The marginal mean of a GBTM is given by

$$\mathbb{E}(y_{i,j}) = \sum_{g=1}^{G} \pi^{(g)} \sum_{k=0}^{K} \alpha_k^{(g)} t_{i,j}^k. \tag{3.5}$$

Here, $\pi^{(g)}$ denotes proportion of group g, with $0 \leq \pi^{(g)} \leq 1$ and $\sum_g \pi^{(g)} = 1$. The model is fitted to the data using ML estimation. The appeal of GBTM comes from the relatively simple group model, combined with a parametric approach that enables researchers to incorporate domain knowledge. Moreover, other factors can be corrected for by the inclusion of time-variant and time-invariant covariates into the model, both in the time profile and the proportion for group g (Nagin and Odgers, 2010a).

Growth mixture modeling Growth mixture modeling (GMM) is a method for identifying heterogeneous subgroups in the data using a mixture of growth curve models (Verbeke and Lesaffre, 1996; Muthén *et al.*, 2002; Muthén and Shedden, 1999). It is a generalization of GBTM by taking the coefficients $\alpha_k^{(g)}$ in (3.4) to be subject-specific, essentially introducing a mixed-effects model in each group g. Thus, a given trajectory $\mathbf{y}_i$ is described by group g by

$$\begin{aligned} y_{i,j}^{(g)} &= \sum_{k=0}^{K} \beta_{k,i}^{(g)} t_{i,j}^k + \varepsilon_{i,j}^{(g)}, \\ \beta_{k,i}^{(g)} &= \alpha_k^{(g)} + \zeta_{k,i}^{(g)}. \end{aligned} \tag{3.6}$$

The group-dependent fixed effects are denoted by $\alpha_k^{(g)}$. In any case, the model complexity of GMM significantly exceeds that of GBTM due to the additional estimation at group level of the random effects $\zeta_{k,i}^{(g)}$, residual $\varepsilon_{i,t}^{(g)}$, and the variance-covariance matrix. In practice, it is desirable to restrict or share some of the parameters across groups to reduce the challenge of finding a numerical solution. The residual is assumed to be independently and normally distributed with zero mean, and uncorrelated with $\zeta_{k,i}^{(g)}$. The random effects are assumed to be normally distributed with zero mean but may be correlated within group g (but not with random effects across groups). The marginal mean of a GMM is computed by (3.4) for $\mathbb{E}\left(\zeta_{k,i}^{(g)}\right) = 0$, with group proportion $\pi^{(g)}$ defined as before.

GMM is commonly used for its flexibility, enabling researchers to specify the random effects and relations between them, in addition to the inclusion of covariates (Frankfurt

et al., 2016). However, this may come at the cost of a greater difficulty in identifying the most appropriate model.

Mixed time-varying effect modeling In a time-varying effect model (TVEM) (Tan *et al.*, 2012), the regression coefficients that describes the relation between covariates and outcome can vary over time. The relation between time and outcome is described by a smooth function $\psi(t)$, thus longitudinal trajectories can be described by the intercept-only TVEM given by

$$y_{i,j} = \psi(t_{i,j}) + \varepsilon_{i,j}. \tag{3.7}$$

The ψ function is modeled using a penalized spline (referred to as a P-spline) (Song and Lu, 2010). A P-spline represents a series of time intervals using low-order polynomials with smooth transitions between intervals. A smooth fit is ensured by imposing a penalty on the second derivative. The P-splines can be represented as a linear model and therefore estimated efficiently using ordinary least-squares (Ruppert, 2002), although smoothing can also be obtained by introducing random effects for the time variable (Lu and Song, 2012).

We evaluate the mixture model proposed by Dziak *et al.* (2015), named MixTVEM, which comprises a mixture of TVEMs. Thus a smooth function $\psi^{(g)}$ of splines is estimated for group g and $\pi^{(g)}$ represents a proportion for group g. Its flexible trajectory estimates make it suitable for intensive longitudinal data, and as an exploration tool for uncovering unforeseen trajectories. The method is a semi-parametric version of GBTM or GMM, depending on whether random effects are considered. Dziak et al. suggest the inclusion of a first-order autoregressive (AR) structure as an alternative to random effects, as this is less computationally intensive and allows for constant heteroskedasticity over time. The correlation between any two measurements $y_{i,j}$ and $y_{i,j'}$ is given by $\rho^{|t_{i,j}-t_{i,j'}|}$, where ρ denotes AR-1 component. For numerical estimation purposes, they introduce an additional variance component that is proportional to the AR-1 component, referred to as the nugget effect, given by

$$\operatorname{cov}(y_{i,j}, y_{i,j'}) = \sigma_\rho^2 \rho^{|t_{i,j}-t_{i,j'}|} + \sigma_\varepsilon^2. \tag{3.8}$$

The ratio of the measurement variance σ_ε^2 to the total variance $\sigma_\rho^2 + \sigma_\varepsilon^2$ is assumed to be fixed across groups.

3.2.1 Number of groups

Determining the number of subgroups to describe the data is a well-known problem in cluster analysis. In practice, clusters are rarely distinct, meaning that the subgroups are not well-separated, and it is therefore difficult to cluster every subject without error. Nagin and Odgers (2010a) suggest to combine the use of an objective criterion for determining the number of groups with domain knowledge to arrive at a reasonable solution. For the mixture methods, the Bayes information criterion (BIC) appears to be generally applicable (Nagin and Odgers, 2010a; Nylund *et al.*, 2007; McNeish and Harring, 2017; Dziak *et al.*, 2015). For consistency, we use the BIC for KML and GCKM too. The likelihood of a k-means solution can be computed by regarding the clusters as a mixture of spherical Gaussians, enabling the computation of the BIC (Pelleg and Moore, 2000).

A recommended alternative to the BIC is the bootstrapped likelihood ratio test (BLRT), which is regarded as a more suitable criterion than the BIC (Nylund *et al.*, 2007; Jung and Wickrama, 2008; Tolvanen, 2007). The BLRT is a computationally intensive criterion, as it requires the estimation of the model on each of the generated bootstrap samples (with a recommendation of 500 bootstrap samples). We will therefore first conduct a preliminary evaluation on GBTM and GMM before evaluating the other methods. The evaluation on GBTM and GMM compares how the BLRT performs on models assuming homogeneous subgroups and heterogeneous subgroups, respectively.

An increase in BIC score of more than 10 is considered to be of significance, meaning that the additional description it provides warrants increased model complexity by introducing an additional group (Raftery, 1995; Frankfurt *et al.*, 2016). A robust alternative to this approach, commonly referred to as the elbow method, investigates the relative improvement in the objective function for a lower number of groups. The improvements tend to decline with an increasing number of groups, but often with a turning point (i.e., the elbow) (Hardy, 1994). While this method is usually assessed visually, we approximate it by estimating a piecewise-linear change point model such that it can be evaluated on the many datasets and scenarios automatically. The model comprises a variable change point and is optimized to maximize the fit to the BIC points over the different number of groups.

3.2.2 Computer software

All methods are evaluated in R 3.4.2 (R Core Team, 2022), running on Intel Xeon E5-2660 (2.6 GHz) processors. The implementation of KML was based on version 2.4.1 of the `kml` package (Genolini *et al.*, 2015). GCKM was evaluated by estimating a GCM using the `lcmm` package (version 1.7.8) (Proust-Lima *et al.*, 2017), and clustering the random effects using the `kml` package[1]. The `lcmm` package is also used to evaluate GBTM and GMM. For the implementation of MixTVEM, we use an R script[2] that has been made available by Dziak *et al.* (2015) (version 1.1), and run it with the default settings. Preliminary tests indicated that MixTVEM generally performed better with the inclusion of the AR-1 structure, and we therefore used it in all evaluations.

The estimation of the models can be a challenging task due to the large number of parameters involved, a problem that grows with the number of groups. The iterative optimization procedures converge to a local optimum, depending on the starting position. This is an issue sometimes observed in GBTM (Skardhamar, 2010; Twisk and Hoekstra, 2012), and especially in GMM (Twisk and Hoekstra, 2012; McNeish and Harring, 2017). The accepted approach in dealing with this involves many repeated random starts (e.g., 100 random starts, although the number depends on the data and model complexity) after which the estimation proceeds with the most likely start (Jung and Wickrama, 2008; McNeish and Harring, 2017), which is a time-wise costly procedure indeed. In view of the many scenarios under which the mixture methods need to be evaluated, we settled for 20 random starts to reduce the computation time. A preliminary evaluation suggested that this does not have a practically significant effect on the performance. For KML and GCKM, the k-means++ algorithm is used for selecting a better starting position, with 25 repeated runs (Arthur and Vassilvitskii, 2007).

[1] Although the random effects are not longitudinal data, the `kml` package was used here to ensure an identical application of k-means.

[2] Version 1.1, available at https://www.methodology.psu.edu/downloads/mixtvem/.

3.3 Simulation

We evaluate the methods across different scenarios using Monte Carlo simulations[3]. The scenarios comprise multiple settings, with each method being evaluated on each permutation of settings on 100 synthetic datasets. The generated group trajectories are consistent between scenarios and settings for the respective number of groups in the data (unless mentioned otherwise). This enables a comparison between scenarios with a higher sensitivity to the effect of changing settings (Burton *et al.*, 2006), with the advantage of requiring fewer simulations.

The datasets are generated using growth curve models describing second-order polynomial trajectories, to comprise a mixture of heterogeneous groups. The measurement times are evenly spaced between $[0, 1]$. The group trajectory $\boldsymbol{y}^{(g)}$ is represented by the model from Equation 3.3 for $K = 2$. The three fixed effects (that is, the intercept $\alpha_0^{(g)}$, slope $\alpha_1^{(g)}$, and quadratic term $\alpha_2^{(g)}$) are sampled from a uniform distribution between -1 and 1. Thus for some datasets we have well-separated group trajectories, and for other datasets there will be overlapping group trajectories. The random effects $\zeta_0^{(g)}$, $\zeta_1^{(g)}$ and $\zeta_2^{(g)}$ for the subjects in each group follow a normal distribution (unless mentioned otherwise) with zero mean and standard deviation of 0.1, 0.2, or 0.3 in the simulation scenarios involving low, medium, and high variability, respectively. For the residuals, we have $\varepsilon_{i,j} \sim N(0, \sigma^2)$, with a standard deviation of 0.01 or 0.1, representing negligible noise and considerable noise, respectively. An example of how a dataset with relatively well-separated group trajectories can vary in difficulty depending on the settings is illustrated in Figure 3.1.

We generate groups of different sizes (i.e., number of subjects), whilst ensuring that the smallest group is of sufficient size to be detected. This is achieved by using group proportions $\pi^{(g)} \propto \sqrt{g}$, normalized by a factor of $\sum_{g=1}^{G} \sqrt{g}$. In the datasets containing six groups, the smallest and largest groups comprise 9% and 55% of the subjects, respectively.

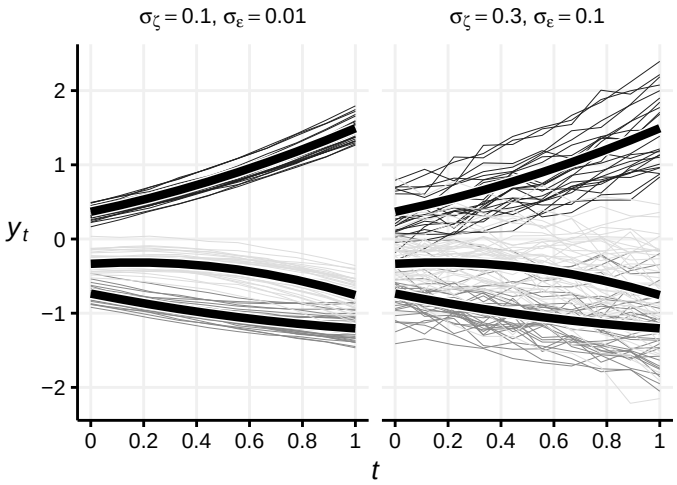
3.3.1 Design

In the first scenario, we investigate how the methods perform given that the number of groups is correctly specified. We compare the performance of the methods under each permutation of settings involving sample size, number of repeated observations, within-group variability, unexplained variability, and number of groups. Drawing from previous research on sample size requirements of the mixture models (Loughran and Nagin, 2006; Nylund *et al.*, 2007; Tolvanen, 2007), we evaluate the sample size at three levels ($N = 200$, 500, 1000). The effect of the number of repeated observations on the performance is evaluated at settings ($T = 4$, 10, 25). The random effects are evaluated for three levels of within-group variability, sampled from a normal distribution with ($\sigma_{\zeta_k} = 0.1, 0.2, 0.3$ where $\sigma_{\zeta_0} = \sigma_{\zeta_1} = \sigma_{\zeta_2}$). Lastly, the within-subject variability is investigated using two levels of white noise ($\sigma_\varepsilon = 0.01$, 0.1), while keeping the variability between subjects constant.

Secondly, we assess the impact of model misspecification on the identification of subgroups. It is known that the model fit of a LME model is sensitive to the assumption on normality in the random effects (Verbeke and Lesaffre, 1996; Muthén and Asparouhov, 2009), but it remains to be seen how this affects the grouping when random effects

[3]The Mersenne Twister algorithm is used for random number generation (Matsumoto and Nishimura, 1998).

Figure 3.1: Two example datasets for low and high within-group variability and measurement error, respectively. The datasets each contain 100 trajectories of 10 observations. The black lines denote the three group trajectories.

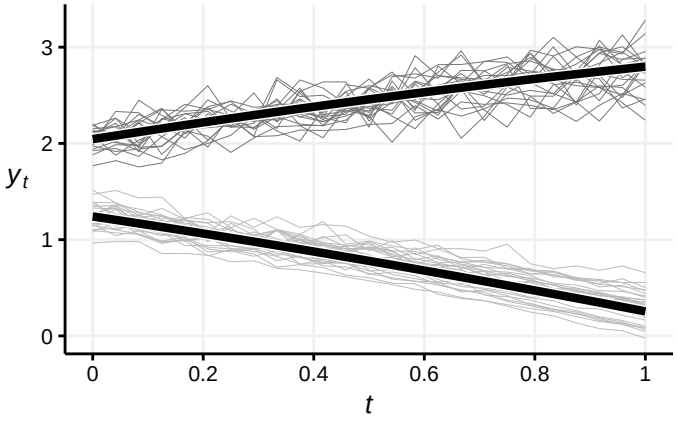


are lognormal. The datasets comprise random effects $\zeta_k^{(g)} \sim \text{lognormal}(\mu, \frac{1}{2})$ with $\mu = \log(0.15)$, $\log(0.30)$, and $\log(0.45)$ for low, medium, and high variability, respectively. The variability of these scenarios approximately corresponds to the similarly named scenarios involving normally distributed random effects. The positive range on the distribution would mean that the group trajectory is the trajectory with the lowest coefficients. We ensure the group trajectory is representative by centering the random effects around zero by subtracting the median $\exp(\mu)$ (0.15, 0.30 and 0.45 for the three settings, respectively).

Similarly, we also study the effect of a misspecified measurement error by introducing proportional within-subject variability (heteroskedastic), a feature that was observed in the case study. Here, the measurement error is specified as $\varepsilon_{i,j} \sim N\left(0, \max(0.01, c \cdot \tilde{y}_{i,j})^2\right)$ with scaling factors $c = (0.01, 0.03, 0.05)$. In this scenario, we take $\alpha_0^{(g)} \sim U(1, 3)$ to ensure that the proportional deviation $c \cdot \tilde{y}_{i,j}$ exceeds the minimum standard deviation of 0.01 in most cases. An example dataset is shown in Figure 3.2.

In the last scenario we assess the preferred fit of each method in terms of the number of groups that achieves the best result according to the model selection criteria described below. Some of the methods may produce a better fit for a different number of groups than the correct number due to differences in the group representation. We generate 100 datasets for each number of groups ranging from 3 to 5, on which the methods are evaluated for 2 to 7 groups. We investigate how well the methods perform at recovering the true number of groups using the BIC, under a low and high random effect variability with $\sigma_\zeta = \{0.1, 0.3\}$, respectively.

Figure 3.2: Example dataset for the proportional noise scenario, with $c = 0.05$, $\sigma_\zeta = 0.1$.



3.3.2 Evaluation

Although the large number of datasets precludes a subjective analysis on the fit of the methods, as is commonly done in applications (Nagin and Odgers, 2010b), the evaluation provides a balanced assessment across different group trajectories and scenarios. We evaluate the fit of the methods to each dataset using three metrics, namely the correctness of the trajectory group membership, the group trajectory, and the number of groups. Cases in which the model could not be estimated are excluded from the evaluation, and we report how often this happens.

NSJ First and foremost, the group assignments of the N trajectories are compared to their true group membership. The split-join distance introduced by Van Dongen (2000) measures the similarity between the partitions $\mathcal{A}$ and $\mathcal{B}$ in terms of the number of subset reassignments that are needed to project the partition onto the other, and back. The partitions comprise sets of subjects that are in the same group g, denoted by $\mathcal{A} = \{a_1, a_2, ..., a_G\}$ and $\mathcal{B} = \{b_1, b_2, ..., b_G\}$. We have $a_g = I_g$ with $\bigcup_{g=1}^{G} I_g = I$ and $I_g \cap I_h = \emptyset$ when $g \neq h$, and the same holds for b_g. The number of groups may differ between the partitions. The number of matching assignments between $\mathcal{A}$ and $\mathcal{A} \cap \mathcal{B}$ is denoted by $N_{\mathcal{A}}(\mathcal{B})$ and computed by

$$N_{\mathcal{A}}(\mathcal{B}) = \sum_{a \in \mathcal{A}} \max_{b \in \mathcal{B}} |a \cap b|, \tag{3.9}$$

where $|a \cap b|$ denotes the number of subjects that occur in both sets. The distance between the partitions is asymmetric, with

$$\begin{aligned} d(\mathcal{A}, \mathcal{A} \cap \mathcal{B}) &= N - N_{\mathcal{A}}(\mathcal{B}), \\ d(\mathcal{B}, \mathcal{A} \cap \mathcal{B}) &= N - N_{\mathcal{B}}(\mathcal{A}). \end{aligned} \tag{3.10}$$

It can be seen that if $\mathcal{A}$ contains sets which are proper subsets of a set of $\mathcal{B}$, then the projection from $\mathcal{A}$ onto $\mathcal{B}$ requires fewer adjustments with respect to these elements than

vice versa. A distance of 0 implies that the partition is a subpartition (Van Dongen, 2000), which makes the metric suitable for comparing partitions with a different number of groups. Combining the pairwise distances, we obtain the split-join distance

$$d(\mathcal{A}, \mathcal{B}) = d(\mathcal{A}, \mathcal{A} \cap \mathcal{B}) + d(\mathcal{B}, \mathcal{A} \cap \mathcal{B}) \tag{3.11}$$

The scale of the metric is dependent on the sample size. To evaluate the split-join distance across simulation scenarios with different sample sizes, we use the normalized metric

$$\text{NSJ}(\mathcal{A}, \mathcal{B}) = \frac{d(\mathcal{A}, \mathcal{B})}{2N}, \tag{3.12}$$

which expresses the distance on a scale from 0 to 1 (lower is better). We will refer to this metric as the normalized split-join (NSJ) distance.

WMMSE While a low NSJ score is expected to be associated with a good fit of the group trajectories, there are cases in which this need not be the case. For example, in the case of overlapping groups it is still possible to obtain proper group trajectory fits, yet there is uncertainty on the subject group membership, impacting the NSJ score. We therefore assess the fit of the group trajectories as a secondary metric. The group trajectories are compared in terms of the mean squared error (MSE) at each observation in time. A challenging aspect of this evaluation is that there is no guaranteed one-to-one mapping between the group and reference group trajectories. We therefore associate each group trajectory $\boldsymbol{y}^{(g)}_{\text{group}}$ with the nearest reference group trajectory $\boldsymbol{y}_{\text{ref}}$, and weigh the score by the group proportion $\pi^{(g)}$. The score, which we shall call the weighted minimum MSE, is denoted by

$$\text{WMMSE} = \frac{1}{T} \sum_{g=1}^{G} \pi^{(g)} \min_{g' \in \boldsymbol{G}_{\text{ref}}} \sum_{j=1}^{T} \left(y^{(g)}_{\text{group},j} - y^{(g')}_{\text{ref},j} \right)^2, \tag{3.13}$$

with $\mathbf{G}_{\text{ref}} = \{1, 2, ..., G_{\text{ref}}\}$, and G_{ref} refers to the true number of groups in the dataset. Due to the relatively small scale of the observed values and the resulting small observation error, the reported WMMSE values are multiplied by 1000.

3.4 Results

3.4.1 Simulations

Numerical convergence

We observed problems with the model estimation across the simulation scenarios. The main effects are reported in Table 3.1. Any convergence issues of GCKM were established by the GCM in the first step, considering that the k-means algorithm is guaranteed to converge. Due to the higher number of parameters in a GMM, it is numerically less stable than KML, GCKM and GBTM. Nevertheless, only MixTVEM exhibits significant convergence problems across scenarios, with an overall nonconvergence rate of 26%. The convergence problems appear to only occur for a large number of observations, independent of the other simulation settings. Whereas a negligible number of problems occur at $T = 4$ (0.22%), for an increasing number of T the convergence problems worsen, with 14% at

$T = 10$, and already 78% at $T = 25$. Moreover, the rate of convergence problems increases with the number of groups, with 4.8% at 2 groups and 18% at 6 groups for $T = 10$. A comparison of MixTVEM configurations revealed that the convergence issues mostly occurred when the AR-1 structure was included, although it is unclear why the estimation is affected for a higher number of repeated observations.

Table 3.1: Percentage of convergence issues across all scenarios and datasets.

Method	Did not converge	Empty groups	Solitary groups	Total
KML	0.0%	0.0%	0.11%	0.11%
GCKM	0.0%	0.0%	0.09%	0.09%
GBTM	0.28%	$< 0.01\%$	0.08%	0.36%
GMM	0.04%	2.0%	0.90%	3.0%
MixTVEM	26%	2.5%	4.3%	31%

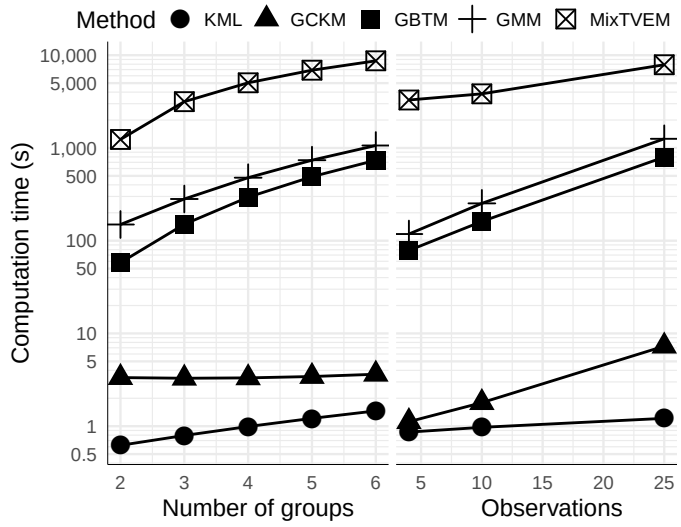
A converged fit is not necessarily without problems. A group can be empty if the posterior class probability of all trajectories is greater for other groups. In the simulations, these problems only occur at a considerable rate for GMM (2.0%) and MixTVEM (2.5%). GMM primarily exhibits this problem for higher number of groups, with 5.5% at 6 groups across scenarios, whereas hardly any problems occur at 2-4 groups ($< 0.48\%$). Furthermore, the problem more often occurs with low measurement error (8.8%), compared to high measurement error (3.8%). A possible explanation for this discrepancy is that the higher measurement error provides additional possible group trajectories due to the increased overlap between trajectories. Log-normally distributed random effects result further in an increased number of solutions with empty groups. The scenario with log-normally distributed random effects at 6 groups with low measurement error exhibits empty groups in 21% of all converged cases. In MixTVEM, the problem occurs mostly in the setting with 25 observations (34% of converged cases), whereas it occurs infrequently (1% of converged cases) at 10 observations.

Another possible problem with a solution is the presence of groups consisting of a single subject trajectory, assuming that such a solution is not meaningful in practice (in the simulations it is not considered to be meaningful). We refer to this as a solitary group. In GMM, this occurs relatively often only for a higher number of groups (3.8% for 6 groups, while below 0.01% for 2 groups), under log-normally distributed random effects (5.9% at 6 groups), and especially when the number of groups does not match the true number of groups (9.0% at 6 groups). For MixTVEM, solitary groups occur frequently on data comprising 25 observations (20%), whereas for 10 observations only 5.3% of converged cases have solitary groups. Independent of the number of observations, solitary groups occur more often on log-normal data (15% compared to 2.6% on the normal data), and with smaller sample size (7.3% for $N = 200$, compared to 2.1% for $N = 1000$).

Computation time

We assess how well the methods scale relative to an increasing volume of data and increasing number of parameters. To illustrate how the methods scale differently, the effects of the number of groups and number of observations on computation time are

Figure 3.3: Computation time per model (in seconds) over the number of groups and observations.



visualized in Figure 3.3. The effect of sample size is not shown because all methods scale in computation time with respect to sample size in the same way. The base computation time differs considerably between methods. Whereas k-means requires only 1 second on average per dataset run, GMM takes 9 minutes. This is largely due to GMM comprising a more complex computational problem. MixTVEM stands out from the mixture methods with computation times of over 1 hour on average, likely due to its relatively unoptimized implementation compared to the mature R packages available for GBTM and GMM. However, MixTVEM scales relatively well with an increasing number of observations compared to GMM, GBTM and GCKM, suggesting that the method may be favorable for a larger number of observations, given a more optimized implementation. As a result of the independent assessment of the methods in each scenario, the first step of GCKM was recomputed each time, and therefore the computation time is higher than it would be in practice, because the results of the first step could be reused. KML is least impacted among the methods by the inclusion of additional groups and observations. The GCM computation in the first step of GCKM accounts for most of the computation time of the method, hence the near-constant time over the number of groups, and the similar scaling to the mixture methods along the number of observations.

3.4.1.1 Group assignment for the correct number of groups

The methods are assessed across simulation settings for all possible combinations. This results in 270 unique cases[4] to be evaluated per method on 100 datasets generated with normally distributed random effects and normal residual. We first evaluate the correctness of the assignment of subject trajectories using the NSJ distance. The main effects of each

[4]We arrived at 270 cases by evaluating all permutations of 5 different number of groups, 3 sample sizes, 3 values for the number of observations, 3 random effect deviations, and 2 measurement errors.

of the independent factors (that make up the settings) are assessed by means of a linear model. Runs in which a model did not converge are excluded. The results are reported in Table 3.2. Due to the small standard errors below 0.01, all differences between methods can be considered statistically significant and we do not report on this further. Instead, we focus on the practical significance of the score differences.

The overall NSJ scores show that GMM and GCKM perform significantly better on average than the other models, with scores of 0.084 and 0.10, respectively. In contrast, KML, GBTM, and MixTVEM achieve an average score of approximately 0.21. By comparing the NSJ of KML and GBTM across the main effects, it becomes clear that their results are identical. Although MixTVEM obtains a similar average score, it deviates from the other two methods in some settings, in particular for the number of observations.

The number of possible assignment errors increases with the number of groups. The performance declines at a similar rate across methods for an increasing number of groups. To put the NSJ scores into perspective, the expected NSJ by random assignment, assuming correct group proportions, are 0.41, 0.58, 0.67, 0.72, and 0.76, for $G = 2, ..., 6$, respectively. In this respect, all methods perform significantly better than random grouping even for a larger number of groups.

GCKM and GMM benefit from a larger sample size, and number of observations under the presence of noise, where GCKM approaches the performance of GMM with an increased number of observations. MixTVEM often fails to find a proper fit for $T = 25$, in addition to the convergence problems reported above. The size of the random effects has a considerable impact on the difficulty of the dataset, yet the differences between the methods are relatively stable. The performance of KML, GBTM, and MixTVEM are unaffected by the presence of measurement error, while GCKM and GMM are negatively affected by it. Still, GCKM and GMM outperform the other methods even in these conditions.

3.4.1.2 Group trajectory estimation with the correct number of groups

In addition to the group assignment accuracy, we investigate the estimation of the group trajectories. The results are reported in Table 3.3. Overall, GCKM and GMM achieve the best group trajectory estimates. The methods have near-identical performance on average, and the same holds for KML and GBTM. Comparing the findings with the NSJ scores of Table 3.2, it is evident that on average a lower NSJ is associated with a better group trajectory fit (i.e., lower WMMSE). In contrast, the group trajectory estimation of GCKM improves with an increasing number of observations, surpassing GMM at $T = 25$, even though this is not reflected in the NSJ scores. Another discrepancy is observed in the worsening WMMSE scores for MixTVEM with an increasing number of groups compared to KML and GBTM, whereas this pattern is not visible when assessing the NSJ scores. This indicates that MixTVEM can achieve a similar subject assignment despite worse group trajectory estimates. The high average WMMSE of MixTVEM arises from the poor model fit at $T = 25$, whereas for fewer observations the WMMSE of MixTVEM is not significantly different from KML and GBTM.

All methods except MixTVEM benefit from an increased sample size, with GCKM and GMM showing the greatest relative improvement. The magnitude of variation of the random effects affects all methods, but KML and GBTM in particular (with a WMMSE of 32 at $\sigma_\zeta = 0.3$ compared to 1.4 at $\sigma_\zeta = 0.1$). Moreover, the associated error with an

Table 3.2: Effects of the scenario settings on group assignment per model, averaged over 100 datasets, as measured by the NSJ distance (lower is better). The 'All' row reports the average performance over all cases.

	KML	GCKM	GBTM	GMM	MixTVEM
All	.21	.10	.21	.084	.22
Number of groups G					
2	.10	.039	.10	.034	.088
3	.17	.069	.17	.058	.17
4	.23	.097	.23	.083	.24
5	.27	.13	.27	.11	.28
6	.30	.15	.30	.14	.30
Sample size N					
200	.22	.099	.22	.087	.22
500	.21	.097	.21	.085	.22
1000	.21	.093	.21	.081	.21
Number of observations T					
4	.20	.11	.20	.090	.19
10	.21	.096	.21	.085	.18
25	.22	.085	.22	.078	.28
Random effects deviation σ_ζ					
.1	.084	.040	.085	.027	.13
.2	.22	.088	.22	.078	.21
.3	.33	.16	.33	.15	.31
Measurement error σ_ε					
.01	.21	.064	.21	.064	.20
.1	.21	.13	.21	.10	.23

increasing number of groups differs significantly per level of σ_ζ. Notably, the fit of GCKM is considerably less accurate than GMM with $\sigma_\zeta = 0.1$ for a low number of observations ($T = 4$).

3.4.1.3 Proportional measurement error

We assess the sensitivity of the methods on longitudinal observations with a proportional measurement error. The performance of the methods is shown in Table 3.4 and 3.5 in terms of the NSJ and WMMSE. The methods are evaluated on datasets comprising 2-6 groups, with a standard deviation on the random effects of 0.1 and 0.3.

The results on group assignments follow the observations from the earlier experiment of Table 3.2 involving two levels of heteroskedasticity, with KML and GBTM being insensitive to the level, and GMM and GCKM benefiting from lower degrees of heteroskedasticity (GCKM from 0.14 to 0.072, GMM from 0.11 to 0.071). MixTVEM shows a relatively small improvement of -0.03 for lower error. Overall, the performance of the models is similar to those on the homoskedastic residual variance. In case of the group trajectory estimation

Table 3.3: Effects of group trajectory estimation error across simulation scenarios, measured by the WMMSE multiplied by 1000.

	KML	GCKM	GBTM	GMM	MixTVEM
All	15	3.5	15	3.4	38
Number of groups G					
2	10	1.4	10	.14	27
3	13	2.4	13	1.8	32
4	15	3.5	15	3.4	38
5	17	4.5	17	5.0	44
6	19	5.5	20	.6.7	49
Sample size N					
200	16	4.6	16	4.4	38
500	15	3.7	15	3.6	38
1000	14	2.1	14	2.2	38
Number of observations T					
4	15	4.4	16	3.1	15
10	15	3.3	15	3.3	15
25	15	2.7	15	3.8	83
Random effects deviation σ_ζ					
.1	1.4	.87	1.4	1.7	19
.2	12	2.2	12	2.6	36
.3	32	7.3	32	5.9	59
Measurement error σ_ε					
.01	15	2.3	15	3.4	32
.1	15	4.6	15	3.4	44

error, the result appears to be unaffected by the level of heteroskedasticity.

3.4.1.4 Log-normal groups

The results of the scenarios involving log-normally distributed random effects are reported in Table 3.6 and 3.7 in terms of the NSJ and WMMSE, respectively. The scores are compared to those under the standard scenario of Table 3.2 and 3.3, respectively. Overall, all methods except MixTVEM achieve a worse performance compared to the standard scenario. Especially GMM is significantly impacted, although it is the best performing method regardless, indicating the importance of the correct specification of the subgroup distribution.

In terms of group trajectory estimation, KML, GCKM and GBTM achieve the best estimates, with an error of 24. GMM has a higher error of 37 on average compared to GCKM, especially for larger between-group variation. MixTVEM performs relatively poorly across all scenarios.

Table 3.4: Averaged effects of group assignment error (NSJ) under proportional measurement error.

	KML	GCKM	GBTM	GMM	MixTVEM
Proportional measurement error c					
.01	.21	.072	.21	.071	.17
.03	.21	.10	.21	.089	.19
.05	.21	.14	.21	.11	.20

Table 3.5: Averaged effects of the group trajectory estimation error (WMMSE $\times 1000$) under proportional measurement error.

	KML	GCKM	GBTM	GMM	MixTVEM
Proportional measurement error c					
.01	16	.2.3	16	3.2	15
.03	16	3.8	17	3.4	16
.05	17	5.3	17	3.6	18

3.4.1.5 Finding the number of groups

We investigate how well the methods can identify the simulated number of groups using the BLRT and BIC. We generated datasets comprising 500 trajectories across 2 to 5 groups, and evaluated the methods with a specification of the number of groups ranging from 2 to 7 groups. We consider two scenarios involving low and high within-group heterogeneity ($\sigma_\zeta = 0.1$ and $\sigma_\zeta = 0.3$, respectively). Each of the scenarios is evaluated with 100 generated datasets, resulting in a total of 800 evaluations per method.

Table 3.8 reports the proportion of correct cases and cases in which the optimal solution was off by one group when using BIC_{min} or $\text{BIC}_{\text{elbow}}$. The results are computed across the two settings for σ_ζ. The NSJ and WMMSE criteria serve as a reference for the number of groups needed to optimally match the group membership assignment and group trajectory fit, respectively. Due to model limitations, it is possible for a criterion to exceed these values. The results of the BLRT evaluation are shown in Table 3.9.

GMM has the highest number of correct cases across all criteria; in particular with the BLRT and BIC_{min}. The BLRT outperforms BIC_{min}, correctly identifying the number of groups in 86% of datasets, as opposed to 71% using BIC_{min}. In contrast, GBTM consistently (99.9%) overestimates the number of groups with both criteria. The same pattern of overestimation is observed when applying BIC_{min} for KML, GCKM, and MixTVEM. In view of these findings and the computationally intensive aspect of BLRT, we therefore do not evaluate the BLRT for the other methods.

KML, GBTM and MixTVEM achieve far better results using $\text{BIC}_{\text{elbow}}$, and come close to the performance with the NSJ as a reference. The solutions of KML and GBTM for minimum WMMSE tend to be closer to the correct number of groups than for the NSJ, indicating that the closest approximation of the group trajectories does not always correspond to a correct group assignment.

The magnitude of σ_ζ has a significant effect on the proportion of correct cases, as

Table 3.6: Averaged effects of group estimation (NSJ) under log-normally distributed random effects.

	KML	GCKM	GBTM	GMM	MixTVEM
All	.26	.14	.26	.14	.23
Log-normal random effects mean $\exp(\mu_\zeta)$					
.15	.13	.060	.13	.051	.13
.30	.28	.13	.27	.14	.24
.45	.37	.22	.36	.23	.31
Measurement error σ_ε					
.01	.26	.11	.25	.12	.22
.1	.26	.17	.26	.16	.24

Table 3.7: Averaged effects of group trajectory estimation error (WMMSE $\times 1000$) under log-normally distributed random effects.

	KML	GCKM	GBTM	GMM	MixTVEM
All	25	25	25	37	51
Log-normal random effects mean $\exp(\mu_\zeta)$					
.15	.6.3	6.8	6.6	7.9	17
.30	23	.21	23	29	46
.45	45	47	45	74	89
Measurement error σ_ε					
.01	25	25	.25	36	55
.1	25	25	25	38	46

can be seen from Table 3.10 and Table 3.9. KML and GBTM are most affected by the large within-group variability, with respect to both the group assignment (from 70% to 15%) and group trajectory fit (from 82% to 26%). GMM achieves a correctness of 93% under low variability for both BLRT and BIC_{min}. Under high variability, the performance degrades to 49% when using BIC_{min} but only to 79% when using BLRT.

Overall, the performance of the methods is consistent between the NSJ and WMMSE criteria except for GCKM, in which the group trajectory estimation does not improve under lower variability. The decrease in performance over σ_ζ is less prominent in the BIC results, demonstrating that the approach is relatively robust. For high variability, the correct cases of KML and GBTM for $\text{BIC}_{\text{elbow}}$ even exceed those of the reference criteria.

3.4.2 Case study

We investigate the usage data of sleep apnea patients undergoing positive airway pressure (PAP) treatment. Weaver *et al.* (2007) have demonstrated the importance of sufficient usage of CPAP therapy, where increasing daily usage was associated with a better outcome. Four hours of usage per day is considered to be the minimum for adequate treatment.

Table 3.8: Percentage of cases in which the solution as determined by the respective criterion corresponds to the true number of groups, computed across all cases. The NSJ and WMMSE provide a reference to how often the optimal fit in terms of group membership assignment and group trajectory fit correspond to the correct number of groups.

	Group error	KML	GCKM	GBTM	GMM	MixTVEM
Ref_{NSJ}	−1	32%	29%	32%	20%	29%
	0	**43%**	**63%**	**42%**	**71%**	**37%**
	+1	1.2%	0.17%	1.0%	1.8%	13%
$\text{Ref}_{\text{WMMSE}}$	−1	17%	9.8%	16%	6.5%	20%
	0	**55%**	**61%**	**53%**	**66%**	**41%**
	+1	10%	19%	11%	20%	19%
BIC_{min}	−1	0%	15%	0%	22%	17%
	0	**0%**	**30%**	**0%**	**71%**	**28%**
	+1	33%	17%	34%	0%	25%
$\text{BIC}_{\text{elbow}}$	−1	32%	35%	32%	32%	30%
	0	**42%**	**46%**	**41%**	**55%**	**36%**
	+1	7.2%	2.7%	7.5%	1.1%	13%

Table 3.9: Percentage of cases in which the solution determined by the BLRT corresponds to the correct number of groups.

Scenario	Method	Group error				
		< -1	-1	0	$+1$	> 1
All	GBTM	0%	0%	**< 1%**	< 1%	99%
	GMM	< 1%	7.3%	**86%**	5.3%	< 1%
$\sigma_\zeta = .1$	GBTM	0%	0%	**< 1%**	1.0%	99%
	GMM	< 1%	< 1%	**93%**	5.3%	0%
$\sigma_\zeta = .3$	GBTM	0%	0%	**0%**	0%	100%
	GMM	1.0%	14%	**79%**	5.3%	< 1%

Table 3.10: Percentage of cases with the correct number of groups according to the respective criterion.

	σ_ζ	KML	GCKM	GBTM	GMM	MixTVEM
Ref_{NSJ}	**.1**	71%	75%	70%	90%	49%
	.3	15%	51%	14%	52%	24%
$\text{Ref}_{\text{WMMSE}}$	**.1**	83%	58%	80%	75%	50%
	.3	27%	64%	26%	56%	33%
BIC_{min}	**.1**	0%	7.7%	0%	93%	40%
	.3	0%	53%	0%	49%	16%
$\text{BIC}_{\text{elbow}}$	**.1**	50%	52%	49%	44%	37%
	.3	34%	40%	33%	37%	34%

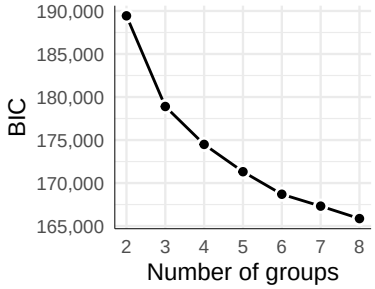
Many patients struggle to accommodate to the therapy, resulting in much lower hours of use, or an abandonment of the therapy, whereas other patients may improve over time. Other patients establish a preferred number of hours of usage early on and remain constant over time. In identifying the common longitudinal patterns of usage, we can describe patients in greater detail and quantify the occurrence of certain patterns. Previous studies have investigated these patterns using a two-step approach (Aloia *et al.*, 2008; Babbin *et al.*, 2015).

The daily usage data was collected from a retrospective observational study in the US over the past six years, comprising 2,686 patients diagnosed with sleep apnea and being on therapy for the first time. The average age of the patients is 60 years ($\sigma = 15$ years). We focus on the first 6 months of therapy, including only patients with at least 6 months of data (1,745 patients). Days on which the device data is missing indicate that the therapy was not used and are therefore represented by usage of zero hours (accounting for 30% of all days). Given that patients used the therapy, the mean daily usage is 6.3 hours ($\sigma = 2.7$ hours). Due to the computational requirements of the mixture methods, we downsample the usage data into weekly averages, resulting in 25 observations per patient.

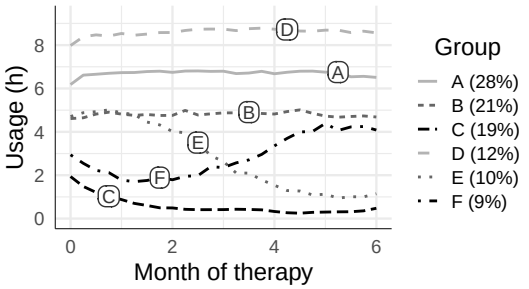
In this exploratory analysis, we are primarily interested in identifying groups of patients that exhibited a change in usage over time, and the patterns of change associated with these groups. We consider the mean level of usage to be of less relevance when it is above 4 hours is this is generally regarded as the minimum for adherence. The balance in relevance of the mean level of usage is challenging to capture in a single metric. We therefore we apply a hybrid approach, basing our decision of the preferred number of groups on a combination of the model information criterion and model results (Van de Schoot *et al.*, 2017). We use the BIC as a starting point towards determining the ideal solution. The solutions close to the preferred number of groups indicated by the BIC are then evaluated regarding the clinical interpretation of the group trajectories (Feldman *et al.*, 2009). A solution involving more groups is preferred only if it contains a group trajectory exhibiting change or a low mean level of usage not present in solutions with a fewer number of groups. Moreover, the solution needs to have groups of considerable size (greater than 5%). Lastly, we assess the confidence in trajectory assignments to the groups, which we measure using relative entropy (Van de Schoot *et al.*, 2017; McNeish and Harring, 2017).

Figure 3.4: Case study analysis using KML.

(a) The BIC per solution.



(b) The group trajectories.



KML

We apply KML as described in Section 3.2.2. Due to the low running time, we can evaluate the method on 1 to 8 groups in a short amount of time, with even the eight-groups solution requiring only 11 seconds to compute. The sequence of BIC values of Figure 3.4a show a consistent but diminishing improvement of the model fit over an increasing number of groups.

We select the solution comprising 6 groups because it provides a balance between the level of detail by which the patients are described, and the number of group trajectories involved. The group trajectories are depicted in Figure 3.4b. The confidence in the classification of the trajectories is strong, with a relative entropy of 0.96. Most patients appear to follow a near-constant trajectory, with the group trajectories A (6.7 hours), B (4.8 hours), C (0.6 hours), and D (8.6 hours) comprising 80% of all patients. With respect to adherence, group B is of particular interest due to variability around the compliance threshold of 4 hours. The proportion of patients with near-zero usage (group C) accounts for 19% of all patients. This group, together with group E, describes patients that either stop using the therapy or use it infrequently. Group F describes patients that start out as non-compliant but improve their usage throughout the therapy.

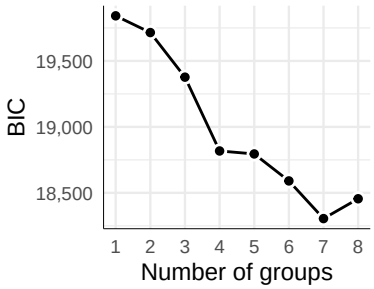
GCKM

The trajectory coefficients underlying the GCKM method are obtained using a GCM with 3^{rd}-order orthogonal polynomial random effects. We selected this model by fitting growth curve models of increasing complexity to the data and selecting the model that minimizes the BIC. The model choice is further supported by the group trajectories found by KML in Figure 3.4b. The k-means algorithm is applied in the same way as was done with KML. The first step of the approach, involving the estimation of the GCM, only needs to be done once. This took only 89 seconds, followed by k-means clustering, of which we evaluated the solutions from 1 to 8 number of groups.

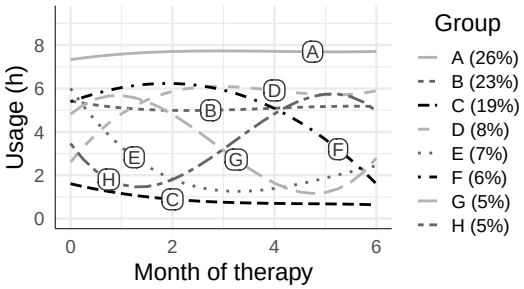
We choose the solution involving eight groups, which is shown in Figure 3.5b. Despite the large number of groups, some groups are of considerable size. Group A, B, and C already comprise 68% of patients, meaning that the remaining groups capture trajectories that occur less frequently. The relative entropy is 0.86, indicating a good separation

Figure 3.5: Case study analysis using GCKM.

(a) The BIC per solution.



