Year : 2018

Latent Markovian Modelling and Clustering for Continuous Data Sequences

Taushanov Zhivko

Taushanov Zhivko, 2018, Latent Markovian Modelling and Clustering for Continuous Data Sequences

Originally published at : Thesis, University of Lausanne

Posted at the University of Lausanne Open Archive http://serval.unil.ch
Document URN : urn:nbn:ch:serval-BIB_D3061B3D4B572

UNIL | Université de Lausanne

Faculté des sciences sociales et politiques

Institut des sciences sociales

Latent Markovian Modelling and Clustering for Continuous Data Sequences

Thèse de doctorat

présentée à la

Faculté des sciences sociales et politiques
de l'Université de Lausanne

pour l'obtention du grade de
Docteur ès mathématiques appliquées aux sciences humaines et sociales

par
Zhivko Taushanov

Directeur de thèse:
professeur André Berchtold

Jury de thèse:
Docteur Jean-Philippe Antonietti, Université de Lausanne
Professeur Gilles Celeux, Université Paris-Sud
Professeur Joan-Carles Surís, Université de Lausanne

Lausanne
July, 2018

UNIL | Université de Lausanne
Faculté des sciences
sociales et politiques

IMPRIMATUR

Le Décanat de la Faculté des sciences sociales et politiques de l'Université de Lausanne, au nom du Conseil et sur proposition d'un jury formé des professeurs

- André BERCHTOLD, directeur de thèse, Professeur à l'Université de Lausanne
- Jean-Philippe ANTONIETTI, Maître d'Enseignement et de Recherche à l'Université de Lausanne
- Gilles CELEUX, Professeur à l'Université Paris Sud
- Joan-Carles SURIS, Professeur à l'Université de Lausanne

autorise, sans se prononcer sur les opinions du candidat, l'impression de la thèse de Monsieur Zhivko TAUSHANOV, intitulée :

« Latent Markovian modelling and clustering of continuous data sequences »

Jean-Philippe LERESCHE
Doyen

Lausanne, le 11 juin 2018

Abstract

Different types of continuous longitudinal data are widely used in social sciences and other fields. These data are referred to as panel data, cohort studies, growth curves or simply time series. This thesis focuses on the modeling and clustering of continuous data sequences, using a latent-level Markovian type model. To improve the separation of the sequences on different clusters, different types of covariates can be included on the visible and on the latent level. In the strict clustering use, this model reduces to a specific type of Gaussian mixture model estimated through time, with means and variances of the components (clusters) that capture the auto-regressive dependence across time periods.

A particular attention is paid to the estimation of the model parameters and the assessment of their variability. Specific procedures are proposed for this purpose. Finally, applications of the model to various datasets are also provided and discussed.

Résumé

Différents types de données longitudinales sont fréquemment utilisés en sciences sociales et diverses autres domaines (données de panel, études de cohortes, courbes de croissances, séries temporelles). Cette thèse se concentre sur la modélisation et la classification de séquences de données continues à l'aide d'un modèle Markovien latent. L'inclusion de diverses variables aux niveaux visible et latent améliore la séparation des séquences en classes distinctes. Lorsqu'il est utilisé pour obtenir une classification stricte, ce modèle revient à une mixture Gaussienne, estimée à travers le temps et dont les moyennes et variances prennent en compte la dépendence au fil du temps.

L'estimation des paramètres du modèle et de leur variabilité sont des sujets principaux de ce travail, et des procédures spécifiques sont proposées afin de les résoudre. Plusieurs applications de ce modèle à des données réelles sont discutées afin d'illustrer la versatilité du modèle.

Acknowledgements

First, I would like to thank my supervisor, professor André Berchtold, for his support and help during my thesis. His indisputable professionalism and expertise in the field of Markov modelling, together with his great personality, make him a perfect supervisor and without his guidance and his commitment to my work, this thesis would not have been possible.

I would like to thank doctor Jean-Philippe Antonietti from the University of Lausanne, professor Joan-Carles Surís from CHUV, and professor Gilles Celeux (Université Paris-Sud) whose work in this field I deeply admire, for accepting to review and analyze my thesis, and for providing their precious feedback and suggestions that were extremely beneficial for the quality of my work. I am glad to have the opportunity to take advantage of their undeniable expertise in all different aspects of my thesis' topic.

I would also like to thank all my colleagues from LINES/LIVES and the University of Lausanne. I will remember my office-mate Serguei Rouzinov, Hannah Klaas, Maïlys Korber, Léïla Eisner, Vanessa Brandalesi, Andrés Guarin, Nadia Girardin, Pierre-Alain Roch, Julie Falcon, Gaëlle Aeby, Carolina Carvalho, Alexandre Camus, Dinah Gross, Ornella Larenza, Danilo Bolano, Matteo Antonini, Nora Dasoki, Annahita Ehsan, Ibrahima Diatta, Emanuele Politi, Emilio Visintin, for their support and very good moments during the last years.

I also need to thank my colleagues statisticians from faculty of GSE namely Jean Golay, Michael Leuenberger, Mohamed Laib and Fabian Guignard for the interesting discussions during the last years. I also would like to thank my former colleague Stéphanie Pin and Edouard Crestin-Billet, with whom I have the pleasure to work, for their trust in me.

A special thanks to my friends Eva, Juliette, Thibault, Lorène, Clotilde and Meret and as well as my old friends Igor and Evgeni and also Krasi Avramov for their support and great time during different moments of my thesis.

Last but not least, I owe this thesis, as well as every achievement in my life, to my mother Ganka who was extremely committed to every single problem I had, my father Kalcho, my brother Georgi and my grand mother Jeliazka who all motivated, supported and encouraged me during my entire life.

Contents

Chapter 1

Introduction

1.1 Clustering longitudinal data sequences

Longitudinal data are a central topic in social sciences surveys. Different types of continuous longitudinal data are widely used in many fields and even though the problematic of modelling and clustering such data can be fairly similar, often different methods are employed in each domain. Various names are also given to this type of data. In addition to continuous longitudinal data, one may discuss about developmental trajectories Jones, Nagin & Roeder [72], panel data, cohort studies Genolini and Falissard [57], growth curves Reinecke and Seddig [123] or simply time series. In some cases of time series clustering, the problematic and assumptions are very similar to the continuous longitudinal data clustering.

Finding an appropriate model to analyse a set of longitudinal data sequences is not trivial, especially when the main purpose is to perform a classification of these sequences in mutually exclusive groups, and especially when the data are continuous. In many cases, the clustering of such data is problematic, because of the scarcity of reliable and well implemented methods to cope with continuous longitudinal data in the social sciences field.

A large variety of clustering methods exist in the literature such as various hierarchical (agglomerative or divisive) and partitional Paterlini and Krink [109] (K-means etc.) algorithms, density-based, mixture Celeux and Govaert [29], and spectral methods Von Luxburg [166]. However, most of them were designed for transversal data and their use with longitudinal data is not straightforward. A common approach for clustering sequences is by seeking some dissimilarity measures that are supposed to quantify the distance between every pair of sequences. For nominal data such measures can be ap-

propriate, for instance the well-known Optimal Matching (OM) procedure described by Abbott and Tsay [1] and frequently used in social sciences.

Although OM has been successfully applied in many social sciences problems, it still suffers from several major issues. The first and most important one is its data-driven nature. The lack of modelling and the subjective choice of the cost function used to evaluate the difference between each couple of sequences make the OM a non-consistent approach. Furthermore, the lack of parameters and inference procedures make this method less useful when we attempt to explain the differences between groups at the population level. Another issue is the difficulty to take into account covariates during the clustering process: OM is only a distance measuring between observed sequences and it ignores any other characteristics of the subjects under stuy. Moreover, OM is applicable to discrete or categorical data only, whereas many datasets are continuous. Finally, regarding specifically the data analysed in social sciences, it has been argued that since the optimal matching procedure was originally proposed for DNA sequences, it may not directly translate to any other kind of data. Wu [171] takes as example a binary sequence in which the cost of the transition from unemployment to employment is the same as the cost of the opposite change if we use OM, even though in social sciences these transitions are considered as completely unequal from psychological and social perspectives. The limits of OM highlight the need for alternative approaches in the continuous sequence clustering problem.

When it comes to discriminating between *continuous* data, a popular approach is to use a continuous distance measures, such as Euclidean, Mahalanobis, Manhatan etc. However, when these measures are adapted to transversal uni- ou multi-variate data, they adapt poorly to longitudinal data. The reason is the ignorance of the time order when applying transversal measures to longitudinal data, an issue we will detail later.

The most appropriate answer to this problem appears to be the model-based clustering, which takes into account the differences between the sequences' distribution through time. The most popular models for clustering are the mixture models. However the ordinary Gaussian mixture models are still not adapted to longitudinal data for the same reasons. This is also the case with many other machine learning and other algorithms that we will mention in this thesis.

Probably the most popular model, properly adapted to clustering of continuous longitudinal data (at least in social sciences), is the Growth Mixture Models (GMM). It represents a mixture of mixed models and the clusters are usually formed according to the specific shapes of the sequences. However depending on the assumptions of the models and their adequacy to the data, GMM should not be the only choice for

researchers. We will also detail this model and provide examples.

Much less popular in social sciences are the methods of clustering of time series. Even though time series clustering is not intended to cope with the exact same problems as longitudinal data (mainly because of the difference in length, number of sequences and dimensionality), these methods could often be applied in the latter case too. Therefore we will also provide a brief overview of some of them.

An alternative consist in using models that can combine a latent and a visible part, the visible part including several models (or components), and the latent part determining which of these models corresponds best to each data sequence. The assumption is that the data distribution is not a single independent process, but rather a mixture of distributions depending on an unobserved variable that determines the current distribution, or state, of the observations. This latent variable may simply indicate the class for each observation, or, as we will see later, may include more than one component to model the same class. For the most part of this thesis however, we will be interested in the former case.

There are different types of latent variable models, but many can only handle transversal and not longitudinal data Vermunt [163]. The main model considered in this thesis belongs to the family of Hidden Markov Models Rabiner [118], whose efficacy has been proven by many applications in different fields, such as speech recognition Rabiner [117], or molecular biology Felsenstein and Churchill [52], Krogh et al. [82] and theoretical developments, such as the Double Chain Markov Model Berchtold [13]. This model named Hidden Mixture Transition Distribution Model (HMTD) is a two-level model: a visible level that models the successive observations of a sequence and a latent level that drives the visible one Berchtold [15], Bolano & Berchtold [24]. The HMTD can take into account covariates at both levels. Since each state of the latent variable implies a different modeling of the observations at the visible level, the model belongs to the class of non-homogeneous Markovian modelings.

The efficacy of the model has been illustrated first by Berchtold [15] on price inflation time series. Later Wang, Smith and Hyndman [167] applied HMTD on the Canadian lynx dataset. They use this popular dataset to compare the HMTD empirical coverages of the prediction intervals, to those obtained using the best specifications of AR, SETAR, GMTD and MAR models obtained by other authors. The results put HMTD on a par with the best among the used models.

Our aim is to present the HMTD model not only as a modelling procedure, but also as a continuous data clustering tool. It represents an alternative to the other methods with different assumptions. A big attention is paid on its complex estimation procedure.

The big problem here, is the need to be adapted to clustering and modelling simultaneously. The estimation algorithm must also support every different specification of the model described in this thesis. This issue is important from a theoretical point of view, but the implementation and the computational part are then especially complex. The algorithm needs to satisfy all possible data types, including various model specifications and covariate types on different levels.

1.2 Main contributions

The main contributions of this thesis are the following:

- First, the estimation procedure of the HMTD model using a Generalized Expectation Maximization is discussed, and a new specific optimization procedure is implemented and compared to the most known heuristic optimization alternatives.
- A general framework of model-based clustering procedure is provided and two different bootstrap implementations are proposed for parameter inference in frequentist estimation.
- Covariates are introduced at the latent level, but instead of influencing the transition matrix, they are used to improve the initial cluster membership probabilities estimation.
- The use of HMTD model for clustering of continuous sequences is illustrated with real world data examples, especially somatic trouble trajectories and internet addiction trajectories.

The thesis is organized as follows:

In Chapter 2, we introduce the concepts related to the HMTD model. We begin by introducing the Markov modelling and go through the latent variable modelling and the models that are closely related to HMTD, such as Mixture Transition Distribution, Hidden Markov Models, Double Chain Markov Models, mixture models etc. Then we present the inclusion of covariates to the model. The covariates can be included on both levels of the model: visible as well as hidden. The latent level covariates are now estimating the initial probability matrix since in (strict) clustering the transition probability matrix is diagonal. In the last Section, we introduce the particularities of the sequence clustering. Then we go through some of the most popular clustering

methods for transversal data, before focusing on the longitudinal data. Some clustering methods used in closely related fields, such as time series methods and mixed effect models, are also briefly presented in this chapter.

Chapter 3 addresses the estimation of the HMTD model. The most important algorithms, such as Forward-Backward, and Viterbi, and the likelihood function of the model are detailed. Since we focus on the frequentist estimation, the different variants of the Expectation-Maximization algorithm are also discussed. Then we discuss the difficulties in deriving the Likelihood function for some specifications of the model and mention possible ways to solve this problem. The chosen estimation algorithm of the model is discussed in details. Since it requires intense computations to optimize the likelihood function, we also discuss the existing methods and present a new optimisation procedure. At the end, this procedure is compared to the most popular heuristic optimizations through examples, including an optimization of an HMTD model likelihood. A short version of this chapter has already been published in 2017 (see Taushanov and Berchtold [152]).