(b) The group trajectories.



of groups. The group trajectories of each of the three major groups is near-constant, with an average usage of 7.7 hours, 5.1 hours, and 0.87 hours, respectively. The latter trend represents patients who were mostly non-compliant throughout the 6 months of therapy, accounting for 19% of patients. Group D describes patients (8%) that were non-compliant at the start of therapy but improved later. The remaining groups describe group trajectories with periods of non-compliance at specific moments in time.

GBTM

The GBTM model is initialized using 50 random starts, and the best candidate model is estimated until either convergence or 500 iterations are reached. We fit GBTMs with 3^{rd}-order orthogonal polynomial group trajectories, based on the reasoning provided in the GCKM analysis described above. In total, the estimation of two groups takes about 6 minutes, whereas the eight-groups model takes almost 2 hours of computation time. The 2 hours of computation time consist of 80 minutes for estimating the random starts, and 39 minutes for optimizing the final model. A converged solution was obtained for all evaluated number of groups.

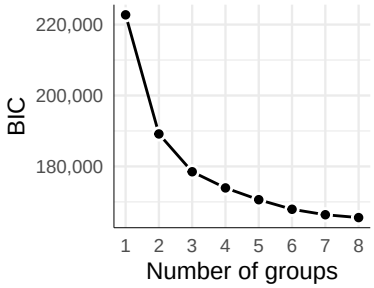
We arrive at the solution in Figure 3.6b, representing 6 group trajectories. With a relative entropy of 0.96, the solution exhibits a strong separation of groups. About 82% of patients are assigned to one of the four near-constant group trajectories (B, C, E, F). Group B represents patients with near-zero usage (0.60 hours on average), comprising 20% of all patients. Group A and D describe patients who are mostly compliant at the start and drop off later, and vice versa.

GMM

We apply GMMs with polynomial group trajectories of degree 3, based on the reasoning that was provided in the GCKM analysis. Furthermore, we specify a random intercept, and a shared diagonal variance-covariance matrix across the latent classes. This is done to limit the added complexity of the model, in view of the large sample size. The model is initialized using 50 random starts, and the best candidate model is iterated until either convergence or 500 iterations are reached. Despite the restrictions imposed on the model, the model estimation remains numerically challenging. This is evident from the time

Figure 3.6: Case study analysis using GBTM.

(a) The BIC per solution.



(b) The group trajectories.

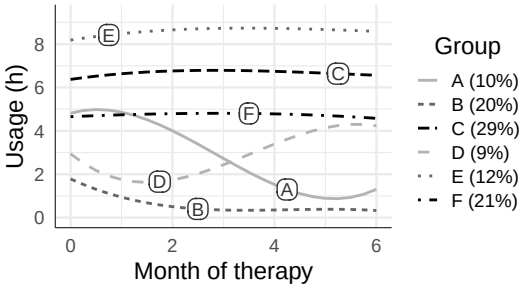
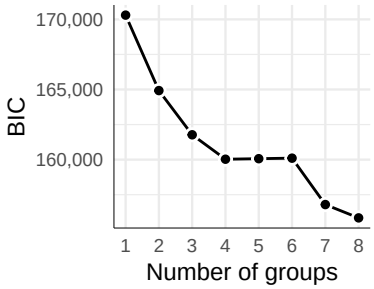
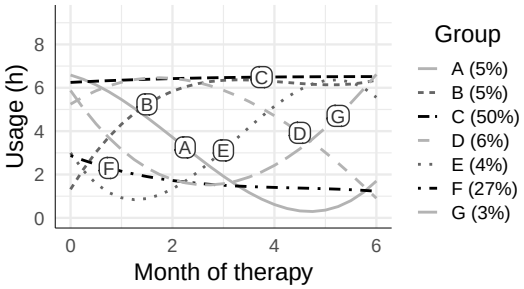


Figure 3.7: Case study analysis using GMM.

(a) The BIC per solution.



(b) The group trajectories.



needed for the model to converge. Whereas the two-class model requires 8 minutes of computation time, the eight-class model takes 4.5 hours to compute. Although GMM converged for all evaluated number of groups, the solution for five and six groups was rather poor, having 7 empty groups.

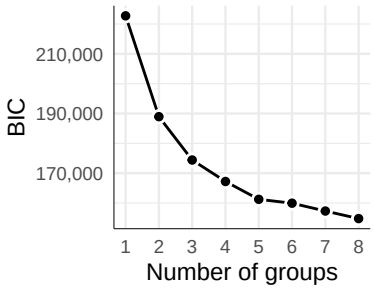
We choose the solution with 7 groups, primarily because the solutions involving a lower number of groups have a group comprising 70% of all patients, with a constant trajectory and large between-patient variability. The preferred solution is shown in Figure 3.7b and consists of two large groups (C and F) describing patients with near-constant usage. Group C is the largest group (50% of all patients) and describes a constant usage around 6 hours. Despite the large number of groups, the relative entropy is high, with a value of 0.93.

MixTVEM

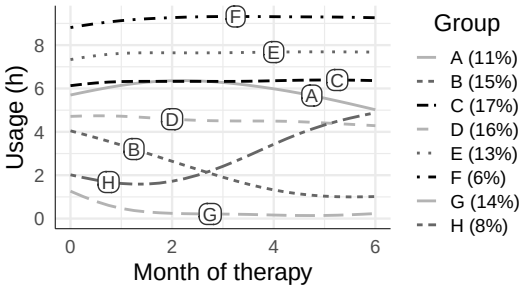
We initialize the model with 50 random starts to ensure a good fit. In contrast to the simulations, we do not include the autocorrelation term here. This was done to reduce the running time, which even with this simplification requires 7 hours to arrive at the 2-groups solution, and 36 hours with 8 groups. Across the different solutions, the group

Figure 3.8: Case study analysis using MixTVEM.

(a) The BIC per solution.



(b) The group trajectories.



trajectories discovered by MixTVEM are mostly constant, likely due to the penalization factor and the variability between patients.

It is not until 8 groups that the solution contains multiple curved group trajectories. The group trajectories are shown in Figure 3.8b. The relative entropy of 0.97 indicates a strong confidence in the classification of the trajectories. Group C, D, E, F and G (66% in total) represent constant usage over time. Out of these, groups G (14%) describes patients with near zero-hour usage. Group A (11%) contains a wide range of trajectories, considering its large within-group variability, and it is questionable whether the group trajectory is representative. Group B and H (23% in total) represent patients who decline or increase over time, respectively.

Evaluation In general, it is difficult, if not impossible, to establish with certainty whether the data truly comprises heterogeneous subgroups, as the observed subgroups could be an artifact from model misspecification (Bauer, 2007). In a more practical approach, referred to as an indirect application of clustering, the data is not assumed to comprise distinct subgroups, but instead comprises a complex spectrum which can be discretized into subgroups for ease of analysis and reporting (using e.g., KML or GBTM) (Nagin and Odgers, 2010a). None of the methods found a solution involving distinct groups, so it is difficult to establish which method achieved the best result. If one regards it as a segmentation problem, where the within-group error should be minimized, KML or GBTM would be preferable. On the other hand, GMM and GCKM focus on grouping similar trajectory shapes (i.e., the coefficients), resulting in larger groups of subjects with similar trajectories, and consequently more groups with varying shapes. Nevertheless, the confidence in the classification of the trajectories, as indicated by the relative entropy, was found to be high for all methods, despite the many groups. Care should be taken in interpreting these varying shapes, as the edges of the polynomial group trajectories may not be representative of the underlying data, but instead could be an artifact of the limited representation (Sher *et al.*, 2011). Using a spline representation, such as the one used in MixTVEM, results in more reliable trajectory estimates.

3.5 Discussion

We evaluated the ability of longitudinal clustering methods to identify group trajectories in synthetic datasets with known groups, displaying different slowly changing longitudinal patterns. The performance of the methods was assessed in terms of the group assignment agreement (via NSJ) and the group trajectory estimation error (via WMMSE). Each combination of conditions involving sample size, number of repeated observations, within-group variability, unexplained variability, and number of groups was explored. This approach allows for an objective assessment of each method, and of the degree to which performance is affected by the conditions. Although it may seem unreasonable to represent real-life datasets by relatively simple synthetic datasets comprising heterogeneous subgroups around a polynomial group trajectory (Raudenbush, 2005; Bauer, 2007), this is exactly how GMM analyses are commonly specified in practice. A similar simulation approach has been undertaken, for example, by Martin and von Oertzen (2015), and McNeish and Harring (2017). While we only evaluated each permutation of settings on 100 synthetic datasets, the actual number of evaluated datasets is much greater, primarily due to the different number of groups across settings. In the scenario involving group assignment with the correct number of groups, 27,000 cases were evaluated.

In the simulations, GMM and GCKM significantly outperform the other methods across all scenarios, both in terms of group assignment and estimation of the group trajectories. The NSJ distances of the other methods were approximately twice as high, meaning that twice as many trajectories were misclassified. The WMMSE was about 4 times higher, which suggests that the other methods were considerably worse at recovering the shape of the group trajectories. The good performance of GMM is not unexpected, after all, the heterogeneous subgroups are its best-case scenario. Moreover, similar findings have been reported in the comparison by Martin and von Oertzen (2015), although they did not evaluate GCKM. In contrast, in the simulation study of Verboon and Pat-El (2022), it was found that KML was more robust than GMM and yielded overall better results. We believe this is due to the homogeneous subgroups simulated in their study, whereas we assessed the performance under heterogeneous subgroups. Such a scenario would be an ideal case for KML and GBTM.

We found that the performance of GMM is closely matched with the two-step approach of GCKM, with the benefit that the computation time of the latter is two orders of magnitude lower. With GCKM being on-par with GMM, the discrepancy in performance of KML and GCKM demonstrates the benefit of dimensionality reduction in the first step, describing the characteristics of the trajectory more concisely. These results contrast with the findings by Twisk and Hoekstra (2012), who concluded that KML and GCKM gave similar results. KML and GBTM were found to have near-identical solutions under all scenarios. This relates to the conclusion of Feldman *et al.* (2009), who found that longitudinal latent class analysis (LLCA), a method that could be regarded as a naive clustering approach such as KML, obtains similar results to GBTM. Our findings suggest that, in general, KML is the preferred choice over GBTM because of its considerable flexibility in describing the trajectories, lower computation time, and better scaling. However, GBTM is preferred when the data contains missing or non-aligned observations, when there is prior knowledge on the shape of the group trajectories, or when covariates are to be included into the trajectories.

MixTVEM performed marginally better than KML and GBTM (a difference in NSJ

distance of -0.02) for 4 and 10 observations, despite its group trajectory estimation error being approximately 2.5 times higher. Its poor performance for 25 observations raised the overall average NSJ distance above the other methods, however. Due to its performance in our simulation and its computational burden, we consider the other methods to be preferable. In contrast to our findings regarding MixTVEM, Yang *et al.* (2019) found that MixTVEM works well in most cases in identifying the number of groups and the group trajectories, even for a larger number of observations. The difference in findings could be due to the higher level of subgroup heterogeneity in our simulations.

Having evaluated the effect of sample size under various settings, all methods except MixTVEM marginally benefited from an increased sample size. This is in agreement with previous studies (Martin and von Oertzen, 2015; Peugh and Fan, 2012). Interestingly, the sample size requirements did not go up with an increasing number of groups (up to 6). This indicates that for datasets with sufficiently distinct group trajectories, a small sample size of 200 trajectories may suffice. Although GMM and GCKM produced similar results, a shortcoming of GCKM becomes evident when comparing the performance across different number of observations. GMM performs relatively well under a low number of observations, whereas GCKM benefits from having more observations to obtain reliable estimates of the random effects. Due to the similar performance to GMM for a higher number of observations, and a better run time scaling with model complexity, GCKM is the favorable option for ILD, due to computational speed.

The assessment of the optimal solution based on the NSJ distance showed that KML, GCKM, GBTM and especially GMM were able to represent the underlying groups well for datasets with low within-group variability. Under larger within-group variability however, only GCKM and GMM were able to do this to a satisfactory degree. Optimizing for the WMMSE, it was found that especially KML and GBTM excel in representing the true underlying group trajectories, although they are surpassed by GCKM and GMM on datasets involving large within-group variability.

Applying GMM in combination with either the BIC or BLRT resulted in the correct selection of the number of groups at a high rate. The BLRT outperformed the BIC in the scenario with high within-group heterogeneity (79% against 49%), whereas the recovery rate was identical under low within-group heterogeneity (93%). The selection of the number of groups turned out to be more difficult for the other methods (around 50% for low group variability) when applying the same metrics. In particular, KML and GBTM tended to consistently overestimate the number of groups when minimizing the BIC (and BLRT in case of GBTM), which is an observation that has also been noted by others (Twisk and Hoekstra, 2012; Feldman *et al.*, 2009). While minimizing the BIC is recommended for GBTM by some (Nylund *et al.*, 2007; Frankfurt *et al.*, 2016), we found that better results were obtained across scenarios by applying the elbow method on the BIC.

The scenario involving a proportional measurement error confirms the insensitivity to measurement error of KML and GBTM that was observed in the standard scenario. The performance of GCKM and GMM degrades with increasing heteroskedasticity, but even under high measurement error the methods perform better than KML, GBTM and MixTVEM.

The evaluation of the log-normally distributed groups demonstrated that the group assignment accuracy of all methods except MixTVEM degraded, although only slightly. GMM showed the highest relative degradation in performance, indicating sensitivity of the

performance to the correct specification of the model, though not substantively (Kreuter and Muthén, 2008). The group trajectory estimation error is elevated for all methods. Most notably, GMM exhibited large group trajectory estimation errors compared to the standard scenario, significantly exceeding the errors of GCKM, KML, and GBTM.

Regarding the evaluation of the case study, the various group trajectories identified by the methods demonstrate a strong level of heterogeneity in the level of therapy adherence among patients over time. All methods identified groups resembling stable users, improving users, and declining users. There is a compelling agreement between KML and GBTM. The similarity of the group trajectories is apparent from Figure 3.4b and 3.6b, and further confirmed by the low WMMSE of 0.91. Consequently, the group assignment agreement is high as well, with an NSJ of only 0.023. In contrast, the NSJ between KML and GCKM is 0.38. The results of GCKMW are closest to those of GMM (NSJ = 0.19, WMMSE = 25), although not as close as the simulation results would suggest. MixTVEM appeared to be conservative in its estimation of the group trajectories, resulting mostly in constant trajectories whereas the other methods showed a more varied range of curves. This is likely a consequence of the regularization term.

Although mixture methods are numerically challenging to estimate, GBTM and GMM experienced few convergence problems on the synthetic datasets. This convergence rate of GMM is in contrast to our case study evaluation, as well as from experiences described by others (Tolvanen, 2007; Feldman *et al.*, 2009; Twisk and Hoekstra, 2012; Frankfurt *et al.*, 2016; McNeish and Harring, 2017), where GMM was found to exhibit convergence problems, especially for more complex specifications. It is for this reason that GBTM is the recommended method by Frankfurt *et al.* (2016). The discrepancy in convergence rate could be due to the satisfaction of all assumptions of the model for the synthetic data, whereas on real-life datasets the groups, if they exist, the subgroups are more heterogeneous. Moreover, we applied GMM with a shared variance-covariance matrix, resulting in a less complex model. The solutions of GMM comprising one or more empty groups could be partly due to a few synthetic datasets comprising duplicate groups, i.e., groups with nearly the same group trajectory. Therefore, these datasets effectively have a lower true number of groups. MixTVEM exhibited convergence problems, especially for a larger number of observations, independent of the other simulation settings. The cause of the frequent convergence problems (78% at 25 observations) is unclear. On the same topic, the numerical complexity increases significantly for the mixture models with an increasing number of observations, which results in poor scalability of these methods on ILD. In fitting a GMM or GBTM with four or 25 observations, we observed a ten-fold increase in computation time. On this aspect, GCKM and KML have a clear advantage.

3.6 Conclusion

The simulations showed that GMM and GCKM outperform KML, GBTM, and MixTVEM on datasets comprising heterogeneous subgroups. In view of the strong assumption of heterogeneous subgroups, the other methods cannot be ruled out in real-life situations for explaining heterogeneity. KML and GBTM were found to have nearly identical results when the group trajectory of GBTM were properly specified, suggesting that KML could provide a good starting point for a GBTM analysis. MixTVEM suffered from significant convergence problems at 25 observations, so under the evaluated specification, GBTM would be preferred. Overall, GMM was found to perform best. Considering the close

results between GMM and GCKM however, we recommend GCKM in ILD applications due to its computational efficiency and scaling.

Acknowledgments

The authors are grateful for the thoughtful comments provided by the anonymous reviewer, which have helped to improve the content of this work.

Supplementary materials

The R code used to run the simulation study and analyze the results and case study can be found at https://github.com/niekdt/comparison-clustering-longitudinal-data. The complete database of simulation results is available from the first author upon request.

Appendix

3.A Case study models

KML

Table 3.11: KML group trajectory parameters for the selected solution.

	Group g					
	A	B	C	D	E	F
$\pi^{(g)}$	28%	21%	19%	12%	10%	9%
$\hat{\mu}_{g,1}$	6.2	4.6	1.9	8.0	4.7	2.9
$\hat{\mu}_{g,2}$	6.6	4.6	1.5	8.4	4.9	2.6
$\hat{\mu}_{g,3}$	6.7	4.8	1.2	8.5	4.9	2.2
$\hat{\mu}_{g,4}$	6.7	4.9	1.0	8.4	5.0	2.1
$\hat{\mu}_{g,5}$	6.7	4.8	.88	8.5	4.9	1.8
$\hat{\mu}_{g,6}$	6.7	4.8	.70	8.5	4.8	1.7
$\hat{\mu}_{g,7}$	6.8	4.8	.60	8.5	4.5	1.8
$\hat{\mu}_{g,8}$	6.8	4.8	.50	8.6	4.3	1.9
$\hat{\mu}_{g,9}$	6.7	4.8	.49	8.6	4.0	1.8
$\hat{\mu}_{g,10}$	6.8	5.0	.43	8.7	4.0	1.9
$\hat{\mu}_{g,11}$	6.8	4.8	.41	8.7	3.5	2.0
$\hat{\mu}_{g,12}$	6.8	4.8	.41	8.7	2.9	2.4
$\hat{\mu}_{g,13}$	6.8	4.9	.42	8.7	2.6	2.4
$\hat{\mu}_{g,14}$	6.7	4.9	.43	8.7	2.1	2.6
$\hat{\mu}_{g,15}$	6.7	4.9	.42	8.8	2.1	2.7
$\hat{\mu}_{g,16}$	6.8	4.8	.40	8.8	1.8	3.0
$\hat{\mu}_{g,17}$	6.7	4.8	.33	8.7	1.5	3.4
$\hat{\mu}_{g,18}$	6.7	4.9	.27	8.7	1.3	3.7
$\hat{\mu}_{g,19}$	6.8	5.0	.25	8.6	1.3	4.0
$\hat{\mu}_{g,20}$	6.8	4.9	.29	8.6	1.1	4.0
$\hat{\mu}_{g,21}$	6.8	4.7	.30	8.7	1.1	4.4
$\hat{\mu}_{g,22}$	6.8	4.7	.31	8.7	.95	4.1
$\hat{\mu}_{g,23}$	6.5	4.7	.32	8.6	1.0	4.2
$\hat{\mu}_{g,24}$	6.6	4.7	.36	8.6	1.0	4.2
$\hat{\mu}_{g,25}$	6.5	4.7	.48	8.6	1.1	4.1

GCKM

Table 3.12: GCKM group trajectory parameters for the selected solution.

	Group g						
	A	B	C	D	E	F	G
$\pi^{(g)}$	44%	23%	9%	8%	5%	5%	5%
$\zeta_0^{(g)}$	5.7	3.0	5.1	3.6	4.1	4.1	4.9
$\zeta_1^{(g)}$	7.0	-44	130	-160	-320	270	-240
$\zeta_2^{(g)}$	-.81	23	-120	190	40	68	-170
$\zeta_3^{(g)}$	.31	-10	56	-37	150	-150	-4.7

GBTM

Table 3.13: GBTM group trajectory parameters for the selected solution. The standard error is reported in brackets.

	Group g					
	A	B	C	D	E	F
$\pi^{(g)}$	10%	20%	29%	9%	12%	21%
$\beta_0^{(g)}$	2.8	.60	6.7	2.9	8.6	4.7
	(.044)	(.021)	(.025)	(.035)	(.033)	(.028)
$\beta_1^{(g)}$	-320	-67	6.0	170	21	-3.7
	(7.6)	(3.8)	(3.3)	(7.8)	(4.6)	(5.1)
$\beta_2^{(g)}$	22	48	-22	78	-23	-13
	(5.8)	(3.6)	(3.0)	(6.3)	(4.5)	(3.8)
$\beta_3^{(g)}$	78	-20	4.8	-73	3.9	-.94
	(5.4)	(3.6)	(3.0)	(5.8)	(4.5)	(3.8)

GMM

Table 3.14: GMM group trajectory parameters for the selected solution. The standard error is reported in brackets.

	Group g						
	A	B	C	D	E	F	G
$\pi^{(g)}$	5%	5%	50%	6%	4%	27%	3%
$\beta_0^{(g)}$	2.7	5.4	6.4	4.9	3.5	1.7	3.3
	(.19)	(.19)	(.071)	(.18)	(.22)	(.11)	(.26)
$\beta_1^{(g)}$	-420	220	16	-290	380	-86	77
	(8.9)	(7.2)	(2.3)	(7.1)	(7.6)	(3.5)	(9.0)
$\beta_2^{(g)}$	150	-170	-5.9	-190	79	37	310
	(7.0)	(7.3)	(2.1)	(6.4)	(7.4)	(3.2)	(11)
$\beta_3^{(g)}$	88	73	.93	17	-180	-14	-24
	(6.9)	(6.6)	(1.9)	(6.8)	(7.2)	(3.1)	(9.0)

MixTVEM

Table 3.15: MixTVEM group trajectory parameters for the selected solution. Here, $B_c^{(g)}(t)$ denote the group-specific spline basis function coefficients (Dziak *et al.*, 2015).

	Group g						
	A	B	C	D	E	F	G
$\pi^{(g)}$	11%	15%	17%	16%	14%	6%	14%
$B_1^{(g)}(t)$	5.3	4.6	5.8	4.6	6.9	8.4	2.2
$B_2^{(g)}(t)$	5.7	4.1	6.2	4.7	7.4	8.8	1.2
$B_3^{(g)}(t)$	6.1	3.5	6.3	4.8	7.6	9.1	.45
$B_4^{(g)}(t)$	6.4	2.9	6.3	4.6	7.7	9.2	.24
$B_5^{(g)}(t)$	6.4	2.2	6.3	4.5	7.6	9.3	.21
$B_6^{(g)}(t)$	6.2	1.6	6.3	4.5	7.7	9.3	.21
$B_7^{(g)}(t)$	5.9	1.2	6.4	4.5	7.7	9.3	.14
$B_8^{(g)}(t)$	5.5	.98	6.4	4.4	7.7	9.3	.13
$B_9^{(g)}(t)$	5.0	1.0	6.4	4.3	7.7	9.3	.23
$B_{10}^{(g)}(t)$	4.5	1.1	6.3	4.2	7.7	9.2	.34

Chapter 4

A latent-class heteroskedastic hurdle trajectory model:

patterns of adherence in obstructive sleep apnea patients on CPAP therapy

N.G.P. Den Teuling, E.R. van den Heuvel, M.S. Aloia, S.C. Pauws
BMC Medical Research Methodology. 2021; 21: 269.
DOI: 10.1186/s12874-021-01407-6

Abstract
Sleep apnea patients on CPAP therapy exhibit differences in how they adhere to the therapy. Previous studies have demonstrated the benefit of describing adherence in terms of discernible longitudinal patterns. However, these analyses have been done on a limited number of patients and did not properly represent the temporal characteristics and heterogeneity of adherence. We illustrate the potential of identifying patterns of adherence with a latent-class heteroskedastic hurdle trajectory approach using generalized additive modeling. The model represents the adherence trajectories on three aspects over time: the daily hurdle of using the therapy, the daily time spent on therapy, and the day-to-day variability. The combination of these three characteristics has not been studied before. Applying the proposed model to a dataset of 10,000 patients in their first three months of therapy resulted in nine adherence groups, among which 49% of patients exhibited a change in adherence over time. The identified group trajectories revealed a non-linear association between the change in the daily hurdle of using the therapy, and the average time on therapy. The inclusion of the hurdle model and the heteroskedastic model into the mixture model enabled the discovery of additional adherence patterns, and a more descriptive representation of patient behavior over time. Therapy adherence was mostly affected by a lack of attempts over time, suggesting that encouraging these patients to attempt therapy daily, irrespective of the number of hours used, could drive adherence. We believe the methodology is applicable to other domains of therapy or medication adherence.

4.1 Background

For clinical efficacy, patients need to adhere to the prescribed medical treatment. The degree to which patients are successful in adhering to their treatment depends on the condition, dosing frequency, treatment duration, and many other factors (Lettieri *et al.*, 2017). Another aspect of interest is the change in adherence over time, of which an improved understanding can contribute to the early prediction of non-adherence and help in selecting the appropriate intervention. Patient adherence can either be modeled in terms of a common time trend from which patients exhibit random structural deviations, or as a stratified analysis comprising subgroups of patients with specific longitudinal patterns.

In this work we explore the longitudinal therapy adherence patterns that obstructive sleep apnea (OSA) patients exhibit during their first three months of continuous positive airway pressure (CPAP) therapy. Identifying common patterns of adherence provides population-level insights on how patients typically use the therapy. It may guide new interventions for targeting the specific adherence behaviors, or help durable medical equipment providers with substantiating their reimbursement claims.

OSA is a chronic disorder involving frequent pauses in breathing during sleep. The disorder is common in the adult population, with the prevalence ranging from 9% to 38% (Senaratna *et al.*, 2017) and increasing with age. The apneas in OSA arise from a collapse of tissue in the airway during sleep. The severity of the condition is typically measured in terms of the number of breathing disturbances per hour of sleep, referred to as the apnea-hypopnea index (AHI), where in severe cases of OSA these disturbances occur over 30 times per hour of sleep. Consequently, excessive daytime sleepiness, reduced quality of life, and increased risk of cardiovascular disease are among the side effects associated with OSA if left untreated (Kendzerska *et al.*, 2014).

CPAP is the first-line therapy for treating OSA. However, in order for the treatment to be effective, patients need to use it daily. The benefits of CPAP (e.g., reduced daytime sleepiness) can diminish after as early as one omitted day (Kribbs *et al.*, 1993). Furthermore, the dose-response relation between hours of usage and daytime sleepiness has been found to be linear, showing improved outcomes with up to 7 hours of usage per day (Weaver *et al.*, 2007). The level of adherence to the therapy is quantified in terms of the daily number of hours the treatment was used.

While most patients (66%) succeed in adjusting to CPAP therapy, others fail to start, give up early, or abandon the therapy within a couple of weeks or months (Rotenberg *et al.*, 2016). Moreover, the consistency in the number of hours used varies between patients. On some days, patients do not initiate therapy, these days are referred to as intermittent days or non-attempts. The complexity of adherence is evident from the numerous factors that have been identified to be indicative of future CPAP (non-)adherence to some degree. This includes demographic factors such as age, sex, BMI, and socioeconomic class (Shapiro and Shapiro, 2010), and equipment-related factors such as the device type (e.g., continuous or automatic PAP), device features (e.g., heated humidification), and therapy-related side effects, e.g., mask discomfort, leakage, or skin abrasion (Wickwire *et al.*, 2013). Moreover, psychological factors have been identified (Shapiro and Shapiro, 2010), for example the knowledge of patients about the therapy, the belief in ability to control one's health, the perceived risk and health benefit of the therapy, and motivation. In addition to the individual factors, external factors such as family, physician, health care professionals and facility all play a role in adherence (Shapiro and Shapiro, 2010).

Earlier studies have handled the heterogeneity of adherence by stratifying the patients on well-defined criteria. An example of this is found in the study by Weaver *et al.* (1997), in which they observed a bimodal distribution for CPAP attempt consistency, with approximately half of the patients being highly consistent (over 90% attempted days). Wohlgemuth *et al.* (2015) stratified patients based on the percentage of nights of usage, nights of usage above 4 hours, average nightly usage, and other factors. For each of these factors, the average was computed over the therapy duration, resulting in a cross-sectional cluster analysis. Using latent class analysis, they identified groups of non-adherers, attempters, and adherers.

Other studies have explored how adherence changes over time across patients, with a focus on the daily time spent on therapy. Aloia *et al.* (2008) investigated first-year CPAP therapy adherence among 71 patients in detail by visualizing individual daily time on therapy and manually grouping similar trajectories on time series characteristics (intercept, variance, slope, autocorrelation, and length). They found seven patterns of adherence. However, a limitation of their approach is that a manual evaluation is infeasible for a large number of patients. Babbin *et al.* (2015) performed a similar time series analysis on 161 patients over 180 days, but they used an automated approach for clustering the adherence trajectories. The trajectories were classified into four clusters using agglomerative hierarchical clustering of the daily time on therapy as independent variables. They identified significant differences between groups on patient characteristics in a post-hoc analysis. Wang *et al.* (2015) applied k-means among 76 patients, identifying three adherence patterns over the first 12 weeks of CPAP therapy. They showed that patients belonging to the cluster with poor adherence could be distinguished reliably from the other clusters at baseline.

Overall, these studies have yielded varying adherence patterns of interest over different ranges of time. However, these analyses involved fewer than 250 patients, which puts an upper bound on the number of groups that can reliably be detected and limits the power of the post-hoc group comparison. With respect to modeling the temporal aspect of adherence, the studies demonstrate the added value of describing adherence in terms of the attempts made and the mean level of usage, as well as the day-to-day variability in time on therapy.

We represent the adherence over time by combining the different approaches taken in previous studies. We model the daily time on therapy using latent-class distributional regression with time as a continuous covariate. Here, the daily patient usage is modeled as a two-stage process, where the daily action of initiating therapy is modeled over time as a hurdle that patients must pass before the time on therapy is modeled. Moreover, we model how the expected mean and variability of time on therapy changes over time. To the best of our knowledge, such an approach has not yet been used for modeling therapy adherence in patients (with sleep-disordered breathing). Hurdle modeling is typically used in areas involving count data, such as economics, epidemiology, healthcare utilization, and ecology. We will identify patterns of adherence in the therapy data on these three aspects by estimating a latent-class hurdle trajectory model, using a generalized additive modeling (GAMLSS) approach (Rigby and Stasinopoulos, 2005). Furthermore, we compare several CPAP therapy-related external variables between groups, following a three-step approach (Bolck *et al.*, 2004; Vermunt, 2010).

4.1.1 Data

In the present study we analyze retrospective data collected from patients in the United States who are on CPAP therapy and registered for and made use of the DreamMapper application made available by Philips Respironics. The DreamMapper application is available on mobile and the web, with the purpose of supporting patients in their first months of PAP therapy (Hardy *et al.*, 2017) and is free to use. Patients can connect their CPAP device or manually upload their CPAP device data to the DreamMapper application to gain insights into their therapy. We obtained CPAP device data from 37,235 patients who have manually uploaded data via Bluetooth or SD card to the DreamMapper application during their first 90 days of therapy and have consented to the use of their uploaded data for research. From the available dataset, we selected a random sample of 10,000 patients to conduct the regression analyses[1].

Patients included in the obtained dataset all meet the following criteria: Firstly, the patients started therapy and manually uploaded data between May 2017 and April 2018. Secondly, the patients started within a week of their DreamMapper registration date with CPAP therapy and were first-time users of the therapy. This was determined by the absence of other user accounts created by them in potentially previous times. Lastly, only patients who have uploaded therapy data beyond their first 90 days of therapy were included. This ensures that they have been on therapy for at least 90 days, regardless of the number of days the therapy was used.

It is important to note that due to the 90-day therapy requirement, the typical level of adherence in our data is higher than what would be expected from a more general patient population. We focus on patients having been on therapy for at least 90 days for two reasons. Firstly, the lack of information on the reason for the data flow stopping means that we could not distinguish between patients who abandoned therapy and those who stopped uploading data but continued their therapy. Secondly, clustering trajectories while including patients of shorter therapy durations confounds the patterns of adherent patients with those who abandoned therapy. Considering that the interpretation and possible applications are different for these two cases, it is preferable to analyze and model the cases separately. Furthermore, this simplifies the required model for our analysis, as we do not need to account for censoring or different drop-out durations.

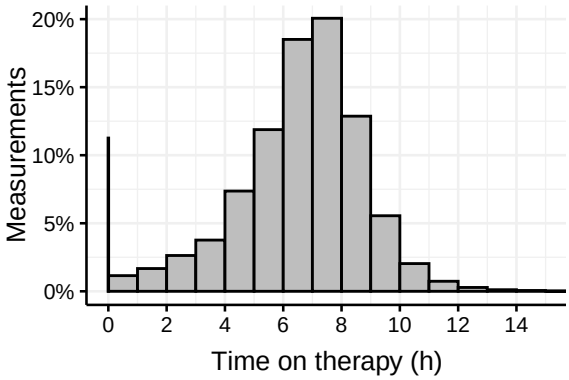
The available data per patient consists of daily aggregated CPAP device data, and a motivation assessment filled in on a voluntary basis. Patient demographics and other relevant baseline information such as the pretreatment AHI are not available for analysis. In addition to the daily amount of time patients were on therapy (the mask-on time, in seconds), as recorded by the CPAP device, we will compare the average residual AHI, average leakage, and pressure settings to identify differences between the identified groups. The motivation assessment is solicited at the very beginning of therapy, during onboarding. Patients are asked to rate their motivation to treat their sleep apnea condition on a scale from 1 to 10, where a 10 represents the highest possible level of motivation. A total of 2,973,759 observation days are available. Missing device data on intermittent days of therapy is assumed to be due to no attempt being made to use the therapy, as technical errors are deemed to be rare.

We computed the summary statistics on the complete dataset. On days during which

[1]This random subset was selected due to computational considerations in the cluster analysis for models with a large number of clusters (up to 10).

patients used the therapy, the time on therapy is approximately normally distributed with a mean of 6.7 hours (SE 0.003) and a standard deviation of 2.1 hours (SE 0.001). The distribution is slightly left-skewed, with a skewness of -0.40 (SE 0.001). Remarkably, the average usage is considerably higher than the estimate of 5.8 hours by Hardy *et al.* (2017) for patients who use the DreamMapper application in their first 90 days of therapy, which we suspect can be attributed to our exclusion of patients that stopped engaging with the app or their therapy before day 90. On average, patients did not use the therapy on 11.3% of days. To correctly model adherence over time, we therefore include these intermittent days as observations with zero hours on therapy. However, this leads to a response variable with an excess of zeroes. The overall distribution of time on therapy is shown in Figure 4.1, where intermittent days are represented by the vertical black line at zero.

Figure 4.1: Distribution of time on therapy, with intermittent days represented by zero hours.



The time on therapy ranges from 0 to 23 hours, but measurements exceeding 15 hours (0.05%) were removed because these relatively extreme values were considered to be unreliable measurements of the actual usage and could affect the model estimation. To improve the robustness of the post-hoc analysis, extreme values from other covariates were removed for the same reason, using conservative thresholds based on the lower and upper 0.01% of values. In total, fewer than 1% of observations are affected by these processing steps.

4.2 Methods

4.2.1 Hurdle model

The excess zeroes that are present in the data cannot be ignored. In count data, this is typically addressed using a zero-inflated model (Dietz and Böhning, 2000), which models the increased probability of observing zeroes, in addition to the zeroes expected from the response distribution, e.g., a Poisson distribution. However, in the present study, the counts of the number of seconds on therapy are more closely represented by a normal distribution with a strictly positive domain (e.g., the log-normal or truncated normal distribution). Additionally, in this context, zeroes have only one interpretation, namely that of the therapy not being initiated on a given day. If we regard initiating treatment

as a hurdle that patients need to overcome daily, we can model the initiation of therapy as a two-step process. This approach is referred to as hurdle modeling. It is generally applicable when a response variable is conditional on the occurrence of an event, with distinct values.

A hurdle model comprises a finite mixture of a point mass at zero, and a distribution with positive domain. Excess zeroes in a count variable can arise from a significant hurdle or a factor preventing the event from happening. This can also happen when the time available for counting the events is too short relative to the frequency of the event occurring. Lee *et al.* (2014) investigated the risk of miscarriage in women with sleep-disordered breathing using truncated Poisson hurdle regression. Hurdle modeling is not restricted to count data, as it can be applied with any distribution that does not contain the hurdle response value (e.g., a truncated distribution). Saberi *et al.* (2011) investigated the percentage of HIV medication non-adherence using a Gamma hurdle model.

Let $\mathbf{y}_i = \{y_{i,1}, y_{i,2}, ..., y_{i,J_i}\}$ denote the adherence trajectory of patient $i \in I$ consisting of $J_i = 90$ observations, for any patient from the set of available patients I. Here, $y_{i,j}$ denotes the time on therapy of patient i on the jth measurement at time t_j. The daily hurdle of initiating therapy can be modeled with a Bernoulli process with probability

$$\Pr\left(H_{i,j} = h_{i,j}\right) = \begin{cases} \nu_j & h_{i,j} = 0 \\ 1 - \nu_j & h_{i,j} = 1 \end{cases}, \tag{4.1}$$

where $h_{i,j} \in \{0, 1\}$ denotes whether the hurdle is overcome for patient i at time t_j, and $\nu_j \in (0, 1)$ represents the probability of failing to pass the hurdle at time t_j.

Truncated normal hurdle model Except for models involving two-sided truncation, as seen in double hurdle modeling, examples of left-sided truncated normal distributions are few in number. Cragg (1971) first proposed a truncated normal hurdle model for modeling the consumer demand of durable goods, to account for periods of time during which no purchases of goods were made. For observations where the hurdle is passed (i.e., $h_{i,j} = 1$), we assume the time on therapy $y_{i,j}$ to be normally distributed with strictly positive values. The probability density function (PDF) is given by

$$f_{\mathrm{N}}(y; \mu, \sigma) = \frac{\phi\left(\frac{(y-\mu)}{\sigma}\right)}{1 - \Phi\left(\frac{-\mu}{\sigma}\right)} \quad y > 0 \tag{4.2}$$

and zero otherwise, where $\phi(\cdot)$ is the standard normal PDF, $\Phi(\cdot)$ is the standard normal CDF, and μ and σ are the mean and standard deviation of the non-truncated normal distribution. If X has a normal distribution, the moments of the truncated normal distribution are then given by (Burkardt, 2014)

$$\mathbf{E}(X|X>0) = \mu + \sigma\frac{\phi(\frac{-\mu}{\sigma})}{1-\Phi(\frac{-\mu}{\sigma})}, \tag{4.3}$$

$$\mathbf{E}(X^2|X>0) = \mu^2 + 2\sigma\mu\frac{\phi(\frac{-\mu}{\sigma})}{1-\Phi(\frac{-\mu}{\sigma})} + \sigma^2\left[\frac{\frac{-\mu}{\sigma}\phi(\frac{-\mu}{\sigma})}{1-\Phi(\frac{-\mu}{\sigma})}+1\right],$$

$$\mathbf{E}(X^3|X>0) = \mu^3 + \left[3\sigma\mu^2 - \mu\sigma^2 + \sigma^3\right]\frac{\phi(\frac{-\mu}{\sigma})}{1-\Phi(\frac{-\mu}{\sigma})} + 3\sigma^2\mu\left[\frac{\frac{-\mu}{\sigma}\phi(\frac{-\mu}{\sigma})}{1-\Phi(\frac{-\mu}{\sigma})}+1\right]. \tag{4.4}$$

The time on therapy for patient i at time t_j is distributed as

$$\Pr\left(y_{i,j} \leq y\right) = \nu_j + (1-\nu_j)F_N(y_{i,j}, \mu_j, \sigma_j) \tag{4.5}$$

with $y \in \mathbf{R}$ and F_N the truncated normal distribution function.

The truncated normal hurdle distribution described here, denoted by TNH, is represented by three parameters. We allow each of the parameters to change over time. As such, each patient i at the jth observation at time t_j is represented by the probability ν_j of failing to pass the hurdle, the expected conditional mean μ_j and standard deviation σ_j. In a single-group analysis the hurdle and conditional time on therapy essentially operate on disjoint data, and therefore can be estimated separately, using logistic regression to model the hurdle, and truncated normal regression for the conditional time on therapy. However, this does not hold when the terms across the models are assumed to be correlated, or when a mixture of hurdle models is being estimated.

4.2.2 Generalized additive modeling for location, scale and shape

GAMLSS is a method for modeling a numerical univariate response variable in terms of a general parametric distribution. Whereas generalized additive modeling (GAM) and generalized linear modeling (GLM) can only handle exponential family distributions and assume a variance as a function of the mean with a constant scaling factor (Nelder and Wedderburn, 1972; Hastie and Tibshirani, 1990), GAMLSS can describe parametric response distributions by their mean (i.e., location μ), variance (i.e., scale σ) and shape (e.g., skewness and kurtosis) in terms of linear predictors and additive functions. Furthermore, through the inclusion of a distributional parameter for the excess zeros, hurdle and zero-inflated distributions can be handled.

GAMLSS was proposed by Rigby & Stasinopoulos (Rigby and Stasinopoulos, 2001, 2005), and developed into a framework implemented in various packages in R (Akantziliotou *et al.*, 2002; Stasinopoulos and Rigby, 2007). To describe our model in terms of GAM, let $\boldsymbol{y}_i^\top = (y_{i,1}, \ldots, y_{i,J_i})$ denote the longitudinal measurements of a patient $i \in I$ among the sets of patients I, with $y_{i,j} \sim \text{TNH}(\mu_{i,j}, \sigma_{i,j}, \nu_{i,j})$. Each of the distributional parameters can be described by a linear model, describing the $J = \sum_{i \in I} J_i$ observations across all patients. The PDF of the complete model is given by $f_Y(y_{i,j}; \mu_{i,j}, \sigma_{i,j}, \nu_{i,j})$. For brevity, the predictor vector of length J for the kth distribution parameter is denoted by $\boldsymbol{d}_k$, with $\boldsymbol{d}_1 = \boldsymbol{\mu}$, $\boldsymbol{d}_2 = \boldsymbol{\sigma}$, and $\boldsymbol{d}_3 = \boldsymbol{\nu}$. The general random effects GAMLSS model (Rigby and Stasinopoulos, 2009) for the kth distributional parameter is given by

$$g_k\left(\boldsymbol{d}_k\right) = \boldsymbol{X}_k\boldsymbol{\beta}_k + \sum_{m=1}^{M_k} \boldsymbol{Z}_{k,m}\boldsymbol{\gamma}_{k,m}, \tag{4.6}$$

where $g_k(\cdot)$ denotes the monotonic link function for the respective distribution parameter. The linear additive terms of the model are represented by a $J \times L_k$ design matrix denoted by $\boldsymbol{X}_k$ for L_k fixed effects, with coefficients $\boldsymbol{\beta}_k^\top = (\beta_{k,1}, \ldots, \beta_{k,L_k})$. The $J \times Q_{k,m}$ design matrix $\boldsymbol{Z}_{k,m}$ models the random effects with $\boldsymbol{\gamma}_{k,m}$ as a vector of $Q_{k,m}$ random variables. These random effects also allow for (penalized) smoothing as a function of an explanatory variable, e.g., cubic splines, P-splines, and fractional polynomials. An advantage of GAMLSS is that the random effects can be included in any of the distributional parameters, although this comes at the cost of increased computational complexity.

We limit the model complexity by only representing each distributional parameter using a linear parametric representation. In addition, the hierarchical nature of the longitudinal data needs to be considered. Patients have different levels of expected usage, variance, and attempts, arising from factors such as sleep schedule, quality of sleep, and tolerance to the therapy. We can account for these patient-specific differences by partitioning the random effects design matrix into patient-specific matrices. We only consider the case of $M_k = 1$ (i.e., a random intercept model for each distributional parameter), so we therefore omit the m subscript from the notation hereafter. The patient-specific random effects design matrix $\boldsymbol{Z}_{k,i}$ of order $J_i \times Q_k$ are concatenated to yield $\boldsymbol{Z}_k^\top = \left[\boldsymbol{Z}_{k,1}^\top \mid \boldsymbol{Z}_{k,2}^\top \mid \cdots \mid \boldsymbol{Z}_{k,|I|}^\top\right]$ (Rigby and Stasinopoulos, 2009). The random effects vector is denoted by $\boldsymbol{\gamma}_{k,i} = (\gamma_{k,1,i}, \ldots, \gamma_{k,Q_k,i})$, with $\boldsymbol{\gamma}_{k,i} \sim N(0, \Sigma_k)$ for each of the distribution parameters, where Σ_k is the variance-covariance matrix for the random effects of the respective distribution parameter.

Although we observe a marginally better fit using smoothing functions of time, modeling change using linear additive terms is preferred in this analysis for its lower complexity, and greatly reducing computation time. We therefore model each of the distributional parameters using a second-order polynomial dependent on time. The identity link function suffices for the mean $\mu_{i,j}$, whereas a log link is used for the variance $\sigma_{i,j}$ to ensure positive values. The hurdle probability $\nu_{i,j}$ is modeled using logistic regression by assuming a logit link $g_3(\nu_{i,j}) = \log\left(\frac{\nu_{i,j}}{1-\nu_{i,j}}\right)$. Accordingly, the random effects model is given by

$$\begin{aligned}
\mu_{i,j} &= \beta_{1,0} + \beta_{1,1}t_{i,j} + \beta_{1,2}t_{i,j}^2 + \boldsymbol{Z}_{1,i,j}\boldsymbol{\gamma}_{1,i}, \\
\log\sigma_{i,j} &= \beta_{2,0} + \beta_{2,1}t_{i,j} + \beta_{2,2}t_{i,j}^2 + \boldsymbol{Z}_{2,i,j}\boldsymbol{\gamma}_{2,i}, \\
\log\frac{\nu_{i,j}}{1-\nu_{i,j}} &= \beta_{3,0} + \beta_{3,1}t_{i,j} + \beta_{3,2}t_{i,j}^2 + \boldsymbol{Z}_{3,i,j}\boldsymbol{\gamma}_{3,i}.
\end{aligned} \tag{4.7}$$

We will use this model to compare against the mixture model described in the next section.