In Chapter 4, we focus on the uncertainty in clustering in general, but especially when inferring the parameters of the models. The main objective is to discuss the stability and the uncertainty when estimating mixture models for clustering, and to provide alternatives for assessing the significance of the parameter estimates under the frequentist framework. We first introduce mixture models for clustering and the role of bootstrap. A general plan for clustering is proposed. Even though our aim is HMTD clustering, this plan may be followed by any model-based clustering of sequences (in the frequentist perspective). The different uncertainty types in clustering are presented. We also review the major issues related to both Bayesian and frequentist estimation of mixture models, such as the label-switching problem, and mention some of the existing solutions. At the end, we stress on the problem of estimating confidence intervals for the visible-level parameters, and we propose two different bootstrap procedures based on a previously obtained clustering solution. We also summarize different methods for clustering validation and stability assessment, and an example using somatic troubles data is presented. A part of this last example was recently accepted for publication by the Swiss Journal of Sociology Berchtold A., Surís J.-C., Meyer T. and Taushanov Z. [19].

In Chapter 5, we provide another example of the use of HMTD clustering on real data. Clustering solutions are obtained using Growth Mixture Modelling and HMTD. Both solutions are discussed and interpreted according to the data. We also provide a general comparison between GMM and HMTD in terms of methodology and we list

the differences between both models and their specificities. nternet addiction data are used for this example. A slightly different version of this chapter has been accepted for publication in the post-proceedings volume of the LaCosa II conference that was held in Lausanne in 2016 (see Taushanov and Berchtold [153]).

A concluding chapter ends the manuscript, summarizing our main findings and opening the way for further researches.

Chapter 2

The HMTD model and related concepts

In this chapter, we begin by briefly recalling a series of important concepts used throughout this thesis. Then we describe in details the main model considered here, the HMTD model. Finally, we discuss some alternative methods for the clustering of continuous data sequences.

2.1 Some important recalls

2.1.1 Markov Chains

In longitudinal data (or in time series) analysis one often attempts to model the data using the previous observations. One model that allows us to do that is the Markov chain. We start by briefly introducing the Markov property and the Markov chain in order to clarify its structure, before discussing more advanced models.

The Markov chain introduced by the Russian mathematician Andrey Markov constitutes the basis of our model framework. It is a probabilistic model that integrates the dependence between the observations of a random variable across time. Many publications describe this model in details (Boussau et al. [25], Kemeny and Snell [75], Kemeny, Snell and Knapp [76]). The stability and the structure of the Markov models are detailed by Meyn and Tweedie [99]. Even in their basic version, these models are still applied in numerous different fields such as genetics Ocone [107], music Pardo and Birmingham [108] and many others.

Let us consider a discrete variable X_t taking values in the finite set $v_t \in \{1, ..., m\}$.

We can often assume that the value at time t is influenced by all past observations of the variable. However, the Markov hypothesis considers that the conditional probability distribution of the future state of the Markov process depends only on the present state and not on any other previous state. This is called the *Markov property*.

The Markov property defines a first order Markov chain, but in general a Markov chain can be of any order. In a Markov chain of order ℓ for instance, the ℓ previous values are used to predict X_t (see Figure 2.1):

$$\begin{aligned} P\left(X_t = v_0 | X_{t-1} = v_1, \ldots, X_0 = v_t\right) &\cong P\left(X_t = v_0 | X_{t-1} = v_1, \ldots, X_{t-\ell} = v_\ell\right) \\ &= a_{v_\ell,\ldots,v_1,v_0} \end{aligned} \tag{2.1}$$

where the probability $a_{v_\ell,\ldots,v_1,v_0}$ is a part of a transition matrix A. In a 1-st order (ℓ=1) model and with m=3, the matrix A has the form:

$$A_1 = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

If we consider a 2-nd order system, still with m=3, the matrix A grows considerably:

$$A_2 = \begin{pmatrix} a_{111} & 0 & 0 & a_{112} & 0 & 0 & a_{113} & 0 & 0 \\ a_{211} & 0 & 0 & a_{212} & 0 & 0 & a_{213} & 0 & 0 \\ a_{311} & 0 & 0 & a_{312} & 0 & 0 & a_{313} & 0 & 0 \\ 0 & a_{121} & 0 & 0 & a_{122} & 0 & 0 & a_{123} & 0 \\ 0 & a_{221} & 0 & 0 & a_{222} & 0 & 0 & a_{223} & 0 \\ 0 & a_{321} & 0 & 0 & a_{322} & 0 & 0 & a_{323} & 0 \\ 0 & 0 & a_{131} & 0 & 0 & a_{132} & 0 & 0 & a_{133} \\ 0 & 0 & a_{231} & 0 & 0 & a_{232} & 0 & 0 & a_{233} \\ 0 & 0 & a_{331} & 0 & 0 & a_{332} & 0 & 0 & a_{333} \end{pmatrix}$$

In general, A is a partially sparse matrix of size $m^\ell \times m^\ell$. Therefore, the major issue with high order Markov chains is the rate of increase of the number of elements in the transition matrix as the order increases. In a matrix of m states, we have $m^{\ell+1}$ elements. As each line of the matrix is a probability distribution summing to 1, the number of parameters to estimate is $m^\ell(m-1)$ and therefore it increases exponentially with the order. This is due to the fact that each combination of the ℓ values preceding the last one has a different influence. In other words, in a Markov chain the influence of

each period's state $X_{t-\ell}$ is not independent from the states observed during the other periods $X_{(t-\ell+1)}...X_{(t-1)}$.

We must note that for simplification reasons we consider that the transition matrix remains the same for any time t, which indicates that the modelling is *homogeneous in time*. This assumption will be relaxed in some models that we will present later in this chapter.

2.1.2 Finite Mixture models

Mixture models in their simplest form represent a combination of several different distributions. The Gaussian Mixture Model (GMM) is among the most popular mixture models. In a GMM a latent variable (represented by a vector s) indicating the membership of each observation to a given distribution (component g) may be added. This latent variable has also an importance for the formulation and the estimation of the model. If μ_g and σ_g^2 are the mean and variance of each component, GMM takes the following linear combination form of Gaussian distributions:

$$p(X|\phi, \mu, \Sigma) = \sum_{g=1}^{k} \phi_g \mathcal{N}(X|\mu_g, \sigma_g^2)$$

where

$$\phi_g = p(s_g = 1)$$

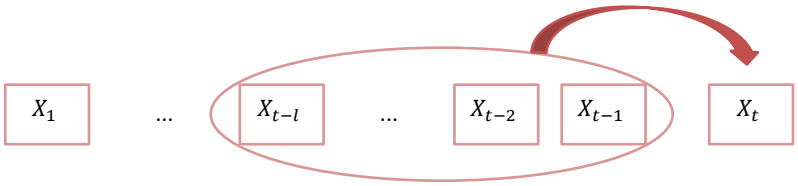
corresponds to the weight of each component and s_g is the g-th binary element of the vector s indicating the distribution membership of each observation, and

$$\sum_{g=1}^{k} \phi_g = 1, \quad \forall \phi_g \geq 0$$

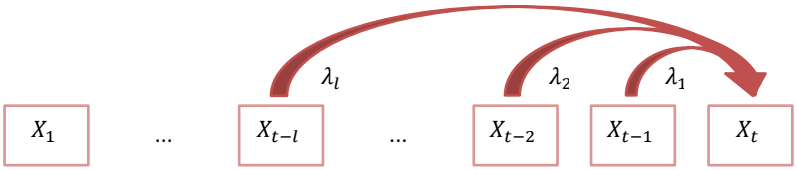
The presence of the latent allocation variable $S = s_{i i=1}^{n}$ indicating the true component that emitted x_i

$$p(x_i|s_i, \mu, \Sigma) = \mathcal{N}(x_i|\mu_{s_i}, \sigma_{s_i}^2)$$

also allows us to use the Expectation Maximisation (EM) algorithm in order to simplify the estimation of the model. Therefore the joint distribution $p(x, s)$ may be used instead of the marginal $p(x)$, and this is especially important for the maximization of the likelihood. One can then work on the individual component distributions within the mixture. More details about the estimation procedures will be provided in chapter 3.



(a) Markov chain of order ℓ: each combination of past states has its own influence on the current state



(b) MTD model of order ℓ: each past state influences the current one independently from the others

Figure 2.1: Structure of a Markov chain and a MTD model

2.1.3 The Mixture Transition Distributions model

MTD As mentioned above, the number of parameters to estimate for a Markov chain of order ℓ is considerable and it grows exponentially with the order ℓ of the chain. Therefore, the estimation time, the size of the transition matrix, and the interpretation issues of the model increase accordingly. A possible solution to this problem is to question the necessity of considering all the possible combinations (interactions) of the past observations of the variable. If one supposes that the previous observations of the variable have an impact that is independent from the other lags, a large number of parameters can be spared. Making this hypothesis transforms the Markov model into the MTD model (Mixture Transition Distribution), introduced by Raftery [119] for the modelling of high-order Markov chains, and developed later by Mehran [96] and by Berchtold [11, 12, 14, 15] among others. Berchtold & Raftery [17] also reviewed the different versions of the model.

The major benefit of this model is that it allows an approximation of high-order Markov chains with fewer parameters than the full model by considering the effect of each lag on the present value as independent from the effect of the other lags. According to this model the effect of each observation is considered separately (see Figure 2.2 and equation 2.2). Let X be a discrete random variable taking values in a finite set N=1,...,n, and consider an ℓ-th order dependence among successive observations of X. Then, the full ℓ-th order Markov chain is approximated as:

$$\begin{aligned} P\left(X_t = x_0 | X_{t-1} = x_{t-1}, \ldots, X_{t-\ell} = x_{t-\ell}\right) &= \textstyle\sum_{i=1}^{\ell} \lambda_i P_i \left(X_t = x_t | X_{t-i} = x_{t-i}\right) \\ &= \lambda_1 P(x_t = x_t | X_{t-1} = x_{t-1}) + \ldots + \lambda_\ell P(x_t = x_t | X_{t-\ell} = x_{t-\ell}) \\ &= \sum_{g=1}^{\ell} \lambda_g a_{x_{t-g} x_t} \end{aligned}$$

where a_{i_{t-g}, i_t} is an element of a transition probability matrix A of the same dimension $m \times m$ as the one of a first-order Markov chain, and that is independent from the order of dependence. A weighting parameter λ_g is associated to the g-th lag of the process. The weighting parameters are interpreted as the relative importance relative of each past period in explaining the value of the variable X measured at time t.

$$\sum_{g=1}^{\ell} \lambda_g = 1$$

The structure of the MTD model may be seen as similar to the one of an autoregressive model. Numerous generalizations applicable to different domains exist. Examples

include an infinite number of past periods, missing data inclusion, spatial models, etc.

A version of the MTD model has also been developed for *continuous* observed variables. The basic equation then becomes:

$$F\left(x_t|x_{t-1}\dots x_1\right) = \sum_{g=1}^{\ell} \lambda_g f_g\left(x_t|x_{t-g}\right) \tag{2.2}$$

The model represents then a mixture of distributions, one distribution for each lag of the model. By extension, each component of the model may use several lags of the variable X_t:

$$F\left(x_t|x_{t-\ell}\dots x_{t-1}\right) = \sum_{g=1}^{\ell} \lambda_g f_g\left(x_t|x_{t-1},\dots,x_{t-\ell}\right)$$

Various distributions may be used for each component of the model. However, we generally assume that they all follow the Gaussian distribution:

$$f_g\left(x_t|x_{t-r_g}\dots x_{t-1}\right) = \Phi\left(\frac{x_t-\mu_{g,t}}{\sigma_{g,t}}\right)$$

Some generalizations have also been proposed for the modelling of continuous time series. For instance the Gaussian MTD (GMTD) (see Le, Martin and Raftery 1996) which includes additional terms for the modelling of outliers, and the mixture autoregressive model (MAR) from Wong and Li (2000).

We must note that MTD is not the only proposed solution to reduce the number of parameters of a Markov chain. Another example is the Variable Length Markov Chains (VLMC) model detailed by Bühlmann and Wyner [26] and by Rissanen [125]. This model assumes that the number of previous values used in the modelling may change according to the state in which the chain is at time t. Some prediction algorithms have been tested in the discrete case Begleiter et al. [9]. Shmilovici et Ben-Gal applied this model to DNA sequences Shmilovici and Ben-Gal [139]. The use of VLMC for the analysis of categorical sequences in social sciences was developed by Gabadinho and Ritschard [55].

2.1.4 The Hidden Markov Model

The Markov chains and derived models such as MTD allow only to model variables that have been really observed, which limits the possibilities. However, a phenomenon

can sometimes be modelled even if it is not directly observed. We speak then of a *latent* process. The observed data are then only the visible manifestation of this phenomenon. An example is the well-being of a person that cannot be directly measured by a specific variable, but, instead, that can be represented by several observable variables such as self-reported health, living conditions, somatic symptoms, etc. In such cases, it may be better to use a more flexible generalization of Markovian models that we obtain by adding a latent variable to the model.

The Hidden Markov Model (HMM) is a two level model that has both a hidden and an observed part. Figure 2.2 (a) illustrates the dependence structure between both levels. The HMM is used to model the probability distribution of time-depending sequences of data. Consider a latent variable S_t taking values in a discrete space {1,...,k} and an observed variable X_t in a finite set {1,...,m}. A latent Markov chain with unobserved states, but whose transition matrix can be estimated, determines the states of the latent variable. Each state of the latent variable generates the observations of the visible variable X according to its distribution function. The conditional distribution of S_t depends only on its ℓ preceding states $P(S_t) = f(S_{t-1}, ..., S_{t-l})$. The observed variable X_t depends only on the corresponding latent variable value S_t. This model may be extended to a continuous variable X.

The HMM has two basic assumptions. The first one is the independence between the successive observations of X. Conditional on the state that generated the observation x_t, x_t is independent from any other observation x. The second assumption is that the latent process respects the Markov property.