Latent-class modeling

The findings from previous studies on CPAP adherence suggest a complex, non-normal distribution of adherence patterns (Aloia *et al.*, 2008; Babbin *et al.*, 2015). We therefore opt for a non-parametric approach to modeling the heterogeneity, by describing the patient-specific deviations from the population mean in terms of a finite number of structural deviations. In a cross-sectional data context, this approach is commonly referred to as finite

mixture modeling (McLachlan and Peel, 2000). This has the added benefit of accounting for the (possibly non-linear) relationship between the distributional parameters through the different clusters. An association can be expected between the attempt probability and the mean level of usage.

Growth mixture modeling (GMM) is an approach to modeling longitudinal change (i.e., a growth curve), accounting for patient heterogeneity by assuming each patient belongs to one of several unobserved (i.e., latent) classes (Verbeke and Lesaffre, 1996; Muthén and Shedden, 1999; Muthén *et al.*, 2002). The class models include patient-specific random effects; therefore the approach essentially assumes the heterogeneous data to consists of a set of heterogeneous subgroups.

The appeal of allowing for patient-specific deviations within the latent classes is that it enables an emphasis on the change of adherence over time as opposed to the expected average time on therapy. Without a random intercept, most of the group trajectories would be representing the differences in mean time of therapy, resulting in many constant group trajectories. To a lesser degree, patients may also exhibit different levels in their attempt probability and conditional standard deviation. However, in consideration of the increased model complexity with an increasing number of latent classes, we opt for simplifying the class model. We therefore only include a random intercept $\gamma_i^{(g)} \sim N(0, \sigma_\gamma)$ for the mean level. Each latent class is described by a model, where the model for class g is described by

$$\begin{aligned}
\mu_{i,j}^{(g)} &= \eta_{1,i,j}^{(g)} = \beta_{1,0}^{(g)} + \beta_{1,1}^{(g)} t_{i,j} + \beta_{1,2}^{(g)} t_{i,j}^2 + \gamma_i^{(g)}, \\
\log \sigma_{i,j}^{(g)} &= \eta_{2,i,j}^{(g)} = \beta_{2,0}^{(g)} + \beta_{2,1}^{(g)} t_{i,j} + \beta_{2,2}^{(g)} t_{i,j}^2, \\
\log \frac{\nu_{i,j}^{(g)}}{1 - \nu_{i,j}^{(g)}} &= \eta_{3,i,j}^{(g)} = \beta_{3,0}^{(g)} + \beta_{3,1}^{(g)} t_{i,j} + \beta_{3,2}^{(g)} t_{i,j}^2.
\end{aligned} \tag{4.8}$$

Each of these class models represents a proportion of the overall heterogeneity in the data. The overall model is given by

$$f(y_{i,j}; \boldsymbol{\Theta}, \boldsymbol{\pi}) = \sum_{g=1}^{G} \pi_g f_g \left(y_{i,j}; \boldsymbol{\beta}_1^{(g)}, \boldsymbol{\beta}_2^{(g)}, \boldsymbol{\beta}_3^{(g)}, \sigma_\gamma \right) \tag{4.9}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(G)}\}$ comprises the group model parameters, f_g denotes the model for group g, and $\boldsymbol{\pi}$ is the vector of group proportions π_g for group g with $\pi_g \geq 0$ and $\sum_g \pi_g = 1$. The class assignment of patients is probabilistic, which contrasts with other approaches such as longitudinal k-means (KML) where the cluster edges are well-defined but arbitrarily selected due to the distance measure used (Genolini and Falissard, 2010).

A few studies have used a similar approach in the context of hurdle modeling. Maruotti (2011) proposed a longitudinal latent-class hurdle mixed effects model that accounts for missing data patterns arising from drop-outs. They applied the model for the analysis of skin cancer counts, of which the data had a considerable number of missing measurements, in addition to zero inflation. Moreover, Ma *et al.* (2018) used a log-normal hurdle mixture to identify patterns of factors contributing to vehicle crash rates. To the best of our knowledge no studies have used a hurdle approach with within-class heterogeneity using GAMLSS up to now, in particular when combined with class-specific temporal heteroskedasticity.

4.2.3 Model estimation

The analysis is performed in R 3.5.0 (R Core Team, 2022) using version 5.1-2 of the `gamlss` package (Rigby and Stasinopoulos, 2005) for the implementation of GAMLSS. The GAMLSS model is fitted using the RS algorithm proposed by Rigby and Stasinopoulos (1996). The algorithm maximizes the (penalized) maximum likelihood of the full model using expectation maximization (EM). The estimation of the random patient factor is based on penalized quasi-likelihood. The zero-truncated normal distribution is available in the `gamlss.tr` package (version 5.1-0) (Stasinopoulos and Rigby, 2018) and was adapted to account for excess zeros by the parameter ν.

We estimate the mixture model specified in Equation 6.1 using a nonparametric maximum likelihood (NPML) approach (Aitkin, 1996; Einbeck and Hinde, 2006), as implemented in the `gamlssNP()` function in the package `gamlss.mx` (version 4.3-5) (Stasinopoulos *et al.*, 2017). This approach describes the data heterogeneity through a non-parametric density function comprising a finite mixture (Laird, 1978; Einbeck and Hinde, 2006; Stasinopoulos *et al.*, 2017). The marginal likelihood for the data is given by

$$f(\boldsymbol{y};\boldsymbol{\Theta},\boldsymbol{\pi}) = \prod_{i\in I}\sum_{g=1}^{G}\left[\pi_g \prod_{j=1}^{J_i} f_g\left(y_{i,j};\boldsymbol{\theta}^{(g)}\right)\right]. \tag{4.10}$$

Here, the group parameters $\boldsymbol{\theta}^{(g)}$ represent the mass points of the non-parametric density, occurring with probability (i.e., the masses) $\pi_1,\ldots,\pi_G$, respectively.

Each trajectory is assumed to have been generated by one of the group models, however, the true group membership is unknown. The membership of the trajectory $\boldsymbol{y}_i$ to group g is indicated by $\delta_{i,g}$, with $\delta_{i,g}=1$ if the trajectory belongs to group g, and $\delta_{i,g}=0$ otherwise. The vector of group indicators for the trajectory i is denoted by $\boldsymbol{\delta}_i=(\delta_{i,1},\ldots,\delta_{i,G})$. We denote the set of all indicator vectors across patients by $\boldsymbol{\delta}$. With this, the likelihood of the model with specified group memberships $\boldsymbol{\delta}$, referred to as the complete model, is given by

$$\begin{aligned} L(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\pi},\boldsymbol{\delta}) &= f(\boldsymbol{y},\boldsymbol{\delta}) \\ &= f(\boldsymbol{y}|\boldsymbol{\delta})f(\boldsymbol{\delta}) \\ &= \prod_{i\in I} f(\boldsymbol{y}_i|\boldsymbol{\delta}_i)f(\boldsymbol{\delta}_i) \\ &= \prod_{i\in I}\prod_{g=1}^{G}\left[\pi_g^{\delta_{i,g}}\prod_{j=1}^{J_i} f_g\left(y_{i,j}\right)^{\delta_{i,g}}\right]. \end{aligned} \tag{4.11}$$

Here, the parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}$ were left out for conciseness. A more detailed derivation is provided by Stasinopoulos *et al.* (2017). The log-likelihood of the complete model is given by

$$\ell_c = \sum_{i\in I}\sum_{g=1}^{G}\delta_{i,g}\log\pi_g + \sum_{i\in I}\sum_{g=1}^{G}\sum_{j=1}^{J_i}\delta_{i,g}f_g(y_{i,j}). \tag{4.12}$$

The complete model with G classes is equivalent to a weighted regression model over repeated data observations for $g=1,...,G$ with an additional covariate indicating the class membership. The observations are weighted by the posterior probability

$$w_{i,g} = \hat{\pi}_{i,g} = \frac{\pi_g \prod_{j=1}^{J_i} f_g\left(y_{i,j}; \boldsymbol{\theta}^{(g)}\right)}{\sum_{g'=1}^{G} \pi_{g'} \prod_{j=1}^{J_i} f_{g'}\left(y_{i,j}; \boldsymbol{\theta}^{(g')}\right)}. \tag{4.13}$$

The latent class proportions of the mixture model are computed from the respective average posterior probability, given by

$$\hat{\pi}_g = \frac{1}{|I|} \sum_{i \in I} w_{i,g}. \tag{4.14}$$

The EM algorithm is initialized by fitting a fixed-effects GAMLSS model. In the E-step, the patient weights are updated, followed by the maximization of the weighted GAMLSS model likelihood in the M-step. The optimization process is halted when the reduction in the deviance, computed as $D = -2 \log L$, falls below a certain threshold. In the analysis, we use a lenient threshold of 0.3, which we determined to provide sufficiently stable results due to the large amount of data, while halting relatively quickly. Details on the algorithm and the initialization are given by Einbeck and Hinde (2006). We observed stable solutions across repeated random starts, which is in agreement with findings by other researchers (Aitkin, 1996; Einbeck and Hinde, 2006). Nevertheless, the model does fail to converge sometimes so repeated random starts are recommended.

4.2.4 Evaluation

Prior to the mixture modeling analysis, we explore several mixed models based on Equation 4.7, with polynomial random effects $\sum_{p=0}^{P} \gamma_{k,p,i} t_{i,j}^p$ of order P in the predictor η_k of the respective distributional parameter. In addition, we estimate a fixed effects model as a baseline.

We assess the model fit of the models by investigating the standardized residuals for normality, using a detrended quantile-quantile (Q-Q) plot. The different models are compared using the Akaike information criterion (AIC). The AIC measures the amount of information lost about the data by the model representation while penalizing overfitting. It is defined as $\text{AIC} = 2m - 2 \log L$, where m is the number of model parameters, and L is the likelihood of the model. This is a specific case of the generalized Akaike information criterion (GAIC) (Akaike, 1983), which is recommended for comparing non-nested models (Rigby and Stasinopoulos, 2005). Only models that converged successfully are evaluated. Likelihood ratio tests were considered but yielded consistent p-values of zero for any improvement in AIC due to the large sample size, and therefore we do not report them in the results section. Lastly, we measure the separation between classes in terms of the relative entropy (Muthén, 2004), given by

$$\text{relative entropy} = 1 - \frac{\sum \sum_{g=1}^{G} -\hat{\pi}_i^{(g)} \log \hat{\pi}_i^{(g)}}{|I| \log G}. \tag{4.15}$$

For the selected mixture model, we compare the subgroups to create distinct descriptors of the groups, and to highlight meaningful differences between the groups in terms of adherence. We assess the group trajectories on each of the distributional parameters visually. Furthermore, we explore whether the groups differ on any of the other available

covariates of interest, which are the residual AHI, leakage, pressure settings, and motivation score. Here, each patient is assigned to the most likely group (i.e., modal assignment).

Although the additional covariates could have been included in the GAMLSS mixture model, this was omitted for practical reasons because the computation time would increase considerably. Furthermore, preliminary tests on a random subset of 1,000 patients yielded mostly the same groups as the mixture model without the inclusion of an additional covariate. We therefore apply a three-step approach, where in the first step the mixture model is estimated (i.e., the measurement model). In the second step, each patient is assigned to their most likely group. Lastly, the patient groups are compared on the covariates of interest. The means are compared using ANOVA F-tests, whereas the medians of skewed distributions are compared using the Kruskal-Wallis test. Due to the large sample size of this study, even small differences between groups are statistically significant. Instead, we will only highlight practically significant differences that are deemed clinically relevant.

The three-step approach has been shown to lead to biased estimates on the effects of external variables (Vermunt, 2010). We therefore considered the modified Bolck-Croon-Hagenaars (BCH) approach, which applies a correction for the misclassification errors (Vermunt, 2010; Bakk *et al.*, 2013). However, when applied to the case study at hand, we observed that the correction did not result in a meaningful difference in the mean estimates between groups, nor different conclusions on statistical significance. This is likely attributable to the large sample size, and low misclassification error due to the large number of observations per trajectory.

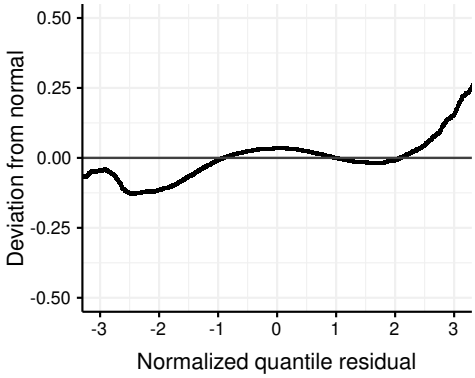
4.3 Results

The model fits for different degrees of polynomials ($P = 0, 1, 2$), and the fixed effects model are reported in Table 4.1 in terms of the AIC. An increasing order of the random effects is associated with an improved model fit. However, the model involving the quadratic random effects failed to converge despite repeated random starts. The detrended Q-Q plot of the linear random effects model shown in Figure 4.2 indicates that the residual deviations from the normal distribution are closely concentrated around zero, suggesting that the normalized quantile residuals are approximately normally distributed. However, the pattern of negative deviation at the tails indicates the presence of many outliers, i.e., heavy tails.

Table 4.1: The single-group model estimates.

Random effects in η_k	AIC
None	4,070,897
Constant	3,267,326
Linear	3,209,083
Quadratic	Did not converge

Figure 4.2: Detrended Q-Q plot of the single-group model with linear random effects.

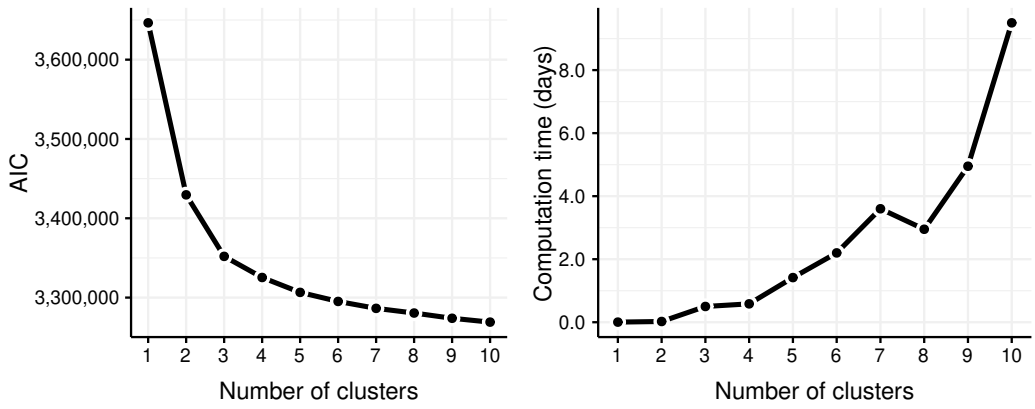


4.3.1 Number of groups

We determine the best fitting mixture model for 1 to 10 classes. For each number of groups, ten models were fitted using random starts, out of which the model with the lowest AIC was selected. Overall, the solutions among the repeated random starts, in the cases where convergence was reached, are stable. This is consistent with observations by Aitkin (1999) for this type of estimation.

Figure 4.3: Metrics for the model solutions for 1 to 10 groups.

(a) AIC. (b) Average computation time (in days).



The AIC of the best model for each number of groups is shown in Figure 4.3a. The monotonically decreasing curve suggests a consistent but diminishing improvement in model fit with an increased number of groups. With the aim of exploring the various ways in which patients adhere to the therapy, and in consideration of the sufficient amount of data for a post-hoc analysis, a solution involving many groups is justified, and supported by the AIC and likelihood ratio test ($p \ll 0.01$). An alternative solution of interest would be the solution involving three groups, which is the solution at which the improvements in

the consecutive models start to diminish.

We choose the nine-groups solution because it provides a better fit than solutions involving fewer groups, as indicated by the AIC. The eight-groups solution lacks the group trajectory exhibiting considerably increase in usage over time which is present in the solution with nine groups. On the other hand, the ten-group solution is almost identical to the preferred solution, except for an additional constant group trajectory which we deem to be not of interest.

The computation time increased considerably with an increasing number of groups, up to the point where model estimation is no longer practical. Whereas the single-group model takes 34 minutes to compute on an Intel Xeon E5-2660 (2.6 GHz) processor, the five-groups model needed 34 hours on average, and the ten-group models completed only after 228 hours on average. The average computation time for each number of groups is shown in Figure 4.3b. The models involving nine groups or less either converged within 50 outer iterations or failed to convergence. The ten-groups models converged within 78 iterations.

4.3.2 Adherence groups

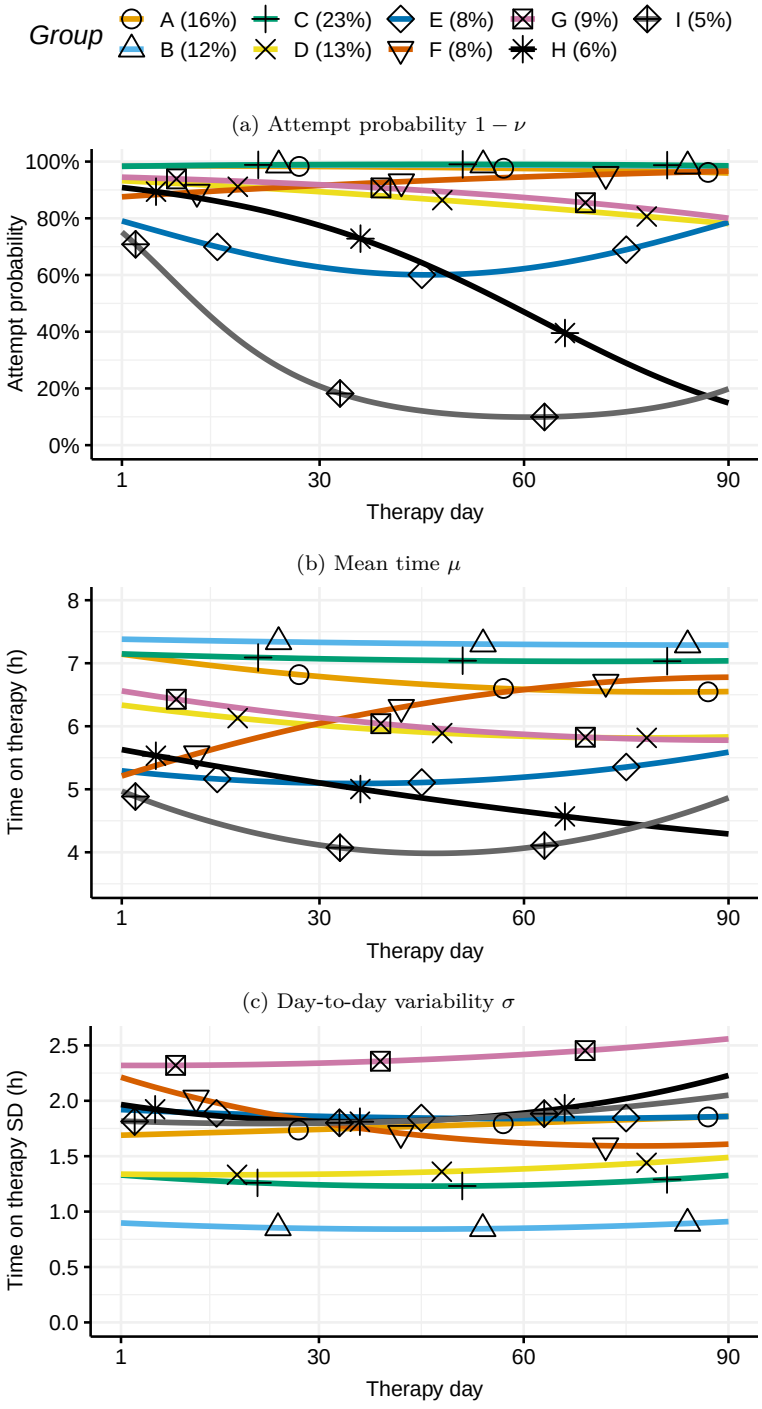
The nine group trajectories are shown in Figure 4.4 for each of the distributional parameters, with the model coefficients shown in Table 4.2. The means of the group trajectories are reported for day 1, 45 and 90 in Table 4.3.

The value range of the group trajectories for the mean time on therapy is surprisingly narrow, with only a 3.5-hour difference between the lowest and highest group. The small proportion of patients that fall outside of this range are accounted for by the random intercept in the group models. The difference between the mean group trajectories is especially small in relation to the day-to-day standard deviation of usage, with standard deviations ranging between 0.84 and 2.4 hours. In addition, there is a considerable spread within groups on the mean intercept, with $\sigma_\gamma = 1.5$ hours. Despite of the high day-to-day variability, the large number of observations available per patient allows for a reliable classification, as indicated by the high relative entropy of 0.93.

The group trajectories show a gradual change in mean usage over time, which is possibly due to patients changing their usage at different moments throughout the therapy. In contrast, the changes in attempt probability are more profound, with a significant group of patients that tend to nearly cease the therapy within the first month. Overall, the attempt probability and its change over time differ considerably between groups, with some groups achieving near-perfect consistency in daily attempts (99%), and other groups using the therapy sporadically (attempts on 15% of days) towards day 90. Several of the groups exhibit a small increase in usage variability over time. In some groups, this change in variability appears to coincide with a change in mean usage, possibly indicating a mean-variance relationship. In general, the group trajectories with higher usage have lower variability.

Group A, B, and C represent highly consistent users, making up the majority of patients (51%). Group B (12%) and C (23%) represent patients that have no trouble adhering to therapy, with a consistent average attempt probability of 99%, usage averaging around 7 hours, and having the lowest day-to-day variability of all groups. The discerning factor between the groups is the day-to-day variability of group C of 1.3 hours, compared to the even lower standard deviation of 0.90 hours for group B. The patients of group A (16%)

Figure 4.4: The identified group trajectories for each distributional parameter.



(a) Attempt probability $1 - \nu$

(b) Mean time μ

(c) Day-to-day variability σ

Table 4.2: Group trajectory coefficients.

Group	$\hat{\pi}_g$	logit ν			μ			log σ		
		$\beta_{2,0}$	$\beta_{2,1}$	$\beta_{2,2}\cdot 10^3$	$\beta_{1,0}$	$\beta_{1,1}$	$\beta_{1,2}\cdot 10^3$	$\beta_{2,0}$	$\beta_{2,1}\cdot 10^2$	$\beta_{2,2}\cdot 10^4$
A Variable users	16%	-4.2	.0032	.093	7.1	-.015	.093	.52	.10	.0057
B Consistent users	12%	-4.1	-.015	.18	7.4	-.0021	.012	-.11	-.31	.36
C Good users	23%	-4.1	-.025	.26	7.1	-.0034	.023	.29	-.36	.39
D Stable decliners	13%	-2.6	.018	-.031	6.3	-.014	.090	.29	-.064	.20
E Strugglers	8%	-1.3	.042	-.46	5.3	-.012	.17	.65	-.15	.12
F Improvers	8%	-2.0	-.015	-.014	5.2	.035	-.19	.79	-.88	.57
G Variable decliners	9%	-2.8	.014	.029	6.6	-.018	.096	.84	-.011	.13
H Dropouts	6%	-2.3	.032	.15	5.6	-.020	.053	.68	-.49	.69
I Early dropouts	5%	-1.1	.11	-.94	5.0	-.044	.47	.60	-.13	.29

Table 4.3: The group trajectories at day 1, 45 and 90.

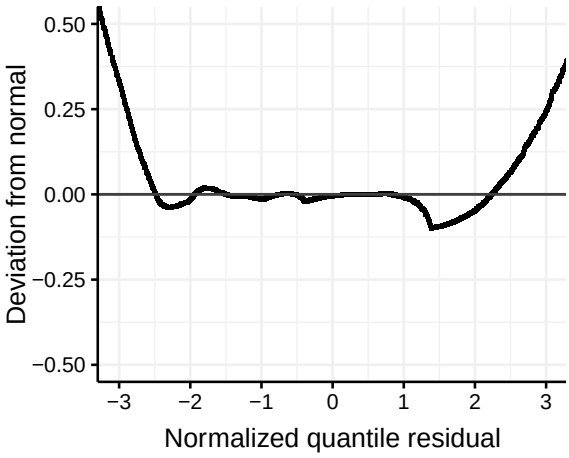
Group	Attempt probability $1-\nu$			Mean time μ (h)			SD time σ (h)		
	Day 1	Day 45	Day 90	Day 1	Day 45	Day 90	Day 1	Day 45	Day 90
A Variable users	99%	98%	96%	7.1	6.7	6.5	1.7	1.8	1.9
B Consistent users	98%	99%	98%	7.4	7.3	7.3	.90	.84	.90
C Good users	98%	99%	99%	7.1	7.0	7.0	1.3	1.2	1.3
D Stable decliners	93%	87%	78%	6.3	5.9	5.8	1.3	1.4	1.5
E Strugglers	79%	59%	77%	5.3	5.1	5.6	1.9	1.8	1.9
F Improvers	88%	93%	97%	5.2	6.4	6.8	2.2	1.7	1.6
G Variable decliners	95%	90%	80%	6.6	6.0	5.8	2.3	2.4	2.6
H Dropouts	91%	65%	15%	5.6	4.9	4.3	2.0	1.8	2.2
I Early dropouts	75%	11%	17%	5.0	3.9	4.8	1.8	1.8	2.0

achieve nearly the same consistency in attempts as group B and C, but show a decrease in usage of half an hour throughout the therapy. Moreover, the standard deviation is considerably higher at 1.8 hours. In terms of usage, group D (13%) and G (9%) follow a trajectory similar to group A, but have a reduced number of attempts by day 90 from 94% to 80%. The difference between these two groups lies in their day-to-day variability. With a standard deviation of around 2.5 hours, patients in group G have the highest variability of all groups by far. Group E, F, H, and I represent struggling patients (for a total of 27% of patients, with 8%, 8%, 6% and 5% respectively). These patients tend to have a lower average usage already at the start of therapy. Whereas the patients from group F improve with time, the usage of the other groups either remains constant or decreases over time. The strugglers in group E exhibit a stable usage over time, but a diminished number of attempts around the second month of therapy. Group H and I comprise patients who decrease in number of attempts, the separating factor between the groups is the time at which attempts are no longer made or only occur sporadically.

We assess the fit of the mixture model to the data using the detrended Q-Q plot shown in Figure 4.5. The normalized quantile residuals closely follow a normal distribution,

except for the heavy tails, indicating that the data contains considerable outliers with respect to expected group trajectory. The deviation around 1.3 can primarily be attributed to observations from patients of the early-dropouts group I, suggesting that more group trajectories are needed to adequately describe the trajectories of these patients, or that a different model is needed for this specific group.

Figure 4.5: Detrended Q-Q plot of the preferred mixture solution.



4.3.3 Group comparison

We compare the identified groups on the covariates described in Section 4.1.1 based on the measurements in the first week, and across all 90 days of therapy. The group mean and median values for each covariate, along with the standard deviation and interquartile range respectively, are reported in Table 4.4. The median attempted days, time on therapy and day-to-day variability are reported for reference. The proportion of compliant days was determined by the number of days with time on therapy exceeding 4 hours out of 90 days. The pressure-related covariates were only available in 675 patients. The minimum and maximum CPAP pressure settings remained unchanged in 81% of patients, therefore the median values in week 1 are not reported. Already in the first week of therapy, group A, B and C comprise relatively more compliant patients than the other groups, with most patients achieving daily compliance. This is in contrast to group E and I with a respective proportion of only 57% and 43%. All groups except F (the improvers) show a decline in compliant days over time. The decline is considerable for the drop-out groups H and I, which can be attributed to the reduced number of attempts over time.

With respect to the residual AHI, the median and lower percentiles between groups is minute. In contrast, the differences at the 75^{th} percentile and higher are more pronounced, where the AHI of groups E, H and I is higher by 1 event/hour. It is worth noting that only the early drop-outs group (I) do not show a decrease in AHI relative to the first week. As patients with consistently high AHI may have abandoned the therapy prematurely, we also investigate the average proportion of patient residual AHI measurements exceeding 15 events/hour across the groups (referred to as high residual AHI in Table 4.4). Differences between the groups are present from week 1 onward, notably between the more adherent

groups (A-D) and the other groups. Even more so, the differences are greater in the period following the first week, with the groups exhibiting struggling or drop-out behavior (groups E, H, I) having over twice the rate of high residual AHI compared to the adherent groups A, B and C.

Leakage was found to be practically identical between groups. We therefore also investigated cases of high leakage. For the high leak analysis, the available research data did not allow for an adjustment of the relevant factors for leakage (most importantly, the type of mask used). Instead, we therefore evaluated the within-patient leakage variability. We computed the standard deviation of the day-to-day differences in leakage for each patient (referred to as SD leakage in Table 4.4). Leakage variability was found to be highest in the drop-out groups (H and I), and lowest in the consistent-users group (B). The drop-out groups (H and I) tend to have a higher proportion of patients with the lowest possible minimum CPAP pressure of 5 cmH_2O compared to the other groups. This could be due to these groups comprising patients with a less severe form of sleep apnea, or a suboptimal device configuration. The motivation score provided by patients during the first week of therapy ranges from 1 to 10. There are considerable proportional differences between groups. The drop-out and struggling groups have a higher proportion of patients who rated their motivation below 4. Conversely, patients in groups with the highest level of adherence, B and C, patients were more likely to be motivated from the start.

Table 4.4: Unadjusted group estimates for the covariates (between-patient standard deviation enclosed in parentheses). The median is reported for covariates with a considerably skewed distribution (lower and upper quartiles enclosed in brackets).

	Variable users (16%)	Consistent users (12%)	Good users (23%)	Stable decliners (13%)	Strugglers (8%)	Improvers (8%)	Variable decliners (9%)	Dropouts (6%)	Early dropouts (5%)
	A	B	C	D	E	F	G	H	I
Attempted days (%)	98 [97, 100]	100 [98, 100]	100 [98, 100]	88 [82, 91]	69 [60, 74]	93 [90, 97]	89 [84, 93]	59 [50, 68]	26 [14, 33]
Time on therapy (h)	6.7 (1.3)	7.3 (.91)	7.1 (1.1)	6.0 (1.2)	5.3 (1.3)	6.3 (1.3)	6.1 (1.4)	5.1 (1.5)	4.7 (1.9)
Time on therapy SD (h)	1.8 (.20)	.86 (.13)	1.3 (.14)	1.4 (.23)	1.8 (.39)	1.8 (.27)	2.3 (.40)	1.8 (.46)	1.6 (.56)
Compliant days (%)									
Week 1	100 [86, 100]	100 [100, 100]	100 [100, 100]	86 [71, 100]	57 [43, 86]	71 [43, 86]	86 [71, 100]	71 [57, 86]	43 [14, 71]
All	90 [83, 96]	99 [98, 100]	98 [94, 99]	79 [69, 86]	50 [36, 61]	81 [71, 89]	70 [60, 80]	41 [29, 53]	11 [4.4, 22]
Residual AHI (events/h)									
Week 1	2.7 [1.5, 4.8]	2.5 [1.3, 4.7]	2.4 [1.2, 4.6]	2.9 [1.5, 5.3]	2.9 [1.5, 5.8]	3.1 [1.5, 5.5]	3.0 [1.7, 5.3]	3.1 [1.7, 5.9]	2.6 [1.3, 5.3]
All	2.3 [1.3, 4.0]	2.1 [1.3, 3.9]	2.1 [1.1, 3.7]	2.5 [1.4, 4.3]	2.8 [1.5, 5.2]	2.4 [1.3, 4.2]	2.7 [1.5, 4.4]	2.8 [1.6, 5.1]	2.6 [1.4, 4.9]
High residual AHI (%)									
Week 1	3.4	3.1	3.0	3.6	4.7	4.9	4.3	4.6	5.1
All	2.1	1.9	1.6	2.4	4.0	2.2	2.6	3.6	4.5
Leakage (L/min)									
Week 1	38 (13)	37 (12)	38 (13)	37 (13)	36 (13)	38 (14)	36 (12)	36 (12)	36 (13)
All	38 (12)	37 (11)	38 (12)	37 (13)	37 (12)	38 (12)	37 (12)	36 (12)	37 (13)
SD leakage (L/min)									
Week 1	4.0 [2.4, 6.9]	3.5 [2.0, 5.9]	3.8 [2.2, 6.5]	3.7 [2.1, 6.9]	4.0 [2.1, 7.5]	4.4 [2.3, 8.0]	3.9 [2.1, 6.7]	4.0 [2.2, 7.0]	3.8 [2.2, 8.2]
All	4.7 [3.1, 7.2]	3.6 [2.4, 5.3]	4.1 [2.7, 6.2]	4.6 [2.8, 7.4]	5.0 [3.1, 8.1]	5.0 [3.0, 7.7]	4.9 [3.2, 7.4]	5.1 [3.0, 8.7]	5.2 [3.3, 9.1]
Min pressure (cmH_2O)	6 [5, 8]	6 [5, 8]	6 [5, 8]	6 [5, 8]	6 [4, 7]	6 [5, 8]	6.0 [5.0, 8.0]	5.0 [4.0, 6.0]	5.0 [4.0, 7.0]
Avg pressure (cmH_2O)									
Week 1	9.4 (3.1)	9.3 (3.2)	9.0 (2.6)	8.4 (2.0)	9.6 (2.9)	9.1 (3.3)	8.3 (2.5)	8.5 (2.4)	9.4 (2.8)
All	9.6 (3.1)	9.3 (3.0)	9.3 (2.9)	8.7 (2.3)	9.5 (3.1)	9.2 (2.9)	8.6 (2.9)	8.8 (2.5)	9.2 (2.9)
Max pressure (cmH_2O)	15 [12, 20]	15 [13, 16]	15 [12, 20]	15 [12, 20]	13 [11, 16]	15 [13, 20]	15 [12, 20]	15 [11, 16]	15 [12, 20]
Patient motivation (%)									
Week 1: score < 4	1.7	1.1	.67	2.4	3.0	1.6	1.5	5.6	6.2
Week 1: $4 \leq$ score ≤ 6	9.2	4.8	5.9	12	17	12	9.6	17	19
Week 1: score > 6	89	94	93	86	80	86	89	77	74

4.4 Discussion

Previous studies have explored CPAP adherence patterns using relatively small datasets involving fewer than 250 patients. Moreover, the behavioral characteristics on which the clusters in these studies are based are more limited in scope, with most studies using the time on therapy as the response to cluster on (Babbin *et al.*, 2015; Wang *et al.*, 2015). Although the number of clusters that could be found in these studies was largely limited by sample size (as pointed out by Wohlgemuth *et al.* (2015)), the fewer model characteristics and lower granularity of measurements have likely also played a role. Despite the different selection of patients, characteristics, and therapy duration, some agreements can be found with other studies on the mean usage over time. In particular, constant and declining patterns of usage are commonly found.

The proposed latent-class hurdle model based on GAMLSS allows for a more detailed description of adherence over time in OSA patients undergoing CPAP therapy, modeling changes in attempts, time on therapy, and day-to-day variability in a single model. The nine identified group trajectories emphasize the complexity surrounding CPAP therapy adherence. A narrow majority of patients who used the DreamMapper application (51%) exhibited a stable adherence pattern across the first 90 days of therapy, with the most distinguishing characteristic being the day-to-day variability. Other patients exhibit a change over time, typically a decline. The group trajectories involving a change over time have similar characteristics in the first week of therapy, suggesting that there are other factors involved that determine how the patient adherence shifts over time. Identifying these contributing factors presents opportunities for early interventions (D'Rozario *et al.*, 2016).

We have identified several possible contributing factors. The largest differences were observed in the motivation score. This score, assessed in the first week of therapy, showed large proportional differences, where patients with low motivation are more likely to belong to the drop-out groups. Including additional psychosocial factors studied in literature would likely help to explain the observed group trajectories further (Crawford *et al.*, 2014; Cayanan *et al.*, 2019). The comparison of residual AHI yielded only minor differences in median AHI between groups. However, the differences were more pronounced in the upper quartile, indicating that struggling and drop-out groups may comprise a subgroup of patients with a higher residual AHI. This is further demonstrated by the different occurrence rates of high residual AHI, with the strugglers and drop-out groups having over twice the rate compared to the most adherent groups. Similarly, variability in leakage was found to differ after the first week of therapy, with the drop-out groups having the highest variability in leakage.

It was essential to model the conditional mean usage by a random intercept because of the variability in intercept between patients with the same change over time. This variance component was not needed for the attempt probability as the patterns of the drop-out and declining groups are much more distinct from the other groups. Although the inclusion of day-to-day variability in the model resulted in the identification of additional groups, the day-to-day variability showed little change over time in most patients, with the improvers from group F being the notable exception.

All group trajectories remained above the minimum compliance threshold of 4 hours, suggesting that even in the group with the lowest average time on therapy (group I, the early drop-outs), patients met the threshold on average. Our findings suggest that

across groups, therapy adherence is mostly affected by a decrease in attempts over time, suggesting that the focus on encouraging patients to attempt the therapy daily is more important than increasing the hours of usage above the compliance threshold.

It is important to note that because most patients that abandon the therapy do so within 90 days, our results are biased towards the more adherent patients. We suspect this is why the groups with average usage below 4 hours identified by Babbin *et al.* (2015), Wang *et al.* (2015) and Wohlgemuth *et al.* (2015) were not found in our analysis. On a similar note, our estimates of therapy factors such as AHI or leakage are also likely to be lower, as significant issues on these aspects could contribute to patients abandoning therapy. Furthermore, due to the selection of patients who used the DreamMapper application, the findings may not be representative of the general sleep apnea population (Hardy *et al.*, 2017).

The residual analysis showed that the model fits the data adequately, with only the tails of the distribution departing from a normal distribution. The heavy tails could likely be accounted for by including more random effects into the class models, allowing a greater range of patient-specific deviations from the group trajectory. Alternatively, a truncated distribution with a heavier tail than the truncated normal distribution (e.g., the t-distribution) could be used. The choice for the normal distribution was a trade-off between model complexity and model fit, as both proposed alternatives would increase the model complexity and estimation time considerably.

Due to the excessive computation time for the nine- and ten-class models, we restricted the regression analyses to a random subset of 10,000 patients out of the available 37,235 patients. To ensure that this would not affect our results, we conducted a preliminary analysis where we visually determined that the group trajectories were sufficiently stable from random samples comprising 5,000 patients each. Considering the large number of data points per patient, a feature-based approach could have possibly provided a similar solution in a significantly shorter amount of time. In such an approach, the patient trajectories are estimated independently, after which latent class analysis is performed on the trajectory coefficients.

Bearing in mind the high day-to-day variability observed within patients, the model fit could be improved further if factors can be determined that explain some of the observed variability. Moreover, the hurdle model assumes that the occurrence of intermittent days are independent events while the factors that affect attempts may last several days (e.g., illness). Modeling intermittent days as a state change lasting one or more days could provide an improved description, especially for patients who are struggling with the therapy.

Overall, the proposed methodology provides a detailed description of patient adherence behavior over time, especially in comparison to earlier studies. Our approach is useful to researchers, clinicians, and durable medical equipment (DME) providers for discovering common patterns of adherence in their (sub)population of interest, and gaining insights into how adherence behavior differs between patients. Moreover, such insights could help DME providers better identify the risk of overpay in reimbursement claims based on adherence levels. Lastly, the proposed model can be used to assign new patients to the most likely adherence pattern, enabling the detection of behaviors of interest. Identifying problematic patterns of adherence may help in better recognizing and targeting patients who are struggling with the therapy.

4.5 Conclusion

We have demonstrated the feasibility and benefits of applying a latent-class heteroskedastic hurdle trajectory model to adherence data with a large number of patients. The inclusion of the hurdle model and the heteroskedastic model into the mixture model enabled the discovery of additional adherence patterns, and a more descriptive representation of patient behavior over time. Most importantly, the analysis revealed a strong non-linear association between the progression of attempts over time, and the average time on therapy. The methodology presented here can be applied to behavioral data in other domains involving the tracking of compliance over time.

Acknowledgments

The authors wish to thank Michael Kane for his advice on CPAP device data, which has helped in designing the data processing steps and interpreting the group comparison findings. We are also grateful for the valuable feedback provided by the anonymous reviewers.

Ethics approval and consent to participate

The data collection and analysis were approved by the Philips Research internal review board; the Internal Committee for Biomedical Experiments (ICBE), Reference number ICBE-2-14816, on 6 May 2019. All methods were carried out in accordance with relevant guidelines and regulations. The users registering with the DreamMapper application were invited to agree with the Terms of Service and provide consent to the use of their data before they could use the application. Before consent was collected, the users were invited to consult the privacy notice to be informed about their data rights, as well as Philips' data practices. The privacy notice was accessible to the users during the consent moment, and after consent via the DreamMapper application. Users could withdraw consent at all times by following the indications in the privacy notice. The privacy notice contained information on the right to withdraw consent, among other rights. Only those users of the DreamMapper application who consented for their data to be used for research and product improvement purposes, were included in the study.

Chapter 5

latrend: A framework for clustering longitudinal data

N.G.P. Den Teuling S.C. Pauws E.R. van den Heuvel
Submitted.

Abstract
Clustering longitudinal data is a useful way to explore differences between subjects on their change over time for a measurement of interest. Various `R` packages have been introduced throughout the years for identifying clusters of longitudinal patterns, summarizing the variability between subject trajectories in terms of one or more common trends. We introduce the `R` package `latrend` as a framework for the unified application of methods for longitudinal clustering, enabling comparisons between methods with minimal coding. The package also serves as an interface to commonly used packages for clustering longitudinal data, including `dtwclust`, `flexmix`, `kml`, `lcmm`, `mclust`, `mixAK`, and `mixtools`. This enables users to easily compare different approaches, implementations, and method specifications. Furthermore, users can build upon the standard tools provided by the framework to quickly implement new cluster methods, enabling rapid prototyping. We demonstrate the functionality and application of the `latrend` package on a synthetic dataset based on the therapy adherence patterns of patients with sleep apnea.

5.1 Introduction

In this work, we consider the case where subjects are measured on the same variable repeatedly over a period of time. This type of data is referred to as longitudinal data. No two subjects are identical, and therefore observations made across subjects may develop differently over time. Modeling the variability between subjects leads to an improved understanding of the different trajectories that may occur.

Usually, a longitudinal dataset is represented by a single general trend, i.e., an average representative trajectory indicating the expected change over time. Here, subjects exhibit a random deviation from the general trend. However, there may be observed and unobserved factors that contribute to structural deviations, causing a single general trend to be an inadequate representation of the trajectories. In other cases, the distribution of random deviations is difficult to model parametrically. In both situations, multiple common trends, i.e., longitudinal clusters, may provide a better representation of the data (Hamaker, 2012). The number of clusters or the definition of the clusters is typically not known in advance and needs to be estimated from the data.

An example of a domain where modeling the between-subject variability is of interest is in the monitoring of therapy adherence of patients with sleep apnea undergoing positive airway pressure (PAP) therapy. Here, therapy adherence is measured in terms of the number of hours of sleep during which the therapy is used, recorded daily. Patients exhibit different levels of adherence to the therapy, depending on many factors such as their sleep schedule, motivation, self-efficacy, and the perceived importance of therapy (Cayanan *et al.*, 2019). Moreover, patients may exhibit a different level of change over time, depending on their initial usage and their ability to adjust to the therapy. Due to the many possibly unobserved factors involved, researchers have used longitudinal clustering to summarize the between-subject variability in terms of longitudinal patterns of therapy adherence (Babbin *et al.*, 2015; Den Teuling *et al.*, 2021; Yi *et al.*, 2022).

Clustering longitudinal data is a practical approach for exploring the variability between subjects. This variability is summarized in terms of a manageable number of common trends, which are identified in an unsupervised manner from the data using a cluster algorithm. Such an approach is especially useful for exploring datasets involving a large number of trajectories, where a visual inspection of the trajectories would be impractical. In essence, the data is assumed to comprise several groups, each with a different longitudinal data generating mechanism. It differs from cross-sectional clustering due to the need to account for the dependency between observations within subjects, and the presence of temporal correlation of the repeated measurements.

A number of packages have been created in `R` (R Core Team, 2022) that can be used for clustering longitudinal data. However, for researchers analyzing a novel case study, choosing the best method or implementation is not straightforward. This is partly due to a lack of guidelines on how the most appropriate method should be selected, but also due to the inherent exploratory nature of such an analysis. Considering that each of these packages have been created to fulfill a gap in the capabilities of other already existing implementations or approaches, there is value in comparing the results for the case study at hand. Be that as it may, the evaluation of different approaches across packages is an activity of significant effort, as the method inputs, estimation procedure, and cluster representations differ greatly between packages.

The aim of the `latrend` package is to facilitate the exploration of heterogeneity in

longitudinal datasets through a variety of cluster methods from various fields of research in a standardized manner. The package provides a unifying framework, enabling users to specify, estimate, select, compare, and evaluate any supported longitudinal cluster method in an easy and consistent way, with minimal coding. Most importantly, users can easily compare results between different approaches, or run a simulation study. The `latrend` package is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=latrend and on GitHub at https://github.com/philips-software/latrend.

A second key aim of the package is extensibility so that users can adapt the framework or methods to their needs. Users are able to extend the framework with new methods or add support for another existing method by creating a new implementation of the framework interface. The effort in implementing new methods is considerably reduced due to the standard longitudinal cluster functionality provided by the framework.

Currently, a total of 18 methods for longitudinal clustering are supported. To provide support for such a variety of approaches, the `latrend` package interfaces with an extensive set of packages that provide methods that are applicable for clustering longitudinal data, including `akmedoids` (Adepeju *et al.*, 2020), `crimCV` (Nielsen, 2018), `dtwclust` (Sardá-Espinosa, 2019), `flexmix` (Grün and Leisch, 2008), `funFEM` (Bouveyron, 2015), `kml` (Genolini *et al.*, 2015), `lcmm` (Proust-Lima *et al.*, 2017), `mclust` (Scrucca *et al.*, 2016), `mixAK` (Komárek, 2009), and `mixtools` (Benaglia *et al.*, 2009). In this way, we build upon the cluster packages created by the `R` community. Support has also been added for MixTVEM; a mixture model proposed and implemented as an `R` script by Dziak *et al.* (2015).

To the best of our knowledge, such a comprehensive package does not yet exist in the context of clustering longitudinal data. The `latrend` package has similar aspirations as the `flexmix` package, which also provide extensible framework for (multilevel) clustering. However, the scope of our package is purposefully broader, to facilitate users to apply approaches from various fields of research. Our framework is agnostic to the specification, estimation, and representation used by the methods.

The chapter is organized as follows. A short overview of different approaches to clustering longitudinal data is given in Section 5.2. In Section 5.3, the design principles and high-level structure of the framework are described. The usage of the package is demonstrated in Section 5.4. Section 5.5 describes three ways in which users can implement their own cluster methods. Lastly, a summary and future steps are presented in Section 5.6.

5.2 Methods

We will briefly describe the general approaches to clustering longitudinal data. Moreover, we summarize the main strengths of these approaches. For brevity, we do not go into the specifics of any particular package. We refer to the accompanying articles of these packages for further details.

We begin by describing the aspects which all the approaches have in common. Let the repeated observations of the trajectory from subject i be denoted by

$$\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, ..., y_{i,J_i}),$$

where $y_{i,j}$ is a numerical value of some variable of interest, $t_{i,j}$ is the measurement time,

and J_i is the number of observations of trajectory $\boldsymbol{y}_i$ for subject i.

Any method for clustering longitudinal data approximates the dataset heterogeneity in terms of a set of K clusters, with each cluster representing a segment of the between-subject heterogeneity. These clusters may be discovered by identifying groupings of similar subjects, based on their trajectory. Typically, a cluster method is estimated for a given number of clusters, specified by the user. Here, users apply a cluster method for a different number of clusters to determine the most appropriate number of clusters for the respective data.

Each cluster represents a proportion of the population, denoted by π_k. The vector of cluster proportions is denoted by $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_K)$, with $\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$. Subjects are generally assumed to belong to a single cluster. Therefore, many cluster methods partition the subjects into k mutually exclusive sets $I_1, I_2, ..., I_K$, where I_k denotes the set of subjects to belong to cluster k, with $\bigcup_{k=1}^{K} I_k = I$. Depending on the application, it may be desirable to identify a representation for each cluster, also referred to as the cluster center, which provides a summary of the cluster. This representation may be obtained from the averaged representation of all the subjects assigned to the respective cluster, by designating a representative subject, or through the internal cluster representation used by the method, if applicable.

If two clusters overlap, there is an inherent uncertainty in the cluster membership of subjects. Some methods account for this uncertainty by estimating the probability or degree (i.e., weight) to which subjects belong to each cluster, depending on their similarity to the respective cluster. In case of well-separated clusters, this subject weight may be practically zero for the other available clusters.

A commonly used type of probabilistic cluster model is the finitemixture model, described by

$$f(\boldsymbol{y}_i) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x}_i, \boldsymbol{\theta}_k), \tag{5.1}$$

where $f(\cdot)$ denotes the density function, $\boldsymbol{\theta}_k$ the representational parameters for cluster k. Using this model, the probability of a subject belonging to cluster k, denoted by $\pi_{i,k}$, is determined from the posterior probability given the subject data and model parameters, given by

$$\Pr(k|\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\Theta}) = \pi_{i,k} = \frac{\pi_k f(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\theta}_k)}{\sum_{k'} \pi_{k'} f(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{\theta}_{k'})}. \tag{5.2}$$

In applications where each subject is assumed to belong to one cluster, subjects are typically assigned to the cluster with the highest subject-specific posterior probability, referred to as modal assignment.

5.2.1 Cross-sectional clustering

Cross-sectional cluster algorithms group objects together based on similarity, measured through a set of object characteristics also referred to as features. In cluster algorithms such as k-means, these features are assumed to be independent, although this is generally not a strict requirement. When applied to longitudinal data, the features represent the different moments in time. The temporal independence assumption of the observations yields a non-parametric description of the trajectories. This makes it a useful approach

for an exploratory analysis without any prior assumptions on the shape of the trajectories. However, the approach does come with some limitations on the type of longitudinal data. Firstly, the observations must be aligned between trajectories, i.e., taken at the same respective moment in time. Secondly, there must be an equal number of observations per trajectory. Consequently, missing observations should be imputed.

An example of a cross-sectional approach is longitudinal k-means (KML). KML applies the k-means cluster algorithm directly to the observations. The cluster trajectories are determined by the averaged observations of trajectories assigned to the respective cluster. The method is implemented in the `kml` package (Genolini *et al.*, 2015).

A cross-sectional mixture approach is seen in longitudinal latent profile analysis (LLPA), otherwise known as longitudinal latent class analysis (Muthén, 2004). Here, latent profile analysis, more commonly referred to as a Gaussian mixture model, is used to describe each moment in time as a normally distributed random variable. A dataset with trajectories each comprising J observations is thus described by J independent normals, each modeling the response distribution at a different moment in time t_j, giving

$$f(\boldsymbol{y}_i) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{N} \phi(y_{i,j}|\mu_{k,j}, \sigma_{k,j}). \tag{5.3}$$

Gaussian mixture models, and thereby LLPA, can be estimated using, for example, the `mclust` package (Scrucca *et al.*, 2016).

5.2.2 Distance-based clustering

As an alternative to clustering based on the repeated measurements directly, distance-based cluster algorithms operate on the pairwise dissimilarity between objects, using a (dis)similarity measure. Such methods take a pairwise dissimilarity matrix as input, where the choice of the dissimilarity metric is left to the user. Examples of cluster algorithms that use this approach include k-medoids and agglomerative hierarchical clustering. Given the trajectories of subject a and b, the dissimilarity metric is denoted by $d(\boldsymbol{y}_a, \boldsymbol{y}_b)$. As an example, the Euclidean distance

$$d(\boldsymbol{y}_a, \boldsymbol{y}_b) = \sqrt{\sum_j (y_{b,j} - y_{a,j})^2}.$$

may be used as the dissimilarity metric.

The approach is commonly used for time series clustering[1], and the list of available dissimilarity metrics that have been proposed over the past decades is extensive (Aghabozorgi *et al.*, 2015). The advantage of distance-based clustering over cross-sectional clustering is that the dissimilarity measure allows for the assessment of the trajectories beyond directly comparing observations. Raw data metrics such as the Euclidean distance assume that the observations are perfectly aligned, whereas in dynamic time warping, the shift in temporal alignment between trajectories is corrected. Other metrics compute the distance based on a different representation, such as the autocorrelation, spectral components, entropy, or a time series model. Many dissimilarity metrics are implemented in the `dtwclust` package (Sardá-Espinosa, 2019).

[1] Clustering longitudinal data can be regarded as a special case of time series clustering where the time series have a common starting point.

5.2.3 Model-based clustering

In model-based clustering, the longitudinal dataset is modeled by a regression model comprising a mixture of submodels. It is also referred to as latent-class trajectory modeling. This approach comprises a versatile class of (semi-)parametric methods. Most importantly, the shape of the trajectories can be represented using a parametric model, requiring fewer parameters compared to a non-parametric approach. Measurements can be taken at different times between subjects, and covariates can be accounted for. Moreover, users can incorporate assumptions into the modeling of the trajectories and clusters, such as the distribution of the response variable, the within-cluster variability, and heteroskedasticity.

A straightforward example of model-based clustering involves modeling the population as a mixture of cluster trajectory models. This is referred to as group-based trajectory modeling (GBTM) or latent-class growth analysis (LCGA). It is essentially a mixture of linear regression models, with

$$y_{i,j} = \boldsymbol{x}_{i,j}\boldsymbol{\beta}_k + \varepsilon_{i,j,k} \quad \text{for } i \in I_k, \tag{5.4}$$

where $\boldsymbol{x}_{i,j}$ is the $N \times B$ design matrix of B terms, $\boldsymbol{\beta}_k$ are the B group-specific coefficients, and $\varepsilon_{i,j,k}$ is the normally distributed residual error with zero mean and variance σ^2. The design matrix contains covariates of time, enabling the model to describe the change in response over time. External covariates can be included to further explain the dependent variable.

The expected value of a measurement $y_{i,j}$ for the model in Equation 5.4 is given by

$$E(y_{i,j}) = \sum_{k=1}^{K} \pi_k[\boldsymbol{x}_{i,j}\boldsymbol{\beta}_k]. \tag{5.5}$$

GBTM is available, for example, in the packages `lcmm` (Proust-Lima *et al.*, 2017) and `crimCV` (Nielsen, 2018).

A popular form of model-based clustering that does consider within-cluster variability is growth mixture modeling (GMM) (Muthén, 2004), which represents a mixture of linear mixed models. The within-cluster variability is modeled through subject-specific random effects, e.g., a random intercept. This allows researchers to assess the deviations between subjects within a cluster. The linear mixed model for cluster k is given by

$$y_{i,j} = \boldsymbol{x}_{i,j}\boldsymbol{\beta}_k + \boldsymbol{z}_{i,j}\boldsymbol{u}_{k,i} + \varepsilon_{i,j,k} \quad \text{for } i \in I_k. \tag{5.6}$$

Here, $\boldsymbol{z}_{i,j}$ is the $N \times U$ design matrix for the U random effects, and $\boldsymbol{u}_{k,i}$ are the subject-specific random coefficients for cluster k. The random effects are assumed to be normally distributed with mean zero and variance-covariance matrix Σ_k. The marginal mean is computed by

$$E(y_{i,j}|\boldsymbol{u}_i) = \sum_{k=1}^{K} \pi_k[\boldsymbol{x}_{i,j}\boldsymbol{\beta}_k + \boldsymbol{z}_{i,j}\boldsymbol{u}_{k,i}]. \tag{5.7}$$

GMM is available in packages such as `lcmm` (Proust-Lima *et al.*, 2017), `mixtools` (Benaglia *et al.*, 2009), and `mixAK` (Komárek, 2009).

A challenge with this approach is the large number of parameters that need to be estimated, which typically increases linearly with the number of clusters. The estimation

may fail to converge or may yield empty clusters. This is usually handled by repeatedly fitting the model with random starts, or by providing better starting values for the coefficients.

5.2.4 Feature-based clustering

In a feature-based approach, each trajectory is independently represented by a set of temporal characteristics (i.e., features, coefficients). The trajectories are then clustered based on the feature values using a cross-sectional algorithm. This is equivalent to applying a distance-based approach with a model-based dissimilarity metric but has the advantage of allowing users to combine arbitrary features. The approach is used, for example, by the anchored k-medoids algorithm provided by the `akmedoids` package. Here, the trajectories are represented using linear regression models, and are clustered based on the model coefficients (Adepeju *et al.*, 2020). Compared to the rather time-intensive model-based clustering approach, the trajectory models only need to be estimated once. A disadvantage compared to model-based clustering is that the reliability of the trajectory coefficients depends on the available data per trajectory. This approach therefore generally requires a larger number of observations per subject to yield similar results.

5.2.5 Identifying the number of clusters

Due to the exploratory nature of clustering, the number of clusters is typically not known. Moreover, most of the cluster methods require the user to specify the number of clusters. The preferred number of clusters for the respective method can be determined by estimating the method for an increasing number of clusters, followed by comparing the solutions by means of an evaluation metric. In such a comparison for a particular method, the interpretation of the metric is consistent across the solutions, as they all originate from the same method specification.

Many metrics are available, depending on the type of method that is being applied. For example, in distance-based methods, the solutions are typically evaluated in terms of the separation between clusters. Cluster separation is measured by the distance between trajectories or cluster trajectories, e.g., using the average Silhouette width (ASW) (Rousseeuw, 1987) or the Dunn index (Arbelaitz *et al.*, 2013). In contrast, a model-based approach typically has no notion of the distance between trajectories, but instead measures the likelihood of the overall model on the given the data, enabling the use of likelihood-based evaluation such as the Bayesian information criterion (BIC), Akaike information criterion (AIC), or likelihood ratio test (van der Nest *et al.*, 2020). Specific to cluster regression methods where the longitudinal observations are modeled at the subject level, assessing the solution in terms of the residual errors of the trajectories may be of interest. Examples of such metrics include the mean absolute error (MAE) and root mean squared error (RMSE). For probabilistic assignments these metrics may be weighted by the posterior probability of the trajectories, denoted by WMAE and WRMSE, respectively.

Overall, the preferred metric depends on the type of method under consideration and the case study domain. Users are advised to follow recommendations from literature for the respective method. Moreover, it is advisable to use the evaluation metric merely as guidance in identifying the preferred solution, as a trade-off between the number of clusters and the interpretability of the solution. Lastly, it is worthwhile to factor in domain

knowledge into the selection of cluster solutions (Nagin *et al.*, 2018).

5.2.6 Comparing methods

The approaches may yield considerably different results, arising from fundamental differences in the temporal representation and similarity criterion of the methods. We provide a high-level summary of strengths and limitations of the approaches in Table 5.1, which helps to guide the user towards an initial selection of applicable approaches relative to the case study at hand. Note that even for methods of the same type of approach, results may differ depending on how the trajectories are represented, trajectory similarity is measured, or how clusters are formed. Considering that the most suitable approach or method is typically not known in advance, it is advisable to evaluate and compare the solutions between methods to identify the most suitable method for the respective case study. The resulting solutions can then be compared using an external evaluation metric.

A useful starting point in comparing the preferred solutions between methods is to evaluate the similarity between the cluster partitions. After all, if both candidate methods find a similar cluster partition, this would indicate that both methods find the same grouping despite representational differences. In contrast, if the cluster partitions are dissimilar, it may suggest that either a hybrid approach could be of interest, or that one method is preferred over the other.

The similarity between cluster partitions of two methods can be assessed using partition similarity metrics such as the adjusted Rand index (ARI) (Hubert and Arabie, 1985), variance of information, or the split-join index. These metrics are applicable to any method and are even applicable when the solutions have a mismatching number of clusters. In some case studies, a ground truth may be available in the form of a reference cluster partition. Partition similarity metrics such as the ARI may then be used to identify the solution that most closely resembles the ground truth. Alternatively, users may obtain a partial ground truth by manually annotating a subset of the trajectories based on domain knowledge.

For methods that have a longitudinal representation of the clusters, it can be insightful to assess the similarity between the cluster trajectories. A possible metric for this is the weighted minimum mean absolute error (WMMAE) (Den Teuling *et al.*, 2021), which evaluates the WMAE for each cluster with its nearest cluster. It is defined as

$$\text{WMMAE} = \frac{1}{J}\sum_{k=1}^{K}\left[\pi_k \min_{k'\in\{1,\ldots,K\}}\sum_{j=1}^{J}|y_{k,j}-\hat{y}_{k',j}|\right], \tag{5.8}$$

where J is the number of observations in cluster trajectory $\hat{\boldsymbol{y}}_k$, and $\hat{y}_{k,j}$ denotes the predicted expected value of the cluster trajectory at time t_j. The reference cluster trajectory observations are denoted by $y_{k,j}$. The interpretation of the value of the WMMAE is relative to the scale of the response variable.

Solutions may be compared further by assessing the compactness of the clusters or the separation between clusters on a common distance metric, for example using the average Silhouette width or the Dunn index. This is useful to identify the method that is best at identifying distinct subgroups.

Table 5.1: Summary of the general strengths of limitations of the different approaches to longitudinal clustering.

Approach	Strengths	Limitations
Cross-sectional	• No assumptions on the shape of the cluster trajectories • Low sample size requirement • Very fast to estimate • Suitable for initial exploration	• Requires time-aligned trajectories of equal length • Requires complete data • Does not account for the temporal relation of observations
Distance-based	• Flexible in the choice of distance metric(s) • Trajectory distance matrix only needs to be computed once • Fast to estimate	• Distance matrix computation is not practical for a large number of trajectories • Pairwise comparison of trajectories is more sensitive to noise • Many distance metrics require time-aligned trajectories
Model-based	• Low sample size requirements due to inclusion of parametric assumptions • Can handle missing data • Can handle trajectories of unequal length and variable time • Can account for covariates • Relatively robust to trajectories that do not fit the representation	• May be challenging to estimate (convergence problems) • Computationally intensive to estimate
Feature-based	• Temporal features only needs to be computed once • Very fast to estimate • Fast alternative to model-based approach given a sufficiently large sample size	• Sensitive to trajectories that do not fit the representation • Trajectory-independent feature estimation is more sensitive to observational outliers

5.3 Software design

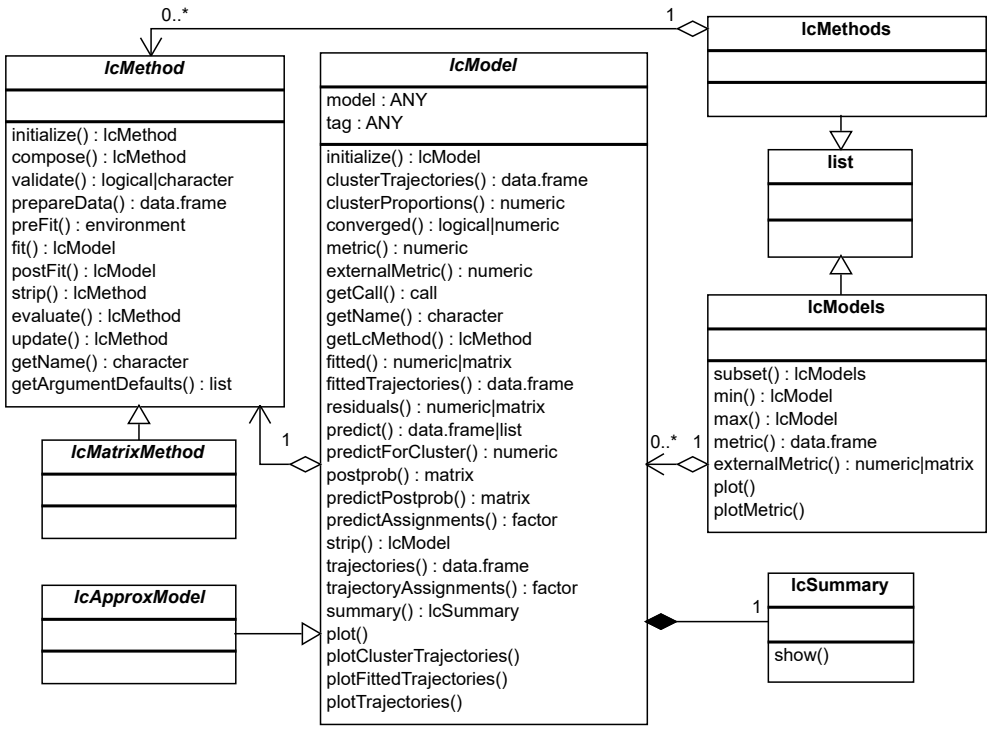
We begin by providing a high-level description of the framework, outlining the main functionality of the classes. The software is built on an object-oriented paradigm using the `S4` system, available in the `methods` package (R Core Team, 2022). We have chosen to use the `S4` paradigm over `S3` due to the more complete set of object-oriented features, including class inheritance, object validation, and method signatures. Functions defined by the package are in camel case unless a generic function is already available in base `R`. A schematic representation of the framework classes is shown in Figure 5.1. While we explain the functionality of the classes and what kind of information the classes store, we avoid describing the internal structure of the classes to the slot level, as these should not be directly accessed or modified by users unless stated otherwise.

The framework is designed to provide a standardized way of specifying, estimating, and evaluating different longitudinal cluster methods, ensuring ease of use regardless of the underlying cluster package being used. This is achieved by defining two main interfaces. Firstly, there is an interface for method specification and estimation, and secondly, there is an interface for representing the estimated method. To enable user extensions with minimal coding, the two interfaces are defined through abstract base classes. These classes provide basic functionality, from which the user can extend certain functions as needed by creating a subclass. For brevity and to distinguish the method specification more clearly from the fitted method, the abstract class for the method specification and estimation interface is named `lcMethod`, whereas the abstract class for the fitted method interface is named `lcModel`. Here, the word "model" should be taken in the broadest sense of the word, where any resulting cluster partitioning represents the data, and thereby is regarded as a model of said data. For example, users can specify a GMM through a `lcMethodLcmmGMM` object, specifying the GMM and the estimation settings. The resulting estimated GMM is represented by a `lcModelLcmmGMM` object.

The `latrend()` function is the main function of the framework by which users can estimate cluster models, taking a `lcMethod` object and dataset as input, and applying the estimation procedure defined in the `lcMethod` object to the input data. Another advantage of having stand-alone estimation functions is that it enables standard validation of the inputs and outputs, which otherwise would need to be implemented for each method. Moreover, it ensures all methods take the same data format as input. While in most cases the complete dataset is used for method estimation, different resampling techniques may be of interest for obtaining a more robust solution, or for validating a solution, i.e., model. The resampling techniques are independent from the method being applied, and therefore this functionality is encapsulated into separate estimation functions in the framework, prefixed by '*latrend*'.

We have selected the `data.frame` in long format as the internal data representation, as is common in `R`. Here, each row represents an observation for a trajectory at a given time, possibly for multiple covariates. The trajectory and time of an observation are indicated in separate columns. This format can represent irregularly timed measurements, a variable number of observations per trajectory, and an arbitrary number of covariates of different types. Since not all datasets are readily available in this format, the `latrend()` estimation functions handle data input by calling the generic `transformLatrendData()` function. Currently, this transformation is only defined for `matrix` input. Users can implement the method to add support for other longitudinal data types.

Figure 5.1: Class diagram of the framework. Standard `S3` methods and private class slots and are not shown.



5.3.1 The lcMethod class

The `lcMethod` class has two purposes. The first purpose is to record the method specification, defined by the method parameters and other settings, referred to as the method arguments. The second purpose is to provide the logic for estimating the method for the specified arguments and given data. `lcMethod` objects are immutable. Users only interact with a `lcMethod` object for retrieving method arguments, or for creating a new specification with modified arguments. This functionality is provided by the base `lcMethod` class.

The base `lcMethod` class stores the method arguments in a list, inside the `arguments` slot. The method arguments can be of any type. The names of subclasses are prefixed by '*lcMethod*'. Subclasses can validate the model arguments against the data by overriding the `validate()` function. Due to the specific internal structure of a `lcMethod` object, constructors are defined for creating `lcMethod` objects of a specific class for a given set of arguments. In `lcMethod` implementations that are a wrapper around an existing cluster package function, the method arguments are simply passed to the package function. The required arguments and their default values are obtained from the formal function arguments of the package function at runtime.

The evaluation of the method arguments is delayed until the method estimation process

is started. This enables a `lcMethod` object to be printed in an easily readable way, where the original argument expressions or calls are shown, instead of the evaluation result. This is useful when an argument takes on a function or complex data structure, and it reduces the memory footprint when a large set of method permutations is generated and serialized, such as in a simulation study.

The estimation process is divided into six steps that process the method arguments, prepare and validate the data, and fit the specified method. The steps are implemented through six generic functions: `prepareData()`, `compose()`, `validate()`, `preFit()`, `fit()`, and `postFit()`. All functions except for `fit()` are optional.

1. The `prepareData()` function transforms the training data into the required format for the internal method estimation code. By default, data is provided in long format in a `data.frame`. For most implementations, no transformation is therefore needed. Cluster methods for repeated-measures data typically require data to be transformed to `matrix` format, however.
2. The `compose()` function evaluates the method arguments and returns an updated `lcMethod` object with the evaluated method arguments. The function can also be used for modifying or even replacing the original `lcMethod` object for the remainder of the estimation process. This is useful when a method is a special case of a more general method and intends to conceal derivative or redundant arguments from the base class.
3. The `validate()` function enables evaluated method arguments to be checked against the input data. This can be used, for example, for checking whether the data contains the covariates specified in the method formula, or whether an argument has a valid value. For implementations which wrap an underlying package function, this validation is usually not needed as the underlying package already performs validation of the input.
4. The `preFit()` function is intended for processing any arguments prior to fitting. In order for these results to be persistent, they should be returned in an `environment` object, which will be passed as an input to the `fit()` function.
5. The `fit()` function is where the internal method is estimated for the given specification to obtain the cluster result. This function is also responsible for creating the corresponding `lcModel` object. The running time of this function is used to determine the method estimation time.
6. The `postFit()` function takes the outputted `lcModel` from `fit()` as input, enabling post-processing to be done. This is used, for example, for computing derivative statistics, or for reducing the memory footprint by stripping redundant data fields from the internal model representation. Preferably, this function is implemented such that it can be called repeatedly, allowing for updates to fitted methods without requiring re-estimation.

These functions are called by the `lcMethod` estimation functions (i.e., the functions prefixed by '*latrend*') in the order in which they are listed above. There are several advantages to this design. Firstly, the structure enables the method estimation process to be checked at each step. Secondly, splitting the estimation logic into processing steps encourages shorter functions with clearer functionality, resulting in more readable code. Thirdly, the steps enable optimizations in the case of repeated method estimation, for which the `prepareData()` function only needs to be called once. Lastly, in case of an update to the `lcModel` post-processing step, the `postFit()` function can be applied to previously obtained `lcModel` objects.

Table 5.2: The list of currently supported methods for clustering longitudinal data, in alphabetical order. The methods in the bottom row represent generic approaches which can be adapted.

Class	Method	Source
`lcMethodAkmedoids`	Anchored k-medoids	`akmedoids`
`lcMethodCrimCV`	Group-based trajectory modeling of count data	`crimCV`
`lcMethodDtwclust`	Dynamic time warping	`dtwclust`
`lcMethodFlexmix`	Interface to FlexMix framework	`flexmix`
`lcMethodFlexmixGBTM`	Group-based trajectory modeling	`flexmix`
`lcMethodFunFEM`	funFEM	`funFEM`
`lcMethodGCKM`	Feature-based clustering using growth curve modeling and k-means	`lme4`
`lcMethodKML`	longitudinal k-means	`kml`
`lcMethodLcmmGBTM`	Group-based trajectory modeling	`lcmm`
`lcMethodLcmmGMM`	Growth mixture modeling	`lcmm`
`lcMethodLMKM`	Feature-based clustering using linear regression and k-means	
`lcMethodMclustLLPA`	Longitudinal latent profile analysis	`mclust`
`lcMethodMixAK_GLMM`	Mixture of generalized linear mixed models	`mixAK`
`lcMethodMixtoolsGMM`	Growth mixture modeling	`mixtools`
`lcMethodMixtoolsNPRM`	Non-parametric repeated measures clustering	`mixtools`
`lcMethodMixTVEM`	Mixture of time-varying effects models	`R` script[2]
`lcMethodRandom`	Random partitioning	
`lcMethodStratify`	Stratification rule	
`lcMethodFeature`	Feature-based clustering	

5.3.1.1 Supported methods

An overview of the currently available methods that can be specified is given in Table 5.2. The `lcMethodGCKM` class implements a feature-based approach, based on representing the trajectories through a linear mixed model specified in the `lme4` package (Bates *et al.*, 2015). Additionally, a partitioning of trajectories can be specified without an estimation step through the `lcModelPartition` and `lcModelWeightedPartition` classes, providing trajectories with a cluster membership or membership weight, respectively.

5.3.1.2 Extended fitting

In general, cluster methods are more challenging to estimate for a greater number of clusters, which may result in the fit procedure failing to find a suitable solution. For example, the estimation algorithm may fail to converge. Even if a solution is obtained, it could comprise one or more clusters with only a few trajectories whereas a more efficient representation may exist. These issues can often be alleviated through the repeated estimation of the method with a different initialization each time. The best obtained result can then be used as the final output of the fit procedure.

Several special methods are defined that can be used to extend the fit procedure of the methods described above. These methods, prefixed by `lcFit`, alter the fit procedure of the

assigned underlying method. For example, the `lcFitConverged()` function defines the fit procedure of the underlying method to be repeated until a converged result is obtained. The `lcFitRepMin()` and `lcFitRepMax()` functions adapt the fit procedure to estimate the underlying method a fixed number of times and then return the best result according to the given metric.

5.3.2 The lcModel class

The `lcModel` class represents the estimated cluster solution. It is designed to function as any other model fitted in `R` (e.g., `lm()` from the `stats` package). Users can apply the familiar functions from the `stats` package where applicable, including the `predict()`, `fitted()`, and `residuals()` functions. Furthermore, `lcModel` objects support functions for obtaining the cluster representation, such as the cluster proportions, sizes, names, and trajectories.

The base `lcModel` class facilitates basic functionality such as providing a solution summary and providing functionality for computing predictions or fitted values. The two most important functions that characterize the class are the `predict()` and `postprob()` functions. These functions are used to derive the cluster trajectories, the posterior probabilities of the trajectories, and cluster proportions.

The base class stores information regarding the model, including the estimated `lcMethod` object, the `call` that was used to estimate the method, the date and time when the method was estimated, the total estimation time, and a text label for differentiating solutions. Users should not update the slots of the base class directly, except for the `tag` slot, which is intended as a convenient way of assigning custom meta data to the `lcModel`.

The names of subclasses are prefixed by '*lcModel*'. Subclasses generally have little need for adding new slots, as most of the functionality resides inside the class functions, such that results and statistics are computed dynamically. This enables fitted `lcModel` objects to be modified retroactively, e.g., for correcting implementation errors that are discovered at a later stage.

In the subclasses that are based on an underlying package implementation, the `lcModel` class serves as a wrapper around the underlying package solution representation. Any abstraction layer inevitably limits some of the capabilities of the underlying packages. Therefore, the internal representation is therefore exposed to the user via the `getModel()` function. This enables users to still benefit from the specialized functionality provided by the underlying package. We encourage users to check the documentation of the original packages to identify which additional functionality is available for a specific method.

5.3.3 The metric interfaces

There is a vast number of metrics available in literature. To provide access to as many metrics as possible, and to enable users to add missing metrics as needed, we define an interface for the computation of metrics. Users can replace or extend the metrics with custom implementations. To ensure a consistent output across all metrics, the output of metric functions must be scalar. Currently, the framework supports any of the applicable metrics from the packages `clusterCrit` (Desgraupes, 2018) and `mclustcomp` (You, 2018). The list of supported internal and external metrics is obtained via the `getInternalMetricNames()` and `getExternalMetricNames()` functions, respectively. Metrics can be added or updated

via the `defineInternalMetric()` and `defineExternalMetric()` functions.

5.4 Using the package

We illustrate the main capabilities of the package through a step-by-step demonstration of an exploratory longitudinal cluster analysis on the `PAP.adh` dataset that is included with the package. The demonstration involves the estimation and comparison of several methods for longitudinal clustering. The goal of the analysis is to identify the common patterns of adherence and to establish the most suitable method for the data out of those considered. For brevity, the description of the package function arguments used in the demonstration below is limited to the main arguments. We refer users to the package documentation to learn more about other optional arguments.

The `PAP.adh` synthetic dataset comprises PAP therapy adherence data of 301 patients in their first 91 days of therapy. The patients were synthesized from one of three clusters, closely resembling the real-life clusters identified by Yi *et al.* (2022). For each patient, the average hours of PAP therapy usage is recorded for each week of therapy. The data is represented by a `data.frame` in long format, with each row representing the observation of a patient at a specific week (1 to 13).

```
library("latrend")
data("PAP.adh")
head(PAP.adh)
```

```
##   Patient Week UsageHours    Group
## 1       1    1   6.298703 Adherers
## 2       1    2   5.916080 Adherers
## 3       1    3   5.022241 Adherers
## 4       1    4   5.788624 Adherers
## 5       1    5   4.758154 Adherers
## 6       1    6   4.222821 Adherers
```

The `Patient` column indicates the trajectory to which the observation belongs. The `UsageHours` column represents the averaged hours of usage in the respective therapy week, denoted by the `Week` column. The true cluster membership per trajectory is indicated by the `Group` column.

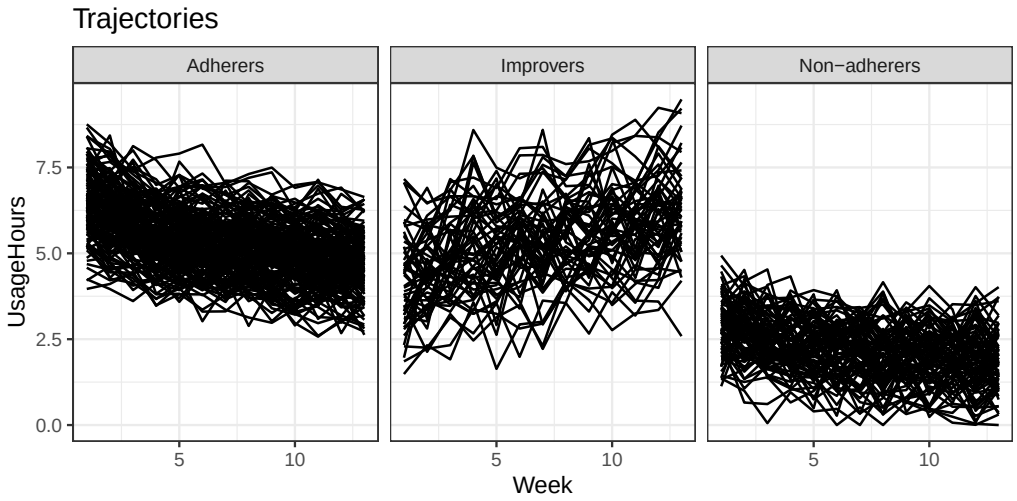
Throughout the analysis, there are several occasions during which the trajectory identifier and time columns would need to be specified. Instead of passing the column names to each function, we can set the default index columns using the `options` mechanism. Keep in mind that this is only recommended during interactive use.

```
options(latrend.id = "Patient", latrend.time = "Week")
```

We can visualize the patient trajectories using the `plotTrajectories()` function, shown in Figure 5.2. As the ground truth is known in our synthetic example, we specified the cluster membership of the trajectories via the `cluster` argument, resulting in a stratified visualization.

```
plotTrajectories(PAP.adh, response = "UsageHours", cluster = "Group")
```

Figure 5.2: The trajectories from the `PAP.adh` dataset, by reference group.



5.4.1 Specifying methods

We first specify the methods to be evaluated. The first method of interest in this case study is KML, selected for its flexibility in identifying patterns of any shape. The KML method is available in the framework through the `lcMethodKML` class, which serves as a wrapper around the `kml()` function of the `kml` package (Genolini *et al.*, 2015). The KML method is specified through the `lcMethodKML()` constructor function.

```
kmlMethod <- lcMethodKML(response = "UsageHours", nClusters = 2)
kmlMethod
```

```
## lcMethodKML specifying "longitudinal k-means (KML)"
##  time:            getOption("latrend.time")
##  id:              getOption("latrend.id")
##  nClusters:       2
##  nbRedrawing:     20
##  maxIt:           200
##  imputationMethod:"copyMean"
##  distanceName:    "euclidean"
##  power:           2
##  distance:        function() {}
##  centerMethod:    meanNA
##  startingCond:    "nearlyAll"
##  nbCriterion:     1000
##  scale:           TRUE
##  response:        "UsageHours"
```

Note that any unspecified arguments have been set to the default values defined by the `kml` package. The method arguments can be accessed using the or `[[` operator. Requested arguments are evaluated unless disabled by the argument `eval = FALSE`. As can be seen in

the method output below, the time index column is obtained from the `options` mechanism by default.

```
kmlMethod$time
```

```
## [1] "Week"
```

```
kmlMethod[["time", eval = FALSE]]
```

```
## getOption("latrend.time")
```

Next, we specify the other methods of interest. We use a variety of approaches that are applicable to this type of data. We evaluate a feature-based approach based on LMKM as implemented in `lcMethodLMKM`, a distance-based dynamic time warping approach via `lcMethodDtwclust` based on the `dtwclust` package, and the model-based approaches via the `lcMethodLcmmGBTM` and `lcMethodLcmmGMM` methods based on the `lcmm` package (Proust-Lima *et al.*, 2017). For LMKM, we model the trajectories using B-splines. We specify the distance-based approach using dynamic time warping. Lastly, GBTM and GMM are specified with an intercept-slope model for the cluster trajectories, and a shared diagonal variance-covariance matrix. The GMM defines a random patient intercept. Note that for methods supporting `formula` input, the response variable is automatically determined from the response of the formula.

```
library("splines")
lmkmMethod <- lcMethodLMKM(formula = UsageHours ~ bs(Week))
dtwMethod <- lcMethodDtwclust(response = "UsageHours",
  distance = "dtw_basic")
gbtmMethod <- lcMethodLcmmGBTM(fixed = UsageHours ~ Week,
  mixture = ~ Week, idiag = TRUE)
gmmMethod <- lcMethodLcmmGMM(fixed = UsageHours ~ Week,
  mixture = ~ Week, random = ~ 1, idiag = TRUE)
```

The method arguments of a `lcMethod` object cannot be modified. Instead, a new specification is created from the existing one with the updated method arguments. Any `lcMethod` object can be used as a prototype for creating a new specification with new, modified, or removed arguments using the `update()` function. As an example, if we would like to respecify KML to identify three clusters, this can be done by updating the existing specification as follows:

```
kml3Method <- update(kmlMethod, nClusters = 3)
```

As the number of clusters is generally not known in advance, we need to fit the methods for a range of number of clusters. Generating specifications for a series of argument values can be done via the `lcMethods()` function, which outputs a `list` of updated `lcMethod` objects from a given prototype. We specify each method for up to six clusters, limited by the computational runtime, using:

```
kmlMethods  <- lcMethods(kmlMethod,  nClusters = 1:6)
lmkmMethods <- lcMethods(lmkmMethod, nClusters = 1:6)
dtwMethods  <- lcMethods(dtwMethod,  nClusters = 2:6)
gbtmMethods <- lcMethods(gbtmMethod, nClusters = 1:4)
gmmMethods  <- lcMethods(gmmMethod,  nClusters = 1:4)
length(gmmMethods)
```

```
## [1] 4
```

5.4.2 Fitting methods

Using the previously created method specifications, we can estimate the methods for the `PAP.adh` data. For estimating a single method, we can use the `latrend()` function. The function optionally accepts an `environment` through the `envir` argument for evaluating the method arguments within a specific environment. The output of the function is the fitted `lcModel` object.

```
lmkm2 <- latrend(lmkmMethod, data = PAP.adh)
summary(lmkm2)
```

```
## Longitudinal cluster model using lmkm
## lcMethodLMKM specifying "lm-kmeans"
##  time:           "Week"
##  id:             "Patient"
##  nClusters:      2
##  center:         `meanNA`
##  standardize:    `scale`
##  method:         "qr"
##  model:          TRUE
##  y:              FALSE
##  qr:             TRUE
##  singular.ok:    TRUE
##  contrasts:      NULL
##  iter.max:       10
##  nstart:         1
##  algorithm:      `c("Hartigan-Wong", "Lloyd", "Forgy", "M
##  formula:        UsageHours ~ bs(Week)
##
## Cluster sizes (K=2):
##           A           B
##  69 (22.9%) 232 (77.1%)
##
## Number of obs: 3913, strata (Patient): 301
##
## Scaled residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.0213 -0.7493  0.2314  0.0000  0.7260  2.4355
```

Instead of needing to update a method prior to calling `latrend()`, the arguments to be updated can also be passed directly to `latrend()`. Here, we estimate the LMKM method for three clusters.

```
lmkm3 <- latrend(lmkmMethod, nClusters = 3, data = PAP.adh)
```

Alternatively, we can achieve the same result by updating the previously estimated two-cluster solution.

```
lmkm3 <- update(lmkm2, nClusters = 3)
```

5.4.2.1 Batch estimation

The `latrendBatch()` function estimates a list of method specifications. This is useful for evaluating a method for a range of number of clusters, as we have defined above using the `lcMethods()` function. Another use case is the improvement of model convergence and the estimation time by tuning the control parameters. Optimizing such parameters may yield considerably improved convergence or considerably reduced estimation time on larger datasets. Many of the methods have settings for the number of random starts, maximum number of iterations, and convergence criteria. However, because such control settings are specific to each method, we will not cover this.

The inputs to the `latrendBatch()` function are a list of `lcMethod` objects, and a list of datasets. The output is an `lcModels` object, representing a list of the fitted `lcModel` objects for each dataset. A seed is specified to ensure reproducibility of the examples.

```
lmkmList <- latrendBatch(lmkmMethods, data = PAP.adh, seed = 1)
lmkmList
```

```
## List of 6 lcModels with
##   .name .method       seed nClusters
## 1     1    lmkm  762473831         1
## 2     2    lmkm 1762587819         2
## 3     3    lmkm 1463113723         3
## 4     4    lmkm 1531473323         4
## 5     5    lmkm 1922000657         5
## 6     6    lmkm 1985277999         6
```

When printing a `lcModels` object, the content is shown as a table of method specifications. By default, only arguments which differ between the models are shown. The table can also be obtained as a `data.frame` by calling `as.data.frame()`. We now fit the other methods in the same manner.

```
dtwList <- latrendBatch(dtwMethods, data = PAP.adh, seed = 1)
```

For the repeated estimation of more computationally intensive methods, we can speed up the process by using parallel computation. By setting `parallel = TRUE`, the `latrendBatch()` function will use the parallel back-end of the `foreach` package (Microsoft and Weston, 2022). To make use of this functionality, we first need to configure the parallel back-end:

```
nCores <- parallel::detectCores(logical = FALSE)
if (.Platform$OS.type == "windows") {
  doParallel::registerDoParallel(parallel::makeCluster(nCores))
} else {
  doMC::registerDoMC(nCores)
}
```

The methods can then be estimated in parallel using:

```
kmlList  <- latrendBatch(kmlMethods,
  data = PAP.adh, parallel = TRUE, seed = 1)
```

```
gbtmList <- latrendBatch(gbtmMethods,
  data = PAP.adh, parallel = TRUE, seed = 1)
gmmList  <- latrendBatch(gmmMethods,
  data = PAP.adh, parallel = TRUE, seed = 1)
```

5.4.3 Evaluation

5.4.3.1 Assessing a cluster result

A cluster result is useful only when it describes the data adequately. There are various aspects on which the cluster result can be evaluated, depending on the method and analysis domain:

- The identified solution may not be reliable when the method estimation procedure did not converge. Convergence can be checked via the `converged()` function.
- The cluster solution may comprise empty clusters or clusters with a negligible proportion of trajectories. In such case, re-estimating the method may yield a better solution. Alternatively, one should consider fitting the method with a lower number of clusters.
- The cluster trajectories may be assessed visually to determine whether the identified patterns are sufficiently distinct.
- The prediction error may help to determine to which degree trajectories are represented by one of the clusters.

As shown in the previous section, the summary of an `lcModel` object shows the method arguments values, cluster sizes, cluster proportions, cluster names, and the standardized residuals. By default, the residuals are computed from the difference between the reference values and the predictions outputted by `fitted()`, conditional on the most likely trajectory assignments. For methods that do not provide trajectory-specific predictions, the fitted values are determined from the cluster trajectories.

The cluster trajectories can be obtained using the `clusterTrajectories()` function, returning a `data.frame`. The cluster trajectories can be plotted via `plot()` or `plotClusterTrajectories()`. The three-cluster LMKM solution is visualized in Figure 5.3. For parametric cluster methods, a more concise representation of the model can be obtained from the model coefficients, using `coef()`.

```
plot(lmkm3, size = 1)
```

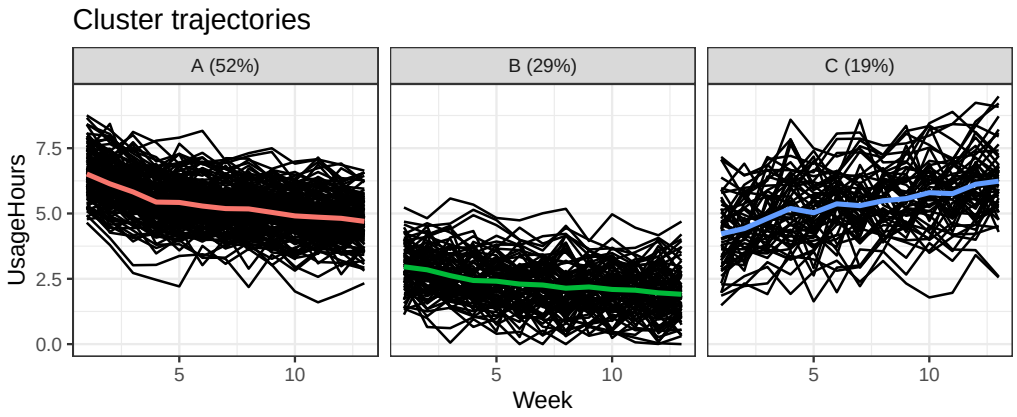
Assigning descriptive names to the clusters can help to increase the readability of the cluster result, which is especially useful for solutions with many clusters. The `clusterNames()` function can be used to retrieve or change the cluster names.

```
clusterNames(lmkm3) <- c("Struggling", "Increasing", "Decreasing")
```

The most likely cluster for each of the trajectories is obtained using the `trajectoryAssignments()` function, which outputs a `factor` with the cluster names as its levels. For soft-cluster representations, the cluster assignments are determined by the cluster with the highest probability, based on the posterior probability matrix. An alternative approach can be specified through the `strategy` argument. For example, the `which.weight()` function assigns a random cluster weighted by the proportions. The `which.is.max()` function from the `nnet` package[3] returns the most likely cluster,

[3] https://CRAN.R-project.org/package=nnet

Figure 5.3: The cluster trajectories of the three-cluster solution identified by LMKM, created using `plot(lmkm2)`.



breaking ties at random.

The posterior probability matrix can be obtained from the `postprob()` function. For probabilistic methods, it can be used to gauge the cluster separation, i.e., the certainty of assignment. The posterior probability is also important in the post-hoc analysis for accounting for the uncertainty in cluster assignment.

When it comes to longitudinal representation, the minimum functionality that is available for all `lcModel` objects is the prediction of the cluster trajectories at the given time points. The prediction has been implemented for packages which lack this functionality. For non-parametric methods such as KML or LLPA, linear interpolation is used when time points are requested which are not represented by the cluster centers.The available functionality differs between methods.