To specify a standard first-order HMM, we need to estimate the probability distribution for the initial states $P(S_1)$ (vector π of length k for a first order Markov chain), the transition matrix A defining $P(S_t|S_{t-1})$ and the output model or emission probabilities $P(X_t|S_t)$. In the discrete case, $m-1$ parameters have to be estimated for each state of S, which makes a total of $k(m-1)$ emission parameters. Since these parameters and models do not change over time, the standard HMM is a *time invariant* model.

In the case of continuous data, if X is a d-dimensional vector of multivariate Gaussian distributions, one element for the mean and the standard deviation of each element of X is estimated, or in total d parameters for the mean and $\frac{d(d+1)}{2}$ for the variance-covariance matrix. In total, we have then $\frac{kd(d+3)}{2}$ emission parameters, plus $k(k-1)$ for the (first-order) latent transition matrix, and $k-1$ for the initial state probabilities.

The estimation of a HMM was detailed by Rabiner [117] in a very influential paper about the applications of HMM to voice recognition. MacDonald and Zucchini [88] also discussed similar applications. HMMs have also been used in various fields including

graphology and biology Boussau et al. [25], Felsenstein and Churchill [52].

Among other developments, Altman [4] proposed the Mixed HMM. This model includes the estimation of fixed and random effects. The author describes both frequentist and bayesian estimation, even though convergence with more than one fixed effect appears to be slow. See also Maruotti [90] for more details about this model.

Continuous state-space models

Unlike the visible part, the latent variable in a HMM is necessarily discrete. If we relax this assumption and we allow S to follow a continuous distribution, we obtain a kind of state-space model called *Kalman filter*, or alternatively a linear Gaussian state-space model that is also referred to as a linear dynamical system Bishop [23]. It was initially been developed in order to estimate the true location of an object in space (Apollo mission) and is widely used nowadays in GPS tracking applications. This framework has been developed in order to estimate a continuous time-varying variable S (for instance coordinates) that cannot be measured precisely and the observations X contain some additional noise ϵ with mean zero. As the process evolves in time, one solution is to estimate it using a weighted average in order to cancel out the independent error terms, but a much more precise way is to define a probabilistic model that captures the evolution in time of the latent variable.

The two levels of a Kalman filter may be represented as

$$S_t = HS_{t-1} + u_t$$
$$X_t = VS_t + w_t$$

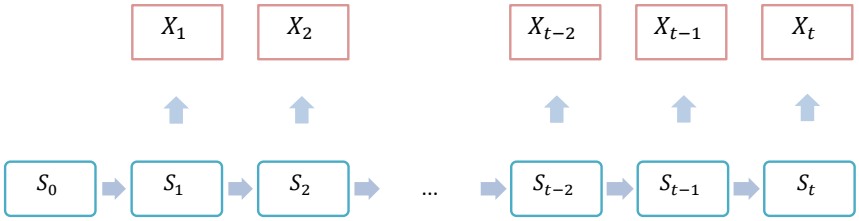
with

$$u \sim N(0, \Gamma)$$
$$w \sim N(0, \Sigma)$$

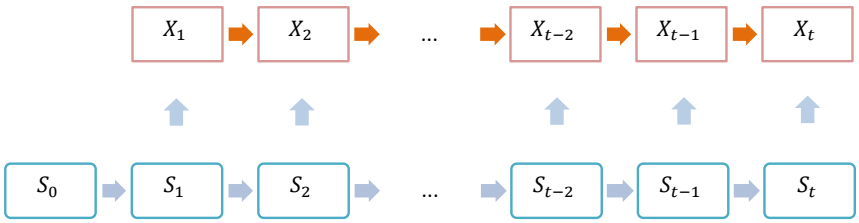
where Γ and Σ are the covariance matrices for the error terms u and w. V represents the visible level model which maps the state space on the observed space and H is responsible for the transition between the hidden states. The parameters of the Kalman filter are usually estimated by maximizing the likelihood using an EM algorithm.

2.1.5 The Double Chain Markov Model

The HMM suffers an important limitation for the representation of social phenomena: the lack of direct influence of the past observed values on the current observation of



(a) HMM



(b) DCMM

Figure 2.2: Structure of the HMM and DCMM models

X. A possible remedy is a generalized version of the HMM called the Double Chain Markov Model (DCMM) (Figure 2.2 (b)). This model contains the same elements as an HMM, but with one extension: the successive realizations of the observed variable are dependant from each other via a second Markov chain. A possible representation of this model consists in a non-homogeneous Markov model, whose transition matrix is able to evolve across time. The DCMM is especially suitable to model animal Berchtold and Sackett [18] and human Chariatte et al. [34] behaviors.

2.2 HMTD model

The Hidden Mixture Transition Distribution (HMTD) model $HMTD$, together with its estimation procedure, inference and applications, is the central topic of this thesis. Similarly to the DCMM, the HMTD model is a two-level model: a visible level that represents the sequences of successive observations (longitudinal data) and a latent level

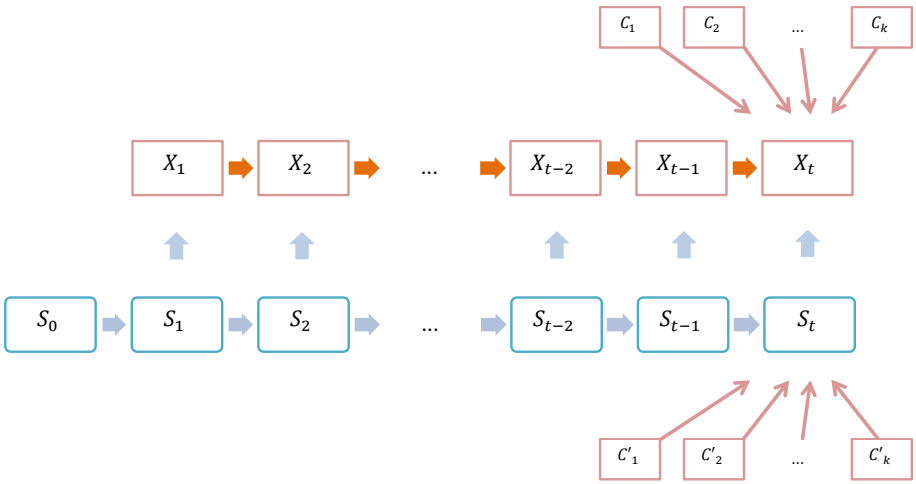


Figure 2.3: Structure of the HMTD model

that drives the visible one **?**] (Figure 2.3). The latent variable is categorical, whereas the observed one can be categorical or continuous, even if we will concentrate in this thesis on continous variables. The transitions across time in both levels are modelled by Markovian processes. Each level of the latent variable implies a different modeling of the observed process, with possibly a different number of lags and different covariates. It is therefore a kind of non-homogeneous modeling. The difference with the DCMM lies in the possibility to use a continuous variable at the observed level, and in the degree of generalization of the model, with the inclusion of various extensions that will be described hereafter.

2.2.1 The visible level

The observations of the visible variable X can be categorical, discrete or continuous. They depend on their past values by the means of a Markovian process of order $\ell > 0$ if the variable X is categorical or discrete ($\ell = 0$ is equivalent to a HMM, as there is no longer visible level modeling), or by Gaussian distributions if X is continuous. The parameters of the visible level (coefficients, number of lags, mean and variance) depend on the current latent state S_t of the hidden Markov chain. Each component (sub-

model) g can take its own p_g lags for the mean and q_g lags for the standard deviation, and $r_g = max(p_g; q_g)$ indicates the total number of lags necessary for the computation of the component g (μ_g or σ_g^2). Therefore, equation 2.2 becomes:

$$F\left(x_t|x_{t-1}\dots x_1\right) = \sum_{g=1}^{k} \lambda_g f_g\left(x_t|x_{t-1},\dots,x_{t-r_g}\right)$$

using Gaussian distributions

$$f_g\left(x_t|x_{t-1},\dots,x_{t-r_g}\right) = \Phi\left(\frac{x_t-\mu_{g,t}}{\sigma_{g,t}}\right)$$

where $\mu_{g,t}$ and $\sigma_{g,t}$ are respectively the mean and the standard deviation of the g-th component at time t.

The dependance between the successive observations is taken into account by considering the mean $\mu_{g,t}$ and possibly the standard deviation of each component as functions of the past. Thus the auto-regressive model is respected for $\mu_{g,t}$:

$$\mu_{g,t} = \phi_{g,0} + \sum_{i=1}^{p_g} \phi_{g,i} x_{t-i}$$

For $\sigma_{g,t}$ several distinct specifications have been proposed by Berchtold [15]:

$$\begin{aligned}
\sigma_{g,t} &= \sqrt{\theta_{g,0} + \sum_{j=1}^{q_g} \theta_{g,j} x_{t-j}^2}, \quad q_g \geq 1 \\
\sigma_{g,t} &= \sqrt{\theta_{g,0} + \sum_{j=1}^{q_g} \theta_{g,j} \left(x_{t-j}-\mu_{g,t}\right)^2}, \quad q_g \geq 1 \\
\sigma_{g,t} &= \sqrt{\theta_{g,0} + \sum_{j=1}^{q_g} \theta_{g,j} \left(x_{t-j}-\bar{x}_{t-q_g}^{t-1}\right)^2}, \quad q_g \geq 2
\end{aligned}$$

Wong and Li [169] also proposed to use an ARCH specification very popular in the finance field:

$$\sigma_{g,t} = \sqrt{\theta_{g,0} + \sum_{j=1}^{q_g} \theta_{g,j}\, \epsilon_{g,t-j}^2}\,, \quad q_g \geq 1$$

with

$$\epsilon_{g,t-j} = x_{t-j} - \mu_{g,t-j}$$

The modelling of the standard deviation of each component in addition to its expectation, allows to take into account the possible heteroskedasticity of the data. However, on the contrary, considering the standard deviation of each component as a constant, we can reduce the number of parameters to be estimated in the model. Note that standard deviations that are fixed as constant in time ($\sigma^2_{k,t} = \sigma^2_k$ for $t \in (1, 2, \ldots, T)$), still often imply different variances for each component: $\sigma^2_{k=1}$ not necessarily equal to $\sigma^2_{k=2}, \ldots \sigma^2_{k=K}$, in a model with K components ($\sigma^2_{k=1} \cap \sigma^2_{k=2} \cap \cdots \cap \sigma^2_{k=K} \neq \emptyset$). It is therefore important to choose between the optimal fitting of the data and the parsimony.

Besides the temporal dependence between successive observations, X can also be influenced by covariates that can be incorporated in the HMTD model at both the latent and the visible level.

2.2.2 Visible level covariates

At the visible level, the covariates are introduced in the model by adding terms to the autoregressive specification of the mean. This has already been proposed by Berchtold & Raftery [17] for the MTD model, and applied in the analysis of missed appointments in a hospital [34], but it was not yet theoretically formalised.

Consider a set of n_{cov} covariates $Y_1, \ldots, Y_n$. The mean of each visible component takes then the following form:

$$\mu_{g,t} = \phi_{g,0} + \sum_{j=1}^{p_g} \phi_{g,j} x_{t-j} + \sum_{cov=1}^{n_{cov}} \phi_{g,cov} y_{cov}$$

These covariates may be time-dependant or fixed. Numerical covariates are introduced directly into the model, when dummy variables are used for categorical covariates. Similarly, covariates could also be included in the modelling of the standard deviation, but this is more complex since it would require a set of constraints in order to keep the standard deviation non-negative.

2.2.3 The latent level

The latent part of the model is a discrete homogeneous Markov chain of any order ℓ (if $\ell = 0$, the HMTD reduces to a mixture model). The discrete latent variable S takes values in a finite set $1, \ldots, k$, where k is the number of components of the model. In the standard case of a first-order Markov chain, k is also the number of states of the hidden process. The parameters of the latent part of the HNTD model are then the

elements of the matrix of transition probabilities $A = [a_{ij}]$ between the k hidden states, and π, the matrix of initial probabilities for each state.

Whereas the visible level allows the representation of any type of observed values, the latent one is separating the observed data into a finite number of different situations (components), each of them corresponding to a distinct sub-part of the observed data. Therefore, the HMTD belongs to the class of non-homogeneous Markovian modeling. Since the latent transition matrix A is responsible for the switching between components, it deserves particular attention especially on the choice of its form which governs the use of the model. As an example, we consider here a particular situation with $k = 4$ components for the latent variable S and a first-order dependence ($\ell = 1$), but the same considerations hold for any other choice of k and ℓ.

- If we consider a free form for A i.e. without imposing any constraints during the estimation, we are in search for the optimal model that describes the data as well as possible. No hypotheses are made on A in this case. If the model requires $k > 1$ latent states (components), the data are considered as non-separable into homogeneous groups.

- If a diagonal constraint is imposed to A, the HMTD model turns into a tool for the clustering of data sequences. The identity matrix A indicates that each sequence is "trapped" into a single component for the whole observation period. The components correspond then to clusters. We must note that this precise specification of the latent part turns the model into a specific mixture model since no transition between the latent states is possible.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Another possibility is to fix the order of transition between the states. This may be done either compulsory, with no possibility to remain in the same state,

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

or gradually, by allowing the individual to either stay in the same state, or to switch to the next state at each period:

$$A = \begin{pmatrix} a_1 & 1-a_1 & 0 & 0 \\ 0 & a_2 & 1-a_2 & 0 \\ 0 & 0 & a_3 & 1-a_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In the latter case, the HMTD model is adapted to represent phenomena that evolve in time without possibility of backtracking. For example, every distinct life part (baby, child, teenager, adult, etc.) can be represented by one different latent state. Leaving the exact moment of the transition to the next state free allows for the assumption that the age of the transition is a characteristic unique to each individual, rather that a general law.