All `lcModel` objects support the standard `R` model functions `fitted()`, `residuals()`, and `predict()`. These functions are primarily of interest for methods that have a notion of a group or individual trajectory prediction error, such as for the model-based approaches like GBTM and GMM. The `fitted()` function returns the expected values for the response variable for the data on which the model was estimated. By default, only the values for the most likely cluster are given. However, for `clusters = NULL`, a matrix of predictions is outputted, where each column represents the predictions of the respective cluster.

The `predict()` function computes trajectory- and cluster-specific predictions for the given input data.

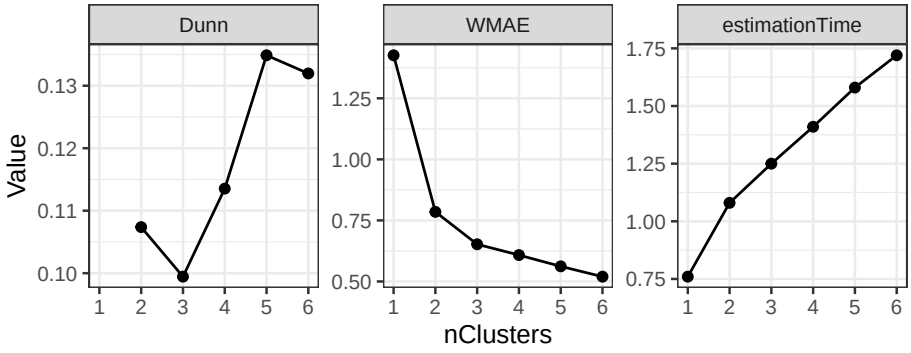
```
predict(lmkm3,
  newdata = data.frame(Week = c(1, 10), Cluster = "Decreasing"))
```

```
##        Fit
## 1 4.222478
## 2 5.797659
```

```
predict(lmkm2, newdata = data.frame(Biweek = c(1, 10),
  Cluster = "Decreasing"))
```

```
##        Fit
```

Figure 5.4: The Dunn index (higher is better), and WMAE (lower is better) metrics for each of the KML solutions from 1 to 6 clusters




```
## 1 5.468127
## 2 4.605169
```

The `predictPostprob()` and `predictAssignments()` functions compute the posterior probability and cluster membership for new trajectories, respectively. As this is not a common use case for cluster methods, most of the underlying packages do not provide this functionality. For demonstration purposes, we have implemented the functionality for the `lcModelKML` class.

Using the metric interface defined in Section 5.3.3, we can compute a variety of internal metrics through the `metric()` function:

```
metric(lmkm3, c("MAE", "RMSE", "Dunn", "ASW"))
```

```
##         MAE        RMSE        Dunn         ASW
## 0.76616319 0.97932759 0.07708328 0.33597008
```

With a model-based regression approach, another aspect that is worthwhile to assess are the residuals of the predicted values. This can be investigated, for example, through a visual inspection using a quantile-quantile (Q-Q) plot, available via the `qqPlot()` function, to assess whether the prediction errors approximately follow a normal distribution.

5.4.3.2 Identifying the number of clusters

Using one or more internal metrics of interest, we can assess how the data representation of a method improves or worsens for an increasing number of clusters. In this case study, we will use the Dunn index as the primary metric for the choice of the number of clusters. The change in metrics for an increasing number of clusters can be visualized via the `plotMetric()` function, and can help to determine the preferred solution. For brevity, we will only provide a detailed view for the KML method. We plot the Dunn index, WMAE, and estimation time (in seconds) for the six KML solutions as follows:

```
plotMetric(kmlList, c("Dunn", "WMAE", "estimationTime"))
```

The resulting plot is shown in Figure 5.4. The Dunn index and WMAE show a rather convincing improvement for an increasing number of clusters[4].

[4]The Dunn index is not defined for a one-cluster solution.

Moreover, we observe that the estimation time increases with the number of clusters. This can be a practical consideration when deciding on the preferred method to use. For much larger datasets, it may be useful to conduct a preliminary analysis on a subset of the data for possibly ruling out methods which are too computationally intensive in relation to the results.

We can obtain the metric values for each of the models by calling the `metric()` function.

```
metric(kmlList, c("Dunn", "WMAE", "estimationTime"))
```

```
##          Dunn      WMAE estimationTime
## 1          NA 1.4261264           0.86
## 2 0.10737225 0.7850566           1.25
## 3 0.09944419 0.6523208           1.47
## 4 0.11353357 0.6081128           1.69
## 5 0.13487175 0.5619086           1.88
## 6 0.13196444 0.5197172           2.05
```

As the preferred solution corresponds to the highest Dunn index, we can obtain the respective model by calling the `max()` function on the `lcModels` list object.

```
kmlBest <- max(kmlList, "Dunn")
```

Alternatively, we can select the preferred model using the `subset()` function. By specifying the `drop = TRUE`, the `lcModel` object is returned instead of a `lcModels` object.

```
kmlBest <- subset(kmlList, nClusters == 5, drop = TRUE)
```

The identification of the number of clusters is a form of model selection. The same approach can therefore be used for identifying the best cluster representation, e.g., evaluating different formulas for a parametric model, or selecting a different method initialization strategy.

5.4.3.3 Comparing methods

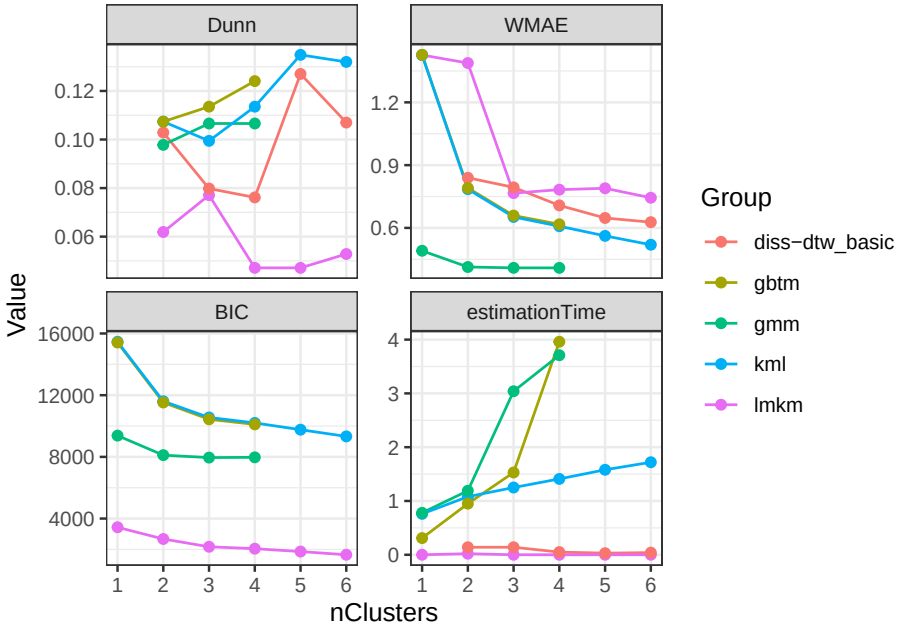
The optimal number of clusters according to the internal metric can be different for other methods or specifications thereof. Depending on the cluster representation, some methods may require fewer or more clusters to represent the heterogeneity to the same degree. By concatenating the lists of fitted methods, we can create a metric plot that is grouped by the type of method as follows:

```
allList <- lcModels(lmkmList, kmlList, dtwList, gbtmList, gmmList)
plotMetric(allList, name = c("Dunn", "WMAE", "BIC", "estimationTime"),
  group = '.method')
```

The WMAE and BIC between GBTM and KML are almost the same, possibly indicating that the methods find a similar solution. If the solutions are found to be practically identical, then one could actually prefer KML due to its considerably favorable computational scaling with the number of clusters.

We explore the best solution of each method further to better understand how the cluster representations differ between the methods. We can select the preferred `lcModel` object corresponding to the selected number of clusters for each of the methods using the `subset()` function.

Figure 5.5: The Dunn index (higher is better), WMAE (lower is better) and BIC (lower is better) for each of the methods and number of clusters




```
kmlBest  <- subset(kmlList,  nClusters == 5, drop = TRUE)
dtwBest  <- subset(dtwList,  nClusters == 5, drop = TRUE)
gbtmBest <- subset(lmkmList, nClusters == 4, drop = TRUE)
lmkmBest <- subset(lmkmList, nClusters == 3, drop = TRUE)
gmmBest  <- subset(gmmList,  nClusters == 3, drop = TRUE)
```

We can then assess the pairwise ARI, described in Section 5.2.6, between each method using the `externalMetric()` function. Calling this function on a `lcModels` list returns a `dist` object representing a distance matrix. We therefore create a list of the best `lcModel` for each method, by which we can then determine the pairwise ARI as follows:

```
bestList <- lcModels(KmL = kmlBest, DTW = dtwBest,
  LMKM = lmkmBest, GBTM = gbtmBest, GMM = gmmBest)
externalMetric(bestList, name = "adjustedRand") |> signif(2)
```

```
##       KmL  DTW LMKM GBTM
## DTW  0.39
## LMKM 0.45 0.38
## GBTM 0.34 0.24 0.61
## GMM  0.49 0.40 0.92 0.60
```

With all pairwise ARI being at least 0.24, all methods demonstrate some degree of similarity between each other. In particular, the very high ARI of approximately 0.92 between GMM and LMKM implies that the methods grouped the trajectories in a highly similar way.

Secondly, we compare the similarity of the cluster trajectories between the methods

using the WMMAE described in Section 5.2.6. The easiest way to compare methods is to compare the cluster trajectories visually. However, this approach is only practical on smaller datasets or solutions with few clusters. As a more scalable alternative, we can use external metrics to measure the pairwise similarity between the cluster trajectories of the methods.

```
externalMetric(bestList, name = "WMMAE") |> signif(2)
```

```
##        KmL   DTW  LMKM  GBTM
## DTW  0.100
## LMKM 0.130 0.130
## GBTM 0.110 0.110 0.029
## GMM  0.130 0.130 0.038 0.036
```

The mean absolute error of 0.029 between the cluster trajectories of GBTM and LMKM is negligible compared to the residual error estimated by GBTM (SD = 1.0), which indicates that both methods have identified practically the same cluster trajectories. The same applies to GMM and LMKM.

5.4.4 Cluster validation

Assessing the stability and reproducibility of a cluster method can help to determine whether the identified cluster solution generalizes beyond the data that was used to estimate the method. This is especially relevant for more complex cluster methods involving a large number of parameters, which may not generalize well to new data. This primarily pertains to the number of clusters the method is estimated for, as the number of parameters increases linearly with the number of clusters. Even relatively simple methods can overfit the data when the representation comprises too many clusters in relation to the sample size.

5.4.4.1 Cluster stability

Many of the estimation algorithms may identify a different solution during each run, depending on the starting values for the model parameters. Hence, it is important to run the estimation repeatedly to identify the best solution. This also helps to assess the stability of the model estimation. Repeated estimation can be done via the `latrendRep()` function, where the number of repetitions is specified via the `.rep` argument. Similar to `latrend()`, the method arguments can be updated within the function. The function returns a `lcModels` object, comprising a `list` of `lcModel` objects.

```
kmlRepList <- latrendRep(kmlMethod, data = PAP.adh,
  nClusters = 5, .rep = 5, .parallel = TRUE)
summary(metric(kmlRepList, c("Dunn", "WMAE")))
```

```
##       Dunn              WMAE
##  Min.   :0.1047   Min.   :0.5599
##  1st Qu.:0.1349   1st Qu.:0.5610
##  Median :0.1349   Median :0.5617
##  Mean   :0.1288   Mean   :0.5618
##  3rd Qu.:0.1349   3rd Qu.:0.5619
##  Max.   :0.1349   Max.   :0.5647
```

Which suggests that the solutions found by KML for the given number of clusters has a small degree of variability, which should be considered during the evaluation of the preferred number of clusters.

5.4.4.2 Comparison to ground truth

We now consider the case where a method is evaluated in a simulation study. In such a study, the ground truth is known, and we can directly evaluate the accuracy of the predicted trajectory cluster memberships of the estimated method. As we have shown in the previous subsection, the ARI can even be used when the number of clusters differs.

We can obtain the vector of trajectory cluster membership of the `PAP.adh` from the `Group` column by selecting the first cluster name of each trajectory, since the cluster membership is stable over time. We then create a `lcModelPartition` from the computed membership vector. In the case where the ground truth contains uncertainty on the cluster membership, the `lcModelWeightedPartition` class could be used. By default, the `lcModelPartition` generates the cluster representations from the means of the trajectories assigned to the respective cluster.

```
refAssignments <- aggregate(Group ~ Patient, data = PAP.adh,
  FUN = head, n = 1L)
refAssignments$Cluster = refAssignments$Group

refModel <- lcModelPartition(data = PAP.adh,
  trajectoryAssignments = refAssignments, response = "UsageHours")
refModel
```

```
## Longitudinal cluster model using part
## lcMethod specifying "undefined"
## no arguments
##
## Cluster sizes (K=3):
##     Adherers    Improvers Non-adherers
##  162 (53.8%)   56 (18.6%)   83 (27.6%)
##
## Number of obs: 3913, strata (Patient): 301
##
## Scaled residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.894748 -0.643670 -0.009533  0.000000  0.634893  3.590377
```

We can now compare our selected method solutions to the reference solution using the ARI.

```
externalMetric(bestList, refModel, name = "adjustedRand", drop = FALSE)
```

```
##      adjustedRand
## KmL     0.4756201
## DTW     0.3925777
## LMKM    0.8996308
## GBTM    0.5800669
## GMM     0.9775104
```

This shows that, for this synthetic dataset, GMM achieved the best result out of the methods considered, with an ARI of 0.98. This result is expected, as Yi *et al.* (2022) used a GMM to identify the clusters from which the `PAP.adh` dataset is simulated. GBTM and especially LMKM achieved a good recovery as well. It is quite likely that GBTM, KML and DTW would have obtained an improved ARI for a greater number of clusters than what was evaluated.

5.4.4.3 Internal validation

A cluster method can also be validated internally, where the model is trained on subsets of the data. The framework includes two resampling techniques for this purpose: bootstrap sampling and cross validation. Such a validation approach can also be used for a more robust type of model selection (Lord *et al.*, 2017).

Bootstrap sampling Bootstrap sampling, also referred to as bootstrapping, involves the repeated estimation of a model on random subsets of the data. These data subsets are obtained by sampling trajectories with replacement. Instead of obtaining a single optimal cluster solution for a dataset, each random subset will have slightly different optimal solutions. This variability between samples can provide an indication of the stability of the cluster model on the overall data (Hennig, 2007).

The `latrendBoot()` function applies bootstrapping to the given method specification. The `samples` argument determines the number of times the data is resampled, and a model is estimated. Setting the `seed` argument ensures that the same sequence of bootstrap samples is generated when redoing the bootstrapping procedure. The output is a `lcModels` list containing the model for each sample. The estimated methods each have a different `call` for the `data` argument such that the original bootstrap training sample can be recreated as needed. This avoids the need for models to store the training data. As an example, we compute 20 bootstrap samples[5] (i.e., repeated fits) in parallel as follows:

```
kmlMethodBest <- update(kmlMethod, nClusters = 5)
kmlBootModels <- latrendBoot(kmlMethodBest, data = PAP.adh,
  samples = 10, seed = 1, parallel = TRUE)
head(kmlBootModels, n = 3)
```

```
## List of 3 lcModels with
##   .name .method                                          data
## 1     1     kml  bootSample(PAP.adh, "Patient", 762473831L)
## 2     2     kml bootSample(PAP.adh, "Patient", 1762587819L)
## 3     3     kml bootSample(PAP.adh, "Patient", 1463113723L)
##         seed
## 1 1062140483
## 2  185557490
## 3  934902099
```

We can now assess the stability of the solutions across the models in terms of metrics of interest. Here, we assess the mean convergence rate, and the quantiles of the WMAE and Dunn metrics.

[5] In practice, a much greater number of bootstrap samples is recommended (at least 100).

```
bootMetrics <- metric(kmlBootModels, c("converged", "Dunn", "WMAE"))
mean(bootMetrics$converged)
```

```
## [1] 1
```

```
summary(bootMetrics$Dunn)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1351  0.1477  0.1506  0.1570  0.1688  0.1852
```

```
summary(bootMetrics$WMAE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5289  0.5490  0.5553  0.5534  0.5587  0.5736
```

As can be seen from the output, there is quite some variability between the estimated solutions across bootstrap samples. This suggests that we should consider estimation with repeated random starts to identify a better and more stable solution.

Lastly, we can compute a similarity matrix for an external metric of interest, containing the pairwise similarity for each model pair.

```
wmmaeDist <- externalMetric(kmlBootModels[1:10], name = "WMMAE")
summary(wmmaeDist)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01392 0.06029 0.07294 0.06841 0.07934 0.11060
```

Showing that there is only a small degree of discrepancy in the cluster trajectories between bootstrap samples.

Cross validation Cross validation (CV) is used to obtain a nearly unbiased estimate of the predictive power of a model, which is useful when the dataset is too small for an independent validation set. This is done by drawing k independent patient folds from the dataset. Each of the k folds are used for testing the model which was trained on the other $k-1$ folds. The `latrendCV()` function applies k-fold CV for a method specification. The `folds` argument determines the number of folds. Setting the `seed` ensures that the same folds are generated across runs. The function output a `lcModels` list with k `lcModel` objects. Like the estimated bootstrap models, the outputted models have a modified data call that returns the respective fold data.

```
kmlFoldModels <- latrendCV(kmlMethodBest, data = PAP.adh,
  folds = 10, seed = 1, parallel = TRUE)
```

The function only estimates the models on the $k-1$ folds. Users should evaluate the model on the appropriate test data as they see fit. The test data for a model can be obtained using:

```
testData1 <- createTestDataFold
  (PAP.adh, model.data(kmlFoldModels[[1]]))
```

This validation scheme can only be used for models which have an implementation of `predict()`, `predictPostprob()` or `predictAssignments()` for new trajectory data. Currently, this is implemented for `lcModelKML` and `lcModelMclustLLPA`.
To compute the performance metric on a single fold, we can use the `predictAssignments()` function to obtain the predicted cluster assignments for the trajectories in the test data of the first fold.

```
predAssign <- predictAssignments(
  kmlFoldModels[[1]], newdata = testData1)
mclustcomp::mclustcomp(as.integer(predAssign),
  as.integer(testData1$Group), types = "adjrand")
```

```
##     types    scores
## 1 adjrand 0.6413792
```

Now that we have shown how to compute the performance on a single fold, we can generalize this to compute the ARI for each fold. We take the list of `lcModel` objects for each fold, and iterate over the folds to compute a `data.frame` with the predicted and reference trajectory assignments.

```
resultsList <- lapply(kmlFoldModels, function(model) {
    testData = createTestDataFold(PAP.adh, model.data(model))
    data.frame(Pred = predictAssignments(model, newdata = testData),
      Ref = testData$Group)
})
foldResultsTable <- data.table::rbindlist(resultsList, idcol = "Fold")
foldScores <- foldResultsTable[, mclustcomp::mclustcomp(
    as.integer(Pred), as.integer(Ref), "adjrand")$scores,
        keyby = Fold]$V1
mean(foldScores)
```

```
## [1] 0.5313103
```

```
sd(foldScores) / sqrt(length(foldScores)) # standard error
```

```
## [1] 0.03956681
```

We obtain a mean ARI of 0.53 ± 0.04, which is close to our performance estimate of 0.48 on the full dataset.

5.5 Implementing new methods

One of the main strengths of the framework is the standard way in which methods are specified, estimated, and evaluated. These aspects make it easy to compare newly implemented methods with existing ones. Using the base classes `lcMethod` and `lcModel`, new methods can be implemented with a relatively minimal amount of code, enabling rapid prototyping.

5.5.1 Stratification

The simplest form of clustering is the stratification of the dataset based on a known factor. This can be the response variable, or any other measure available for each trajectory. This is useful for case studies where there is prior knowledge or expert guidance on how the trajectories should be grouped: Either by another factor (e.g., age or gender), or a characteristic of the trajectory (e.g., the intercept, slope, average, or variance).

A stratification approach can be specified using the `lcMethodStratify()` function, which takes an `R` expression as input. The expression is evaluated within the `data.frame` at the trajectory level during the method estimation, so any column present in the data

can be used. The expression should resolve to a number or category, indicating the stratum for the respective trajectory.

As an example, we stratify the trajectories by thresholding on the mean hours of usage. This expression returns a `logical` value which determines the cluster assignment. For categorizing trajectories into more than two clusters, the `cut()` function can be used. The cluster trajectories are computed by aggregating the trajectories of each cluster at the respective time points. By default, the average is computed, but an alternative center function can be specified via the `center` argument.

```
stratMethod <- lcMethodStratify(response = "UsageHours",
  stratify = mean(UsageHours) > 4)
stratModel <- latrend(stratMethod, data = PAP.adh)
clusterProportions(stratModel)
```

```
##         A         B
## 0.3156146 0.6843854
```

5.5.2 Feature-based clustering

Feature-based clustering is a flexible and fast approach to clustering longitudinal data, with an essentially limitless choice of trajectory representations. The framework includes a generic feature-based clustering class named `lcMethodFeature` for quickly implementing this approach.

A `lcMethodFeature` specification requires two functions: A representation function outputting the trajectory representation `matrix`, and a cluster function that applies a cluster algorithm to the matrix, returning an `lcModel` object. To illustrate the method, we represent each trajectory using a linear model, and we cluster the model coefficients using k-means. In the representation step, `lm()` is applied to each trajectory, and the model coefficients are combined into a `matrix` with the trajectory-specific coefficients on each row. We parameterize the `lcMethod` implementation by enabling the user to define a `formula` argument. The representation function is as follows:

```
repStep <- function(method, data, verbose) {
  repTraj <- function(trajData) {
    lm.rep <- lm(method$formula, data = trajData)
    coef(lm.rep)
  }
  dt <- as.data.table(data)
  coefData <- dt[, as.list(repTraj(.SD)), keyby = c(method$id)]
  coefMat <- as.matrix(subset(coefData, select = -1))
  rownames(coefMat) <- coefData[[method$id]]
  coefMat
}
```

We implement the cluster step to return a `lcModelPartition` object based on the cluster assignments outputted by `kmeans()`. We have parameterized the function by obtaining the number of clusters for k-means from the `nClusters` model argument. The cluster function is as follows:

```
clusStep <- function(method, data, repMat, envir, verbose) {
  km <- kmeans(repMat, centers = method$nClusters)
  lcModelPartition(response = responseVariable(method),
  method = method, data = data,
  trajectoryAssignments = km$cluster, center = mean)
}
```

We can now specify and estimate the feature-based method, including the additionally required arguments. Comparing the estimated model to the preferred KML model, we see that the solutions have a relatively high degree of overlap.

```
tsMethod <- lcMethodFeature(response = "UsageHours",
  formula = UsageHours ~ Week, representationStep = repStep,
  clusterStep = clusStep)
tsModel <- latrend(tsMethod, data = PAP.adh, nClusters = 5)
externalMetric(tsModel, kmlBest, "adjustedRand")
```

```
## adjustedRand
##    0.4859513
```

```
externalMetric(tsModel, kmlBest, "WMMAE")
```

```
##     WMMAE
## 0.1083389
```

5.5.3 Implementing a method

We will describe the high-level steps that are involved in adding support for a method to the framework, so that users can extend or implement new methods to address their use case. Considering the number of lines of code for even a relatively simple cluster model, we do not cover a complete example here. Instead, we only outline the typical set of functions (`fit()`, `getArgumentDefaults()`, `getName()`, `getShortName()`) that need to be implemented, together with any relevant input and output assumptions of these functions. A step-by-step example of implementing a statistical method in the framework can be found in the vignette included with the package.

The implementation of a method requires a new `lcMethod` class to be created, which we will name `lcMethodExample` here. Usually, a `lcModel` class needs to be implemented for representing the representation of the fitted method, which we will name `lcModelExample` here. If the method estimation only outputs a partitioning, then the `lcModelPartition` class may be used instead.

5.5.3.1 Extending the method class

Defining a new method involves creating a subclass of the `lcMethod` class, defining its default arguments, its name, and any logic needed for the fitting procedure. We start by defining the `lcMethodExample` class.

```
setClass("lcMethodExample", contains = "lcMethod")
```

Any method can be specified by instantiating the respective class through the `new()` function. It is recommended to rely on the object initialization mechanism of the base `lcMethod` class for this, as it takes care of collecting all arguments and adding default

values for missing arguments. Defining new method arguments in custom class slots would hinder users from passing specialized or new optional arguments to the underlying estimation call.

Given that the base class handles the initialization of our `lcMethodExample` class, all we need to do is to define the default argument values in a named list. By adding `formals(stats::kmeans)` to the named list, our method will inherit all arguments from the `kmeans()` function.

```
setMethod("getArgumentDefaults", "lcMethodExample", function(object) {
  c(
    formals(stats::kmeans),
    time = quote(getOption("latrend.time")),
    id = quote(getOption("latrend.id")),
    nClusters = 2,
    callNextMethod()
  )
})
```

Method arguments can be of any class. However, we recommend that methods are specified using scalar arguments. This results in a more easily readable method summary, and greatly simplifies the permutation of argument options in a simulation study.

For identification purposes, it is recommended to specify a name and an abbreviated name for the method. This can be done by implementing the `getName()` and `getShortName()` functions, returning the names as `character`.

```
setMethod("getName", "lcMethodExample",
  function(object) "simple example method")

setMethod("getShortName","lcMethodExample",function(object) "example")
```

We can now specify our example method by instantiating an object through the `new()` function, providing optional arguments as additional inputs.

```
new("lcMethodExample", nClusters = 3)
```

```
## lcMethodExample specifying "simple example method"
##  iter.max:       10
##  nstart:         1
##  algorithm:      c("Hartigan-Wong", "Lloyd", "Forgy", "Ma
##  trace:          FALSE
##  time:           getOption("latrend.time")
##  id:             getOption("latrend.id")
##  nClusters:      3
```

Lastly, the relevant steps of the estimation process outlined in Section 5.3.1 need to be implemented. At the very least, we need to define a `fit()` function which uses the `lcMethodExample` object passed via the `method` argument and the data to estimate the specified model. The function returns a new `lcModelExample` object based on the internal model.

```
setMethod("fit", "lcMethodExample",
  function(method, data, envir, verbose, ...) {
```

```
    fittedRepresentation <- CODE_HERE
    new("lcModelExample", data = data, model = fittedRepresentation,
      method = method,
      clusterNames = make.clusterNames(method$nClusters)
)})
```

In case an external estimation function should be called with the defined method arguments, one can apply `as.list()` to the `lcMethod` object to obtain a named list of argument values. The external function can then be called using `do.call()`.

Checking for missing arguments and for the correct argument type or valid values avoids late and confusing errors during the estimation process. It is therefore recommended to implement a validation mechanism of the method specification. This can be done by assigning a validation function to the class via `setValidity()` as part of the `S4` system, or by implementing `validate()`. The latter function allows for easier validation as all arguments are already evaluated, and the arguments can be validated against the input data.

5.5.3.2 Extending the model class

We begin by defining the `lcModelExample` class. One can consider adding slots for representing, for example, the representational coefficients.

```
setClass("lcModelExample", contains = "lcModel")
```

The `postprob()` function is used to determine the cluster assignments and cluster proportions, so every `lcModel` subclass should provide it. In case of hard-cluster models, the posterior probability consists of zeros and ones.

```
setMethod("postprob", "lcModelExample", function(object) {
  ppMatrix <- CODE_HERE
  colnames(ppMatrix) <- clusterNames(object)
  return (ppMatrix)
})
```

The `predict.lcModel()` function is relatively complex due to the different types of inputs and outputs it supports. As these cases generalize across methods, the `lcModel` class provides a suitable standard implementation. For implementing new `lcModel` classes, it is therefore advisable to implement the `predictForCluster()` function instead of `predict()`, as it is called by `predict.lcModel()`. This function should provide a prediction for each row of the `data.frame` of the `newdata` argument, conditional on the given cluster membership.

```
setMethod("predictForCluster", "lcModelExample",
  function(object, newdata, cluster, ...) {
    predData <- CODE_HERE
    return (predData)
})
```

Lastly, implementing the `predictPostprob()` function enables the model to predict the posterior probability for new data. The output should be a matrix matching the number of rows of `newdata` and indicating the cluster-specific probabilities in the respective columns.

```
setMethod("predictPostprob", "lcModelExample",
  function(object, newdata, ...) {
    ppMat <- CODE_HERE
    colnames(ppMat) <- clusterNames(object)
    return (ppMat)
})
```

It is also possible to override the `predictAssignments()` function. However, the default function already uses the output of `predictPostprob()`, so overriding it is only of use for implementing a more extensive or method-specific classification strategy.

5.6 Summary and outlook

The `latrend` package facilitates the standardized yet flexible exploration of heterogeneity in longitudinal datasets, with a minimal amount of coding effort. The framework provides functionality for specifying, estimating, and assessing models for clustering longitudinal data. The package builds upon the efforts of the `R` community by providing an interface to the many methods for clustering longitudinal data across packages. Perhaps most importantly, the `latrend` package makes it easy to compare between any two cluster methods, enabling users to identify the most suitable method to their use case. To ensure transparent and reproducible research, all decisions and settings that are relevant to the analysis should be reported. A useful checklist for reporting on latent-class trajectory studies is provided by Van de Schoot *et al.* (2017), which is also relevant to longitudinal cluster analyses in general.

Users can implement new methods within the framework or add support for other packages, enabling rapid prototyping for the case study at hand. Additionally, the standard functionality provided by the framework also reduces the effort needed in implementing a longitudinal cluster model.

We encourage the framework to be used as a first exploratory step in clustering longitudinal data, after which the identified preferred method can then be applied directly from the original package, which typically provides special tools or options not provided by the framework. To illustrate one such limitation, consider the initialization or prior specification of a longitudinal cluster model. This is generally an important aspect of model estimation that can improve the identified model solution but is challenging to facilitate in a standardized way.

Although the package allows for the automatic comparison and selection across methods through various metrics, it is advisable to assess whether the identified cluster solution is meaningful. It is a useful practice to consider domain knowledge when evaluating the solution (Nagin, 2005), both in the choice of metrics as well as the interpretation of the clusters. For example, in some applications, the change over time is more of interest than the mean level, and vice versa. Along similar lines, a solution comprising a very small cluster (i.e., with few subjects) provides little additional descriptive power of the heterogeneity, unless the presence of outliers is of significant interest.

The framework is currently focused on the modeling of a single continuous response variable, whereas some of the supported cluster packages already support multivariate trajectory modeling. The possible support for multivariate trajectories has been accounted for in the design of the software. Similarly, while the single response is required to

be numerical, support could be added for categorical outcomes such as those used in longitudinal latent class analysis. These features are planned for a future version.

Overall, we intend the framework to bridge the different approaches to clustering longitudinal data that exist from the various areas of research. We encourage users and package developers to create interfaces for their methods, as the availability of a standard framework for performing a longitudinal cluster analysis lowers the barrier to evaluating and comparing methods for applied researchers.

Computational details

The examples and figures in this paper were obtained using `R` 4.2.2 (R Core Team, 2022) with the packages `latrend` 1.5.1, `ggplot2` 3.4.0 (Wickham, 2016), and `data.table` 1.14.6 (Dowle and Srinivasan, 2020). The KML method was estimated based on the `kml` 2.4.1 package. The distance-based method utilized the `dtwclust` 5.5.11 package. The GBTM analysis was performed based on the `lcmm` 2.0.0 package, with the parallel computation achieved using the `foreach` 1.5.2 package.

`R` and all packages used within the article and the `latrend` package are available from the Comprehensive `R` Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

The development of this framework builds upon the work of the `R` community. The authors would like to express their appreciation for the numerous longitudinal cluster packages that have been developed, as these packages have made `R` a versatile platform for clustering longitudinal data. Moreover, we gratefully incorporated many of the cluster metrics by using the packages `clusterCrit` (Desgraupes, 2018) and `mclustcomp` (You, 2018).

Chapter 6

Latent-class trajectory modeling with a heterogeneous mean-variance relation

N.G.P. Den Teuling, F. Ungolo, S.C. Pauws, E.R. van den Heuvel
Submitted.

Abstract
This work investigates the benefit of addressing heteroskedastic residual variances across trajectories with the purpose of finding clusters of longitudinal trajectories. We propose models that account for class-specific heteroskedasticity through a mean-variance relation or random residual variance, thereby accounting for trajectory-specific variance. The latent-class trajectory models we evaluate are an extension of growth mixture models (GMM). We assess the estimation bias of the model parameters and the recoverability of the number of latent classes under various data-generating models and settings by means of a simulation study. Furthermore, we show the empirical applicability of these models by analyzing the time-varying number of COVID-19 cases across counties in the United States. Overall, the class-specific mean-variance could be reliably estimated by the proposed models in datasets comprising 250 trajectories. In addition, we have found that the extended GMM accounting for the residual random variance had an improved group trajectory estimation over the standard GMM.

6.1 Introduction

Longitudinal data collection provides opportunities for assessing change over time within a population, and for exploring the differences between and within subjects. Such heterogeneous data can be analyzed using a linear mixed modeling (LMM) (Hartley and Rao, 1967; Laird and Ware, 1982) approach. However, if the heterogeneity is structured or too complex to model parametrically, a more flexible approach may be desirable. Growth mixture models (GMM) (Muthén, 2004) are a common approach for describing population heterogeneity in terms of a finite number of latent classes, each representing a group trajectory from which subjects may deviate within the classes (Bauer, 2007). This model can be represented as a mixture of LMMs, each representing a latent class. Whereas the expected value of trajectories is modeled in detail, simpler assumptions are typically made about the residual error, assuming homoskedasticity and homogeneity.

Ignoring the variance structure can potentially lead to an inadequate estimation of the mean, as well as wrong conclusions about the estimated model (Carroll, 2003). In particular, if the residual error scale differs structurally between subjects but is unaccounted for, it can result in biased estimates of the trajectory coefficients and variance components (Enders and Tofighi, 2008) and affect model selection accuracy (Diallo *et al.*, 2016). In contrast, by accounting for residual heteroskedasticity, one may obtain additional latent classes of interest, as a new dimension of heterogeneity is considered (Foulley and Quaas, 1995; De Kort *et al.*, 2017). Finally, modeling the variance structure yields more reliable prediction intervals for the mean trajectories (Davidian and Giltinan, 1993).

A source of such heteroskedasticity is the presence of a mean-variance relation, as seen in data with observations spanning orders of magnitude. For instance, this can happen when analyzing count data due to its lower bound of zero. Examples include the mean number of weekly alcoholic drinks in a study on alcoholic dependence (Zhu *et al.*, 2017), and the abundance of species or the count of the number of observed species in ecological research (Tsou, 2011). If the heteroskedasticity due to the mean-variance relationship is not accounted for, the estimate of the mean could be affected (Foulley, 2004).

Modeling approaches for residual heteroskedasticity are underrepresented in research, especially those regarding GMMs. Researchers tend to focus on identifying latent classes with respect to changes in the mean level of the response variable over time, while the possible dynamics of within-subject variance are overlooked. Few papers have investigated the joint modeling of the mean and variance in a multilevel mixture model such as GMM. For instance, De Kort *et al.* (2017) investigated heteroskedastic multilevel mixture models with a smooth or step function for the variance. They found that ignoring heteroskedasticity under nonlinearity resulted in a biased estimation of the regression relation. Moreover, it is key to evaluate different functions, as the residual variance relation is sensitive to the specified heteroskedastic function. Diallo *et al.* (2017) assessed the performance of GMM with a time-varying covariate under time-dependent variance. They concluded that such heteroskedasticity only had a minimal effect on the model selection of GMM.

In this work, we explore the application of GMM under two forms of variance. First, we investigate the GMM with class-specific mean-variance relations as an exploratory tool for when the degree to which the variance depends on the mean is suspected to vary within the population. Accounting for a mean-variance relation is necessary since variability may depend on the level of the measurement, which is common in the life sciences. Then, we explore the application of GMM accounting for random residual variance (that is,

subject-specific deviations from the class-specific variance). The random residual variance is used to deal with heteroskedasticity that is unrelated to the measurement level, but related to individual characteristics that are not necessarily measured. It is not always clear why certain people show very stable trajectories, and others show a greater variability over time.

We investigate the estimation of the group trajectories, mean-variance relations, and random residual variance. In addition, we assess the impact of the estimation when ignoring the mean-variance relation or random residual variance present in the dataset. The proposed models are estimated by means of a fully Bayesian analysis. The posterior distribution is estimated by using Hamiltonian Monte Carlo sampling (Neal, 2011). We show that the specification of these models can be done in a straightforward way.

We illustrate the models to the analysis of the number of confirmed COVID-19 cases across counties in the United States of America over time. The geographic, demographic, cultural and regulatory differences between counties are likely to contribute to this heterogeneity. In addition, the variability in the daily cases depends on the number of contagious people and may therefore affect the variability around growth trajectories in confirmed cases over time. We expect that the mean and variance are related in some form, and we will evaluate different mean-variance coefficients between latent classes.

This chapter is organized as follows. We describe the growth mixture models in Section 6.2. Section 6.4 describes the simulation study and the results thereof. The case study analysis is described in Section 6.5. Section 6.6 concludes.

6.2 Models

For a given set of independent subjects I, let $\boldsymbol{y}_i = (y_{i,1}, \cdots, y_{i,n_i})$ denote the trajectory of subject $i \in I$ with n_i measurements over time and $y_{i,j} \in \mathbb{R}$. Individuals may have a different number of observations over time, and the time at which a measurement $y_{i,j}$ was taken, denoted by $t_{i,j}$, may differ between trajectories. Absent observations are assumed to be missing at random.

We identify two sources of variability; the between-subject variability and the within-subject variability, which can be handled using LMM (Laird and Ware, 1982). We approximate the heterogeneity between subjects using a mixture of LMMs (Verbeke and Lesaffre, 1996; McLachlan and Peel, 2000). Here, each of the class models represents a different data-generating process for the trajectories. We assume that each trajectory originates from one of the classes and does not change class membership over time. Given that a subject i belongs to class k, the trajectory $\boldsymbol{y}_{k,i}$ follows the class-specific LMM given by

$$\begin{aligned} \boldsymbol{y}_{k,i} &= \alpha_k + \boldsymbol{X}_i\boldsymbol{\beta}_k + \boldsymbol{Z}_i\boldsymbol{b}_{k,i} + \boldsymbol{\varepsilon}_{k,i} \\ \boldsymbol{b}_{k,i} &\sim \text{MVN}(0, \boldsymbol{\Sigma}_k) \\ \boldsymbol{\varepsilon}_{k,i} &\sim N(0, \sigma^2_{\varepsilon,k}). \end{aligned} \tag{6.1}$$

Here, $\boldsymbol{X}_i \in \mathbb{R}^{n_i \times p}$ and $\boldsymbol{Z}_i \in \mathbb{R}^{n_i \times q}$ denote the known design matrices for the fixed and random effects, respectively, comprising an intercept, covariates dependent on $t_{i,j}$, and possibly other covariates. The class-specific fixed-effects coefficients are denoted by $\boldsymbol{\beta}_k$. The random effects $\boldsymbol{b}_{k,i}$ are typically assumed to be normally distributed with zero mean and a class-specific $q \times q$ variance-covariance matrix $\boldsymbol{\Sigma}_k$. For simplicity, we assume the random effects to be independent, representing $\boldsymbol{\Sigma}_k$ as a diagonal matrix with diagonal

$\boldsymbol{\sigma}_{b,k} = \{\sigma_{1,k}, \ldots, \sigma_{q,k}\}$. Lastly, the residual errors $\boldsymbol{\varepsilon}_{k,i}$ are assumed to be independent from the random effects, and normally distributed with $\boldsymbol{\varepsilon}_{i,k} \sim N(0, \sigma^2_{\varepsilon,k})$. In other applications, the residual variance component is represented by a variance-covariance matrix $\boldsymbol{\Lambda}_k$ imposing, for example, a correlation structure such as an autoregressive model. More elaborate correlation structures for the residuals in the presence of random effects may lead to non-identifiability issues (Regis *et al.*, 2019).

The mixture of class models are weighted according to the probability $\Pr(C_i = k) = \pi_k$ of a subject i belonging to class k, where $\boldsymbol{\pi} = \{\pi_1, ..., \pi_K\}$ with $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$. For a trajectory with unknown class assignment or for the marginal distribution, the trajectory is modeled as the weighted sum of models with

$$\boldsymbol{y}_i \sim \sum_{k=1}^{K} \pi_k \text{MVN}(\boldsymbol{X}_i\boldsymbol{\beta}_k, \boldsymbol{Z}_i\boldsymbol{\Sigma}_k\boldsymbol{Z}_i^T + \boldsymbol{\Lambda}_k). \tag{6.2}$$

It should be noted that the class-specific residual variance $\sigma^2_{\varepsilon,k}$ in the GMM of Equation 6.1 already accounts for some degree of heteroskedasticity.

6.2.1 GMM with mean-variance relation

In heteroskedastic models, the residual variance of subjects is modeled using a variance function. This function typically describes the variance in terms of one or more explanatory variables or the subject mean. In case the variance function is not known, model selection could be used to evaluate several candidates (Ledwina and Mielniczuk, 2010). As an alternative, a non-parametric or semi-parametric approach to describing the variance function has been shown to perform well (Carroll, 1982; Fan and Yao, 1998; Liitiäinen *et al.*, 2010).

We focus on modeling a mean-variance relation as this comprises a parsimonious model for residual heteroskedasticity, suitable in an exploratory setting. Accounting for class-specific heteroskedasticity including explanatory variables would introduce considerable challenges, both in terms of model specification and the number of parameters to be estimated. We regard this work as a precursor to including other variance functions in an exploratory mixture modeling setting (Tong and Wang, 2005; Ledwina and Mielniczuk, 2010).

To the best of our knowledge, no other researchers have explored a heterogeneous mean-variance relation through a GMM. Here, we consider the function $g(\mu_{i,j}, \boldsymbol{\phi})$, with the mean of the trajectory described by $\mu_{i,j} = X_{i,j}\boldsymbol{\beta}_k + Z_{i,j}\boldsymbol{b}_{k,i}$, $\boldsymbol{\phi}$ a vector of unknown parameters, and g a known function. Nevertheless, the literature on mean-variance modeling in mixed regression analysis provides useful insights that likely translate to a mixture setting. Sugasawa and Kubokawa (2017) explored the modeling of an unknown mean-variance relation through arbitrary variance functions.

The mean-variance function $g(\mu_{i,j}, \boldsymbol{\phi})$ is usually modeled under the assumption of a positive mean, where $\mu_{i,j} \in \mathbb{R}_+$. This allows for a direct linear relation to the variance, e.g., assuming $g(\mu_{i,j}, \boldsymbol{\phi}) = \phi_0 + \phi_1\mu_{i,j}$ (Davidian and Giltinan, 1993) or a power relation $g(\mu_{i,j}, \boldsymbol{\phi}) = \phi_0\mu_{i,j}^{\phi_1}$.

In this work, we consider a more generally applicable mean-variance relation allowing for $\mu_{i,j} \in \mathbb{R}$. Specifically, we model the mean-variance relation as a multiplicative effect on the base variance, assuming the power relation $\log \sigma_{\varepsilon,k,i,j} = \gamma_{0,k} + \mu_{k,i,j}\gamma_{1,k}$. Here,

$\gamma_{0,k}$ represents the baseline class variance, and $\gamma_{1,k}$ is the base for the class-specific mean-variance power relation. The advantage of this relation is that it allows for negative values of $\mu_{k,i,j}$, and only requires the estimation of one additional coefficient over a homoskedastic model. In addition, $\exp[2\gamma_{1,k}]$ has the intuitive interpretation as the amount of proportional change of $\sigma^2_{\varepsilon,k,i,j}$ per unit change of $\mu_{k,i,j}$.

The resulting model, which we shall refer to as MV-GMM, is given by

$$\begin{aligned} \boldsymbol{y}_{k,i} &= \boldsymbol{\mu}_{k,i} + \boldsymbol{\varepsilon}_{k,i} \\ \boldsymbol{\mu}_{k,i} &= \alpha_k + \boldsymbol{X}_i\boldsymbol{\beta}_k + \boldsymbol{Z}_i\boldsymbol{b}_{k,i} \\ \log \boldsymbol{\sigma}_{\varepsilon,k,i} &= \gamma_{0,k} + \boldsymbol{\mu}_{k,i}\gamma_{1,k} \\ \boldsymbol{b}_{k,i} &\sim \text{MVN}(0, \boldsymbol{\Sigma}_k) \\ \boldsymbol{\varepsilon}_{k,i}|\boldsymbol{b}_{k,i} &\sim N(0, \boldsymbol{\sigma}^2_{\varepsilon,k,i}), \end{aligned} \tag{6.3}$$

where $\boldsymbol{\mu}_{k,i}$ represents the true mean individual trajectory with $E\left(\boldsymbol{y}_{k,i}|\boldsymbol{b}_{k,i}\right) = \boldsymbol{\mu}_{k,i}$. By modeling a class-varying mean-variance relation, the model accounts for a correlation between the longitudinal profile (i.e., latent class) and the strength of the mean-variance association.

A key aspect of estimating the mean-variance relation is the accuracy of the estimate of $\boldsymbol{\mu}_{k,i}$. We therefore propose to specify random effects for all fixed effects, i.e., $\boldsymbol{Z}_i = \boldsymbol{X}_i$, ensuring a better estimation of the trajectory-specific mean $\boldsymbol{\mu}_{k,i}$. This considerably increases the dimensionality of the model compared to models with fewer random effects in the mean. Nevertheless, the simulation study in Section 6.4 shows that the estimation strategy of this work can handle such high-dimensionality.

6.2.2 GMM with random residual variance

In growth mixture modeling, a large emphasis is placed on representing the subject heterogeneity on the expected response. This is justified by the argument of ergodicity, that is, the expectation or observation that no two subjects are the same, and therefore no two subjects follow exactly the same trajectory. For the same reason, it can be argued that the heterogeneity in the variance across subjects should also be considered (Hamaker, 2012). We therefore consider a GMM with a random residual variance effect that may be independent or partially dependent with respect to the random effect used in the mean response. The random effect included in the variance structure accounts for subject-specific systematic errors (Carroll, 2003). The joint estimation of the mean and variance structures requires more data due to the larger number of degrees of freedom. Fortunately, with the increase in the typical sample size of longitudinal datasets, these models can be estimated reliably. As such, random residual variance models are becoming more commonplace (Hamaker *et al.*, 2018).

We focus on modeling the residual variance per subject assuming that the subject variance is time-invariant. Each subject is assumed to have their own magnitude of variability, regarded as a systematic deviation from the group-level variability. The deviation is included as an unconstrained coefficient $\omega_{k,i}$ in the log-linear model for the subject-specific variance, and is assumed to be normally distributed. Therefore, the random residual variance scaling factor $\exp[\omega_{k,i}]$ follows a lognormal distribution. This log-linear model for residual variance has also been used by (Hedeker *et al.*, 2012) in a

single-class setting. The model, which we shall refer to as RV-GMM, is outlined in relation to the GMM of equation (6.1):

$$\begin{aligned} \log \boldsymbol{\sigma}_{\varepsilon,k,i} &= \gamma_{0,k} + \omega_{k,i} \\ (\omega_{k,i}, \boldsymbol{b}_{k,i}) &\sim \text{MVN}(0, \boldsymbol{\Sigma}_k) \\ \boldsymbol{\varepsilon}_{k,i} | \omega_{k,i}, \boldsymbol{b}_{k,i} &\sim N(0, \boldsymbol{\sigma}^2_{\varepsilon,k,i}). \end{aligned} \tag{6.4}$$

Here, $\boldsymbol{\Sigma}_k$ denotes the covariance structure of the scale of the random residual variance and random effects for the mean. The estimation of this model turns out to be computationally more demanding due to the addition of $|I|$ random variables representing the trajectory-specific variances per latent class.

6.2.3 GMM with mean-variance relation and random residual variance

The reasoning provided for the specification of the RV-GMM from the argument of ergodicity applies equally to the MV-GMM. Therefore, we combine the previously described models into one that accounts for both the heterogeneous residual variance and a mean-variance relation within groups. We refer to this model as random-mean-variance GMM (RMV-GMM). The model for the residual variance is given by

$$\log \boldsymbol{\sigma}_{\varepsilon,k,i} = \gamma_{0,k} + \boldsymbol{\mu}_{k,i}\gamma_{1,k} + \omega_{k,i} \tag{6.5}$$

where the variance is modeled as a multiplicative effect of the baseline coefficient γ_0, the estimated mean, and the random residual variance.

6.3 Estimation

The fully Bayesian analysis of this work focuses on the estimation of the posterior distribution of the parameter vector Θ, conditional on the observable data $\mathbf{y}$ and $\mathbf{X}$, denoted as $\Pr(\boldsymbol{\Theta}|\boldsymbol{y}, \boldsymbol{X})$. Since $\Pr(\boldsymbol{\Theta}|\boldsymbol{y}, \boldsymbol{X})$ cannot be obtained in closed form, we estimate it by using samples from this distribution. Due to the complexity of the models hereby presented, we use Hamiltonian Monte Carlo (HMC, (Neal, 2011)), implemented using Stan[1]. It uses the No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014), which automatically tunes the HMC.

HMC requires the computation of the gradient of the log-posterior. Stan addresses this aspect by providing reverse-mode automatic differentiation. The use of HMC has several benefits over more traditional sampling algorithms such as Gibbs sampling and the Metropolis-Hasting (MH) algorithms (Monnahan *et al.*, 2016). Indeed, the algorithm has a greater sampling efficiency, requiring fewer iterations to converge. The HMC sampler allows for a more efficient exploration of the posterior distribution, which is a critical aspect when dealing with high-dimensional models with correlated parameters.

[1]Stan is a freely available open-source program that can estimate complex statistical models, such as hierarchical models, on large datasets. It is available at https://mc-stan.org.

6.3.1 Model inference

As Stan does not support discrete parameters, the discrete latent class assignments are marginalized out of the model. An advantage of this approach is that the tails of the multimodal distributions are better explored, as the assignment to each class is considered in every iteration. In order to speed up the posterior sampling, we avoid the integration of the random effects on location and scale by treating the random effects as nuisance parameters. Here, we make use of the capability of HMC to sample efficiently in high-dimensional space. An added benefit of specifying the model in this way is that the different GMMs are straightforward to implement in Stan.

We denote the set of all parameters and random variables for the sth draw from the posterior distribution by $\boldsymbol{\Theta}^{(s)} = \{\boldsymbol{\pi}, \boldsymbol{b}, \boldsymbol{\omega}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$. Here, $\boldsymbol{\pi}$ is the vector of class proportions, $\boldsymbol{\theta}_k^{(s)} = \{\alpha_k, \boldsymbol{\beta}_k, \gamma_{0,k}, \gamma_{1,k}, \boldsymbol{\Sigma}_k\}$ are the class-specific parameters, and $\boldsymbol{b}$ and $\boldsymbol{\omega}$ represent the random effects of the trajectories and residuals. For brevity, we omit the sample number indicator when denoting the model parameters. The conditional likelihood for any given draw from the posterior distribution $\Pr(\boldsymbol{\Theta}|\boldsymbol{y}, \boldsymbol{X}) \propto \Pr(\boldsymbol{\Theta}) \Pr(\boldsymbol{y}|\boldsymbol{\Theta}, \boldsymbol{X})$ is computed by

$$\begin{aligned} L_c^{(s)} &= \Pr(\boldsymbol{y}|\boldsymbol{\Theta}, \boldsymbol{X}) \\ &= \prod_{i \in I} \sum_{k=1}^{K} \Pr(C_i = k) \Pr(\boldsymbol{y}_i | C_i = k, \boldsymbol{\Theta}, \boldsymbol{X}) \\ &= \prod_{i \in I} \sum_{k=1}^{K} [\pi_k f(\boldsymbol{y}_i | \boldsymbol{\theta}_k, \boldsymbol{b}_{i,k}, \omega_{i,k}, \boldsymbol{X})] \\ &= \prod_{i \in I} \sum_{k=1}^{K} \left[\pi_k \prod_{j=1}^{N_i} f(y_{i,j} | \boldsymbol{\theta}_k, \boldsymbol{b}_{i,k}, \omega_{i,k}, \boldsymbol{X}) \right], \end{aligned} \tag{6.6}$$

where C_i denotes the class membership of trajectory i. The posterior probability $\hat{\pi}_{i,k}$ of a trajectory i belonging to class k at any given draw is computed by normalizing the class likelihood over all classes, with

$$\begin{aligned} \hat{\pi}_{i,k} &= \Pr\left(C_i = k | \boldsymbol{y}_i, \boldsymbol{\Theta}, \boldsymbol{X}_i\right) \\ &= \frac{\Pr\left(\boldsymbol{y}_i | C_i = k, \boldsymbol{\Theta}_{k,i}, \boldsymbol{X}_i\right) \Pr(C_i = k)}{\sum_{k'=1}^{K} \Pr\left(\boldsymbol{y}_i | C_i = k', \boldsymbol{\Theta}_{k'i}, \boldsymbol{X}_i\right) \Pr(C_i = k')}. \end{aligned} \tag{6.7}$$

The posterior classification of trajectory i corresponds to the class with the highest associated posterior probability, given by $\hat{c}_i = \arg\max_k \hat{\pi}_{i,k}$.

We center the covariates vector X and the response variable y to improve sampling efficiency, following the guide provided by Stan developers (Stan Development Team, 2020b). More precisely, we center the response variable around zero to $\boldsymbol{y}_i' = \boldsymbol{y}_i - \bar{\boldsymbol{y}}$ (here, $\bar{\boldsymbol{y}}$ denotes the mean response across all trajectories) and do the same with the p-dimensional vector of covariates. Then, we decorrelate the centered covariates using a thin QR decomposition. In this way, the design matrix is decomposed into an orthogonal $N \times p$ matrix $\boldsymbol{Q}'$ with uncorrelated columns, and an upper-triangular $p \times p$ matrix $\boldsymbol{R}'$ such that $\boldsymbol{X}' = \boldsymbol{Q}' \cdot \boldsymbol{R}'$. We normalize the matrices by scaling to $\boldsymbol{Q} = \boldsymbol{Q}' \sqrt{N-1}$ and $\boldsymbol{R} = \boldsymbol{R}' / \sqrt{N-1}$ (Stan Development Team, 2020b). The expected value for the centered

response, with $\boldsymbol{Z}_i = \boldsymbol{X}_i$, is given by

$$\begin{aligned}\mathbb{E}(\boldsymbol{y}'_{i,k}) &= \tilde{\alpha}_k + \boldsymbol{X}'_i(\boldsymbol{\beta}_k + \boldsymbol{b}_{k,i}) \\ &= \tilde{\alpha}_k + \boldsymbol{Q}_i\boldsymbol{R}(\boldsymbol{\beta}_k + \boldsymbol{b}_{k,i}) \\ &= \tilde{\alpha}_k + \boldsymbol{Q}_i\left(\tilde{\boldsymbol{\beta}}_k + \tilde{\boldsymbol{b}}_{k,i}\right),\end{aligned} \tag{6.8}$$

with adjusted intercept $\tilde{\alpha}_k$, fixed effects coefficients $\tilde{\boldsymbol{\beta}}_k = \boldsymbol{R}\boldsymbol{\beta}_k$, and $\tilde{\boldsymbol{b}}_{k,i} \sim \text{MVN}\left(0, \tilde{\boldsymbol{\Sigma}}_k\right)$. Note that for $K = 1$, we have $\tilde{\alpha}_1 \simeq 0$. We restrict the random effects to be independently distributed with respect to the decomposed scale for simplicity, that is, $\tilde{\boldsymbol{\Sigma}}_k = \tilde{\boldsymbol{\sigma}}_k^2 I$, with $\tilde{\boldsymbol{\sigma}}_k^2$ denoting the scales of the random effects. The linear transformation from the original model does not affect the likelihood of the model, nor the residual scale. The original intercept α_k and fixed effects coefficients $\boldsymbol{\beta}_k$ are recovered via $\alpha_k = \tilde{\alpha}_k - \boldsymbol{Q}\tilde{\boldsymbol{\beta}}_k + \bar{\boldsymbol{y}}$ and $\boldsymbol{\beta}_k = \boldsymbol{R}^{-1} \cdot \tilde{\boldsymbol{\beta}}_k$. The variance-covariance matrix on the original scale is obtained as $\boldsymbol{\Sigma}_k = \boldsymbol{R}^{-1}\tilde{\boldsymbol{\Sigma}}_k\left(\boldsymbol{R}^{-1}\right)^T$.

Label switching Without a defined ordering of classes, there are $K!$ possible class permutations yielding the same likelihood. This presents a challenge during sampling and model comparison, referred to as the label switching problem (Richardson and Green, 1997). We therefore constrain the class ordering by an increasing order on the class intercepts, with $\tilde{\alpha}_1 < \tilde{\alpha}_2 < \ldots < \tilde{\alpha}_K$ (Richardson and Green, 1997). This is achieved by mapping the ordered intercepts to an unconstrained vector $\boldsymbol{\delta}$ of log-increments (Carpenter *et al.*, 2017). Using the constrained class ordering, the mixing of classes across the MCMC iterations occurs less frequently. We use a model-agnostic approach, which consists of relabeling the classes based on the posterior probabilities $\hat{\pi}_{i,k}^{(s)}$ of the trajectories using the ECR algorithm[2] (Rodríguez and Walker, 2014; Papastamoulis, 2016).

Convergence Parameter convergence is assessed using the potential scale reduction factor (PSRF) (Gelman *et al.*, 1992), where a value below 1.1 is considered to be acceptable (Gelman *et al.*, 2013).

6.3.2 Prior specification

We specify weakly informative priors for the model parameters $\boldsymbol{\Theta}$, meaning that we set priors that are reasonable in a general exploratory setting. Specifically, a Dirichlet prior is used on the class proportions with $\boldsymbol{\pi} \sim \text{Dir}(a_1, \ldots, a_K)$. In general, it is recommended to use values of $a_k \geq 1$ to avoid empty classes (Frühwirth-Schnatter, 2006), so we specify $a_k = 1\ \forall k$ to allow for variable class proportions. We specify the class-specific priors to be equal across classes. The decomposed and centered intercepts $\tilde{\boldsymbol{\alpha}}$ and fixed-effects coefficients $\tilde{\boldsymbol{\beta}}_k$ are assumed to follow a standard normal distribution. A generalized half-$t(3, 0, 1)$ prior is used for the scale of the shared independent random effects $\sigma_{b,1}, \ldots, \sigma_{b,p}$ and σ_ω and for the residual error $\sigma_{\varepsilon,k}$ of GMM. The log-linear model intercept γ_0 for RV-GMM, MV-GMM and RMV-GMM is assumed to follow a standard normal distribution. In order to avoid overestimating the mean-variance association, we assume $\gamma_{1,k} \sim t(3, 0, 0.1)$.

[2]We apply ECR with iterative relabeling strategy 1.

6.3.3 Model selection

The number of mixture components K is learnt through a model selection procedure which identifies the model with the lowest marginal value of the Widely Applicable Information Criterion (WAIC) (Watanabe, 2009). The WAIC generalizes the use of the AIC to models with complex hierarchical layers as is the case for the models of this work. The WAIC is given by

$$\text{WAIC} = -\sum_{i \in I} \log \left[\frac{1}{S} \sum_{s=1}^{S} f\left(\boldsymbol{y}_i | \boldsymbol{\Theta}^{(s)}\right) \right] + p_{\text{WAIC}}, \tag{6.9}$$

$$p_{\text{WAIC}} = 2 \sum_{i \in I} \left[\log \left(\frac{1}{S} \sum_{s=1}^{S} f\left(\boldsymbol{y}_i | \boldsymbol{\Theta}^{(s)}\right) \right) - \frac{1}{S} \sum_{s=1}^{S} \log f\left(\boldsymbol{y}_i | \boldsymbol{\Theta}^{(s)}\right) \right], \tag{6.10}$$

$$f(\boldsymbol{y}_i | \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \int f(\boldsymbol{y}_i | \boldsymbol{\Theta}_k) f\left(\boldsymbol{b}_{k,i} | \boldsymbol{\Sigma}_k\right) d\boldsymbol{b}_{k,i}, \tag{6.11}$$

where $f\left(\boldsymbol{y}_i | \boldsymbol{\Theta}^{(s)}\right)$ denotes the density for subject i, and p_{WAIC} penalizes the model for the effective number of parameters (Gelman *et al.*, 2013).

The conditional likelihood we used for inference is a numerical trick to recover the parameters, as it is more efficient from a sampling point of view. However, for model selection, it is preferable to use the marginal likelihood (Tong *et al.*, 2022). The full marginal likelihood of subject i is given in Equation 6.11 for GMM and MV-GMM, requiring an integration over the random effects $\boldsymbol{b}_{k,i}$. For the marginal likelihood of RV-GMM and RMV-GMM, an additional integration over $\omega_{k,i}$ is needed.

Since the integral of Equation 6.11 cannot be solved in closed form for MV-GMM and RMV-GMM, we compute it by means of an approximation. For the sth MCMC sample $\boldsymbol{\Theta}^{(s)}$, we use draws of $\boldsymbol{b}_{k,i}$ from a distribution parameterized by the covariance of the random effects $\boldsymbol{\Sigma}_k^{(s)}$. The likelihood is computed conditional on the respective subject belonging to their most likely class during the sth sample, denoted by $\hat{c}_i^{(s)}$. The approximation is computed by

$$f\left(\boldsymbol{y}_i | k = \hat{c}_i^{(s)}; \boldsymbol{\Theta}^{(s)}\right) = \int f\left(\boldsymbol{y}_i | k = \hat{c}_i^{(s)}; \boldsymbol{\Theta}_k^{(s)}\right) f\left(\boldsymbol{b}_{k,i} | \boldsymbol{\Sigma}_k\right) d\boldsymbol{b}_{k,i} \tag{6.12}$$

$$\approx \frac{1}{M} \sum_{m=1}^{M} f\left(\boldsymbol{y}_i | \boldsymbol{b}_{k,i}^{(m)}; k = \hat{c}_i^{(s)}; \boldsymbol{\Theta}_k^{(s)}\right), \tag{6.13}$$

where M denotes the number of draws for $\boldsymbol{b}_{k,i}^{(m)} \sim f\left(\boldsymbol{b}_{k,i}^{(s)} | \boldsymbol{\Sigma}_k\right)$. The marginal likelihood of each subject can therefore be computed in constant time, independent from the number of classes estimated by the model.

6.3.4 Software

We conduct a simulation study and analyze a case study using R 3.6.1 (R Core Team, 2022), running on Intel Xeon E5-2660 processors. The code used is available in the

supplementary materials. The R package `rstan` (version 2.19.3) (Stan Development Team, 2020a) was used to interface with Stan.

The models were implemented in the longitudinal clustering framework of the `latrend` package (version 1.5.0) (Den Teuling, 2022). The `label.switching` package (version 1.8) created by Papastamoulis (2016) was used for the implementation of the ECR algorithm for dealing with the label switching problem described in Section 6.3.1.

6.4 Simulation study

We investigate the performance of our clustering approach in the presence of heteroskedasticity through a simulation study. We assess the ability of the MV-GMM and RMV-GMM models in identifying the heterogeneous mean-variance relations of the latent classes, assuming that the number of classes is known. Furthermore, we explore the effects of misspecification of the mixture model, where we consider type I and type II errors for heteroskedasticity arising from a mean-variance relation or latent variance. These aspects are evaluated by estimating GMM, MV-GMM, RV-GMM, and RMV-GMM on a set of simulated datasets under different data-generating processes.

In a second simulation study, we evaluate the identification of the number of classes for GMM, MV-GMM and RMV-GMM under different class separation conditions, the presence of a mean-variance relation, and the presence of random residual variance.

6.4.1 Settings

We evaluate the GMM, MV-GMM, RV-GMM and RMV-GMM models on datasets comprising 250 trajectories, each having ten observations with $\boldsymbol{t}_i$ drawn from a Unif$[0, 2]$ distribution. We consider scenarios involving two and three classes, where each trajectory is generated according to a particular class. The class proportions are given by $\pi_k \propto \sqrt{k}$, yielding classes of different sizes[3]. Each of the simulation scenarios is evaluated on 500 randomly generated datasets.

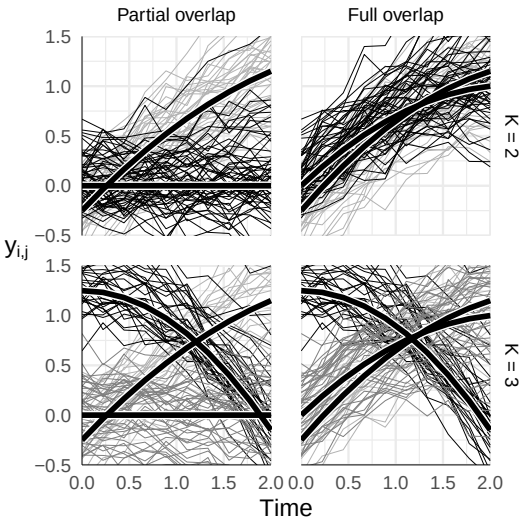
The group trajectories of the expected value for each class are generated from second-order polynomials. We distinguish between two degrees of class separation. In the first case, referred to as *Partial overlap* (PO), two of the group trajectories have a similar intercept, and diverge over time. This results in a significant overlap of the expected values of the trajectories of both classes around the intercept. In the second case, named *Full overlap* (FO), two group trajectories start at approximately the same intercept and have a similar time profile. This results in a large overlap between the two classes, and thus there is almost no class separation on the mean. In this scenario, the classes can only reliably be recovered through the heterogeneous mean-variance relation. The coefficients of the group trajectories under the different data scenarios are reported in Table 6.1. For simplicity, we specify independent random effects for the parameters in the time profiles with equal scale within each of the classes. We generate the trajectories with a moderate degree of within-class heterogeneity, with $\boldsymbol{\sigma}_b = \{.16, .071, .032\}$. The standard deviation for the Gaussian noise is also set to be equal across classes, with $\sigma_{\varepsilon,k} = .05$ for the homoskedastic scenarios, and $\sigma_{\gamma,0,k} = \log(.05)$ accordingly for the heteroskedastic

[3]In the case of two classes, 104 trajectories (41.6%) are assigned to class A, and 146 (58.4%) trajectories are assigned to class B. With three classes, 60 trajectories (24%) will be assigned to class A, 85 trajectories (34%) to class B, and 105 trajectories (42%) to class C.

Table 6.1: Group trajectory coefficients used for generating the datasets. Each row represents the coefficients for a different class.

Classes	Dataset	
	Partial overlap	Full overlap
2	$\boldsymbol{\beta} = \begin{bmatrix} -.25 & 1 & -.15 \\ 0 & 0 & 0 \end{bmatrix}$	$\boldsymbol{\beta} = \begin{bmatrix} -.25 & 1.1 & -.2 \\ 0 & .9 & -.2 \end{bmatrix}$
3	$\boldsymbol{\beta} = \begin{bmatrix} -.25 & 1 & -.15 \\ 0 & 0 & 0 \\ 1.25 & 0 & -.35 \end{bmatrix}$	$\boldsymbol{\beta} = \begin{bmatrix} -.25 & 1.1 & -.2 \\ 0 & .9 & -.2 \\ 1.25 & 0 & -.35 \end{bmatrix}$

Figure 6.1: Generated datasets comprising 100 trajectories, based on the four sets of group trajectories provided in Table 6.1. The thick lines denote the group trajectories.



scenarios. A visualization of the generated trajectories and group trajectories for the different data scenarios is shown in Figure 6.1.

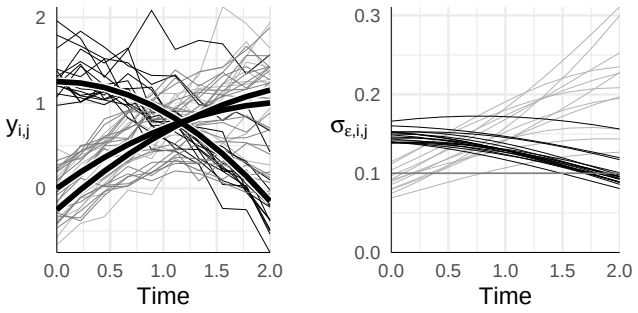
We evaluate the models on datasets with and without a heterogeneous mean-variance relation. In the scenarios with data showing a mean-variance relation, we use $\boldsymbol{\gamma}_1 = \{.6, 0\}$ and $\boldsymbol{\gamma}_1 = \{.6, .3, 0\}$ for data with two and three classes, respectively. An example of a dataset showing a class-varying mean-variance relation is shown in Figure 6.2. Finally, we consider the effect of the presence of high random residual variance, with $\sigma_\omega = .3$ and $\gamma_0 \equiv \log \sigma_\varepsilon$.

For the second simulation study on the identification of the number of classes, we simulate 200 datasets for each condition, where observations come from two classes. The models are estimated for $K = 1, 2, 3$ and the best number of classes as identified by the model selection procedure described in Section 6.3.3 is counted.

Figure 6.2: An example dataset with a heterogeneity in the mean-variance relation. Fifty trajectories were generated with high random scale, high noise, and low class-specific mean variance settings. Note that one of the classes has no mean-variance relation and is therefore depicted by a horizontal line in (b).

(a) The generated trajectories. The thick lines denote the group trajectories.

(b) The trajectory-specific standard deviation $\sigma_{\varepsilon,i,k}$, dependent on the mean.



6.4.2 Evaluation

We only consider converged models in the evaluation, that is, models for which the posterior samples of each parameter were deemed to have converged according to the PSRF criterion, described in Section 6.3. For each scenario, we evaluate the trajectory classification agreement using the adjusted Rand index (ARI) (Hubert and Arabie, 1985). A score of 1 indicates a perfect agreement, whereas a score of 0 indicates an agreement that is no better than random chance. A perfect recovery of the trajectory classes is not expected in the scenarios under consideration, due to the large overlap between classes. We investigated our approaches under extreme settings to check their robustness. Finally, we analyze the mean absolute error of the posterior mean with respect the true parameter value.

Convergence On rare occasions, the class intercept parameters were found to be strongly correlated during HMC sampling, resulting in poor parameter recovery. We therefore discard estimated models which exhibited highly correlated ($|\rho| > .95$) samples between the intercepts. We consider a sampled model to have converged when all model parameters converged, and the intercept parameters are not correlated. Across all scenarios, a converged result was achieved in 92% of all simulated datasets.

6.4.3 Results

6.4.3.1 Trajectory classification agreement

The average trajectory classification agreement in each of the scenarios is reported in Table 6.2. In the partial overlap scenarios, all models achieved a good recovery (ARI $>$ 0.80) of the trajectory class membership, regardless of the presence of heteroskedasticity. The recovery of MV-GMM, RV-GMM and RMV-GMM are on par or better than GMM, indicating that the models are not sensitive to the absence of the assumed mean-variance

Table 6.2: Trajectory classification agreement (adjusted Rand index) per model across the simulation scenarios. The RRV column indicates whether random residual variance was simulated.

Overlap	K	MV	RRV	Model			
				GMM	RV-GMM	MV-GMM	RMV-GMM
Partial	2	No	No	.86	.86	.86	.85
			Yes	.85	.85	.83	.84
		Yes	No	.83	.84	.87	.87
			Yes	.82	.84	.85	.86
	3	No	No	.92	.92	.92	.92
			Yes	.91	.92	.88	.92
		Yes	No	.91	.92	.93	.93
			Yes	.91	.91	.90	.92
Full	2	Yes	No	.035	.030	.30	.22
			Yes	.032	.029	.22	.10
	3	Yes	No	.55	.55	.66	.64
			Yes	.55	.54	.63	.57

relation or random residual variance. Under the presence of a mean-variance relation, MV-GMM and RMV-GMM outperformed the other models, suggesting that the methods can make use of the additional information about the trajectories. However, MV-GMM showed to be sensitive to the presence of random residual variance. This is apparent in the three-class scenario with residual random variance, where MV-GMM obtained an ARI of 0.88 compared to 0.91 by GMM. Overall, RMV-GMM outperformed or matched the performance of the other models in almost all partial overlap scenarios, even in the absence of random residual variance.

In the full overlap scenarios, MV-GMM and RMV-GMM consistently outperformed GMM and RV-GMM. The latter two models could not discern between the two classes for which the group trajectories fully overlap, as indicated by the ARI that is near zero. MV-GMM outperformed RMV-GMM in all four settings, even in the scenarios involving residual random variance, with a sizable difference in ARI of 0.12 and 0.06 for two and three classes, respectively.

6.4.3.2 Parameter estimation

The bias in the estimation of the intercept and slope are reported in Table 6.3. We first explore the partial overlap scenario without a mean-variance relation. RV-GMM consistently achieved the lowest bias across the scenarios. For two classes, all models achieved practically the same level of bias (± 0.005). In the three-class scenarios with random residual variance, MV-GMM exhibited an increased bias, especially in the intercept of the first class ($+0.08$).

In the scenario with partial overlap and the presence of a mean-variance relation, MV-GMM and RMV-GMM performed marginally better (± 0.005) than the other two

methods on the two-class datasets. In the three-class datasets, no method obtained a clearly better parameter recovery than another model. Across all scenarios, GMM showed a slightly higher bias on average.

The greater difficulty of the full overlap scenario is apparent from the relatively high magnitudes of biases for the intercept and slope of the first class compared to the previous scenarios. MV-GMM and RMV-GMM achieved a considerably lower bias in these scenarios (50% lower bias for most parameters), as the models benefit from the additional information from the mean-variance relation. Remarkably, MV-GMM outperformed RMV-GMM even under the presence of random residual variance. This is especially prominent in the three-class datasets, where the intercept bias differs by 0.2.

As shown in Table 6.4, the bias of the mean-variance relation for MV-GMM and RMV-GMM was within acceptable levels in nearly all scenarios. Notably, the absence of a mean-variance relation was correctly identified by both models (bias below 0.04). Here, RMV-GMM outperformed MV-GMM under the presence of random residual variance with a consistent bias below 0.015. Under the presence of a mean-variance relation, both models achieved a satisfactory recovery with a bias below 0.05 in nearly all cases. In the full overlap scenarios, the presence of random residual variance presents a challenge to both models, resulting an increased bias of 0.1. MV-GMM performed well under the absence of residual random variance, with bias below 0.06 across the parameters. RMV-GMM did not outperform MV-GMM under the presence of random residual variance. The recovery was especially poor in the three-class dataset for the first class (bias of -0.26).

The bias in the parameters of the random residual variance for RV-GMM and RMV-GMM is reported in Table 6.5. In the absence of a mean-variance relation in the data, both random heteroskedastic models only had a relatively small bias (0.05) under constant variance. Both models correctly identified the random residual variance scale when present, with a bias below 0.007. In the presence of a mean-variance relation, RV-GMM tends to adjust for the heteroskedasticity with a larger residual variance scale, resulting in a large bias above 0.20. In contrast, RMV-GMM correctly accounted for the mean-variance relation, resulting in approximately the same level of minimal bias as in the partial overlap scenarios without a mean-variance relation.

6.4.4 Identification of number of classes

The convergence rates for each of the models in the different scenarios is shown in Table 6.6. Whereas all models converged consistently for $K = 1$, the convergence rate for models with additional classes appears to be attributable to the ability of the model to discern between the classes. For this reason, all models achieved a good convergence rate for $K = 2$ on the partial overlap datasets ($> 94\%$), but some models failed to converge in the more challenging scenarios under full class overlap. MV-GMM was able to consistently converge in the full overlap scenario under a heterogeneous mean-variance relation (99%), and RMV-GMM also showed a relatively high convergence rate (76%). Across the models estimated for $K = 3$, convergence rates are low especially in the full overlap datasets (2.3%–27%). The low convergence for three classes did not appear to affect the model selection, as we observed identical enumeration rates between the converged models and non-converged models across all scenarios (data not shown).

The model selection rates are shown in Table 6.7. All models performed well on the partial overlap datasets involving a mean-variance relation, achieving high rates (93% or

Table 6.3: Bias of the intercept and slope for the models across scenarios. The RRV column indicates whether random residual variance was simulated.

			Partial overlap without MV				Partial overlap with MV				Full overlap with MV			
K	RRV	Par.		RV-GMM	MV-GMM	RMV-GMM		RV-GMM	MV-GMM	RMV-GMM		RV-GMM	MV-GMM	RMV-GMM
2	No	$\beta_{1,0}$	.026	.026	.028	.029	.032	.031	.025	.027	.16	.17	.054	.082
		$\beta_{1,1}$	-.027	-.026	-.026	-.027	-.033	-.032	-.025	-.026	-.082	-.091	-.033	-.043
		$\beta_{2,0}$	-.018	-.019	-.018	-.019	-.024	-.021	-.015	-.016	-.13	-.13	-.045	-.067
		$\beta_{2,1}$	.016	.015	.017	.016	.023	.018	.012	.012	.049	.043	.030	.024
	Yes	$\beta_{1,0}$	.028	.026	.031	.031	.034	.031	.028	.028	.16	.18	.072	.12
		$\beta_{1,1}$	-.029	-.026	-.032	-.031	-.035	-.031	-.033	-.030	-.080	-.082	-.039	-.070
		$\beta_{2,0}$	-.021	-.019	-.020	-.020	-.025	-.020	-.017	-.017	-.13	-.13	-.058	-.077
		$\beta_{2,1}$	.018	.015	.024	.020	.025	.017	.016	.014	.049	.037	.043	.023
3	No	$\beta_{1,0}$	.088	.065	.078	.093	.085	.065	.061	.080	.41	.56	.12	.21
		$\beta_{1,1}$	-.025	-.018	-.040	-.024	-.031	-.032	-.019	-.022	-.28	-.39	-.068	-.12
		$\beta_{2,0}$	-.043	-.027	-.036	-.046	-.036	-.022	-.033	-.042	-.011	-.016	-.045	-.026
		$\beta_{2,1}$	.0026	.0022	.0036	.0032	.0038	.0016	.0045	.0058	.0016	-.0010	.0068	.0032
		$\beta_{3,0}$	-.014	-.014	-.014	-.015	-.018	-.015	-.0092	-.011	-.095	-.089	-.010	-.045
		$\beta_{3,1}$	.021	.019	.022	.020	.025	.017	.013	.018	.054	.057	.034	.035
	Yes	$\beta_{1,0}$	.092	.058	.14	.080	.092	.085	.11	.091	.37	.50	.12	.34
		$\beta_{1,1}$	-.037	-.026	-.092	-.021	-.029	-.040	-.038	-.033	-.26	-.37	-.063	-.23
		$\beta_{2,0}$	-.038	-.023	-.032	-.043	-.039	-.028	-.038	-.033	-.014	-.015	-.051	-.021
		$\beta_{2,1}$	.0038	.0023	.0058	.0028	.0023	.0042	.0025	.0011	.0032	.0030	.0053	.0023
		$\beta_{3,0}$	-.015	-.013	-.016	-.016	-.019	-.016	-.011	-.012	-.092	-.089	-.015	-.073
		$\beta_{3,1}$	.019	.016	.042	.019	.026	.022	.024	.016	.057	.053	.035	.037

Table 6.4: Bias of the mean-variance coefficients $\boldsymbol{\gamma}_1 = \{\gamma_{1,k}\}$ for $k = 1, \ldots, K$ across scenarios for MV-GMM and RMV-GMM, denoted by the MV and RMV columns, respectively.

			Partial overlap		Partial overlap + MV		Full overlap + MV	
K	RRV	Par.	MV	RMV	MV	RMV	MV	RMV
2	No	$\gamma_{1,1}$	.0063	.0075	.0078	.0063	-.026	-.099
		$\gamma_{1,2}$	-.0010	-.0029	-.015	-.012	.011	.045
	Yes	$\gamma_{1,1}$	.0053	.013	.031	.0069	.14	-.16
		$\gamma_{1,2}$	.016	-.015	-.065	-.021	-.081	.071
3	No	$\gamma_{1,1}$	.0010	.0028	-.017	-.022	-.055	-.13
		$\gamma_{1,2}$	-.0089	-.0097	-.027	-.026	-.023	-.029
		$\gamma_{1,3}$	.0024	.0010	.0015	-.0025	.013	.045
	Yes	$\gamma_{1,1}$	.032	.0088	.0010	-.022	.11	-.26
		$\gamma_{1,2}$	-.026	-.014	-.028	-.045	-.017	-.046
		$\gamma_{1,3}$	.029	-.0051	-.020	-.0042	-.11	.086

Table 6.5: Bias of the random residual variance scale coefficient σ_ω across scenarios for RV-GMM and RMV-GMM, denoted by the RV and RMV columns, respectively.

		Partial overlap		Partial overlap + MV		Full overlap + MV	
K	RRV	RV	RMV	RV	RMV	RV	RMV
2	No	.050	.050	.25	.052	.26	.12
	Yes	-.0058	-.0064	.082	-.0068	.086	.037
3	No	.058	.057	.21	.059	.21	.089
	Yes	-.0045	-.0053	.057	-.0059	.059	.022

Table 6.6: Convergence rate during estimation of the models for different values of K, with the data comprising two classes. The "Hsk." column indicates the type of heteroskedasticity under which model selection was simulated.

Overlap	Hsk.	Model	$K = 1$	$K = 2$	$K = 3$
Partial	MV	GMM	100%	100%	41%
		MV-GMM	99%	98%	43%
		RMV-GMM	100%	80%	38%
	RRV	GMM	100%	99%	30%
		MV-GMM	100%	94%	70%
		RMV-GMM	100%	98%	41%
Full	MV	GMM	100%	32%	8.3%
		MV-GMM	100%	99%	28%
		RMV-GMM	100%	76%	23%
	RRV	GMM	100%	17%	2.3%
		MV-GMM	100%	63%	27%
		RMV-GMM	100%	20%	5.7%

more) of identifying the correct number of classes. GMM showed to be insensitive to heteroskedasticity under well-separated classes, as it performed well in both the mean-variance and residual random variance cases, with selection rates of 96% and 98%, respectively. MV-GMM often overestimated the number of classes (79%) under the presence of residual random variance. For the datasets with fully overlapping class trajectories, MV-GMM and RMV-GMM were able to consistently recover the number of classes, indicating that the models can discern the heterogeneity in the mean-variance relation between classes. As expected, GMM identified only a single class on practically all full-overlap datasets, as there was practically no difference in the class trajectories. Similarly, under the absence of a mean-variance relation, the two classes are not discernible, and one therefore would expect all models to identify only a single class. MV-GMM often (84%) overestimated the number of classes here.

6.5 Case study

We use the proposed models to explore the heterogeneity in the number of weekly confirmed COVID-19 cases per county throughout the United States of America. The purpose of the analysis is to identify in which ways the number of cases has changed over time per county. Providing insights into the historical developments of different counties can support decision makers in tailoring policies to similar counties. By county, we refer to counties and county-equivalent administrative regions. We consider the 15-week period starting from June 1st to September 13th, 2020, which we selected for the relatively stagnant growth surrounding the period.

Across the many counties in the USA, we expect to find heterogeneity in the number of cases over time. Such heterogeneity could arise from different policies per county, geographical factors, and demographic factors. Similarly, there may be heterogeneity in

Table 6.7: Model selection rates across data scenarios and models. The "Hsk." column indicates the type of heteroskedasticity under which model selection was simulated. The column K in **bold** represents the expected number of classes for the respective data scenario.

Overlap	Hsk.	Model	$K = 1$	$K = 2$	$K = 3$
Partial	MV	GMM	0.0%	**96%**	4.0%
		MV-GMM	0.0%	**100%**	0.0%
		RMV-GMM	0.0%	**93%**	7.0%
	RRV	GMM	0.0%	**98%**	2.0%
		MV-GMM	0.0%	**21%**	79%
		RMV-GMM	0.0%	**98%**	2.0%
Full	MV	GMM	99%	**1.0%**	0.0%
		MV-GMM	0.0%	**99%**	1.0%
		RMV-GMM	0.0%	**96%**	4.0%
	RRV	GMM	**100%**	0.0%	0.0%
		MV-GMM	**16%**	76%	8.0%
		RMV-GMM	**100%**	0.0%	0.0%

the variance between counties. Modeling county-specific variability has the advantage of providing more reliable prediction intervals. Lastly, we expect to see differences in heteroskedasticity between classes, because the variability in number of new cases directly depends on the number of infectious people. For example, considering the case of a county with few infected people with one or two new cases per day, against a city with thousands of infectious people, and hundreds of new cases per day.

6.5.1 Data

The case data is obtained from the COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (Dong *et al.*, 2020)[4]. The repository comprises daily cumulative confirmed cases and deaths aggregated from US public health departments at the state and county level. We include data from all fifty states, and the District of Columbia, comprising a total of 3,150 counties. We exclude 29 counties where no cases were reported during the period of interest[5]. In order to compute the normalized number of cases, counties for which the population size was unavailable in the dataset were therefore also excluded, which occurred in only seven counties[6].

On 5,787 observation days (1.73%), the cumulative number of confirmed cases are lower than the preceding day, likely arising from later corrections. We address overreporting by applying a centered median filter with a window of 3 days, which leaves the data unaffected if the daily total counts numbers are monotonic. For count corrections that were applied after more than one day, we truncate any observations exceeding future observations.

[4]Available at https://github.com/CSSEGISandData/COVID-19.

[5]No cases were reported in counties across Alaska (2), Hawaii (1), Massachusetts (2), Nevada (1), Texas (1), and Utah (22).

[6]Population size data was absent from one county in Massachusetts, and six counties in Utah.

We use the corrected daily cumulative confirmed new cases of COVID-19 per county to compute the number of new cases per week. This was needed as many counties only provide updates on specific days of the week. The weekly number of new cases are computed from the difference in cumulative cases between Sundays. Across counties, 17% of weeks had zero new cases within the time period of interest. The median weekly number of new cases, ignoring zeros, is 64.6 per 100,000 inhabitants, with the 99th percentile at 555 new cases per 100,000 inhabitants. The highest observed number of weekly new cases is approximately 10,000 per 100,000 inhabitants.

Lastly, it is necessary to address the difference in the numbers between counties of several orders of magnitude. We therefore model the logarithm of the weekly normalized new cases per 100,000 inhabitants instead. Using this representation, the group trajectories are representative across counties, as changes in the number of cases are modeled relative to the order of magnitude. We treat observation days with zero new cases as having 0.5 cases per 100,000 inhabitants. This ensures that the zero-case observations are not too distant in magnitude from the other observations. The range of the processed data is $[-0.693, 9.21]$.

6.5.2 Model specification

We model the log-normalized new cases over time using the GMM, MV-GMM, and RMV-GMM models as defined in Equation 6.1, Equation 6.3, and Equation 6.5, respectively. We apply uninformative priors, computed in the same way as defined for the models used in the simulation study. In view of the volume of data, most of the priors will have little effect on the posterior distribution.

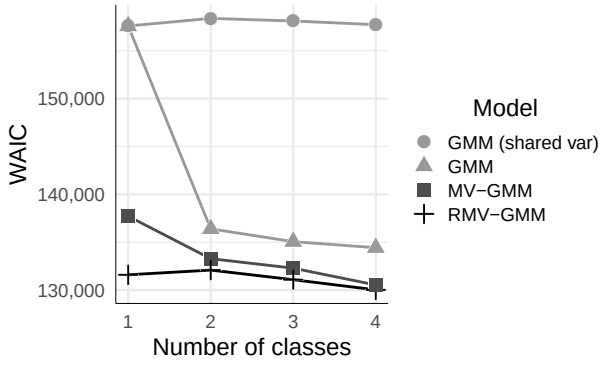
In a preliminary analysis, we observed a poor fit when using the second-order polynomial representation as was used in the simulation studies. We therefore opt for using a basis spline (B-spline) of degree 3 (Hastie and Tibshirani, 1990). Each county trajectory is therefore represented by a piecewise second-order polynomial function. The B-spline provides a smoother fit while being able to account for more shapes than a polynomial of the same degree. We evaluate GMM for both a shared and class-specific residual error scale. We tested the MV-GMM and RMV-GMM models with class-specific residual error scales as well, but the sampling was not stable in three- and four-class models. Here, the sampling of the residual scale parameter would often diverge for one of the classes. We therefore specify a shared residual error scale parameter, as done in the simulation analysis.

We sample each of the models for 1 to 4 classes. We restricted the maximum number to four classes due to the computational time needed especially for RMV-GMM considering the large sample size.

6.5.3 Model evaluation

We evaluate each model using multiple Markov chains, which enables a reliable assessment of convergence and diagnosis of the posterior distribution (Gelman *et al.*, 1992). As the possibility for multimodality increases with the number of latent classes, an increasing number of Markov chains were used (up to 20). A solution was determined to be reliable when the same solution was found in multiple chains. We report the model parameters in terms of the mean posterior estimates, and 80% highest density intervals (HDI). We

Figure 6.3: The model fit for each of the models, across the different number of classes (lower is better).



compare the models using the same approach as the model selection analysis above. As recommended by Van de Schoot *et al.* (2017), we measure the classification agreement of the model for individual trajectories. This is typically assessed using relative entropy, which is a measure of class separation in terms of the uncertainty of the class-specific posterior probabilities. A score of 0 indicates that the assignment of counties to classes is maximally uncertain (i.e., uniform posterior class probabilities), whereas a score of 1 indicates a perfect separation between classes (i.e., each county is assigned to a single class).

6.5.4 Results

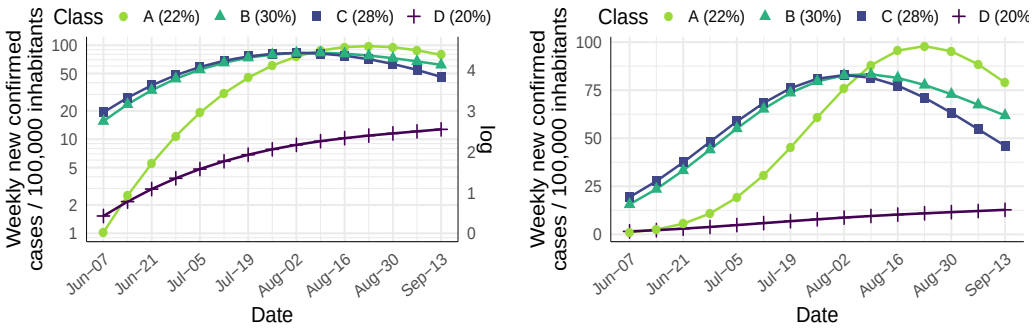
Number of classes The WAIC for each of the class solutions is shown in Figure 6.3. Although the single-class models already account for heterogeneity, all models achieve an improved fit with an additional number of classes. The importance of accounting for heteroskedasticity in this case study is evident from the considerable improvement in WAIC between the GMM with shared variance and the GMM with class-specific variance. Even more so, the MV-GMM and RMV-GMM models provide a significantly better fit than the GMM model, demonstrating the value of modeling the mean-variance relation in this case study. MV-GMM and RMV-GMM achieve a similar fit for the three- and four-class solutions. We therefore manually compare these solutions between the models. The solutions of RMV-GMM comprise one or two small classes with a proportion below 5% which did not capture a sufficiently distinguishing longitudinal aspect. In contrast, the class proportions of the MV-GMM solutions are relatively balanced. We select the four-class MV-GMM solution as the preferred solution due to the significant improvement in WAIC, the introduction of a relatively distinct class over the three-class solution, and the consistency of the class allocations across the chains.

Preferred solution The four group trajectories identified by MV-GMM are shown in Figure 6.4, ordered by the number of new weekly cases in the last week. The relative entropy of 0.82 indicates the classes are well-separated (Diallo *et al.*, 2016). Figure 6.5 shows the expected standard deviation for the respective mean of each class over time.