- Several groups of incompatible latent states can also co-exist by using a matrix of the following form:

$$A = \begin{pmatrix} a_1 & 1-a_1 & 0 & 0 \\ a_2 & 1-a_2 & 0 & 0 \\ 0 & 0 & a_3 & 1-a_3 \\ 0 & 0 & a_4 & 1-a_4 \end{pmatrix}$$

This structure may be used in the case of a clustering into two mutually exclusive groups of individuals, keeping a non-homogeneous modeling of observed data within each group. This may also be interpreted as a kind of *fuzzy* clustering in which it is possible for a given observed sequence to be represented by different visible models, with a non-null probability to switch anytime from the curent model to another one. In the example, the sequences are clustered in two different, but not necessarily homogeneous groups. Within each group they can still exhibit particularities by switching between the two states or not. This approach opens a possibility for simultaneously modeling and clustering the data sequences.

- The last matrix can also be modified in order to allow a transition from one group to another, but without transition in the opposite sense (one way only transition

between groups):

$$A = \begin{pmatrix} a_1 & 1-a_1 & 0 & 0 \\ a_{21} & a_{22} & 1-a_{21}-a_{22} & 0 \\ 0 & 0 & a_3 & 1-a_3 \\ 0 & 0 & a_4 & 1-a_4 \end{pmatrix}$$

This can correspond to individuals that suffered a major and irreversible event in their lives, like an accident with severe disability consequences. The first two latent states correspond then to the life before the accident, and the two others to the post accident situation.

Besides A, we also need an initial probability matrix π that determines the probability of appartenance of the first ℓ (order of the latent Markov chain) observations of each sequence to each group. In the case of a first-order matrix, π is a vector of size k with one probability for each component of the model, but in the case of higher-order models, π becomes a multidimensional array of size $(k^{\ell-1}, k, \ell)$.

The aforementioned examples of A illustrate the large flexibility of HMTD modeling and its adaptability to various situations. However, a further step can be taken by adding covariates at the latent level too.

2.2.4 Latent level covariates

One motivation for the inclusion of latent covariates is the unequal probabilities accross individuals with different characteristics to belong to a given cluster. Such is the case, for instance, for the access of men and women to high responsibility positions, or for kids from different minorities to achieve higher education in some countries. These examples clearly suggest a difference in the initial cluster probabilities based on known individual characteristics.

Since clustering is the main objective of this thesis, the matrix A is most often constrained to be diagonal, even though the other specifications discussed previously could also be useful in some cases. Therefore, the only possible influence (and the most intuitive one) of covariates on the latent part of the HMTD model is via the matrix of initial probabilities π. When A is diagonal, it is necessarily of first-order, and $\pi = [\pi_1, \ldots, \pi_k]$ is a $(1 \times k)$ vector.

In our specification of the HMTD model, the latent covariates influence the initial probabilities π via a multinomial logistic regression. For each covariate, we then need

to estimate $k-1$ additional parameters ($c \times (k-1)$ in total if c is the number of latent covariates)

$$f(g,i) = \beta_{0,g} + \beta_{1,g}x_{1,i} + \beta_{2,g}x_{2,i} + \cdots + \beta_{c,g}x_{c,i}$$

where $f(g,i)$ is a linear function that predicts the probability of sequence i to be assigned to component (or cluster) g, $\beta_{c,g}$ is the regression coefficient associated to the c-th latent covariate to explain membership to the g-th component. In matrix form, we have

$$f(g,i) = \beta_{0g} + \beta_g x_i$$

$$f(g,i) = \ln\left(\frac{P(i=g)}{P(i=ref)}\right)$$

where ref denotes the reference component chosen for the multinomial regression.

Since the individual corresponding to each observed data sequence can have its own characteristics, hence its own covariate values, a different π vector has to be estimated for each individual i with probabilities

$$\hat{\pi_{ig}} = P(i=g) = \frac{\exp(f(g,i))}{\sum_{g=1}^{g=k}\exp(f(g,i))}$$

As the use of multinomial regression allows, any type of covariates may be used (continuous, discrete as well as nominal). Indeed a non-linear relationship with covariates may also be added i.e. $\beta f(x)$ instead of βx, for example to investigate a possible quadratic effect of age. However the linearity in parameters is still respected.

The output of the latent covariates model is a $n \times k$ matrix $\hat{\pi}$ of estimated probabilities, where each line contains the probabilities of individual i to belong to each cluster. As a consequence, we do not dispose of a single vector of latent initial probabilities π, but instead we have one $\hat{\pi}_i$ for every individual. However, notice that whatever the number of individuals or sequences n, the number of parameters to be estimated through the multinomial regression is always $c(k-1)$.

We must note that if only categorical covariates are introduced at the latent level, we will have at most $\prod_{g=1}^{g=c} m_g$ distinct estimated vectors $\hat{\pi}_i$ instead of n (provided that $\prod_{g=1}^{g=c} m_c < n$), where m_g is the number of modalities of the g-th covariate, and c the total number of covariates. In other words, individuals with the exact same values on all covariates will have the same initial probability vector π. On the other hand, when at least one covariate is truly continuous, with a different value for each individual, then this case will not happen and each individual will have its own $\hat{\pi}_i$.

The advantage of the inclusion of latent-level covariates is that the reestimation of the initial probabilities π can take into account the own characteristics of each individual. In the clustering framework discussed in this thesis, this is even more crucial, since π is the only parameter at the latent level that is not fixed (A being an identity matrix in the most standard specification).

The π_i vectors of initial probabilities influence the membership of each observed sequence to a specific component of the model, but the final assignation of each sequence to the most likely component is realized through the Viterbi algorithm Viterbi [165]. This algorithm will be discussed later in the chapter dedicated to the estimation of the HMTD model.

2.2.5 Multi-sequence datasets

Even though if is perfectly possible to apply the HMTD model on a single long time series, in the context of social sciences it is more common to apply it to a large number of sequences in parallel, each one corresponding to a different individual in the sample. Moreover, when using the HMTD for the purpose of clustering, it is of course mandatory to have more than sequence. In order to obtain a sample that is representative of the population uner study, it is common to associate a sampling weights to each sequence. A correct analysis of the data requires these weights to be taken into account. In the case of HMTD, each sequence may be of a different length and it only influences the final results proportionally to this length. Therefore, if we replace the information on length by a vector of weights, it is straightforward to integrate sampling weights during the estimation of an HMTD model.

2.2.6 Choice of Gaussian distributions

One decisive advantage of the HMTD approach is that it is extremely versatile, with the possibility to work with different types of data, to model and to cluster data sequences, and to include covariates and large order time dependence into the modelling, while remaining very parsimonious. For continuous data, Gaussian distributions are generally used at the visible level, but any other continuous distribution could be chosen as well. However, following Rossi [128], there are two main reasons why Gaussian distributions are often preferred: one of them is the possibility to model univariate data as well as multivariate data, but the most important one is the ability to approximate any other distribution by a mixture of Gaussian distributions, using only a relatively small number of components.

Identifiability of the model

Generally a model is identifiable if every parameter set defines a unique density function, that is:

$$f(x, \Theta_i) = f(x, \Theta_j)$$

if and only if $\Theta_i = \Theta_j$.

In mixture models this definition includes two types of identifiability, as discussed in the literature: the one of the probability density function of the mixture, but also one of the parameters permutation (a.k.a. labelling identifiability, source of the label-switching problem). In the latter meaning of the term, the parameters of a mixture model are not identifiable because there exist $k!$ different permutations of the k components that all result in the same cumulative density function. It will be discussed later in the label-switching section.

Here we will explore the identifiability in terms of uniqueness of density of the mixture for each parameter set. This represents a major issue that comes prior to the parameter inference and answers the question whether distinct sets of parameters result in distinct mixture distributions (regardless of the parameters' order). A mixture is defined as identifiable if every distinct set of parameters ($\Theta_i \neq \Theta_j$) generates a distinct mixture density. As we deal with longitudinal data, we also need to discuss the identifiability of parameters across time. Tse and Anton [159] provide a necessary condition for identifiability of parameter Θ in this case: there exists an infinite set of non-negative numbers S such that for $t \in S$ with non-zero probability

$$P(x_t|X^{t-1}, \Theta_i) \neq P(x_t|X^{t-1}, \Theta_j)$$

where $X^{t-1} = \{x_1, x_2, \ldots, x_{t-1}\}$, $\Theta_i \neq \Theta_j$.

This concept is important because if a mixture model is identifiable, sampling from it would converge to the exact correct parametrization as the sample size increases. However, we must note that the lack of identifiability does not necessarily mean that the model is difficult to estimate.

Recall that in our case, the model that we use for clustering is a generalization of mixture of Gaussian distributions. The particularity of the model structure, compared to a Gaussian mixture, is the presence of auto-regressive dependence on past observations included in the calculation of the parameters (means and variances) of each Gaussian distribution within the mixture. Therefore we can separate the identifiability problem in two sub-problems: identifiability of a finite Gaussian mixture model and of an auto-regressive AR process.

In his paper from 1961, Teicher [154] is the first author to define the identifiability conditions for a mixture model. Later Teicher [155] (1963) also proved that a finite mixture of Gaussian distributions is identifiable (for more details see also Titterington, Smith and Makov [156] and Yakowitz and Spragins [173]).

This reduces the question about the identifiability of the model to the problem whether the parameter equations for the means $\mu_{g,t} = \varphi_{g,0} + \sum_{i=1}^{p_g} \varphi_{g,i} x_{t-i}$ and the variances $\sigma^2_{g,t} = \theta_{g,0} + \sum_{j=1}^{q_g} \theta_{g,j} x^2_{t-j}$ of each component g are identifiable.

If we take the example with the mean expression, it can be rewritten as the following $AR(p_g)$ process:

$$X_t = \sum_{i=1}^{p_g} \varphi_{g,i} X_{t-i} + \varphi_{g,0} + \epsilon_t$$

where $\varphi_{g,0}$ is a constant and ϵ_t is normally distributed with 0 mean and finite variance $\sigma^2_{g,t}$

Equivalently the process is represented as:

$$\varphi(B) X_t = \epsilon_t + \varphi_{g,0}$$

where $\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_{p_g} B^{p_g}$.

This process has a unique solution (p_g roots), which is also stationary if the roots of $\varphi(B)$ are not located on the unit circle: $|B| \neq 1$ or $\varphi(B) = 1 - \varphi_1 B \cdots - \varphi_{p_g} B^{p_g} \neq 0$.

To summarize, we have an identifiable process (AR) that determines the parameters of another identifiable model (finite Gaussian mixture), therefore the complete model is identifiable. The parameters of the Gaussian distributions (μ and σ) are not necessarily fixed and can be recalculated on each time period. Here we need to mention that the identifiability is a theoretical property of a model which does not depend on the observed data but only on the model. It states that by increasing the number of observations to infinity, the parameter estimates converge to the unique set of parameters which determine a unique density function.

Finally it is important to mention that despite the fact that the property is respected, identifiability issues may appear when over-fitting the mixture model by adding too many components. In this case either some of the components become null and do not include any observations, or a single cluster is covered by two or more similar components (see McLachlan and Peel [94]).

2.3 Alternative clustering and classification methods

2.3.1 Unsupervised versus supervised clustering

The clustering and classification of time-varying data is one of the central topics of this thesis. In this section, we provide a small overview of clustering methods that have been applied on continuous sequences. However at first a brief clarification of the distinction between clustering and classification must be done.

Put simply, a classification is the task of learning how to assign sequences to predefined classes, whereas in clustering these classes do not exist a priori and they have to be found from the data. In a machine learning perspective, one opposes *supervised* learning (classification) where a training dataset defines the classes, and *unsupervised* learning (clustering) where no data class membership is observed. In the former, one attempts to find the features of the data that match a defined partitioning output. In the latter, one attempts to find the partitioning that regroups the observations at best.

2.3.2 Sequence clustering: The longitudinal data and time series problematic

As we discuss the clustering of continuous longitudinal data, we cannot omit one very important particular case of longitudinal data: time series. Often, the problematic in time series clustering is very similar to the one of longitudinal data. In the former usually, the number of time periods t is larger than the number of sequences n, whereas in the latter $n > d$. Another belief that is generally assumed (but not necessarily always correct) is that longitudinal data are a collection of iid variables observed over time, whereas time series are observations of stochastic processes that might be dependant. Since there is no clear distinction between both cases, we will discuss the different approaches that can treat both data types, while still focussing on the ones that have also been applied to social sciences longitudinal data.

Three particular types of clustering are known in time series clustering according to Aghabozorgi, Shirkhorshidi & Wah [2]: “whole time-series clustering”, “subsequence clustering” and “time point clustering”. The first type aims to partition different series with the use of dissimilarity measures. The second type includes clustering of different smaller parts (time windows) of the same long sequence, and the time point clustering is different from “whole time-series clustering” only by the fact that some points may

be considered as noise instead of being a part of a cluster. Therefore, only the "whole time-series clustering" is applicable to our research.

On the contrary of time series, in longitudinal data, and especially in the social sciences, it is rare to cope with very long sequences, and therefore dimensionality reduction is less important than in time series clustering, except when the data are multidimensional with a very large number of possibly correlated variables. In this case, the multidimensionality would require feature selection (or extraction) prior to the partitioning of the data, but this is relatively rare in social sciences.

Another particularity of longitudinal data clustering is the frequent use in practice of well known and approved transversal methods instead of proper longitudinal ones. A possible reason may be the scarcity of methods really adapted to continuous longitudinal data, the smaller number of software implementations of such methods, or the more limited popularity of the longitudinal approaches. In the next section, we present several frequently used clustering alternatives. We also discuss the advantages and disadvantages of their use when dealing with data sequences.

2.3.3 Aims and assumptions in continuous sequence clustering

One of the particularities of continuous data clustering in practice lies in the sensitivity of the partition results with respect to the clustering method, the number of clusters, or the parameter estimates. The variety of approaches for the clustering of continuous data sequences necessarily results in large difference among results (partitions). One reason is the different underlying assumptions of each method, or in other words, the characteristics chosen by the researcher in order to define the similarity or dissimilarity between two sequences.