Figure 6.4: The estimated group trajectories for the preferred solution.

(a) Log-scale.

(b) Original scale.



The point estimates of the model coefficients and the 80% HDI are shown in Table 6.8.

Class A (22%) represents counties with a considerable increase in the number of cases over time, with a decreasing standard deviation as the number of infectious people becomes greater. Class B (30%) and class C (28%) follow almost exactly the same trend, but counties in class C have a 50% higher standard deviation in relation to the mean. Class D (20%) comprises counties with relatively few cases and high uncertainty on a week-to-week basis.

The classified counties are visualized on a map in Figure 6.6. Despite the independent classification of counties, the map shows strong regional similarities, with neighboring counties tending to belong to the same cluster. These findings could be used to create a minimal set of tailored policies, each addressing specific intercounty and interstate regions of the USA with a similar development in cases.

6.6 Discussion

In this work, we compared the performance of different growth mixture models under heteroskedasticity arising from heterogeneous mean-variance relations and random residual variance. We evaluated four GMM variants: the standard GMM, a GMM accounting for random residual variance between trajectories, a GMM that accounts for a heterogeneous mean-variance relation, and a GMM that handles both aspects. We demonstrated the feasibility of recovering the correct number of classes and the group trajectories under these conditions. Notably, we obtained these results on datasets of only 250 trajectories with ten observations, which is relatively small compared to other longitudinal clustering studies. Although we did not investigate the impact of the sample size, it is likely that the recovery of the group trajectories will improve with a greater number of observations and trajectories, as has been shown in previous simulation studies (Den Teuling *et al.*, 2021; Martin and von Oertzen, 2015). MV-GMM and RMV-GMM were able to correctly identify the heterogeneous mean-variance relations or absence thereof in all but the most difficult scenario involving full class overlap with random residual variance. Moreover, the group trajectory recovery of MV-GMM and RMV-GMM in the absence of heteroskedasticity

Figure 6.5: The estimated group trajectories for the preferred solution with the predicted standard deviation on the expected value.

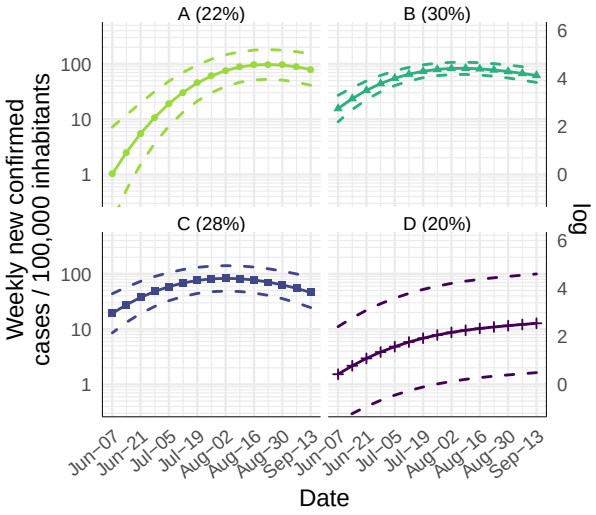
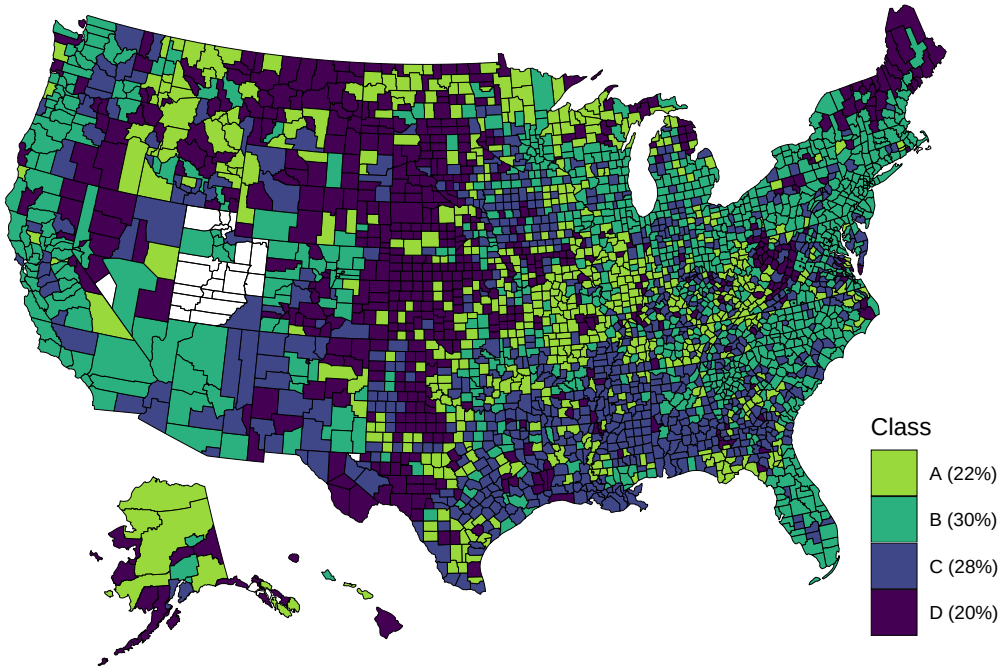


Figure 6.6: The counties of the USA as classified by MV-GMM. Counties colored in white either had no available data or had no recorded cases throughout the selected period.



was not negatively affected, suggesting that the model can be used in an exploratory setting where the presence thereof may not be known in advance. Under the presence of high random residual variance, MV-GMM overestimated the number of classes due to the mean-variance relation being used to explain the heterogeneity in the variance between trajectories. Fortunately, this issue can be detected from the classes having similar coefficients of variation. As an alternative to using RMV-GMM, this issue may be alleviated by constraining the coefficient of variation to be considerably different between classes.

For GMM, group trajectory estimation was marginally affected in the presence of heteroskedasticity, but this did not affect the trajectory class-membership recovery in the partial overlap scenarios. In the full overlap scenarios, the standard GMM consistently underestimated the number of classes, demonstrating the benefit of accounting for class-specific heteroskedasticity through MV-GMM or RMV-GMM. The group trajectory estimation of RV-GMM in the partial overlap scenarios was on par with the models that did account for a mean-variance relation, suggesting it is a suitable model for exploratory analyses where a mean-variance relation is not expected.

Previous studies have investigated the estimation of heteroskedastic mixture models with variance functions. Accounting for heteroskedasticity is important for obtaining correct prediction intervals. The selection of the correct variance functions remains a challenge, especially when accounting for a different function per class. MV-GMM requires only one additional parameter per class compared to other models that account for heteroskedasticity. It therefore could serve as a practical starting point for the evaluation of heterogeneous heteroskedasticity with a dependency on the mean.

The sampling of the mixture models using NUTS was found to be exceptionally stable despite the multimodality of the posterior distribution and overlapping classes. Label switching was not observed within Markov chains, thus it only needed to be addressed between Markov chains. While this makes the samples of the Markov chains easy to interpret, it prevents the sampling algorithm from fully exploring the posterior. As such, more Markov chains need to be sampled to properly assess the extent of multimodality.

In the case study, the MV-GMM and RMV-GMM were found to better model the data than GMM. Notably, the strength of the mean-variance relation was found to differ significantly between classes, yielding different prediction intervals per county. The models underestimated the group trajectory close to zero weekly cases due to the excess number of lower-truncated observations. A zero-inflated mixture model accounting for the excess number of case-free days may have provided a better fit and more reliable estimates for this class of counties. RMV-GMM generally described the data better than the other two models in case of the one- and two-class solutions. However, for three and four classes, the estimate of the mean-variance relation was not stable in one of the classes, resulting in poor convergence across parameters.

We conducted the simulation study with the models as the data generating processes. It therefore remains unclear how well the mean-variance models handle the misspecification of, for example, the random effects and the mean-variance relation. The case study comprised these aspects and provided a positive indication in this regard. We opted to apply random effects to all fixed effects to reliably model the trajectory-specific means, but a more parsimonious random effects model (i.e., fewer random effects) may have sufficed. The normal distribution is a favorable choice for the random effects from the perspective of the central limit theorem, but in case of a skewed distribution may result in group

trajectory estimation errors and an overextraction of the number of classes (Den Teuling *et al.*, 2021; Van Horn *et al.*, 2012; Bauer and Curran, 2003).

Overall, we recommend the application of MV-GMM as a first exploratory step when heteroskedasticity is suspected or of interest, as it is parsimonious compared to other models that account for heteroskedasticity. If such heteroskedasticity is established, then the RMV-GMM could be fitted thereafter to establish the significance of the mean-variance relations. In case that only random heteroskedasticity is suspected, the RV-GMM is preferable over GMM. Due to multimodality, it is strongly recommended to fit the model repeatedly and to compare the respective models on parameter convergence, model fit (WAIC), and the obtained group trajectories and class proportions. In case of the absence of a mean-variance relation, we recommend the application of RV-GMM over GMM in applications with considerable within-subject variance, as the former yielded a consistently better group trajectory estimation in our simulation study.

Supplementary materials

The Stan code used to specify the various models, as well as the R code that has been used to run the simulation study and perform the case study analysis, can be found at https://github.com/niekdt/meanvar-clustering-longitudinal-data.

Appendix

6.A Case study

Table 6.8: Point estimates (posterior means), 80% HDI, and PSRF of the model parameters of the preferred solution.

Par.	Mean	Lower bound	Upper bound	$\hat{R}$
π_A	.22	.21	.23	1.00
π_B	.31	.29	.32	1.00
π_C	.28	.26	.29	1.01
π_D	.20	.19	.21	1.00
$\beta_{A,0}$	.023	-.10	.15	1.03
$\beta_{A,1}$	4.5	4.2	4.8	1.01
$\beta_{A,2}$	5.0	4.8	5.1	1.00
$\beta_{A,3}$	4.3	4.2	4.5	1.01
$\beta_{B,0}$	2.7	2.7	2.8	1.01
$\beta_{B,1}$	2.1	2.0	2.2	1.02
$\beta_{B,2}$	1.8	1.7	1.9	1.01
$\beta_{B,3}$	1.4	1.3	1.4	1.04
$\beta_{C,0}$	3.0	2.7	2.8	1.03
$\beta_{C,1}$	1.8	1.6	1.9	1.01
$\beta_{C,2}$	1.7	1.6	1.8	1.00
$\beta_{C,3}$	.87	.78	.94	1.04
$\beta_{D,0}$	.42	.33	.53	1.02
$\beta_{D,1}$	1.8	1.5	2.1	1.01
$\beta_{D,2}$	1.9	1.7	2.1	1.00
$\beta_{D,3}$	2.1	2.0	2.3	1.02
ϕ_0	.68	.66	.69	1.00
$\phi_{1,A}$	-.25	-.26	-.25	1.02
$\phi_{1,B}$	-.46	-.47	-.46	1.00
$\phi_{1,C}$	-.30	-.30	-.29	1.00
$\phi_{1,D}$	.017	.0082	.026	1.00
$\sigma_{\beta,0}$	.84	.82	.86	1.04
$\sigma_{\beta,1}$	.35	.34	.36	1.00
$\sigma_{\beta,2}$	.29	.28	.30	1.01
$\sigma_{\beta,3}$	.27	.26	.27	1.01

Chapter 7

Discussion and future work

Through the clustering of longitudinal data, researchers can obtain a better understanding of the differences between subjects over time within a heterogeneous population. The identification of groups of subjects with similar longitudinal characteristics is a pragmatic and valuable tool for providing a more detailed description of the population than a general common trend or average. The resulting clusters may also be helpful for addressing a specific proportion of the population. For example, it can be used for the improvement of PAP therapy adherence management for sleep apnea patients, where it facilitates better triage and patient-tailored intervention, leading to improved adherence and health outcomes. As another example, the incidence of COVID-19 across regions in a country can be better understood by identifying regions with similar developments over time. This could help policy makers to identify regions that would benefit from the same policies, thereby minimizing the overall number of required policies.

In this thesis, we present a comparison of various methodologies for clustering longitudinal data, and we propose extensions for jointly clustering the variance and other distributional aspects over time. We selected, applied, and adapted methods for the analysis of patient PAP therapy adherence during the first three months of therapy, and the incidence over time of COVID-19 across regions of the United States of America. Furthermore, we have created statistical software, named `latrend`[1], that provides a framework for researchers to compare different approaches for clustering longitudinal data in a standardized way, and evaluating new methods.

7.1 The current state

Researchers have a vast selection of methods at their disposal for exploring and modeling data heterogeneity using longitudinal clustering. However, adjacent academic disciplines have their own terminology, their own idiosyncratic use of tools and methods, and journals with different focus areas. This makes it difficult for researchers to familiarize themselves with the broad range of methods that are available for clustering longitudinal data. For that reason, we have created a broad scoping review in Chapter 2 of different approaches to longitudinal clustering, presented in the form of a tutorial. Here, we categorized the

[1]The `latrend` R package is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=latrend and on GitHub at https://github.com/philips-software/latrend.

available methods into general approaches ranging from a naive cross-sectional approach to a model-based approach. The chapter contains a demonstration for each of the approaches in a case study, involving one or more methods and noting similarities and differences between the clusters identified by the methods.

With applicable methods originating from different and disconnected areas of research, there are relatively few cross-disciplinary comparison studies. We aimed to provide guidelines for researchers in determining the appropriate method depending on the type of longitudinal dataset. In Chapter 3, we therefore investigated the similarities, strengths, and limitations of different approaches in a simulation study and on a real-life case study. We have demonstrated that longitudinal k-means (KML) and group-based trajectory modeling (GBTM) yield similar clusters, given that the shape of the trends was correctly specified for GBTM. This suggests that KML is favorable for use as an initial exploratory analysis, and may be followed up by GBTM once there is more clarity on the expected shape of the trends. Our findings also suggest that the feature-based GCKM method could be a viable alternative to growth mixture modeling (GMM) when applied to intensive longitudinal data. The case study showed that the resulting clusters may differ greatly depending on the choice of method. This highlights the importance on the choice of methods, but even more so that researchers should be careful in interpreting the clusters from such an analysis as being distinct or the only possible representation. Applying multiple methods with different assumptions provides insights about the population heterogeneity from different angles.

In setting up the simulation and case studies of Chapter 2 and 3, we observed the difficulty of dealing with the different inputs and outputs of the respective R package of each of the methods. This presents a considerable barrier to researchers when it comes to comparing and evaluating different methods on a longitudinal dataset with the aim of identifying the most suitable method. We therefore developed a general framework to facilitate a standardized way of estimating and analyzing methods for clustering longitudinal data, regardless of the type of approach or underlying software used. The resulting software, named `latrend`, is described in Chapter 5. To the best of our knowledge such a generic framework for clustering longitudinal data does not yet exist. The software currently supports 18 methods and can be extended further by the user. Researchers can now evaluate and compare different methods originating from various R packages with minimal coding effort. We expect the software will contribute to the awareness and accessibility of a variety of methods to researchers who are not yet familiar with the field.

7.2 Proposed approaches

For most of the methods that we have discussed, the similarity between trajectories is determined by the mean value over time. However, as we have demonstrated in Chapter 4 and 6, accounting for heterogeneity on multiple longitudinal aspects may yield a more accurate representation of the trajectories, and can considerably affect the result of the cluster analysis. In both chapters, we used a model-based approach based on GMM, which was done for two primary reasons: Firstly, the approach has an intuitive interpretation as a set of heterogeneous subgroups. Fewer clusters are then needed to capture high and complex heterogeneity compared to methods that assume homogeneous clusters like KML or GBTM. Secondly, the regression approach used by GMM can be extended to distributional regression for modeling the mean, and other (longitudinal) aspects such

as heteroskedasticity. With these two chapters, we filled some of the gaps regarding the estimation of GMM under the presence of different types of heteroskedasticity: temporal heteroskedasticity, a mean-variance relation, and subject-specific random residual variance.

Previous studies have demonstrated considerable heterogeneity in patient PAP therapy adherence, but have done so using either methods with low temporal granularity, or by only modeling the aggregated hours of usage over time. In Chapter 4, we aimed to explore the heterogeneity of daily patient PAP therapy adherence in their first three months of therapy in much more detail. Here, we jointly modeled the daily hours of usage of patients on three longitudinal aspects (attempt probability, mean hours of usage, and usage variability). Our approach was based on distributional regression, specified using a GMM approach based on a mixture of generalized additive models for location, scale and shape (GAMLSS).

The choice of GMM was further motivated by the case study. In PAP therapy, any daily usage above 4 hours is generally regarded as sufficient, whereas a large part of the heterogeneity comprises subjects around 6 hours of usage. Considering that we were interested in identifying clusters of patients that exhibited different changes in usage over time, the mean subject level was therefore of less interest than the relative changes in usage over time. This aspect is addressed through the inclusion of a subject-specific random intercept. This consequently lowers the contribution of the patient intercept to the model fit, resulting in fewer clusters that merely differ on the mean level of usage. Arguably, a feature-based approach where patient trajectories are estimated independently could have yielded similar results in a significantly shorter amount of time. This approach comes with its own caveats however, as researchers need to put effort into the assessment and handling of model estimation errors for a proportion of subjects.

Our case study analysis revealed a non-linear relation between attempt probability and usage time, demonstrating of the importance of using a hurdle modeling approach for describing PAP therapy adherence data. The modeling of heteroskedasticity as a function of time also turned out to be of added explanatory power. For the identified clusters which exhibited a change over time, a corresponding change in variance over time was observed, suggesting the presence of heterogeneous mean-variance relation in the population. Overall, the methodology could be of interest in other areas of therapy adherence research as well.

Lastly, in Chapter 6 we explored the impact of heterogeneity in the heteroskedasticity on the estimation of GMM. We proposed extensions to GMM for handling heteroskedasticity under a heterogeneous mean-variance relation, subject-specific random residual variance, and a combined model. Through a simulation study, we showed that the proposed models were able to reliably recover the heteroskedasticity in most scenarios. Moreover, we found that under the absence of a mean-variance relation, the relevant models did not wrongly identify such a relation. These findings suggest that the models are suitable for exploratory purposes on populations where heterogeneity is expected on location and scale. With these promising results, we applied the models to a real-life case study, exploring the heterogeneity of the number of new COVID-19 cases across all counties in the USA. The results showed geographically correlated clusters exhibiting different levels of variance.

7.3 Future work

In this work we compared different methods for clustering longitudinal data and proposed some extensions for modeling distributional heterogeneity. Yet, many interesting research questions remain. For instance, in the scoping review of Chapter 2 and the comparison study in Chapter 3, we applied a selection of the most common methods. However, we did not touch upon all available domains for clustering longitudinal data, such as methods from the field of functional data analysis. More cross-disciplinary comparison studies are needed. This connects different areas of research, to the benefit of all domains. Having access to a broader choice of methods enables researchers to identify a more suitable method, but also increases the difficulty of identifying such method.

Few studies address heterogeneity in combination with heteroskedasticity other than assuming a different constant variance per cluster. Modeling heteroskedasticity is primarily useful for ensuring cluster- or subject-specific confidence intervals around the prediction of the mean, a task that is generally not the focus in clustering. Yet, as we have shown in Chapter 4 and 6, modeling the variance can affect the resulting clusters, and may lead to new insights about the population. The field would benefit from a greater focus on jointly modeling longitudinal characteristics, involving the location, scale, and other relevant longitudinal aspects. With the increasing availability of ILD, modeling developments, and increase in computational capabilities, such models have become feasible to estimate.

The heteroskedastic mixture models proposed in Chapter 6 achieved a favorable recovery of the parameters with low bias even under homoskedasticity. The simulation study was conducted under the assumption of the correct specification of the mean-variance relation (if any), but in real-world studies the exact relation is unlikely to be known. It remains unclear how the model behaves under the misspecification of the mean-variance relation, and whether the correct relation can be recovered.

Data collection and storage capabilities are growing faster than the computational capabilities. The computational efficiency of methods needs to be improved. Some of the more traditional methods have poor computational scaling with sample size. Even today, the estimation of some multilevel mixture models on intensive longitudinal datasets is infeasible due to the long computation times involved. Challenges with respect to the computation time have been a recurring topic throughout this thesis, as we have sought the limit in terms of dataset sizes in Chapter 4 and 6 that are still practical to estimate in terms of computation time.

One such computational optimization is the dimensionality reduction of traditional model-based methods. Here, trajectories are represented through model coefficients instead of the raw measurements. The unpooled estimation of trajectories is more susceptible to estimation errors, but under a large number of observations this problem should be less of a concern. We have shown in Chapter 3 that a feature-based approach can achieve similar results to GMM. It should be noted that this was observed under perfect model data conditions in which all subject trajectories could be represented by the same trajectory model. In practice, we have seen that the feature-based approach is sensitive to outliers (i.e., trajectories that do not meet the model assumptions), resulting in erroneous coefficient estimates which at best is identified as latent classes of outliers, and at worst results in a misrepresentation of the underlying data structure. A feature-based approach involving Bayesian (regularized) inference may provide a useful direction here, enabling the mixture model to consider the uncertainty of the estimation of the trajectory model coefficients.

In this thesis, we have contributed to a better understanding of the strengths and limitations of the various methods for longitudinal clustering, which will benefit researchers in better exploring longitudinal data heterogeneity. We hope that our overview, comparison study and software contribute to an improved connection between the different fields of research that are addressing clustering longitudinal data. The proposed models that we have demonstrated in the case studies may inspire to broaden the scope of longitudinal clustering: accounting for heterogeneity in other relevant longitudinal aspects besides the expected value.

Bibliography

Adepeju M, Langton S, Bannister J (2019). *akmedoids: Anchored Kmedoids for Longitudinal Data Clustering*. R package version 0.1.2, URL https://CRAN.R-project.org/package=akmedoids.

Adepeju M, Langton S, Bannister J (2020). "Akmedoids R package for generating directionally-homogeneous clusters of longitudinal data sets." *Journal of Open Source Software*, **5**(56), 2379. DOI: 10.21105/joss.02379.

Aghabozorgi S, Shirkhorshidi AS, Wah TY (2015). "Time-series clustering – A decade review." *Information Systems*, **53**, 16–38. ISSN 0306-4379. DOI: 10.1016/j.is.2015.04.007.

Aitkin M (1996). "A general maximum likelihood analysis of overdispersion in generalized linear models." *Statistics and computing*, **6**(3), 251–262. ISSN 0960-3174. DOI: 10.1007/bf00140869.

Aitkin M (1999). "A general maximum likelihood analysis of variance components in generalized linear models." *Biometrics*, **55**(1), 117–128. ISSN 0006-341X. DOI: 10.1111/j.0006-341x.1999.00117.x.

Akaike H (1983). "Information measures and model selection." *Bulletin of the International Statistical Institute*, **50**, 277–290.

Akantziliotou K, Rigby R, Stasinopoulos D (2002). "The R implementation of generalized additive models for location, scale and shape." In "Statistical modelling in Society: Proceedings of the 17$^{\text{th}}$ International Workshop on statistical modelling," volume 54, pp. 75–83. Statistical Modelling Society.

Aloia MS, Goodwin MS, Velicer WF, Arnedt JT, Zimmerman M, Skrekas J, Harris S, Millman RP (2008). "Time series analysis of treatment adherence patterns in individuals with obstructive sleep apnea." *Annals of Behavioral Medicine*, **36**(1), 44–53. ISSN 0883-6612. DOI: 10.1007/s12160-008-9052-9.

Alvarez E, Brida JG, Limas E (2020). "Comparisons of COVID-19 dynamics in the different countries of the World using Time-Series clustering." *medRxiv*.

Ansari A, Jedidi K, Jagpal S (2000). "A hierarchical Bayesian methodology for treating heterogeneity in structural equation models." *Marketing Science*, **19**(4), 328–347. DOI: 10.1287/mksc.19.4.328.11789.

Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I (2013). "An extensive

comparative study of cluster validity indices." *Pattern recognition*, **46**(1), 243–256. ISSN 0031-3203. DOI: 10.1016/j.patcog.2012.07.021.

Arthur D, Vassilvitskii S (2007). "k-means++: the advantages of careful seeding." In "Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms," SODA 2007, pp. 1027–1035. ACM, New York, Philadelphia, PA, USA. ISBN 978-0-898716-24-5. DOI: 10.1145/1283383.1283494.

Asparouhov T, Muthén B (2014). "Auxiliary variables in mixture modeling: Three-step approaches using M plus." *Structural Equation Modeling: A Multidisciplinary Journal*, **21**(3), 329–341.

Axén I, Bodin L, Bergström G, Halasz L, Lange F, Lövgren PW, Rosenbaum A, Leboeuf-Yde C, Jensen I (2011). "Clustering patients on the basis of their individual course of low back pain over a six month period." *BMC Musculoskeletal Disorders*, **12**(1), 99. ISSN 1471-2474. DOI: 10.1186/1471-2474-12-99.

Babbin SF, Velicer WF, Aloia MS, Kushida CA (2015). "Identifying longitudinal patterns for individuals and subgroups: An example with adherence to treatment for obstructive sleep apnea." *Multivariate Behavioral Research*, **50**(1), 91–108. ISSN 0027-3171. DOI: 10.1080/00273171.2014.958211.

Bakk Z, Tekle FB, Vermunt JK (2013). "Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches." *Sociological methodology*, **43**(1), 272–311. DOI: 10.1177/0081175012470644.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. DOI: 10.18637/jss.v067.i01.

Bauer DJ (2007). "Observations on the use of growth mixture models in psychological research." *Multivariate Behavioral Research*, **42**(4), 757–786. ISSN 0027-3171. DOI: 10.1080/00273170701710338.

Bauer DJ, Curran PJ (2003). "Distributional assumptions of growth mixture models: Implications for overextraction of latent trajectory classes." *Psychological Methods*, **8**(3), 338–363. ISSN 1939-1463. DOI: 10.1037/1082-989x.8.3.338.

Benaglia T, Chauveau D, Hunter DR, Young D (2009). "mixtools: An R Package for Analyzing Finite Mixture Models." *Journal of Statistical Software*, **32**(6), 1–29. DOI: 10.18637/jss.v032.i06.

Bezdek JC (1981). *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York-London. ISBN 0-306-40671-3. DOI: 10.1007/978-1-4757-0450-1.

Boker S, Neale M, Maes H, Wilde M, et al (2011). "OpenMx: an open source extended structural equation modeling framework." *Psychometrika*, **76**(2), 306–317. ISSN 0033-3123. DOI: 10.1007/s11336-010-9200-6.

Bolck A, Croon M, Hagenaars J (2004). "Estimating latent structure models with categorical variables: One-step versus three-step estimators." *Political Analysis*, **12**(1), 3–27. ISSN 1047-1987. DOI: 10.1093/pan/mph001.

Bollen KA, Curran PJ (2006). *Latent curve models: A structural equation perspective*, volume 467. John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 978-0-471-45592-9.

Bouveyron C (2015). *funFEM: Clustering in the Discriminative Functional Subspace*. R package version 1.1, URL https://CRAN.R-project.org/package=funFEM.

Bruckers L, Molenberghs G, Pulinx B, Hellenthal F, Schurink G (2018). "Cluster analysis for repeated data with dropout: Sensitivity analysis using a distal event." *Journal of Biopharmaceutical Statistics*, **28**(5), 983–1004. DOI: 10.1080/10543406.2018.1428612.

Bryk AS, Raudenbush SW (1987). "Application of hierarchical linear models to assessing change." *Psychological Bulletin*, **101**(1), 147–158. ISSN 1939-1455. DOI: 10.1037/0033-2909.101.1.147.

Burkardt J (2014). "The truncated normal distribution." *Technical report*.

Bürkner PC (2017). "brms: An R package for Bayesian multilevel models using Stan." *Journal of Statistical Software*, **80**(1), 1–28. DOI: 10.18637/jss.v080.i01.

Burton A, Altman DG, Royston P, Holder RL (2006). "The design of simulation studies in medical statistics." *Statistics in medicine*, **25**(24), 4279–4292.

Bushway SD, Sweeten G, Nieuwbeerta P (2009). "Measuring long term individual trajectories of offending using multiple methods." *Journal of Quantitative Criminology*, **25**(3), 259–286. DOI: 10.1007/s10940-009-9070-1.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A probabilistic programming language." *Journal of Statistical Software*, **76**(1), 1–32. ISSN 1548-7660. DOI: 10.18637/jss.v076.i01.

Carroll RJ (1982). "Adapting for Heteroscedasticity in Linear Models." *The Annals of Statistics*, **10**(4), 1224–1233. DOI: 10.1214/aos/1176345987.

Carroll RJ (2003). "Variances are not always nuisance parameters." *Biometrics*, **59**(2), 211–220. DOI: 10.1111/1541-0420.t01-1-00027.

Cayanan EA, Bartlett DJ, Chapman JL, Hoyos CM, Phillips CL, Grunstein RR (2019). "A review of psychosocial factors and personality in the treatment of obstructive sleep apnoea." *European Respiratory Review*, **28**(152), 190005.

Celeux G, Forbes F, Robert CP, Titterington DM (2006). "Deviance information criteria for missing data models." *Bayesian analysis*, **1**(4), 651–673. DOI: 10.1214/06-ba122.

Chen C, Hogg-Johnson S, Smith P (2007). "The recovery patterns of back pain among workers with compensated occupational back injuries." *Occupational and Environmental Medicine*, **64**(8), 534–540. ISSN 1351-0711. DOI: 10.1136/oem.2006.029215.

Collins LM, Lanza ST (2010). *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences*, volume 718. John Wiley & Sons, Inc., Hoboken, New Jersey. ISBN 978-0-470-22839-5.

Cragg JG (1971). "Some statistical models for limited dependent variables with application to the demand for durable goods." *Econometrica*, **39**(5), 829–844. ISSN 0012-9682. DOI: 10.2307/1909582.

Crawford MR, Espie CA, Bartlett DJ, Grunstein RR (2014). "Integrating psychology and medicine in CPAP adherence – New concepts?" *Sleep Medicine Reviews*, **18**(2), 123–139. ISSN 1087-0792. DOI: 10.1016/j.smrv.2013.03.002.

Davidian M, Giltinan DM (1993). "Some simple methods for estimating intraindividual variability in nonlinear mixed effects models." *Biometrics*, **49**(1), 59–73. DOI: 10.2307/2532602.

De Kort JM, Dolan CV, Lubke GH, Molenaar D (2017). "Studying the strength of prediction using indirect mixture modeling: Nonlinear latent regression with heteroskedastic residuals." *Structural Equation Modeling: A Multidisciplinary Journal*, **24**(2), 301–313. DOI: 10.1080/10705511.2016.1250636.

Den Teuling N (2022). *latrend: A Framework for Clustering Longitudinal Data*. R package version 1.5.0, URL https://CRAN.R-project.org/package=latrend.

Den Teuling NGP, Pauws SC, van den Heuvel ER (2021). "A comparison of methods for clustering longitudinal data with slowly changing trends." *Communications in Statistics - Simulation and Computation*. DOI: 10.1080/03610918.2020.1861464.

Den Teuling NGP, van den Heuvel ER, Aloia MS, Pauws SC (2021). "A latent-class heteroskedastic hurdle trajectory model: patterns of adherence in obstructive sleep apnea patients on CPAP therapy." *BMC Medical Research Methodology*, **21**(1), 1–15. DOI: 10.1186/s12874-021-01407-6.

Depaoli S (2013). "Mixture class recovery in GMM under varying degrees of class separation: Frequentist versus Bayesian estimation." *Psychological Methods*, **18**(2), 186–219. ISSN 1939-1463. DOI: 10.1037/a0031609.

Depaoli S, Clifton JP, Cobb PR (2016). "Just Another Gibbs Sampler (JAGS): Flexible Software for MCMC Implementation." *Journal of Educational and Behavioral Statistics*, **41**(6), 628–649. DOI: 10.3102/1076998616664876.

Desgraupes B (2018). *clusterCrit: Clustering Indices*. R package version 1.2.8, URL https://CRAN.R-project.org/package=clusterCrit.

Diallo TM, Morin AJ, Lu H (2017). "Performance of growth mixture models in the presence of time-varying covariates." *Behavior Research Methods*, **49**(5), 1951–1965. DOI: 10.3758/s13428-016-0823-0.

Diallo TMO, Morin AJS, Lu H (2016). "Impact of Misspecifications of the Latent Variance–Covariance and Residual Matrices on the Class Enumeration Accuracy of Growth Mixture Models." *Structural Equation Modeling: A Multidisciplinary Journal*, **23**(4), 507–531. DOI: 10.1080/10705511.2016.1169188.

Dietz E, Böhning D (2000). "On estimation of the Poisson parameter in zero-modified Poisson models." *Computational Statistics & Data Analysis*, **34**(4), 441–459. ISSN 0167-9473. DOI: 10.1016/s0167-9473(99)00111-5.

Ding M (2019). "Development of a Mixture Model (SMM) Allowing for Smoothing Functions of Trajectories." *medRxiv*.

Dong E, Du H, Gardner L (2020). "An interactive web-based dashboard to track COVID-19 in real time." *The Lancet Infectious Diseases*, **20**(5), 533–534. DOI: 10.1016/S1473-3099(20)30120-1.

Donnat C, Holmes S (2021). "Modeling the heterogeneity in COVID-19's reproductive

number and its impact on predictive scenarios." *Journal of Applied Statistics*, **0**(0), 1–29. DOI: 10.1080/02664763.2021.1941806.

Dowle M, Srinivasan A (2020). *data.table: Extension of data.frame*. R package version 1.13.0, URL https://CRAN.R-project.org/package=data.table.

D'Rozario AL, Galgut Y, Bartlett DJ (2016). "An update on behavioural interventions for improving adherence with continuous positive airway pressure in adults." *Current Sleep Medicine Reports*, **2**(3), 166–179. ISSN 2198-6401. DOI: 10.1007/s40675-016-0051-2.

Dunn JC (1973). "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters." *J. Cybernet.*, **3**(3), 32–57. ISSN 0022-0280. DOI: 10.1080/01969727308546046.

Dziak JJ, Li R, Tan X, Shiffman S, Shiyko MP (2015). "Modeling intensive longitudinal data with mixtures of nonparametric trajectories and time-varying effects." *Psychological Methods*, **20**(4), 444–469. ISSN 1939-1463. DOI: 10.1037/met0000048.

Einbeck J, Hinde J (2006). "A note on NPML estimation for exponential family regression models with unspecified dispersion parameter." *Austrian Journal of Statistics*, **35**(2&3), 233–243.

Elmer J, Jones BL, Nagin DS (2018). "Using the Beta distribution in group-based trajectory models." *BMC medical research methodology*, **18**(1), 152. DOI: 10.1186/s12874-018-0620-9.

Elmer J, Jones BL, Zadorozhny VI, Puyana JC, Flickinger KL, Callaway CW, Nagin D (2019). "A novel methodological framework for multimodality, trajectory model-based prognostication." *Resuscitation*, **137**, 197–204.

Enders CK (2011). "Missing not at random models for latent growth curve analyses." *Psychological methods*, **16**(1), 1.

Enders CK, Tofighi D (2008). "The impact of misspecifying class-specific residual variances in growth mixture models." *Structural Equation Modeling: A Multidisciplinary Journal*, **15**(1), 75–95. DOI: 10.1080/10705510701758281.

Ernst AF, Timmerman ME, Jeronimus BF, Albers CJ (2019). "Insight into individual differences in emotion dynamics with clustering." *Assessment*. ISSN 1073-1911. DOI: 10.1177/1073191119873714.

Fan J, Yao Q (1998). "Efficient estimation of conditional variance functions in stochastic regression." *Biometrika*, **85**(3), 645–660. DOI: 10.1093/biomet/85.3.645.

Feldman BJ, Masyn KE, Conger RD (2009). "New approaches to studying problem behaviors: A comparison of methods for modeling longitudinal, categorical adolescent drinking data." *Developmental Psychology*, **45**(3), 652–676. ISSN 1939-0599. DOI: 10.1037/a0014851.

Ferguson TS (1973). "A Bayesian Analysis of Some Nonparametric Problems." *The Annals of Statistics*, **1**(2), 209–230. ISSN 00905364. DOI: 10.1214/aos/1176342360.

Fitzmaurice GM, Laird NM, Ware JH (2011). *Applied Longitudinal Analysis*. John Wiley & Sons, Inc. ISBN 9780470380277. DOI: 10.1002/9781119513469.

Foulley JL (2004). "Including mean–variance relationships in heteroskedastic mixed linear models: theory and application." *InterStat*.

Foulley JL, Quaas R (1995). "Heterogeneous variances in Gaussian linear mixed models." *Genetics Selection Evolution*, **27**(3), 211. ISSN 1297-9686. DOI: 10.1186/1297-9686-27-3-211.

Francis B, Elliott A, Weldon M (2016). "Smoothing group-based trajectory models through b-splines." *Journal of Developmental and Life-Course Criminology*, **2**(1), 113–133. DOI: 10.1007/s40865-016-0025-6.

Frankfurt S, Frazier P, Syed M, Jung KR (2016). "Using group-based trajectory and growth mixture modeling to identify classes of change trajectories." *The Counseling Psychologist*, **44**(5), 622–660. ISSN 0011-0000. DOI: 10.1177/0011000016658097.

Franklin JM, Shrank WH, Pakes J, Sanfélix-Gimeno G, Matlin OS, Brennan TA, Choudhry NK (2013). "Group-based trajectory models: A new approach to classifying and predicting long-term medication adherence." *Medical Care*, **51**(9), 789–796. ISSN 0025-7079. DOI: 10.1097/mlr.0b013e3182984c1f.

Frühwirth-Schnatter S (2006). *Finite mixture and Markov switching models*. Springer Science & Business Media, New York, 1 edition. ISBN 978-0-387-32909-3. DOI: 10.1007/978-0-387-35768-3.

Fulcher BD, Jones NS (2014). "Highly comparative feature-based time-series classification." *IEEE Transactions on Knowledge and Data Engineering*, **26**(12), 3026–3037. ISSN 1041-4347. DOI: 10.1109/tkde.2014.2316504.

Fulcher BD, Little MA, Jones NS (2013). "Highly comparative time-series analysis: The empirical structure of time series and their methods." *Journal of The Royal Society Interface*, **10**(83). ISSN 1742-5689. DOI: 10.1098/rsif.2013.0048.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013). *Bayesian Data Analysis*. New York: Chapman and Hall/CRC. ISBN 9781439898208. DOI: 10.1201/b16018.

Gelman A, Rubin DB, *et al.* (1992). "Inference from iterative simulation using multiple sequences." *Statistical Science*, **7**(4), 457–472. DOI: 10.1214/ss/1177011136.

Genolini C, Alacoque X, Sentenac M, Arnaud C (2015). "kml and kml3d: R packages to cluster longitudinal data." *Journal of Statistical Software*, **65**(4), 1–34. ISSN 1548-7660. DOI: 10.18637/jss.v065.i04.

Genolini C, Falissard B (2010). "KmL: *k*-means for longitudinal data." *Comput. Statist.*, **25**(2), 317–328. ISSN 0943-4062. DOI: 10.1007/s00180-009-0178-4.

Gompertz B (1820). "A Sketch of an Analysis and Notation Applicable to the Estimation of the Value of Life Contigencies." *Philosophical Transactions of the Royal Society of London*, **110**, 214–294. DOI: 10.1098/rstl.1820.0018.

Green MJ (2014). "Latent class analysis was accurate but sensitive in data simulations." *Journal of Clinical Epidemiology*, **67**(10), 1157–1162. ISSN 0895-4356. DOI: 10.1016/j.jclinepi.2014.05.005.

Green PJ, Richardson S (2001). "Modelling heterogeneity with and without the Dirichlet process." *Scandinavian journal of statistics*, **28**(2), 355–375. DOI: 10.1111/1467-9469.00242.

Greenberg DF (2016). "Criminal careers: Discrete or continuous?" *Journal of Developmental and Life-Course Criminology*, **2**(1), 5–44. DOI: 10.1007/s40865-016-0029-2.

Grimm KJ, Mazza GL, Davoudzadeh P (2017). "Model selection in finite mixture models: A k-fold cross-validation approach." *Structural Equation Modeling: A Multidisciplinary Journal*, **24**(2), 246–256. DOI: 10.1080/10705511.2016.1250638.

Grimm KJ, Ram N, Estabrook R (2010). "Nonlinear Structured Growth Mixture Models in M plus and OpenMx." *Multivariate behavioral research*, **45**(6), 887–909. DOI: 10.1080/00273171.2010.531230.

Grün B, Leisch F (2008). "FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters." *Journal of Statistical Software*, **28**(4), 1–35. ISSN 1548-7660. DOI: 10.18637/jss.v028.i04.

Gude T, Odd EH (2000). "More than one way to change: A study of course heterogeneity during and after short-term psychiatric in-patient treatment." *Scandinavian Journal of Psychology*, **41**(2), 91–100. ISSN 1467-9450. DOI: 10.1111/1467-9450.00176.

Hamaker EL (2012). In MR Mehl, TS Conner (eds.), "Handbook of Research Methods for Studying Daily Life," chapter Why researchers should think "within-person": A paradigmatic rationale, pp. 43–61. The Guilford Press. ISBN 9781609187477.

Hamaker EL, Asparouhov T, Brose A, Schmiedek F, Muthén B (2018). "At the Frontiers of Modeling Intensive Longitudinal Data: Dynamic Structural Equation Models for the Affective Measurements from the COGITO Study." *Multivariate Behavioral Research*, **53**(6), 820–841. DOI: 10.1080/00273171.2018.1446819.

Hardy A (1994). "An examination of procedures for determining the number of clusters in a data set." In E Diday, Y Lechevallier, M Schader, P Bertrand, B Burtschy (eds.), "New Approaches in Classification and Data Analysis," pp. 178–185. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-51175-2.

Hardy W, Powers J, Jasko JG, Stitt C, Gary Lotz M, Aloia MS (2017). "A mobile application and website to engage sleep apnea patients in PAP therapy and improve adherence to treatment."

Harrington M, Velicer WF, Ramsey S (2014). "Typology of alcohol users based on longitudinal patterns of drinking." *Addictive Behaviors*, **39**(3), 607–621. ISSN 0306-4603. DOI: 10.1016/j.addbeh.2013.11.013.

Hartley HO, Rao JN (1967). "Maximum-likelihood estimation for the mixed analysis of variance model." *Biometrika*, **54**(1-2), 93–108.

Hastie T, Tibshirani R (1990). *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, Ltd., London, 1st edition. ISBN 0412343908.

Hastie T, Tibshirani R (1993). "Varying-coefficient models." *J. Roy. Statist. Soc. Ser. B*, **55**(4), 757–796. ISSN 0035-9246.

Hedeker D, Mermelstein RJ, Demirtas H (2012). "Modeling between-subject and within-subject variances in ecological momentary assessment data using mixed-effects location scale models." *Statistics in Medicine*, **31**(27), 3328–3336. DOI: 10.1002/sim.5338.

Heinzl F, Tutz G (2013). "Clustering in linear mixed models with approximate Dirichlet process mixtures using EM algorithm." *Statistical Modelling*, **13**(1), 41–67. DOI: 10.1177/1471082X12471372.

Hennig C (2007). "Cluster-wise Assessment of Cluster Stability." *Computational Statistics & Data Analysis*, **52**(1), 258–271. ISSN 0167-9473. DOI: 10.1016/j.csda.2006.11.025.

Hoeppner BB, Goodwin MS, Velicer WF, Mooney ME, Hatsukami DK (2008). "Detecting longitudinal patterns of daily smoking following drastic cigarette reduction." *Addictive Behaviors*, **33**(5), 623–639. ISSN 0306-4603. DOI: 10.1016/j.addbeh.2007.11.005.

Hoffman MD, Gelman A (2014). "The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." *Journal of Machine Learning Research*, **15**(1), 1593–1623. DOI: 10.5555/2627435.2638586.

Hoover DR, Rice JA, Wu CO, Yang LP (1998). "Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data." *Biometrika*, **85**(4), 809–822. ISSN 0006-3444. DOI: 10.1093/biomet/85.4.809.

Huang M, Yao W, Wang S, Chen Y (2018). "Statistical inference and applications of mixture of varying coefficient models." *Scandinavian Journal of Statistics*, **45**(3), 618–643. DOI: 10.1111/sjos.12316.

Hubert L, Arabie P (1985). "Comparing partitions." *Journal of Classification*, **2**(1), 193–218. ISSN 1432-1343. DOI: 10.1007/BF01908075.

Islam MA, Alizadeh BZ, van den Heuvel ER, Bruggeman R, Cahn W, de Haan L, Kahn RS, Meijer C, Myin-Germeys I, van Os J, Wiersma D (2015). "A comparison of indices for identifying the number of clusters in hierarchical clustering: A study on cognition in schizophrenia patients." *Communications in Statistics: Case Studies, Data Analysis and Applications*, **1**(2), 98–113. ISSN 2373-7484. DOI: 10.1080/23737484.2015.1103670.

Jara A, Hanson T, Quintana F, Müller P, Rosner G (2011). "DPpackage: Bayesian Semi- and Nonparametric Modeling in R." *Journal of Statistical Software*, **40**(5), 1–30. DOI: 10.18637/jss.v040.i05.

Jebb AT, Tay L, Wang W, Huang Q (2015). "Time series analysis for psychological research: Examining and forecasting change." *Frontiers in Psychology*, **6**, 727. ISSN 1664-1078. DOI: 10.3389/fpsyg.2015.00727.

Jensen HH, Mortensen EL, Lotz M (2014). "Heterogeneity of treatment changes after psychodynamic therapy within a one year follow-up: A replication study." *Scandinavian Journal of Psychology*, **55**(2), 168–179. ISSN 0036-5564. DOI: 10.1111/sjop.12104.

Jones BL, Nagin DS (2007). "Advances in group-based trajectory modeling and an SAS procedure for estimating them." *Sociol. Methods Res.*, **35**(4), 542–571. ISSN 0049-1241. DOI: 10.1177/0049124106292364.

Jones BL, Nagin DS (2013). "A note on a Stata plugin for estimating group-based trajectory models." *Sociological Methods & Research*, **42**(4), 608–613.