Even if there is no need for a dimensionality reduction to be performed, time-series (as well as longitudinal data) clustering diverges in general in function of the point of view taken by the researcher, and in function of the assumptions of the different methods regarding the notion of similarity between sequences. Aghabozorgi, Shirkhorshidi & Wah [2] distinguish three ways of time series partitioning:

- In time - One is interested in finding sequences that change together or simultaneously. Their correlation is measured during the same time periods.

- In shape - Similar paterns of evolution are researched, despite possible differences in timing. An example is the Dynamic Time Warping approach.

- In change (structural) - A typical behaviour accross time, such as an autoregressive structure for instance, is researched within each sequence. This is the case of the HMTD model clustering that we are interested in, but also of HMM, ARIMA, GARCH models, etc. A model is fitted to each sequence, and the dissimilarity between sequences is captured by the difference in the parameters that we obtain.

  Another example of "in change" partitioning is based on global features that have been extracted from every sequences Wang, Smith and Hyndman [167]. Simple features such as skewness, kurtosis, seasonality, trend etc. are extracted from the sequences and used as input for other clustering approaches (though usually transversal) such as Self-Organizing Maps (SOM) or K-means for instance. We can qualify this method as a feature extraction procedure and therefore it could be more useful in large data (in terms of number of sequences but especially in number of periods). However, covariates appear to be not usable in this procedure, what we consider as an important limitation.

Aghabozorgi, Shirkhorshidi & Wah [2] state that similarity measures based on the shape of the sequences are more often used for short sequences, whereas structure-based approaches are more appropriate for long time series. However, no objective threshold is discussed for the respective efficacy of each approaches, and there is no objective rule for choosing. Approaches based on dynamic programming appear to be most effective according to the authors.

The shape-based approaches are also based on raw (untransformed) data. They usually use distance similarity measures combined with conventional clustering methods that are adapted to transversal data (like SOM for instance). The adaptation to time-varying data is made through the selection of proper similarity measures.

2.3.4 Dimensionality reduction

Many longitudinal data studies are characterised by a large number of variables whose mutual (in)dependence is not known and must be considered. Surveys are overwhelmed with potential correlations and therefore clustering is also often combined with dimension reduction methods, i.e. feature extraction or feature selection. Besides the case when there are too many potentially not important variables that need to be reduced in order to perform proper clustering, feature extraction may be performed before a standard clustering method also in order to neutralise outliers. A Principal Component Analysis (PCA) for instance can be used to combine several variables into one, but by

doing this it also reduces the influence of the outliers when they are projected to the resulting components. A clustering on these components instead of the original variables may then be a simpler and more efficient task, with more interpretable results. Various other feature extraction or reduction techniques exist and some of them can also be used when dealing with very long sequences. However, this topic is less discussed in social sciences, and it is not central in this thesis.

2.3.5 Transversal methods

A time serie $X = X_1, \ldots, X_T$ can also be represented as a unique point in a multidimensional space, where each dimension d represents an observation of the serie at time t. Therefore, the number of dimensions represent the number of measures in time $D = T$. By making this representation, we lose the notion of sequence, which implies numerous problems especially related to the interpretation of the clusters. For example, groups can often be formed simply because they are in similar states in one or more disjoint periods, what is a non-sense in terms of interpreting the partitions. Therefore, it is important to take into account that a time serie (and longitudinal data in general) cannot be represented as simple collection of points in a T-dimensional space, because often in reality the T distinct dimensions are far from being independent. Since we do not consider the crucial information of the ordering of the measures, the time dependence between the observation is lost.

However, it is not uncommon for these methods to be used on time-dependent data, so we briefly present some examples. Some researchers are ready to sacrifice this loss of information for the sake of simplicity and the possibility to use more advanced transversal methods. This is most often done with discrete or categorical data. But there exist also a non negligible number of continuous longitudinal data clustering problems that have been treated using transversal tools. Therefore, it is necessary to briefly overview some of the frequently used transversal models.

Some of these methods make use of a given metric indicating a distance between sequences. After the distance between every pair of sequences is computed, a distance matrix is constructed and used to create clusters. Multidimensional clustering methods are often directly applied on the sequences, neglecting the time order. Depending on the approach, the distance can be measured between static points in a multivariate space, rather than between ordered time-varying sequences. An issue of such procedure is the possibility to form clusters based just on the proximity at a single time point (for instance at t=5), that can be considered as the "most discriminant dimension" of the

data. This procedure can be criticised, because it is based on distance measures that are difficult to interpret for time-dependant data, since they neglect the evolution and dependence in time. Some specific procedures are described hereafter.

Optimal matching

One of the most used algorithm for measuring the distance between data sequences, especially in the social sciences, is the Optimal Matching (OM). We already mentioned the important inconvenients of this method, but it is important to mention that it does not belon to the model-based clustering procedures, since it is a data-driven approach. Therefore, no inference on the results seems feasible. It is also more appropriate for discrete and nominal data, but applications on discretized continuous data are also frequent. Once the dissimilarities between data sequences are found, classical clustering tools may be applied.

Multidimensional scaling

Multidimensional Scaling (MDS) is an alternative method generally used on transversal data. This is a form of non-linear dimensionality reduction. It is frequently used as a data visualization technique taking a distance matrix as input and returning a coordinate matrix, using eigenvalue decomposition, which minimizes a loss function.

K-means

One of the simplest and most popular transversal clustering method is the K-means algorithm. It separates data observations into k different clusters, where each observation is associated to the cluster with the closest mean μ_j.

Even though it is mostly associated with means, what the k-means algorithm performs is essentially variance minimization. The function $F()$ (representing the sum of squared errors within the clusters) is minimised:

$$F = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (||x_i^{(j)} - \mu_j||)^2$$

where n_k is the number of data points in the k-th cluster and x are the observations. K-means (like the EM algorithm) iterates between two repetitive steps until convergence. After randomly attributing each observation to a class, one first computes the mean of each class (E-like step) and then the observations are re-assigned to the nearest cluster

by minimizing the distance to the cluster means (M-like step). The algorithm stops when no more observation moves from one class to another (see Bishop [23]) .

Though being basically a transversal data-driven method, a version of k-means (called KmL) was developed for longitudinal data clustering by Genolini and Falissard [57] in 2010. However this method seems to ignore the time sequencing of the observations, by computing the Euclidean or Manhatan distance between sequences. Thus we return to the situation in which a sequence is viewed as a single point in a multidimensional space, with all its related issues. Moreover, this approach can also be criticised, since it uses the Gower adjustment of the Euclidian distance, and this adjustment simply ignores the time periods where missig observations occurs:

$$D_{Gower}(x_{it}, x_{jt}) = \sqrt{\frac{1}{\sum \omega_{ijt}} \sum_{t=1}^{T} (x_{it} - x_{jt})^2 * \omega_{ijt}}$$

where $\omega_{ijt} = 0$ if one of the sequences (x_i or x_j) in unobserved on time t, and 1 otherwise. That leads to a clustering based only on a part of the available data.

Neural networks and self organizing maps

Neural Networks, inspired by the central nervous system in biology, are statistical learning models that are mainly used in machine learning and particularly in supervised learning where we usually need to observe some output before using the model. Therefore, they are more suited for classification rather than for clustering. Some related models, such as Self Organizing Maps (SOM) invented by Kohonen [81] and Adaptive Resonance Theory, were however developed to perform clustering. The latter has also been applied to the clustering of time-varying data Tomida, Hanai, Honda & Kobayashi [157].

SOM are a tool for the visualization of high-dimensional data, and it attempts to "convert complex non-linear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display" Kohonen [81]. Therefore, it is a dimensionality reduction tool (most often on two dimensions) and it relates to a non-parametric regression model.

A distance measure is used by SOM in order to select the best node (best matching unit). This is again an important part of the procedure which indicates its perfect adaptation to multivariate data, but also an inconvenient when dealing with time-varying sequences. Nevertheless, SOM has been applied for time-varying data clustering, for example by Cherif, Cardot & Boné [35] and Sarlin [133]. The conclusion of these au-

thors was that the results depend on the seasonality and on the characteristics of the series.

2.3.6 Longitudinal methods

Even if they are less known than models for the clustering of transversal data, there exist clustering models that are adapted to time-varying continuous data. Even though some of them were developed in other fields and are not popular in social sciences, we introduce now several of these models and we discuss briefly their advantages and disadvantages.

Functional clustering

The basic idea of Functional Data Analysis (FDA) is to represent each data sequence by a smooth function. Functional data techniques are also often applied to time series Pérez, Cao & Vilar-Fernández [110]. Each sequence X_t is seen as a curve and it is expressed as a linear combination of basis functions b_f (like splines for example):

$$X_{i,t} = \sum_f c_{i,t} b_f(t) + \epsilon_{i,t}$$

Each sequence is approximated using traditional methods. For the sake of computational simplicity, the linear methods are generally preferred Rossi [127]. The analysis is then performed on the functional representations, instead of on the sequences of original observations.

Chiou and Li [36] group longitudinal data using functional clustering (a method they called k-centres functional clustering). As a functional analysis method, the aim is to find a particular shape of trajectory, or a representative curve pattern, that fits well as many sequences as possible. At the beginning, data curves are fitted using spline approximations. One attempts to find the cluster to which an observed curve x_i belongs to by minimizing the distance to the cluster curve (truncated Karhunen-Loève expansion $\tilde{x}^k$ with its own mean and covariance structure) over all clusters $k \in \{1, \ldots, k\}$:

$$\hat{k}(x_i) = argmin||x_i - \tilde{x}^k||$$

Functional methods provide interesting results in long time series, but they are less used when the length of the sequences is small, because in this case the possibility of over-fitting is important. This problem is also very well known in spline approximation,

when the bandwidth is reduced, and this is also the case when we work with short sequences which are typical in social sciences longitudinal surveys.

A popular approach that can be applied in FDA are the *wavelet transforms* that can also be used in time series clustering. Applications of wavelets for clustering exist in the recent literature such as Antoniadis et al. [5]. The clustering strategy they implemented consists in four steps. At first the data are preprocessed and paths are estimated. Then the feature extraction is performed and the most relevant features are selected. The number of clusters is chosen and, finally, the sequences are clustered using a k-means algorithm with the selected features.

Other examples are provided by Song et al. [142] who first determine the functional principal components (FPCs) using basis function expansions, then perform a clustering based on the FPC scores, while Leng and Müller [85] represent the expression profiles using a linear combination of FPCs and perform a functional logistic regression of the scores to classify the expression profiles into groups. The main problem lies in the choice of the number of basis functions and of the nots (the joint points of these functions), which introduces the problem of under- or over-fitting the data.

As a kind of summary, Jacques and Preda [68] separate functional data clustering methods into four groups:

1. In *raw-data methods*, one does not need to reconstruct the functional form of the data since the function is considered to be directly observed on a large number of points. However no place for observation errors is left in these methods and they are not recommended by the authors.

2. In *filtering methods*, the curves are first approximated into a finite basis of functions (a form of dimension reduction) using splines or functional PCA for instance. Then, a clustering (such as k-means or SOM) is performed using the resulting parameters or coefficients.

3. The *adaptive methods* perform simultaneously dimensionality reduction of the curves and clustering, leading to a functional representation of data depending on clusters. Instead of taking the coefficients on the last point as fixed, they are considered as random variables following a given distribution that is proper to each cluster. A probabilistic modelling either on expansion coefficients or on functional PCA scores is performed.

4. In *distance-based methods*, the clustering methods use measures of distance or

dissimilarity between two functions. Depending on these measures, a relation with the first two methods is possible.

Dynamic time warping

Very popular in speech recognition, Dynamic Time Warping (DTW) is an algorithm for measuring the similarity between two time series (discrete or continuous). A particularity of this method is that the similarity may vary in speed between the series. For instance, a slow or a fast pronunciation of the same sentence are equally recognisable.

If we take an example of two sequences $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., x_m\}$, in order to align both series using DTW, one first constructs an $n * m$ matrix D of squared distances where the element $D_{i,j}$ indicates the distance between the corresponding points $d(x_i, y_j)$. If we name p a given path from $d(1, 1)$ to $d(n, m)$ defined as $p = \{p_1, ..., p_l, ..., p_L\}$, $p_l = d(x_{n_l}, y_{m_l})$, $L = n + m - 1$, the DTW would correspond to the path trough the distance matrix that minimises its total distance:

$$DTW = \min_p \{\sum_{i=1}^{L_p} d_{n_i,m_i}\}, \quad p \in P^{nm}$$

where P is the set of different paths trough the matrix D. If we see the distance path as a cost function, the aim of the DTW is to find the optimal alignment between a pair of sequences by minimizing the cost function. Therefore, the path trough the lowest values ("valleys") of the distance matrix is found. This indicates that one finds the smallest distances, even if the time indexes of one of the series are delayed. Being a distance measure, DTW needs to be combined with a clustering algorithm (such as the k-nearest-neighbour for instance) in order to cluster sequences.

Mixed effect analysis models

Coffey, Hinde and Holian [38] use a mix between functional analysis and mixture model called *curve-based clustering methods* for time-course gene expression data. In a first step, they attempt to smooth the sequences in order to eliminate the noise from the measurement and to recover the missing data. A spline regressions approach (penalized spline) is chosen by the authors and a penalization of the curvature is used to prevent over-fitting. Then, they use mixtures of mixed effect models for the clustering of the smoothed series. In this case, each model includes a given time effect. However, more complex non-linear time effects could be difficult to capture by this approach.