Jones BL, Nagin DS, Roeder K (2001). "A SAS procedure based on mixture models for estimating developmental trajectories." *Sociol. Methods Res.*, **29**(3), 374–393. ISSN 0049-1241. DOI: 10.1177/0049124101029003005.

Jung T, Wickrama KAS (2008). "An introduction to latent class growth analysis and growth mixture modeling." *Social and Personality Psychology Compass*, **2**(1), 302–317. ISSN 1751-9004. DOI: 10.1111/j.1751-9004.2007.00054.x.

Kalpakis K, Gada D, Puttagunta V (2001). "Distance measures for effective clustering of ARIMA time-series." In "Proceedings 2001 IEEE International Conference on Data Mining," pp. 273–280. IEEE, IEEE Comput. Soc. ISBN 0-7695-1119-8. DOI: 10.1109/icdm.2001.989529.

Kendall M, Stuart A, Ord JK (1983). *The advanced theory of statistics. Vol. 3*, volume 3. Macmillan, Inc., New York, fourth edition. ISBN 0-02-847860-6.

Kendzerska T, Mollayeva T, Gershon AS, Leung RS, Hawker G, Tomlinson G (2014). "Untreated obstructive sleep apnea and the risk for serious long-term adverse outcomes: A systematic review." *Sleep medicine reviews*, **18**(1), 49–59. ISSN 1087-0792. DOI: 10.1016/j.smrv.2013.01.003.

Kim SY, Kim JS (2012). "Investigating stage-sequential growth mixture models with multiphase longitudinal data." *Struct. Equ. Model.*, **19**(2), 293–319. ISSN 1070-5511. DOI: 10.1080/10705511.2012.659632.

Kiwuwa-Muyingo S, Oja H, Walker SA, Ilmonen P, Levin J, Todd J (2011). "Clustering based on adherence data." *Epidemiologic Perspectives & Innovations*, **8**(1), 3. ISSN 1742-5573. DOI: 10.1186/1742-5573-8-3.

Klijn SL, Weijenberg MP, Lemmens P, van den Brandt PA, Lima Passos V (2017). "Introducing the fit-criteria assessment plot–A visualisation tool to assist class enumeration in group-based trajectory modelling." *Statistical methods in medical research*, **26**(5), 2424–2436. DOI: 10.1177/0962280215598665.

Knauert M, Naik S, Gillespie MB, Kryger M (2015). "Clinical consequences and economic costs of untreated obstructive sleep apnea syndrome." *World Journal of Otorhinolaryngology-Head and Neck Surgery*, **1**(1), 17–27. ISSN 2095-8811. DOI: 10.1016/j.wjorl.2015.08.001.

Komárek A (2009). "A New R package for Bayesian Estimation of Multivariate Normal Mixtures Allowing for Selection of the Number of Components and Interval-Censored Data." *Computational Statistics & Data Analysis*, **53**(12), 3932–3947. DOI: 10.1016/j.csda.2009.05.006.

Komárek A, Komárková L (2014). "Capabilities of R Package mixAK for Clustering Based on Multivariate Continuous and Discrete Longitudinal Data." *Journal of Statistical Software*, **59**(12), 1–38. DOI: 10.18637/jss.v059.i12.

Košmelj K (1986). "A two-step procedure for clustering time varying data." *The Journal of Mathematical Sociology*, **12**(3), 315–326. DOI: 10.1080/0022250X.1986.9990017.

Košmelj K, Batagelj V (1990). "Cross-sectional approach for clustering time varying data." *Journal of Classification*, **7**(1), 99–109. DOI: 10.1007/BF01889706.

Kreuter F, Muthén B (2008). "Analyzing criminal trajectory profiles: Bridging multilevel and group-based approaches using growth mixture modeling." *Journal of Quantitative Criminology*, **24**(1), 1–31. ISSN 1573-7799. DOI: 10.1007/s10940-007-9036-0.

Kribbs NB, Pack AI, Kline LR, Getsy JE, Schuett JS, Henry JN, Maislin G, Dinges DF (1993). "Effects of one night without nasal CPAP treatment on sleep and sleepiness in patients with obstructive sleep apnea." *American Review of Respiratory Disease*, **147**(5), 1162–1168. ISSN 0003-0805. DOI: 10.1164/ajrccm/147.5.1162.

Laird N (1978). "Nonparametric maximum likelihood estimation of a mixing distribution." *Journal of the American Statistical Association*, **73**(364), 805–811.

Laird NM, Ware JH (1982). "Random-effects models for longitudinal data." *Biometrics*, **38**(4), 963–974. ISSN 0006-341X. DOI: 10.2307/2529876.

Lazarsfeld PF, Henry NW, Anderson TW (1968). *Latent structure analysis*. Houghton Mifflin Boston.

Ledwina T, Mielniczuk J (2010). "Variance function estimation via model selection." *Applicationes Mathematicae*, **37**(4), 387–411. DOI: 10.4064/am37-4-1.

Lee EK, Gutcher ST, Douglass AB (2014). "Is sleep-disordered breathing associated with miscarriages? An emerging hypothesis." *Medical hypotheses*, **82**(4), 481–485. ISSN 0306-9877. DOI: 10.1016/j.mehy.2014.01.031.

Lee JY, Brook JS, Finch SJ, Brook DW (2016). "Trajectories of cigarette smoking beginning in adolescence predict insomnia in the mid thirties." *Substance use & misuse*, **51**(5), 616–624. ISSN 1082-6084. DOI: 10.3109/10826084.2015.1126747.

Lennon H, Kelly S, Sperrin M, Buchan I, Cross AJ, Leitzmann M, Cook MB, Renehan AG (2018). "Framework to construct and interpret latent class trajectory modelling." *BMJ open*, **8**(7), e020683.

Lettieri CJ, Williams SG, Collen JF, Wickwire EM (2017). "Treatment of obstructive sleep apnea: Achieving adherence to positive airway pressure treatment and dealing with complications." *Sleep Medicine Clinics*, **12**(4), 551–564. DOI: 10.1016/j.jsmc.2017.07.005.

Li F, Duncan TE, Duncan SC, Hops H (2001). "Piecewise growth mixture modeling of adolescent alcohol use data." *Structural Equation Modeling: A Multidisciplinary Journal*, **8**(2), 175–204. ISSN 1070-5511. DOI: 10.1207/s15328007sem0802_2.

Liang H, Wu H, Carroll RJ (2003). "The relationship between virologic and immunologic responses in AIDS clinical research using mixed-effects varying-coefficient models with measurement error." *Biostatistics*, **4**(2), 297–312. ISSN 1465-4644. DOI: 10.1093/biostatistics/4.2.297.

Liao TW (2005). "Clustering of time series data—a survey." *Pattern Recognition*, **38**(11), 1857–1874. ISSN 0031-3203. DOI: 10.1016/j.patcog.2005.01.025.

Liitiäinen E, Corona F, Lendasse A (2010). "Residual variance estimation using a nearest neighbor statistic." *Journal of Multivariate Analysis*, **101**(4), 811–823. DOI: 10.1016/j.jmva.2009.12.020.

Little RJ (1995). "Modeling the drop-out mechanism in repeated-measures studies." *Journal of the american statistical association*, **90**(431), 1112–1121.

Liu S (2017). "Person-specific versus multilevel autoregressive models: Accuracy in parameter estimates at the population and individual levels." *British Journal of Mathematical and Statistical Psychology*, **70**(3), 480–498. DOI: 10.1111/bmsp.12096.

Liu Y, Liu H, Zheng X (2018). "Piecewise Growth Mixture Model with More than One Unknown Knot: An Application in Reading Development." *Nonlinear dynamics, psychology, and life sciences*, **22**(4), 485–507.

Lo AY (1984). "On a class of Bayesian nonparametric estimates: I. Density estimates." *The Annals of Statistics*, **12**(1), 351–357. DOI: 10.1214/aos/1176346412.

Lo Y, Mendell NR, Rubin DB (2001). "Testing the number of components in a normal mixture." *Biometrika*, **88**(3), 767–778. DOI: 10.1093/biomet/88.3.767.

Lock EF, Kohli N, Bose M (2018). "Detecting multiple random changepoints in Bayesian piecewise growth mixture models." *Psychometrika*, **83**(3), 733–750. DOI: 10.1007/s11336-017-9594-5.

Lord E, Willems M, Lapointe FJ, Makarenkov V (2017). "Using the Stability of Objects to Determine the Number of Clusters in Datasets." *Information Sciences*, **393**, 29–46. ISSN 0020-0255. DOI: 10.1016/j.ins.2017.02.010.

Loughran T, Nagin DS (2006). "Finite Sample Effects in Group-Based Trajectory Models." *Sociological Methods & Research*, **35**(2), 250–278. DOI: 10.1177/0049124106292292.

Lu X, Huang Y (2014). "Bayesian analysis of nonlinear mixed-effects mixture models for longitudinal data with heterogeneity and skewness." *Statistics in medicine*, **33**(16), 2830–2849. DOI: 10.1002/sim.6136.

Lu Z, Song X (2012). "Finite mixture varying coefficient models for analyzing longitudinal heterogenous data." *Stat. Med.*, **31**(6), 544–560. ISSN 0277-6715. DOI: 10.1002/sim.4420.

Lu ZL, Zhang Z, Lubke G (2011). "Bayesian inference for growth mixture models with latent class dependent missing data." *Multivariate Behavioral Research*, **46**(4), 567–597. ISSN 0027-3171. DOI: 10.1080/00273171.2011.589261.

Lunn D, Spiegelhalter D, Thomas A, Best N (2009). "The BUGS project: Evolution, critique and future directions." *Statistics in Medicine*, **28**(25), 3049–3067. DOI: 10.1002/sim.3680.

Ma L, Zhang S, Yan X, Wei C (2018). "A hurdle finite mixture lognormal crash rate estimation model for addressing heterogeneous characteristics of influential factors." *Journal of Transportation Safety & Security*, **11**(5), 443–463. ISSN 1943-9962. DOI: 10.1080/19439962.2017.1419524.

MacQueen J (1967). "Some methods for classification and analysis of multivariate observations." In "Proceedings of the fifth Berkeley symposium on mathematical statistics and probability," volume 1, pp. 281–297. Oakland, CA, USA., Univ. California Press, Berkeley, Calif.

Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K (2019). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.0.

Magidson J, Vermunt J (2002). "Latent class models for clustering: A comparison with K-means." *Canadian Journal of Marketing Research*, **20**(1), 36–43.

Maleki M, Mclachlan G, Gurewitsch R, Ray M, Pyne S (2020). "A Mixture of Regressions Model of COVID-19 Death Rates and Population Comorbidities." *Statistics and Applications*, **18**, 295–306.

Malsiner-Walli G, Frühwirth-Schnatter S, Grün B (2016). "Model-based clustering based on sparse finite Gaussian mixtures." *Statistics and computing*, **26**(1-2), 303–324. DOI: 10.1007/s11222-014-9500-2.

Marcoulides KM, Khojasteh J (2018). "Analyzing longitudinal data using natural cubic smoothing splines." *Structural Equation Modeling: A Multidisciplinary Journal*, **25**(6), 965–971. DOI: 10.1080/10705511.2018.1449113.

Martin DP, von Oertzen T (2015). "Growth mixture models outperform simpler clustering algorithms when detecting longitudinal heterogeneity, even with small sample sizes." *Struct. Equ. Model.*, **22**(2), 264–275. ISSN 1070-5511. DOI: 10.1080/10705511.2014.936340.

Maruotti A (2011). "A two-part mixed-effects pattern-mixture model to handle zero-inflation and incompleteness in a longitudinal setting." *Biometrical Journal*, **53**(5), 716–734. ISSN 0323-3847. DOI: 10.1002/bimj.201000190.

Matsumoto M, Nishimura T (1998). "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator." *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **8**(1), 3–30. ISSN 1049-3301. DOI: 10.1145/272991.272995.

Matthews JW (2015). "Group-based modeling of ecological trajectories in restored wetlands." *Ecological Applications*, **25**(2), 481–491. ISSN 1051-0761. DOI: 10.1890/14-0390.1.

McLachlan G, Peel D (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. John Wiley & Sons, New York. ISBN 0471006262. DOI: 10.1002/0471721182.

McNeish D, Harring JR (2017). "The effect of model misspecification on growth mixture model class enumeration." *J. Classification*, **34**(2), 223–248. ISSN 0176-4268. DOI: 10.1007/s00357-017-9233-y.

Microsoft, Weston S (2022). *foreach: Provides Foreach Looping Construct*. R package version 1.5.2, URL https://CRAN.R-project.org/package=foreach.

Moffitt TE (2003). "Adolescence-limited and life-course-persistent antisocial behavior: A developmental taxonomy." In "Biosocial theories of crime," pp. 69–96. Routledge.

Monnahan CC, Thorson JT, Branch TA (2016). "Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo." *Methods in Ecology and Evolution*, **8**(3), 339–348. DOI: 10.1111/2041-210X.12681.

Muthén B (2004). "Latent variable analysis: Growth mixture modeling and related techniques for longitudinal data." In "The SAGE Handbook of Quantitative Methodology for the Social Sciences," pp. 346–369. SAGE Publications, Inc. DOI: 10.4135/9781412986311.n19.

Muthén B, Asparouhov T (2009). "Growth mixture modeling: Analysis with non-Gaussian random effects." In "Longitudinal data analysis," Chapman & Hall/CRC Handb. Mod. Stat. Methods, pp. 143–165. CRC Press, Boca Raton, FL. DOI: 10.1201/9781420011579.ch6.

Muthén B, Asparouhov T (2015). "Growth mixture modeling with non-normal distributions." *Stat. Med.*, **34**(6), 1041–1058. ISSN 0277-6715. DOI: 10.1002/sim.6388.

Muthén B, Asparouhov T, Hunter AM, Leuchter AF (2011). "Growth modeling with nonignorable dropout: alternative analyses of the STAR* D antidepressant trial." *Psychological methods*, **16**(1), 17.

Muthén B, Brown CH, Masyn K, Jo B, Khoo ST, Yang CC, Wang CP, Kellam SG, Carlin JB, Liao J (2002). "General growth mixture modeling for randomized preventive interventions." *Biostatistics*, **3**(4), 459–475. DOI: 10.1093/biostatistics/3.4.459.

Muthén B, Shedden K (1999). "Finite mixture modeling with mixture outcomes using the EM algorithm." *Biometrics*, **55**(2), 463–469. ISSN 0006-341X. DOI: 10.1111/j.0006-341x.1999.00463.x.

Muthén LK, Muthén BO (1998–2012). "Mplus user's guide." Seventh Edition. Los Angeles, CA: Muthén & Muthén.

Nagin D (2005). *Group-based modeling of development*. Harvard University Press, Cambridge. ISBN 978-0-674-04131-8. DOI: 10.4159/9780674041318.

Nagin DS (1999). "Analyzing developmental trajectories: A semiparametric, group-based approach." *Psychological Methods*, **4**(2), 139–157. ISSN 1082-989X. DOI: 10.1037/1082-989x.4.2.139.

Nagin DS, Jones BL, Passos VL, Tremblay RE (2018). "Group-based multi-trajectory modeling." *Statistical Methods in Medical Research*, **27**(7), 2015–2023.

Nagin DS, Land KC (1993). "Age, criminal careers, and population heterogeneity: Specification and estimation of a nonparametric, mixed Poisson model." *Criminology*, **31**(3), 327–362. ISSN 1745-9125. DOI: 10.1111/j.1745-9125.1993.tb01133.x.

Nagin DS, Odgers CL (2010a). "Group-based trajectory modeling in clinical research." *Annual Review of Clinical Psychology*, **6**(1), 109–138. ISSN 1548-5943. DOI: 10.1146/annurev.clinpsy.121208.131413.

Nagin DS, Odgers CL (2010b). "Group-based trajectory modeling (nearly) two decades later." *Journal of Quantitative Criminology*, **26**(4), 445–453. ISSN 1573-7799. DOI: 10.1007/s10940-010-9113-7.

Nagin DS, Tremblay RE (2005). "Developmental trajectory groups: Fact or a useful statistical fiction?" *Criminology*, **43**(4), 873–904. ISSN 0011-1384. DOI: 10.1111/j.1745-9125.2005.00026.x.

Nasserinejad K, van Rosmalen J, de Kort W, Lesaffre E (2017). "Comparison of criteria for choosing the number of classes in Bayesian finite mixture models." *PloS one*, **12**(1), e0168838. DOI: 10.1371/journal.pone.0168838.

Neal RM (2011). *Handbook of Markov Chain Monte Carlo*, chapter MCMC using Hamiltonian dynamics, p. 50. Chapman and Hall/CRC, New York, 1 edition. ISBN 9780429138508. DOI: 10.1201/b10905.

Nelder JA, Wedderburn RWM (1972). "Generalized linear models." *Journal of the Royal Statistical Society: Series A (General)*, **135**(3), 370–384. DOI: https://doi.org/10.2307/2344614.

Nielsen JD (2018). *crimCV: Group-Based Modelling of Longitudinal Data.* R package version 0.9.6, URL https://CRAN.R-project.org/package=crimCV.

Nielsen JD, Rosenthal JS, Sun Y, Day DM, Bevc I, Duchesne T (2014). "Group-based criminal trajectory analysis using cross-validation criteria." *Comm. Statist. Theory Methods*, **43**(20), 4337–4356. ISSN 0361-0926. DOI: 10.1080/03610926.2012.719986.

Ning L, Luo W (2018). "Class Identification Efficacy in Piecewise GMM with Unknown Turning Points." *The Journal of Experimental Education*, **86**(2), 282–307.

Nylund KL, Asparouhov T, Muthén BO (2007). "Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study." *Struct. Equ. Model.*, **14**(4), 535–569. ISSN 1070-5511. DOI: 10.1080/10705510701575396.

Papastamoulis P (2016). "label.switching: An R Package for Dealing with the Label Switching Problem in MCMC Outputs." *Journal of Statistical Software, Code Snippets*, **69**(1), 1–24. DOI: 10.18637/jss.v069.c01.

Pelleg D, Moore AW (2000). "X-means: Extending k-means with efficient estimation of the number of clusters." In "Proceedings of the Seventeenth International Conference on Machine Learning," volume 1 of *ICML 2000*, pp. 727–734. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISSN 1-55860-707-2.

Peugh J, Fan X (2012). "How well does growth mixture modeling identify heterogeneous growth trajectories? A simulation study examining GMM's performance characteristics." *Structural Equation Modeling: A Multidisciplinary Journal*, **19**(2), 204–226. ISSN 1070-5511. DOI: 10.1080/10705511.2012.659618.

Peugh J, Fan X (2013). "Modeling unobserved heterogeneity using latent profile analysis: A Monte Carlo simulation." *Structural Equation Modeling: A Multidisciplinary Journal*, **20**(4), 616–639.

Proust-Lima C, Joly P, Dartigues JF, Jacqmin-Gadda H (2009). "Joint modelling of multivariate longitudinal outcomes and a time-to-event: a nonlinear latent class approach." *Computational statistics & data analysis*, **53**(4), 1142–1154. DOI: 10.1016/j.csda.2008.10.017.

Proust-Lima C, Philipps V, Liquet B (2017). "Estimation of extended mixed models using latent classes and latent processes: the R package lcmm." *Journal of Statistical Software*, **78**(2), 1–56. ISSN 1548-7660. DOI: 10.18637/jss.v078.i02.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Raftery AE (1995). "Bayesian model selection in social research." *Sociological Methodology*, **25**, 111–163. DOI: 10.2307/271063.

Ram N, Grimm KJ (2009). "Growth mixture modeling: A method for identifying differences in longitudinal change among unobserved groups." *International Journal of Behavioral Development*, **33**(6), 565–576. DOI: 10.1177/0165025409343765.

Raudenbush SW (2005). "How do we study "what happens next"?" *The ANNALS of the American Academy of Political and Social Science*, **602**(1), 131–144. DOI: 10.1177/0002716205280900.

Regis M, Brini A, Nooraee N, Haakma R, van den Heuvel ER (2019). "The t linear mixed model: model formulation, identifiability and estimation." *Communications in Statistics - Simulation and Computation*, **0**(0), 1–25. DOI: 10.1080/03610918.2019.1694153.

Reinecke J, Meyer M, Boers K (2015). "Stage-sequential growth mixture modeling of criminological panel data." In "Dependent data in social sciences research," pp. 67–89. Springer International Publishing. DOI: 10.1007/978-3-319-20585-4_3.

Richardson S, Green PJ (1997). "On Bayesian analysis of mixtures with an unknown number of components (with discussion)." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **59**(4), 731–792. DOI: 10.1111/1467-9868.00095.

Rigby R, Stasinopoulos D (2001). "The GAMLSS project: A flexible approach to statistical modelling." In "New trends in statistical modelling: Proceedings of the 16$^{\text{th}}$ international workshop on statistical modelling," volume 337, p. 345. University of Southern Denmark.

Rigby R, Stasinopoulos D (2009). "A flexible regression approach using GAMLSS in R." *London Metropolitan University, London*.

Rigby RA, Stasinopoulos DM (2005). "Generalized additive models for location, scale and shape." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **54**(3), 507–554. DOI: https://doi.org/10.1111/j.1467-9876.2005.00510.x.

Rigby RA, Stasinopoulos MD (1996). "Mean and dispersion additive models." In W Härdle, MG Schimek (eds.), "Statistical theory and computational aspects of smoothing," pp. 215–230. Physica-Verlag HD, Heidelberg.

Rights JD, Sterba SK (2016). "The relationship between multilevel models and nonparametric multilevel mixture models: Discrete approximation of intraclass correlation, random coefficient distributions, and residual heteroscedasticity." *British Journal of Mathematical and Statistical Psychology*, **69**(3), 316–343. DOI: 10.1111/bmsp.12073.

Rodríguez CE, Walker SG (2014). "Label switching in Bayesian mixture models: Deterministic relabeling strategies." *Journal of Computational and Graphical Statistics*, **23**(1), 25–45. DOI: 10.1080/10618600.2012.735624.

Rotenberg BW, Murariu D, Pang KP (2016). "Trends in CPAP adherence over twenty years of data collection: A flattened curve." *Journal of Otolaryngology - Head & Neck Surgery*, **45**(1), 43. ISSN 1916-0216. DOI: 10.1186/s40463-016-0156-0.

Rousseeuw PJ (1987). "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *Journal of computational and applied mathematics*, **20**, 53–65.

Ruppert D (2002). "Selecting the number of knots for penalized splines." *J. Comput. Graph. Statist.*, **11**(4), 735–757. ISSN 1061-8600. DOI: 10.1198/106186002321018768.

Ryoo JH, Konold TR, Long JD, Molfese VJ, Zhou X (2017). "Nonlinear growth mixture models with fractional polynomials: an illustration with early childhood mathematics ability." *Structural Equation Modeling: A Multidisciplinary Journal*, **24**(6), 897–910. DOI: 10.1080/10705511.2017.1335206.

Saberi P, Johnson MO, McCulloch CE, Vittinghoff E, Neilands TB (2011). "Medication adherence: Tailoring the analysis to the data." *AIDS and Behavior*, **15**(7), 1447–1453. ISSN 1090-7165. DOI: 10.1007/s10461-011-9951-9.

Sardá-Espinosa A (2019). "Time-Series Clustering in R Using the dtwclust Package." *The R Journal*. DOI: 10.32614/RJ-2019-023.

Scrucca L, Fop M, Murphy TB, Raftery AE (2016). "mclust 5: clustering, classification and density estimation using Gaussian finite mixture models." *The R Journal*, **8**(1), 205–233.

Senaratna CV, Perret JL, Lodge CJ, Lowe AJ, Campbell BE, Matheson MC, Hamilton GS, Dharmage SC (2017). "Prevalence of obstructive sleep apnea in the general population: A systematic review." *Sleep medicine reviews*, **34**, 70–81. ISSN 1087-0792. DOI: 10.1016/j.smrv.2016.07.002.

Serang S, Zhang Z, Helm J, Steele JS, Grimm KJ (2015). "Evaluation of a Bayesian approach to estimating nonlinear mixed-effects mixture models." *Struct. Equ. Model.*, **22**(2), 202–215. ISSN 1070-5511. DOI: 10.1080/10705511.2014.937322.

Shapiro GK, Shapiro CM (2010). "Factors that influence CPAP adherence: An overview." *Sleep and Breathing*, **14**(4), 323–335. ISSN 1520-9512. DOI: 10.1007/s11325-010-0391-y.

Shedden K, Zucker RA (2008). "Regularized finite mixture models for probability trajectories." *Psychometrika*, **73**(4), 625–646. DOI: 10.1007/s11336-008-9077-9.

Sher KJ, Jackson KM, Steinley D (2011). "Alcohol use trajectories and the ubiquitous cat's cradle: cause for concern?" *Journal of abnormal psychology*, **120**(2), 322–335. DOI: 10.1037/a0021813.

Shiyko MP, Li Y, Rindskopf D (2012). "Poisson growth mixture modeling of intensive longitudinal data: An application to smoking cessation behavior." *Struct. Equ. Model.*, **19**(1), 65–85. ISSN 1070-5511. DOI: 10.1080/10705511.2012.634722.

Skardhamar T (2010). "Distinguishing facts and artifacts in group-based modeling." *Criminology*, **48**(1), 295–320. ISSN 0011-1384. DOI: 10.1111/j.1745-9125.2010.00185.x.

Song XY, Lu ZH (2010). "Semiparametric latent variable models with Bayesian P-splines." *J. Comput. Graph. Statist.*, **19**(3), 590–608. ISSN 1061-8600. DOI: 10.1198/jcgs.2010.09094.

Sperrin M, Jaki T, Wit E (2010). "Probabilistic relabelling strategies for the label switching

problem in Bayesian mixture models." *Statistics and Computing*, **20**(3), 357–366. ISSN 1573-1375. DOI: 10.1007/s11222-009-9129-8.

Spiegelhalter DJ, Best NG, Carlin BP, Linde A (2014). "The deviance information criterion: 12 years on." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**(3), 485–493. DOI: 10.1111/rssb.12062.

Spiegelhalter DJ, Best NG, Carlin BP, Van Der Linde A (2002). "Bayesian measures of model complexity and fit." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64**(4), 583–639. DOI: 10.1111/1467-9868.00353.

Spiegelhalter DJ, Freedman LS, Parmar MK (1994). "Bayesian approaches to randomized trials." *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pp. 357–416. DOI: 10.2307/2983527.

Stan Development Team (2020a). "RStan: the R interface to Stan." R package version 2.19.3, URL http://mc-stan.org/.

Stan Development Team (2020b). *Stan Modeling Language Users Guide and Reference Manual*. https://mc-stan.org, 2.25 edition.

Stasinopoulos DM, Rigby RA (2007). "Generalized additive models for location scale and shape (GAMLSS) in R." *Journal of Statistical Software*, **23**(7), 1–46. ISSN 1548-7660. DOI: 10.18637/jss.v023.i07.

Stasinopoulos M, Rigby B (2018). *gamlss.tr: Generating and fitting truncated 'gamlss.family' distributions*. R package version 5.1-0, URL https://CRAN.R-project.org/package=gamlss.tr.

Stasinopoulos MD, Rigby RA, Heller GZ, Voudouris V, De Bastiani F (2017). *Flexible Regression and Smoothing*. Chapman and Hall/CRC, London, 1st edition. DOI: 10.1201/b21973.

Sterba SK, Baldasaro RE, Bauer DJ (2012). "Factors affecting the adequacy and preferability of semiparametric groups-based approximations of continuous growth trajectories." *Multivariate Behavioral Research*, **47**(4), 590–634. ISSN 0027-3171. DOI: 10.1080/00273171.2012.692639.

Sugasawa S, Kubokawa T (2017). "Heteroscedastic nested error regression models with variance functions." *Statistica Sinica*, **27**, 1101–1123. DOI: 10.5705/ss.202015.0318.

Sun J, Herazo-Maya JD, Kaminski N, Zhao H, Warren JL (2017). "A Dirichlet process mixture model for clustering longitudinal gene expression data." *Statistics in Medicine*, **36**(22), 3495–3506. DOI: 10.1002/sim.7374.

Tan X, Shiyko MP, Li R, Li Y, Dierker L (2012). "A time-varying effect model for intensive longitudinal data." *Psychological Methods*, **17**(1), 61–77. ISSN 1939-1463. DOI: 10.1037/a0025814.

Todo N, Usami S (2016). "Fitting Unstructured Finite Mixture Models in Longitudinal Design: A Recommendation for Model Selection and Estimation of the Number of Classes." *Structural Equation Modeling: A Multidisciplinary Journal*, **23**(5), 695–712.

Tofighi D, Enders CK (2008). *Identifying the correct number of classes in growth mixture models*, chapter 13, pp. 317–341. Information Age Publishing, Inc, Greenwhich, CT. ISBN 978-1-59311-847-1.

Tolvanen A (2007). *Latent growth mixture modeling: a simulation study*. University of Jyväskylä. ISBN 978-951-39-2971-8.

Tong T, Wang Y (2005). "Estimating residual variance in nonparametric regression using least squares." *Biometrika*, **92**(4), 821–830. DOI: 10.1093/biomet/92.4.821.

Tong X, Kim S, Ke Z (2022). "Impact of Likelihoods on Class Enumeration in Bayesian Growth Mixture Modeling." In M Wiberg, D Molenaar, J González, JS Kim, H Hwang (eds.), "Quantitative Psychology," pp. 111–120. Springer International Publishing, Cham. ISBN 978-3-031-04572-1. DOI: 10.1007/978-3-031-04572-1_9.

Tsou TS (2011). "Determining the mean-variance relationship in generalized linear models — A parametric robust way." *Journal of Statistical Planning and Inference*, **141**(1), 197–203. ISSN 0378-3758. DOI: 10.1016/j.jspi.2010.05.029.

Twisk J, Hoekstra T (2012). "Classifying developmental trajectories over time should be done with great caution: A comparison between methods." *Journal of Clinical Epidemiology*, **65**(10), 1078–1087. ISSN 0895-4356. DOI: 10.1016/j.jclinepi.2012.04.010.

Van de Schoot R, Sijbrandij M, Winter SD, Depaoli S, Vermunt JK (2017). "The GRoLTS-Checklist: Guidelines for Reporting on Latent Trajectory Studies." *Structural Equation Modeling: A Multidisciplinary Journal*, **24**(3), 451–467. DOI: 10.1080/10705511.2016.1247646.

Van Den Bergh M, Vermunt JK (2017). "Building latent class growth trees." *Structural Equation Modeling: A Multidisciplinary Journal*, **25**(3), 331–342.

Van Den Bergh M, Vermunt JK (2019). "Latent Class Trees with the three-step approach." *Structural Equation Modeling: A Multidisciplinary Journal*, **26**(3), 481–492.

van der Nest G, Lima Passos V, Candel MJ, van Breukelen GJ (2020). "An overview of mixture modelling for latent evolutions in longitudinal data: Modelling approaches, fit statistics and software." *Advances in Life Course Research*, **43**, 100323. ISSN 1040-2608. DOI: 10.1016/j.alcr.2019.100323.

Van Dongen S (2000). "Performance criteria for graph clustering and Markov cluster experiments." *techreport INS-R0012*, Amsterdam, The Netherlands.

Van Horn ML, Smith J, Fagan AA, Jaki T, Feaster DJ, Masyn K, Hawkins JD, Howe G (2012). "Not quite normal: consequences of violating the assumption of normality in regression mixture models." *Structural Equation Modeling: A Multidisciplinary Journal*, **19**(2), 227–249. DOI: 10.1080/10705511.2012.659622.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC." *Statistics and computing*, **27**(5), 1413–1432.

Vendramin L, Campello RJ, Hruschka ER (2010). "Relative clustering validity criteria: A comparative overview." *Statistical analysis and data mining: the ASA data science journal*, **3**(4), 209–235.

Verbeke G, Lesaffre E (1996). "A linear mixed-effects model with heterogeneity in the random-effects population." *Journal of the American Statistical Association*, **91**(433), 217–221. ISSN 0162-1459. DOI: 10.2307/2291398.

Verbeke G, Molenberghs G (2000). *Linear mixed models for longitudinal analysis*. New York: Springer-Verlag.

Verboon P, Pat-El R (2022). "Clustering Longitudinal Data Using R: A Monte Carlo Study." *Methodology*, **18**(2), 144–163. DOI: 10.5964/meth.7143.

Verhulst PF (1845). "Recherches mathématiques sur la loi d'accroissement de la population." *Nouveaux mémoires de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles*, **18**, 14–54.

Vermunt JK (2010). "Latent class modeling with covariates: Two improved three-step approaches." *Political analysis*, **18**(4), 450–469.

Vermunt JK, Magidson J (2002). "Latent class cluster analysis." *Applied latent class analysis*, **11**, 89–106.

Vermunt JK, Magidson J (2016). "Technical guide for Latent GOLD 5.1: Basic, advanced, and syntax." Belmont, MA: Statistical Innovations Inc.

Von Luxburg U, Williamson RC, Guyon I (2012). "Clustering: Science or art?" In "Proceedings of ICML Workshop on Unsupervised and Transfer Learning," pp. 65–79.

Walls TA, Schafer JL (2006). *Models for intensive longitudinal data*. Oxford University Press, Oxford, New York, NY. DOI: 10.1093/acprof:oso/9780195173444.001.0001.

Wang CP, Hendricks Brown C, Bandeen-Roche K (2005). "Residual diagnostics for growth mixture models: Examining the impact of a preventive intervention on multiple trajectories of aggressive behavior." *Journal of the American Statistical Association*, **100**(471), 1054–1076.

Wang X, Mueen A, Ding H, Trajcevski G, Scheuermann P, Keogh E (2013). "Experimental comparison of representation methods and distance measures for time series data." *Data Min. Knowl. Discov.*, **26**(2), 275–309. ISSN 1384-5810. DOI: 10.1007/s10618-012-0250-5.

Wang X, Smith K, Hyndman R (2006). "Characteristic-based clustering for time series data." *Data Min. Knowl. Discov.*, **13**(3), 335–364. ISSN 1384-5810. DOI: 10.1007/s10618-005-0039-x.

Wang Y, Geater AF, Chai Y, Luo J, Niu X, Hai B, Qin J, Li Y (2015). "Pre- and in-therapy predictive score models of adult OSAS patients with poor adherence pattern on nCPAP therapy." *Patient Preference and Adherence*, **9**, 715–723. ISSN 1177-889X. DOI: 10.2147/ppa.s83105.

Watanabe S (2009). *Algebraic Geometry and Statistical Learning Theory*. Cambridge University Press, Cambridge, UK. ISBN 9780511800474. DOI: 10.1017/CBO9780511800474.

Weaver TE, Grunstein RR (2008). "Adherence to continuous positive airway pressure therapy: the challenge to effective treatment." *Proceedings of the American Thoracic Society*, **5**(2), 173–178. ISSN 1546-3222. DOI: 10.1513/pats.200708-119mg.

Weaver TE, Kribbs NB, Pack AI, Kline LR, Chugh DK, Maislin G, Smith PL, Schwartz AR, Schubert NM, Gillen KA, Dinges DF (1997). "Night-to-night variability in CPAP use over the first three months of treatment." *Sleep*, **20**(4), 278–283. ISSN 1550-9109. DOI: 10.1093/sleep/20.4.278.

Weaver TE, Maislin G, Dinges DF, Bloxham T, George CFP, Greenberg H, Kader G, Mahowald M, Younger J, Pack AI (2007). "Relationship between hours of CPAP use and achieving normal levels of sleepiness and daily functioning." *Sleep*, **30**(6), 711–719. ISSN 0161-8105. DOI: 10.1093/sleep/30.6.711.

Wei Y, Tang Y, Shireman E, McNicholas PD, Steinley DL (2017). "Extending Growth Mixture Models Using Continuous Non-Elliptical Distributions." *arXiv preprint arXiv:1703.08723*.

Weybright EH, Caldwell LL, Ram N, Smith EA, Wegner L (2016). "Trajectories of adolescent substance use development and the influence of healthy leisure: A growth mixture modeling approach." *Journal of Adolescence*, **49**, 158–169. ISSN 0140-1971. DOI: 10.1016/j.adolescence.2016.03.012.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2nd edition. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wickwire EM, Lettieri CJ, Cairns AA, Collop NA (2013). "Maximizing positive airway pressure adherence in adults: A common-sense approach." *Chest*, **144**(2), 680–693. ISSN 0012-3692. DOI: 10.1378/chest.12-2681.

Wishart J (1938). "Growth-rate determinations in nutrition studies with the bacon pig, and their analysis." *Biometrika*, **30**(1/2), 16–28. ISSN 0006-3444. DOI: 10.2307/2332221.

Wohlgemuth WK, Chirinos DA, Domingo S, Wallace DM (2015). "Attempters, adherers, and non-adherers: latent profile analysis of CPAP use with correlates." *Sleep medicine*, **16**(3), 336–342. DOI: 10.1016/j.sleep.2014.08.013.

Yang J, Shao M, Cai G (2019). "On the performance of MixTVEM: a simulation study." *Communications in Statistics - Simulation and Computation*, **48**(9), 2830–2844. DOI: 10.1080/03610918.2018.1468458.

Ye M, Lu ZH, Li Y, Song X (2019). "Finite mixture of varying coefficient model: Estimation and component selection." *Journal of Multivariate Analysis*, **171**, 452–474. DOI: 10.1016/j.jmva.2019.01.013.

Yi H, Dong X, Shang S, Zhang C, Xu L, Han F (2022). "Identifying longitudinal patterns of CPAP treatment in OSA using growth mixture modeling: Disease characteristics and psychological determinants." *Frontiers in Neurology*, **13**, 1063461. DOI: 10.3389/fneur.2022.1063461.

You K (2018). *mclustcomp: Measures for Comparing Clusters*. R package version 0.3.1, URL https://CRAN.R-project.org/package=mclustcomp.

Zhu H, Luo S, DeSantis SM (2017). "Zero-inflated count models for longitudinal measurements with heterogeneous random effects." *Statistical methods in medical research*, **26**(4), 1774–1786. DOI: 10.1177/0962280215588224.

Summary

On approaches for clustering longitudinal data, with extensions for modeling therapy adherence of sleep apnea patients

Longitudinal data comprises repeated measurements over time of subjects or other independent sources. Longitudinal studies are commonly used in domains such as sociology, psychology, medicine, and ecology. For example, researchers may be interested to learn how patients change their level of adherence to a treatment over time. Using longitudinal data measured from many subjects, researchers can discern between the natural differences between subjects and the subject-specific variability between their measurements over time. Historically, the collection of longitudinal data has been difficult and costly, resulting in studies with a few fixed moments of measurement for a limited number of subjects. However, with the improved data collection and storage capabilities, longitudinal datasets are becoming larger, both in terms of the number of subjects, and the number of measurements per subject. This growing volume of data presents new analysis opportunities. Having more measurement moments enables temporal characteristics to be modeled in more detail, and having a greater number of subjects allows for a more detailed exploration of population heterogeneity. In this thesis, we investigated different approaches to clustering longitudinal data, and we proposed extensions to modeling therapy adherence and counts data.

Clustering longitudinal data is a flexible way to explore differences in changes over time within a population. Subjects are grouped based on the similarity of their longitudinal characteristics such as the expected average over time, thereby representing the population in terms of a manageable number of clusters. This thesis was motivated by the study of daily therapy adherence in sleep apnea patients on positive airway pressure (PAP) therapy. Each patient follows the therapy in a unique way due to a mix of factors, including behavioral, therapy-related, support and environmental factors. For example, patients may differ considerably in their number of treated days, mean hours of usage, changes over time, day-to-day variability, and other longitudinal aspects. To better understand the common ways in which patients follow their therapy, we explore a more detailed longitudinal representation of patient adherence and population heterogeneity. A second case study of interest is the spread of COVID-19 across counties in the United States of America. Here, clustering the weekly number of new cases helps to group counties based on similarities in the development over time. This allows for the discovery of discrepancies between geographical regions and may provide policy makers with guidance on which regions to enact a specific set of policies.

In this thesis, we review and compare various approaches to clustering longitudinal data

and provide recommendations on the applicability of certain methods. We have created software that features a general framework for clustering longitudinal data, supporting the common approaches and methods. Secondly, we propose extensions to a model-based approach in order to account for population heterogeneity on multiple longitudinal aspects. We bring special attention to assessing and accounting for a heterogeneous mean-variance relation and the presence of subject-specific random residual variance.

In Chapter 2 we provide an overview of commonly used approaches for clustering longitudinal data. We demonstrated the selection of methods on a synthetic dataset allowing for a transparent comparison. In Chapter [comparison], we thoroughly compared a selection of methods in many scenarios, contributing to a better understanding of the strengths and limitations of the methods, and the similarities between them. Growth mixture modeling was found to be preferred in the simulation study and case study, with a feature-based alternative achieving promising results on intensive longitudinal datasets. With the software that we have developed, described in Chapter 5, we have contributed to a facilitating a more standardized approach to performing longitudinal cluster analyses. This allows researchers to experiment more easily with methods from different disciplines or evaluate new methods, with minimal coding.

Chapter 4 proposes a model-based approach to modeling heterogeneity in PAP therapy adherence on multiple longitudinal aspects. We apply a hurdle modeling approach for representing skipped therapy days, and we modeled the mean and variance of daily hours of usage over time. The identified adherence profiles demonstrated the benefit of our approach, as the identified adherence profiles revealed considerable differences on all aspects. Notably, the hurdle modeling approach revealed a strong association between the likelihood of reaching adherence and skipped therapy days. Overall, PAP therapy adherence was mostly affected by skipped attempts rather than a low average hours of usage. The proposed methodology is applicable to other domains that involve the tracking a level of adherence over time.

Lastly, in Chapter 6 we explored the heterogeneity in heteroskedasticity in more detail, proposing mixture models accounting for a heterogeneous mean-variance relationship, and additional heterogeneity in the variance. Ignoring the mean-variance relation was found to only have a marginal effect on the parameter bias. However, the identification of the number of classes and the class recovery of trajectories was found to be severely impacted under scenarios with high variance. We applied the models for the analysis of weekly new COVID-19 cases across counties, showing an improved fit from the inclusion of a heterogeneous mean-variance relationship. This model allowed for a more condensed representation of the COVID-19 developments, and provided more reliable predictive intervals due to the improved modeling of the variance.

Acknowledgments

First and foremost, I would like to thank my supervisors Edwin van den Heuvel and Steffen Pauws. I am grateful for your guidance and extensive feedback on my research and writing throughout these years. It has been invaluable and has helped me to raise the bar on my work. I like to think that our combined sense of humor made our meetings all the more pleasant, especially those on the early Monday morning or late Friday afternoon. Edwin, thank you for regularly freeing up time for me in your increasingly busy calendar. You helped me stay focused, and our in-depth discussions have been invaluable. Steffen, thank you for the mentorship throughout all these years, and for helping me to not lose sight of the clinical side of my research. You have taught me a lot on clinical analyses, reporting, and writing. This has helped in shaping the dissertation to be more accessible and of interest to a broader audience. I would also like to thank Francesco Ungolo for his contributions and the technical discussions as part of Chapter 6.

I want to thank Joerg Habetha for providing me the opportunity to do a PhD within Philips Research. Also, thanks to my later department leads Sybo Dijkstra and Aleksandra Tesanovic, and project lead Henning Maass, for their support and enabling me to continue to work on my PhD alongside other work. Special thanks go to my former project lead Mareike Klee for her support and guidance throughout the years. I am grateful that I have been able to work in the sleep apnea domain all this time. It is exciting that there are still so many research topics to explore, and with even more opportunities arising as technology advances.

Many thanks go to the many colleagues that I have met over the years at Philips Research for their support, discussions, and interest in my topic. I have learned a lot from you all. Special thanks go to Jorn, Christian, Tobias, and Marco for reviewing parts of my dissertation. I would also like to thank the colleagues I have worked with at Philips Sleep & Respiratory Care. In particular, I want to thank Mike Kane and Mark Aloia for the many insightful discussions. I have always found pleasure in tackling a domain-specific problem and the challenges and caveats that come with it. To Rich Sofranko and Dave Smith, thank you for supporting my research topic.

To my loving parents Marij and George, whom without their support and encouragement I do not think that I would have persevered through these challenging years. Thank you for everything. I am also grateful to my dearest friends Thijs, Maarten, and Fabien for providing the regular yet needed distractions from PhD and work life. Mustafa, Gijs-Jan, and Enno, thanks for the insightful and engaging discussions on machine learning and algorithms on many occasions. Marjolein, thanks for your support and helping me see things in perspective. Lastly, I want to thank my friends and family who I did not mention yet for showing interest in my research and progress. It has been a consistent topic of discussion at every meet-up, and in a sense, it will feel strange yet exhilarating to having closed this significant chapter of my life.

About the author

Niek Den Teuling was born in the year 1990 in Brunssum, the Netherlands. Following the completion of a VWO pre-university education at Sint-Janscollege in Hoensbroek, the Netherlands in 2008, he commenced his studies at Maastricht University in the Netherlands. He obtained a bachelor's degree in Knowledge Engineering in 2011, and completed his master's degree in Operations Research at the same university in 2013. During his study, he followed extracurricular courses from the Artificial Intelligence master, and he has worked as a research intern at Philips Research on two occasions. During the first internship, he developed predictive models for fall risk in elderly people (Jul–Dec 2012). For his master thesis project, the second internship was on unobtrusive deep sleep stage classification (Mar–Aug 2013), for which he received the Best Thesis Award. Parts of this work were published in an international journal. Starting in October 2013, he was employed by Philips Research in Eindhoven, the Netherlands to work as a research scientist. He has been working on big data analysis and machine learning involving PAP device data of patients on sleep apnea, and risk models for heart failure readmission detection for remote patient monitoring. In collaboration with Philips Research and Eindhoven University of Technology, he formally started his PhD project in September 2015 under the supervision of prof. dr. Edwin van den Heuvel and prof. dr. Steffen Pauws. During his PhD, he continued research into PAP therapy adherence, with a focus on modeling patient heterogeneity. Parts of this effort are included in the dissertation. This has led to two publications in international journals. Niek will defend his PhD thesis at Eindhoven University of Technology on July 5, 2023.