Growth mixture models

GMM Model-based clustering is very popular since several decades. In particular, different variants of mixtures of linear models have been applied for the clustering of longitudinal data Ciampi et al. [37]. For instance, Celeux et. al [31] and McNicholas and Murphy [95] applied mixtures of linear mixed models to cluster gene expression data. Celeux et. al [31] chose the LMM in order to take into account data variability and considered a mixture of LMM in order to cluster the gene sequences. McNicholas and Murphy used a modified Gaussian Mixture model (the "expanded parsimonious Gaussian mixture model (EPGMM) family") in order to obtain a specific covariance structure. However, the most popular example of mixture modelling (especially used in longitudinal data studies in the social sciences, but also in medical research etc.) are the Growth mixture models (GMM [1]). In order to evaluate the performance of the HMTD approach as a tool for clustering sequences of continuous data, a gold standard alternative has to be used, and the GMM approach appears to be the most obvious choice especially in the domain of social sciences.

Growth modelling includes many often similar models that aim to discover the patterns of (and model) individual change in a longitudinal data framework Reinecke and Seddig [123] McArdle and Epstein [92]. Basic growth models assume that all trajectories belong to the same population and may be approximated by a single average growth trajectory using a single set of parameters. There exist several models with similar assumptions. Such are the latent class growth analysis LCGA which assumes null variance-covariance for the growth trajectory within each class Nagin [105]Jung and Wickrama [74], and the heterogeneity model Verbeke and Lesaffre [161] which goes a bit further, but still imposes the same variance-covariance structure within each group of subjects. Therefore, the more flexible GMM will be discussed in this section and used later on in our analysis as gold-standard.

The GMM developed by Bauer [8], Muthén and Shedden [104], Wang and Bodner [168] is a model designed to discover and describe the unknown groups of sequences that share a similar pattern. This method may be represented as a *mixture of mixed-effects models* and each of the unknown sub-populations follows a distinct linear mixed effect model. Its main advantage over other similar models (like the heterogeneity model Verbeke and Lesaffre [161]) is that it allows for the estimation of a specific variance-covariance structure within each class Francis and Liu [53]. Within-class inter-

[1]Note: from now on **GMM** denotes **Growth Mixture Models** and no longer Gaussian Mixture Models

individual variation is allowed for the latent variables via distinct intercept and slope variances, represented by a class-specific fixed effects and random effects distribution. In other words, the variation around the group-specific expected trajectory is distinct for each group, which implies heterogeneity in the growth trajectories. These advantages made the model a reference in the continuous longitudinal data modelling with various applications in criminology Francis and Liu [53], Reinecke and Seddig [123], health and medicine Muthén and Shedden [104], Ram and Grimm [121], psychology and social sciences Muthén [102] among others.

The GMM contains two parts and uses both observed and latent variables. The observed ones consist in a p-dimensional vector of continuous dependant variables X (often a variable with repeated measurements) and a q-dimensional vector of covariates Y. The latent variables are represented as a continuous m-dimensional vector η. Finally, to indicate the group into which each subject is classified, a dummy variable with multinomial distribution is used and stored in a k-dimensional binary vector c Muthén and Shedden [104]. The equation of the GMM for an individual i is then

$$X_i = \Lambda \eta_i + \epsilon_i \tag{2.3}$$

where Λ is a $p \times m$ parameter matrix (or matrix with basis vectors) that can be seen as a matrix of factor loadings; η_i is a vector of latent continuous variables and ϵ_i is the error term vector with zero mean.

In our case, the matrix Λ of latent variable parameters has one column with parameters for the latent factor accounting for the intercept and another one with parameters for the latent factor accounting for the slope.

The general equation for every η is:

$$\eta_i = Ac_i + \Gamma y_i + \zeta_i \tag{2.4}$$

where Γ is a $m \times q$ parameter matrix, ζ_i is a $m \times 1$ vector of zero mean residuals (and covariance matrix Ψ). A is a matrix containing columns of intercept parameters for each class and c_i is a vector of dummy variables indicating which latent group the sequence i belongs to.

If we make the assumption that some time-independent covariates z could influence the distribution of the group membership c_i, a multinomial logistic regression is considered (with parameters a and b):

$$P(c_i = k|z_i) = \frac{\exp^{a_k + b_k z_i}}{\sum_{c=1}^{k} \exp^{a_c + b_c z_i}}$$

An alternative notation of the first part of the model for a subject i being part of class k at time t is:

$$X_{i,t|c_i=k} = Y_{1i}(t)^T\beta + Y_{2i}(t)^T\gamma_k + V_i(t)^T u_{ik} + w_i(t) + \epsilon_{i,t} \tag{2.5}$$

where Y_{1i} is vector of covariates with common fixed effects β, Y_{2i} is vector of covariates with class-specific fixed effects γ_k, V_i is a set of covariates with individual class-specific random effects u_{ik}. Finally, $w_i(t)$ is an autocorrelated Gaussian process with null mean and covariance equal to $cov(w_i(t)w_i(s) = \sigma_w^2 \exp(-\rho|t-s|)$. Note that the equation 2.5 is equivalent to the more general equation 2.3.

Further developments of the GMM exist, such as the non-parametric GMM which uses a non-parametric distribution for the random effects (NGMM Muthen and Asparouhov [103]), but they will not be discussed here because they do not compare directly to the HMTD.

The GMM is estimated by maximization of the likelihood using an ordinary EM algorithm. The continuous latent variables η and the group membership variables c are considered as missing data. We used the R package lcmm Proust-Lima, Philipps, and Liquet [113] to compute the GMM.

Chapter 3

Estimation of the HMTD model

In this chapter, I discuss the estimation of the HMTD model and I propose a new ad hoc heuristic adapted to the specificities of the model. [1]

3.1 Introduction

Depending on its specification, the estimation of the HMTD model is not straightforward. Considering that the GMTD model, the MAR model and even a basic finite mixture model could be represented as special cases of HMTD by fixing some parameters to zero, we understand how adaptable the model is. However its versatility and adaptable nature present also an important challenge when we attempt to implement an estimation procedure able to cope with all possible specifications of the model. As we will discuss in this chapter, such general estimation procedure should not rely directly on the Likelihood derivatives. One of the ways to achieve this goal is the development of a heuristic optimization method that should be as fast as possible, but also it needs to reach an optimum of the likelihood.

Many studies deal with the problem of finding the optimum of a function without using its derivatives. Although numerous methods cope with this problem, some more popular than others, there is no unique method that can optimally cope with all the situations. In this paper, we present a search method with hill-climbing features specifically designed to deal with the maximization of the log-likelihood of a hidden mixture transition distribution (HMTD) model for continuous variables, but which could also

[1] A short version of this chapter has been published as Taushanov Z & Berchtold A (2017). A direct local search method and its application to a markovian model. *Statistics, optimization and information computing* 5:19-34. doi:10.19139/soic.v5i1.253 Taushanov and Berchtold [152].

be used in many other problems that have similar characteristics. One advantage of this method appears when we do not have a fixed set of constraints, but we cannot accept all mathematically correct solutions. For instance, an extremely high value for the autoregressive coefficient of the mean appears occasionally as the most likely solution through the Nelder-Mead method, but such a value is non-interpretable with regard to the data meaning. This occurs especially in the case of datasets containing either short sequences or a small number of sequences. As a local search, our approach begins to explore the neighbourhood of the initial solution without going too far in the parameter space, thus avoiding aberrant solutions or numerical irregularities in the objective function.

In this chapter, we explore the potential of the aforementioned method to improve the estimation of the parameters of an HMTD model for continuous variables. This model can be used to both describe (model) longitudinal data and cluster multiple sequences. The most time-consuming part of the modelling is the estimation of the parameters that maximize the log-likelihood. As the log-likelihood equation is not easy to derive explicitly, we need a procedure that allows us to rapidly maximize our log-likelihood function without using any derivatives. In the following sections, we briefly present the HMTD estimation tools and procedure, followed by our heuristic estimation procedure, which we compare with some other well-known heuristic methods, and finally, we present and analyse the results of different numerical experiments.

3.2 Estimation principles

3.2.1 Log-likelihood computation and the EM algorithm

Before introducing the likelihood equations of the model, we need to present the context of its computation. Let us first consider the case of the simpler HMM that implies no dependence between the observations, but on a latent level. In this case the likelihood is obtained by marginalizing over the latent variables S: $p(X) = \sum_S f(X, S|\Theta)$. As stated by Bishop [23], the problem is that we have to sum over exponentially many paths (k^n) through one latent chain of length n, which is impossible even for a small size dataset. Furthermore, the likelihood function consists in a summation over the visible models for the different possible settings of the latent variable, which results in complex expressions when maximizing the likelihood. Since the latent states that generate the data have different distributions at each t, depending on the previous states, we need to pass through every possible paths (among T^k paths in total). Therefore, in order to

find a more efficient way to maximize the likelihood, one must estimate the distribution of the unobserved part of the model. For this reason a version of the Expectation-Maximization (EM) algorithm first published by Dempster, Laird & Rubin [41] is used.

In its basic form, the *EM* algorithm consists in two steps alternating until convergence: First, initial values for all the visible level parameters $\Theta^0 = \{\theta^0_{0,1} \dots \theta^0_{q,k}, \phi^0_{0,1} \dots \phi^0_{p,k}\}$ are chosen. Then, in the Expectation step, using this initialization, the posterior distribution of the latent states is estimated $P(S|X, \Theta^0)$ (specified by the transition matrix A and initial probabilities vector π). The so-obtained $\hat{\pi}$ and $\hat{A}$ are used to calculate the expectation of the complete data log-likelihood

$$Q(\Theta|\Theta^0) = E_{S|X,\Theta^0}(\ln P(X, S|\Theta)).$$

In the M-step one attempts to find the visible parameters Θ ($\Theta = \{\theta, \phi\}$ in the HMTD) that maximize the quantity:

$$\Theta^1 = \arg\max_{\Theta} Q(\Theta|\Theta^0)$$

Then the latent parameters are estimated again and the log-likelihood is recomputed, and so on until convergence.

Let us now illustrate the likelihood computations with the HMTD equations. To begin, we consider only the visible level of the model. Considering that the model is a mixture of Gaussians with λ_g indicating the weight of the g-th mixture component

$$F\left(x_t|x_{t-1} \dots x_1\right) = \sum_{g=1}^{k} \lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}}\right),$$

we can write the likelihood of a series $x_{r+1} \dots x_T$ knowing the previous r observations as:

$$\mathcal{L}(\Theta|X) = f_\Theta(X) = \prod_{t=r+1}^{T} \sum_{g=1}^{k} \lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}}\right)$$

the *incomplete* log-likelihood expression then becomes:

$$\begin{aligned}
\log \mathcal{L} &= \sum_{t=r+1}^{T} \log \left[\sum_{g=1}^{k} \lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}} \right) \right] \\
&= \sum_{t=r+1}^{T} \log \left[\sum_{g=1}^{k} \lambda_g \frac{1}{\sqrt{2\pi\sigma_{g,t}^2}} \exp \left(-\frac{(x_t - \mu_{g,t})^2}{2\sigma_{g,t}^2} \right) \right] \\
&= (T-r) \log(\frac{1}{\sqrt{2\pi}}) + \sum_{t=r+1}^{T} \log \left[\sum_{g=1}^{k} \frac{\lambda_g}{\sigma_{g,t}} \exp \left(-\frac{(x_t - \mu_{g,t})^2}{2\sigma_{g,t}^2} \right) \right] \\
&= \frac{(r-T)}{2} \log(2\pi) + \sum_{t=r+1}^{T} \log \left[\sum_{g=1}^{k} \frac{\lambda_g}{\sigma_{g,t}} \exp \left(-\frac{(x_t - \mu_{g,t})^2}{2\sigma_{g,t}^2} \right) \right]
\end{aligned}$$

In this equation we have the likelihood of the data considering only the observed part of the model and ignoring the latent one. In EM framework this is considered as *incomplete* data.

Let us now consider a vector z_t indicating the "true" component g that generated the observation x_t with $z_{g,t} = 1$ if $X_t \sim f_g$ and 0 elsewhere. In this case we can note $P(Z_i = z_i) = \lambda$ and the distribution of Z_i is multinomial:

$$Z_i \sim Mult_g(1, \lambda)$$

where $\lambda = \{\lambda_1, \ldots, \lambda_g\}^T$. Then, if we have both $\{x_{r+1} \ldots x_T\}$ and the corresponding latent vectors $\{z_{r+1} \ldots z_T\}$, we can compute the *complete data* log-likelihood (see Berchtold [15]) for one sequence:

$$\mathcal{L}_c(\Theta|X) = f_{c\Theta(X)} = \prod_{t=r+1}^{T} \sum_{g=1}^{k} z_{g,t} \left(\lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}} \right) \right)$$

$$\begin{aligned}
\log \mathcal{L}_c &= \sum_{t=r+1}^{T} \log \left(\sum_{g=1}^{k} z_{g,t} \lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}} \right) \right) \\
&= \sum_{t=r+1}^{T} \log \left[\sum_{g=1}^{k} z_{g,t} \frac{\lambda_g}{\sqrt{2\pi\sigma_{g,t}^2}} \exp \left(-\frac{(x_t - \mu_{g,t})^2}{2\sigma_{g,t}^2} \right) \right]
\end{aligned}$$

As $z_{g,t}$ is a binary indicator, at time t only one $z_{g,t}$ is one, the others being equal to 0. Therefore we have

$$\log \left[\sum_{g=1}^{k} z_{g,t} \lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}} \right) \right] = \log \left[0 + \cdots + 0 + 1 \times \lambda_g f_g \left(\frac{x_t - \mu_{g,t}}{\sigma_{g,t}} \right) + 0 \ldots \right]$$

Thus we can split the elements of the logarithm by ignoring (taking out of the log) the sum sign

$$\begin{aligned}\implies &= \sum_{t=r+1}^{T}\sum_{g=1}^{k} z_{g,t}\log\left[\frac{\lambda_g}{\sqrt{2\pi}\times\sigma_{g,t}}\exp\left(-\frac{(x_t-\mu_{g,t})^2}{2\sigma_{g,t}^2}\right)\right]\\ &= (T-r)\log(\frac{1}{\sqrt{2\pi}}) + \sum_{g=1}^{k} log(\lambda_g)\sum_{t=r+1}^{T} z_{g,t} - \sum_{t=r+1}^{T}\sum_{g=1}^{k} z_{g,t}\log(\sigma_{g,t})\\ &- \sum_{t=r+1}^{T}\sum_{g=1}^{k} z_{g,t}\frac{(x_t-\mu_{g,t})^2}{2\sigma_{g,t}^2}\end{aligned}$$

$$\begin{aligned}\log\mathcal{L}_c = \frac{(r-T)}{2}\log(2\pi) + \sum_{g=1}^{k}\log(\lambda_g)\sum_{t=r+1}^{T} z_{g,t}\\ - \sum_{t=r+1}^{T}\sum_{g=1}^{k} z_{g,t}\log(\sigma_{g,t}) - \sum_{t=r+1}^{T}\sum_{g=1}^{k} z_{g,t}\frac{(x_t-\mu_{g,t})^2}{2\sigma_{g,t}^2}\end{aligned}$$

The log-likelihood computation would be straightforward if we knew the complete data. However, typically we only have the incomplete data X and our only knowledge of the latent variables Z is their posterior distribution $p(Z|X,(\phi,\theta))$ determined by $\hat{A}$ and $\hat{\pi}$. This is where the EM algorithm is useful. During the expectation step (E), the visible parameters of the models are considered as known and the expectation of the unobserved variables is computed:

$$\hat{z}_{g,t} = \frac{\frac{\lambda_g}{\sigma_{g,t}} f_g\left(\frac{x_t-\mu_{g,t}}{\sigma_{g,t}}\right)}{\sum_{g=1}^{k}\frac{\lambda_g}{\sigma_{g,t}} f_g\left(\frac{x_t-\mu_{g,t}}{\sigma_{g,t}}\right)}, \quad \text{for each} \quad g = 1\ldots k$$

During the maximization step (M), the so-obtained expectations of the latent states $\hat{z}_{g,t}$ are plugged in the $\log\mathcal{L}_c$ equation. Then all the model parameters are reestimated.

The component weights are reestimated as

$$\hat{\lambda}_g = \frac{\sum_{t=r+1}^{T}\hat{z}_{g,t}}{T-r}$$

If we take as example a model with $\mu_{g,t} = \phi_{g,0} + \phi_{g,1} x_{t-1}$, the first derivative of the log-likelihood with respect to ϕ_1 is:

$$\begin{aligned}
\frac{\partial \log \mathcal{L}_c}{\partial \phi_{g,1}} &= \frac{\partial}{\partial \phi_{g,1}} \sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \frac{(x_t - \phi_{g,0} - \phi_{g,1} x_{t-1})^2}{2\sigma_{g,t}^2} + const \\
&= \frac{\partial}{\partial \phi_{g,1}} \sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \frac{(x_t^2 + \phi_{g,0}^2 + \phi_{g,1}^2 x_{t-1}^2 - 2\phi_{g,0} x_t - 2\phi_{g,0}\phi_{g,1} x_{t-1} - 2\phi_{g,1} x_t x_{t-1})}{2\sigma_{g,t}^2} \\
&+ const \\
&= \sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \frac{2(\phi_{g,1} x_{t-1}^2 - \phi_{g,0} x_{t-1} - x_t x_{t-1})}{2\sigma_{g,t}^2} \\
&= \phi_{g,0} \sum_{t=r+1}^{T} \frac{z_{g,t} x_{t-1}}{\sigma_{g,t}^2} - \phi_{g,1} \sum_{t=r+1}^{T} \frac{z_{g,t} x_{t-1}^2}{\sigma_{g,t}^2} - \sum_{t=r+1}^{T} \frac{z_{g,t} x_t x_{t-1}}{\sigma_{g,t}^2} \quad = 0
\end{aligned}$$

$$\implies \sum_{t=r+1}^{T} \frac{z_{g,t} x_t x_{t-1}}{\sigma_{g,t}^2} = \sum_{s=1}^{p_g} \phi_{g,0} \sum_{t=r+1}^{T} \frac{z_{g,t} x_{t-1}}{\sigma_{g,t}^2} - \sum_{s=1}^{p_g} \phi_{g,1} \sum_{t=r+1}^{T} \frac{z_{g,t} x_{t-1}^2}{\sigma_{g,t}^2}$$

If we generalize this result, the mean parameters $\phi_{g,j}$ are estimated from the roots of the following $\sum_{g=1}^{k} p_g$ equations (supposing that each component may include a different number p_g of lags for the mean):

$$\text{for each lag} \quad j = 1, \ldots, p_g : \quad \sum_{t=r+1}^{T} \frac{z_{g,t} x_t w_j}{\sigma_{g,t}^2} = \sum_{s=0}^{p_g} \hat{\phi}_{g,s} \left(\sum_{t=r+1}^{T} \frac{\hat{z}_{g,t} w_j w_s}{\sigma_{g,t}^2} \right)$$

$$\text{where} \quad \omega_j = \begin{cases} 1 \quad \text{for} \quad j = 0 \\ x_{t-j} \quad \text{otherwise.} \end{cases}$$

The root of these equations can be computed.

The last set of parameters to be re-estimated for each component of the model separately are the variance parameters $\theta_{g,j}$. If the variances are not constant, the roots of another set of $\sum_{g=1}^{k} q_g$ equations must be found. For instance, deriving the log-likelihood with respect to $\theta_{g,1}$ and considering that $\sigma_{g,t}^2 = \theta_{g,0} + \theta_{g,1} x_{t-1}^2$, we obtain:

$$\frac{\partial \log \mathcal{L}_c}{\partial \theta_{g,1}} = -\frac{\partial}{\partial \theta_{g,1}} \sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \underbrace{\log(\sqrt{\theta_{g,0} + \theta_{g,1} x_{t-1}^2})}_{f(g(\theta_{g,1}))}$$
$$+ \frac{\partial}{\partial \theta_{g,1}} \sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \frac{(x_t - \mu_{g,t})^2}{2(\theta_{g,0} + \theta_{g,1} x_{t-1}^2)} + const$$
$$= -\frac{\partial}{\partial \theta_{g,1}} \sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \underbrace{\frac{1}{\sqrt{\theta_{g,0} + \theta_{g,1} x_{t-1}^2}} \frac{x_{t-1}^2}{2\sqrt{\theta_{g,0} + \theta_{g,1} x_{t-1}^2}}}_{f'(g(\theta_{g,1}))g'(\theta_{g,1})}$$
$$+ \sum_{t=r+1}^{T} \frac{z_{gt}(x_t - \mu_{g,t})^2 x_{t-1}^2}{2(\theta_{g,0} + \theta_{g,1} x_{t-1}^2)^2}$$
$$= -\sum_{t=r+1}^{T} \sum_{g=1}^{k} z_{gt} \frac{x_{t-1}^2}{2(\theta_{g,0} + \theta_{g,1} x_{t-1}^2)} + \sum_{t=r+1}^{T} \frac{z_{gt}(x_t - \mu_{g,t})^2 x_{t-1}^2}{2(\theta_{g,0} + \theta_{g,1} x_{t-1}^2)^2} \quad = 0$$

$$\Longrightarrow \sum_{t=r+1}^{T} \frac{z_{gt} x_{t-1}^2}{\theta_{g,0} + \theta_{g,1} x_{t-1}^2} = \sum_{t=r+1}^{T} \frac{z_{gt}(x_t - \mu_{g,t})^2 x_{t-1}^2}{(\theta_{g,0} + \theta_{g,1} x_{t-1}^2)^2}$$

Again, deriving with respect to any lag leads to the following general equation Berchtold [15]:

$$\text{for each lag} \quad j = 1, \ldots, q_g : \quad \sum_{t=r+1}^{T} \frac{z_{g,t} u_j}{\hat{\theta}_{g,0} + \underbrace{\sum_{s=1}^{q_g} \hat{\theta}_{g,s} x_{t-s}^2}_{=0 \text{ if constant } \sigma_{g,t}}} = \sum_{t=r+1}^{T} \frac{\hat{z}_{g,t}(x_t - \mu_{g,t})^2 u_j}{(\hat{\theta}_{g,0} + \underbrace{\sum_{s=1}^{q_g} \hat{\theta}_{g,s} x_{t-s}^2}_{=0 \text{ if constant } \sigma_{g,t}})^2}$$

$$\text{where} \quad u_j = \begin{cases} 1 \quad \text{for} \quad j = 0 \\ x_{t-j}^2 \quad \text{otherwise (if we choose the first specification of } \sigma_{g,t}) \end{cases}$$

From the above equations, we obtain that if the standard deviation of a component g is specified as constant, ($\sigma g, t = \theta_0$ and $\theta_{g,1} = \theta_{g,2} = \cdots = \theta_{g,q_g} = 0$), we need to solve the equation:

$$\hat{\theta}_{g,0} = \frac{\sum_{t=r+1}^{T} \hat{z}_{g,t}(x_t - \mu_{g,t})^2}{\sum_{t=r+1}^{T} \hat{z}_{g,t}}$$

However, if this is not the case and $\sigma_{g,t}$ is in its autoregressive form, no single solution can be found. The impossibility to compute these solutions indicates that we cannot employ the ordinary M-step of the regular EM algorithm, since it requires these solutions of the derivatives in order to maximize the log-likelihood function. Therefore, we need to find another way to insure the log-likelihood maximization with respect to all parameters.

3.2.2 GEM algorithm and alternatives

GEM algorithm

One solution is to implement an EM algorithm in which the M-step avoid the use of the derivatives. We will use a form of the Generalised EM algorithm (GEM) (see McLachlan and Krishnan [93]), with a maximization procedure related to the one proposed by Berchtold [14]. The idea behind this approach is to perform the maximization of $log\mathcal{L}$ within the M-step by using an optimization procedure that does not rely on the derivative equations. It is designed for cases where a solution of the M-step does not exist in a closed form. In this procedure, the only requirement for the maximization step when choosing the new visible parameters Θ^{i+1} is to fulfill the inequality:

$$Q(\Theta^{(i+1)}, \Theta^{(i)}) \geq Q(\Theta^{(i)}, \Theta^{(i)})$$

instead of maximising Q w.r.t. Θ. This implies that necessarily $L(\Theta^{i+1}) \geq L(\Theta^{i})$ i.e. the likelihood does not decrease at each step.

This is the procedure on which we base the estimation of the model and for the rest of this chapter we will focus on implementing an optimal heuristic optimization for the M-step. GEM has already been implemented for HMTD using a genetic algorithm Berchtold [15], but a gradient-type algorithm can also be used McLachlan and Krishnan [93].

CEM, SEM and alternatives

The Expectation Conditional Maximization (ECM) algorithm is another interesting class of GEM algorithm. It has been applied on HMTD (see Wang, Smith and Hyndman [167]) with positive results. The difference with EM is that instead of the usual M step, we have several different steps that are computed consecutively. Thus the monotone convergence property of EM is present since ECM still maximizes the complete-data likelihood, even though instead of one M-step, one implements several (s) CM steps

(often one for each parameter of the model $\Theta = \{\theta_1 \dots \theta_s\}$). Each of these CM steps is performed over a constrained space in which the parameters that are not being calculated are considered as fixed. The parameters are then estimated separately, using either a close form or an iterative solution. Thus, if $\Theta^{i+1,s}$ is the solution after the s-th CM step of the $i+1$-th iteration, the maximization step respects:

$$Q(\Theta^{(i+1,s)}, \Theta^{(i)}) \geq Q(\Theta^{(i+1,s-1)}, \Theta^{(i)}) \geq Q(\Theta^{(i)}, \Theta^{(i)})$$

Other EM-type algorithms exist, such as the SEM (see Celeux, Chauveau & Diebolt [28]) that was first designed to compute the likelihood for finite mixture models. It contains an additional stochastic step between the E and M steps in which the values of z_t are drawn at random, i.e. the data are associated to the components and the coefficients are estimated accordingly. The random drawing within the S part prevents the SEM from converging towards the nearest optimum, especially since Θ_{i+1} may actually decrease $Q(\Theta^{(i+1)}, \Theta^{(i)})$.

We need to highlight that the estimation procedure may also be more complex, depending on the presence and nature of covariates on the visible, but also on the latent level of the model, in addition to the number of components and lags. The specification of the latent level, namely the constrained form of the transition matrix, but also the order of its corresponding Markov chain, have an influence on the computational part. For instance, any order >1 of the latent level Markov chain imposes the use of a partially sparse transition matrix (ex: the matrix of order 2 in Figure 2.1.1). On the other side, when ordinary clustering is the aim of the analysis, a diagonal matrix of ones is used, which transforms the model into a simpler mixture model and the latent part estimation is no longer necessary except for the initial probabilities π_i for each state.

3.2.3 Properties of the (G)EM algorithm

3.2.4 Forward-backward algorithm and latent parameters estimation

Before considering the possible versions of the M-step of the EM algorithm, we need to detail how the likelihood is computed during the E-step. From the initial values (and after each re-estimation) of the visible-level parameters, one needs to compute the transition probabilities of the latent states given the entire observed sequences and the specification of the model. This is done using the Forward-Backward algorithm introduced by Rabiner [117].

Once the visible-level parameters $\phi_{g,t}$ and $\theta_{g,t}$ are initialised or re-estimated, one needs to estimate the latent parameters. For this purpose, the forward-backward algorithm (FB) is employed. The main objective is to estimate the latent state probabilities (transition probabilities A and initial probabilities π) when the visible-level parameters are considered as known. It is a commonly used tool in HMM which computes the posterior marginal probability of the latent variables given the observed sequences (and the current model M with the visible part parameters) $P_M(S_t|X_{1:T})$ at every time $t \in 1, ..., T$. The algorithm consists in two dynamic computation passes: a forward pass and a backward one. The computation is carried out a first time forward, starting from t=1, and then a second time backward, starting from $t = T$. Both sets of probabilities are then combined by "smoothing" the information obtained from the forward pass and the one obtained from the backward pass.

The forward part is estimating the probabilities to be in a given latent state at a given time, knowing the observations up to this time: $P_M(S_t|X_{1:t})$. But in order to do this, we first need to estimate the joint probabilities

$$\alpha_t(j) = P_M(X_0, \ldots, X_t, S_t = j).$$

Given a vector of initial probabilities for each state $\pi_j = P(S_0 = j)$, at $t = 0$ we have:

$$\alpha_0(j) = P_M(X_0, S_0 = j) = P(X_0|S_0 = j)P(S_0 = j) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_t - \mu_{j,t})^2}{2\sigma_j^2}\right) \pi_j$$

Knowing that $P(X_0|S_0 = j) = f_j(X_0)$ is a Gaussian distribution with parameters μ_j and σ_j, and that $P(S_0 = j) = \pi_0(j)$, we obtain

$$\alpha_0(j) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_t - \mu_{j,t})^2}{2\sigma_j^2}\right) \pi_j$$

As $P(S_t = j) = \sum_{i=1}^{K} a_{ij}\alpha_{t-1}(i)$, with a_{ij} an element of the transition matrix A, we obtain the following equation that is solved consecutively for each t until $t = T$:

$$\alpha_t(j) = P_M(X_0, \ldots, X_t, S_t = j) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_t - \mu_{j,t})^2}{2\sigma_j^2}\right) \sum_{i=1}^{K} a_{ij}\alpha_{t-1}(i)$$

The probabilities of the latent states at each time t, given the observations up to this time are:

$$P_M(S_t = j|X_1, \ldots, X_t) = \frac{P_M(X_0, \ldots, X_t, S_t = j)}{P_M(X_1 \ldots X_t)} = \frac{\alpha_t(j)}{P_M(X_1 \ldots X_t)}$$

After calculating the forward probabilities α_t, we need to proceed in the exact same manner for the backward pass, but starting from the end of the sequence $t = T$ up to $t = 1$. This will provide us with the backward probabilities β_t. We start from a given latent state and we look for the probabilities of observing all the future observations up from this state. We consider the initial state as known and therefore each $\beta_t(i) = 1$. Continuing backwards we obtain:

$$\beta_t(i) = P_M(X_{t+1}, \ldots, X_T | X_t, S_t = i) = \sum_{j=1}^{K} a_{ij}\beta_{t+1}(j)\frac{1}{\sigma_j\sqrt{2\pi}}\exp\left(-\frac{(x_{t+1} - \mu_{j,t+1})^2}{2\sigma_j^2}\right)$$

A normalization is applied in the computations of $\alpha_t(j)$ and $\beta_t(i)$ to correct the numerical problems that occur if one state has an excessively small probability.

After passing the algorithm in both directions, we can compute the marginal probabilities γ_t of the latent states at any time, knowing the entire sequence of observations:

$$\begin{aligned}
\gamma_t(i) &= P_M(S_t = i | X_{1:T}) = P_M(S_t = i | X_{1:t}, X_{(t+1):T}) \\
&= \frac{P_M(S_t = i, X_{1:t}, X_{(t+1):T})}{P_M(X_1 \ldots X_T)} \\
&= \frac{P_M(X_{1:t}, X_{(t+1):T} | S_t = i) P(S_t = i)}{P_M(X_1 \ldots X_T)} \\
&= \frac{P_M(X_{1:t} | S_t = i) P(X_{(t+1):T} | S_t = i) P(S_t = i)}{P_M(X_1 \ldots X_T)} \\
&= \frac{P_M(X_{1:t}, S_t = i) P_M(X_{(t+1):T} | S_t = i)}{P_M(X_1 \ldots X_T)} \\
&= \frac{\alpha_t(i)\beta_t(i)}{L(X_0 \ldots X_T)}
\end{aligned}$$

Combining the forward and backward probabilities results in a "smoothing" probability computation of γ_t. The latter represent an estimation of the most probable state of the latent variable at each time t of the observed sequence. However, this does not result in the most probable *sequence* of hidden states. The reason is that even though the latent level transition probabilities are used in the calculation of α_t and β_t, they are not respected when combining both to obtain γ_t. In other words, we have the most probable states independently, but we do not know how likely they are to occur successively in this exact sequence i.e. $P(S_t = i)P(S_{t+1} = j) \neq P(S_t = i, S_{t+1} = j)$. Fortunately, there exists another tool called the *Viterbi algorithm*, which can provide us with this optimal latent sequence and which we will describe later.

After computing γ_t over the entire sequence, there is one more set of probability that we need in order to estimate the latent transition matrix A. The joint probability of two successive states (i and j) given the entire sequence of observations is called $\epsilon_t(i,j)$ and represents a three dimentional array of size $[k^\ell \times k \times n-1]$, where ℓ is the order of dependence of the hidden Markov chain. It is computed as:

$$\begin{aligned}
\epsilon_t(i,j) &= P_M(S_t = i, S_{t+1} = j | X_0, \ldots, X_T) \\
&= \frac{P(X_{1:t}, X_{(t+1):T} | S_t = i, S_{t+1} = j) \quad P(S_t = i, S_{t+1} = j)}{P(X_{1:T})} \\
&= \frac{P(X_{1:t}|S_t = i) P(X_{t+1}|S_{t+1}) P(X_{(t+2):T}|S_{t+1} = j) P(S_{t+1} = j|S_t = i) P(S_t = i)}{P(X_{1:T})} \\
&= \frac{P(X_{1:t}|S_t = i) P(S_t = i) P(X_{t+1}|S_{t+1}) P(X_{(t+2):T}|S_{t+1} = j) P(S_{t+1} = j|S_t = i)}{P(X_{1:T})} \\
&= \frac{\alpha_t(i) \quad \frac{1}{\sigma_j \sqrt{2\pi}} \exp(-\frac{(x_{t+1} - \mu_{j,t+1})^2}{2\sigma_j^2}) \quad \beta_{t+1}(j) \quad a_{ij}}{L(X_0 \ldots X_T)}
\end{aligned}$$

After computing all the probabilities γ_t and ϵ_t for every $t \in \{1, \ldots, T\}$, it becomes easy to re-estimate the latent part transition probability array A. Its estimation is provided by the ratio of the sums over all periods of all ϵ_t-s and γ_t-s:

$$\begin{aligned}
\hat{a}_{ij} &= \sum_{t=1}^{T-1} P(S_{t+1} = j | S_t = i, X_0, \ldots, X_T) \\
&= \frac{\sum_{t=1}^{T-1} P_M(S_t = i, S_{t+1} = j | X_{1:T})}{\sum_{t=1}^{T-1} P_M(S_t = i | X_{1:T})} \\
&= \frac{\sum_{t=1}^{T-1} \epsilon_t(ij)}{\sum_{t=1}^{T-1} \gamma_t(i)}
\end{aligned}$$

For what concerns the vector of initial probabilities for each latent state π_i at time $t = 0$, they are computed from the sums of all γ_t:

$$\pi_i = \frac{\sum_{t=1}^{T-1} \gamma_t(i)}{T-1}$$

It is important to precise that since longitudinal data are often composed of multiple data sequences, the latent level parameters A and π_i are estimated separately on each sequence and then aggregated at the end:

$$A^{tot} = \sum_{i=1}^{n} w_i A^{(i)}$$

where w_i is the weight of a sequence i and $A^{(i)}$ indicates its corresponding estimation of A. The weights are either provided with the data (from the design of the survey), or proportional to the length of each sequence otherwise.

The above formulas consider the estimation in the most common latent specification where the order of the hidden Markov chain is $\ell = 1$. However in general, ϵ_t denotes the probability of $\ell + 1$ successive states of the latent chain. Therefore, for a second order chain for instance, we have an array of size $(k^2 \times k \times n - 1)$ for $\epsilon_t(i, j, k) = P_M(S_t = i, S_{t+1} = j, S_{t+2} = k|X_0, \ldots, X_T)$, from which one can estimate the matrix A of size $(k^2 \times k^2)$ giving the transition probabilities conditionally on the two previous states.

After re-estimation of all latent parameters, the log-likelihood equation is:

$$L(X_0 \ldots X_T) = \sum_{i=1}^{K} \alpha_t(i) \sum_{j=1}^{K} a_{ij}\beta_{t+1}(j) \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_{t+1} - \mu_{j,t})^2}{2\sigma_j^2}\right)$$

To provide a more concrete example with a specified model, if all components have two lags for both the mean and the standard deviation ($p_k = 2$ and $q_k = 2$) and two visible-level covariates c_1 and c_2, then the above equation becomes:

$$\begin{aligned} L(X_0 \ldots X_T) = &\sum_{i=1}^{K} \alpha_t(i) \sum_{j=1}^{K} a_{ij}\beta_{t+1}(j) \\ &\times \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left(-\frac{(x_{t+1} - (\phi_{j,0} + \phi_{j,1}x_t + \phi_{j,2}x_{t-1} + \phi_{cov_1}c_1 + \phi_{cov_2}c_2))^2}{2(\theta_{j,0} + \theta_{j,1}x_{t-1}^2 + \theta_{j,2}x_{t-2}^2)}\right) \end{aligned}$$

As seen before, solving the likelihood derivative equations is complex for the standard deviation parameters, because of the lack of unique solutions. An additional complexity is that every component may often use its own numbers of lags for the mean and the standard deviation. Depending on the data and the objectives, it is possible to choose a component with constant mean and variance, together with another one with a two period memory for the mean and one for the standard deviation (for instance we may have: $\mu_{g=1} = \phi_{1,0}$ and $\sigma_{g=1} = \theta_{1,0}$ for the first component and $\mu_{g=2} = \phi_{2,0} + \phi_{2,1} \times X_{t-1} + \phi_{2,2} \times X_{t-2}$ and $\sigma_{2,t} = \sqrt{\theta_{2,0} + \theta_{2,1}x_{t-1}^2}$). Thus, in order to

allow the HMTD model to be as flexible as possible (allow heterogenious modelling), we attempted to implement an estimation procedure that is not the fastest for some given specifications, but that is as generalisable as possible over the variety of model specifications and uses. This is why we explored the use of heuristic methods (that does not use the derivatives of the likelihood function) within the E-step of a GEM algorithm, in order to optimize the log-likelihood function.

3.2.5 Viterbi algorithm

In HMMs, one is often interested in the most probable sequence of latent states that could lead to the observed sequence. The most popular solution to this problem is the algorithm proposed by Andrew Viterbi Bishop [23], Viterbi [165] (and other authors simultaneously).

Suppose we have a sequence of length T and k latent states. This leads us to a set of k^T possible paths, a number that grows exponentially with the number of time periods. Even though we could compute the path probability using the initial π_i, the transition probability matrix A and the probability distribution for each state, it would be difficult to do this for all the paths. The Viterbi algorithm makes the task easier computationally by following only k paths at each time. Suppose that we need to find the optimal path up to time t for state $S_t = i$. Even though many paths lead to this point, only one of them is the most probable. Therefore, at t we need to consider only k optimal paths. While we move to $t+1$, this number becomes k^2, but again only one of them is the most likely for each state, and therefore we keep only k of them. At time T, only one state will be the most likely, and only one optimal path will lead to it. If we call $\mathcal{V}_{t,i}$ the probability of the path that is the most likely up to the state i at time t, we can calculate these probabilities iteratively, starting with

$$\mathcal{V}_{i,1} = P(X_1|S_1 = i) \times \pi_i$$

and maximizing

$$\mathcal{V}_{j,t} = \max_{j\in\{1,\dots,K\}} P(X_t|S_t = j) \times a_{i,j} \times \mathcal{V}_{i,t-1} \quad \text{for each} \quad t \in \{1,\dots,T\}$$

By tracking all the optimal paths, we can then find the sequence corresponding to $\mathcal{V}^*_{k,T}$.

3.2.6 Maximization of the log-likelihood function

The procedure for log-likelihood maximization follows the general principle of the expectation-maximization (EM) algorithm, even if an alternative Markov Chain Monte Carlo approach for hidden Markov models estimation was discussed by Ryden [132] and Scott [135]. In Figure 3.1, we briefly illustrate the main steps of the estimation procedure. After initializing the visible parameters (step 1), we apply the aforementioned forward-backward algorithm in the E-step (2) to re-estimate the parameters of the latent part of the model on the basis of the parameters of the visible part. Then, we calculate the resulting log-likelihood (3). During the maximization step (4), we try to improve the log-likelihood by changing the parameters of the visible part of the model, using the heuristic procedure described in the next Section. These modified values of the visible parameters are in turn used in step 2 to compute the corresponding log-likelihood value, which indicates whether the changes in the visible parameters were beneficial. The algorithm iterates until a stopping criterion is satisfied.

The difficulties in deriving the log-likelihood equation (for non-constant variance specifications) compels us to use a heuristic (or direct) estimation procedure (in step 4) to find parameter estimates that do not decrease the log-likelihood. Because of that modification of the M-step (optimization methods instead of having equations to maximize), our procedure is qualified as a generalized EM algorithm (GEM) rather than a standard EM.

This procedure can also be applied on multiple sequences or longitudinal data. Likelihood computations are performed separately on each independent sequence, and results are then averaged over all sequences.

3.3 Visible parameters estimation procedure

We discuss now the estimation of the parameters of the visible part of the HMTD model. We start with an unoriented search of the maximum of the log-likelihood function. This implies the introduction of an arbitrarily chosen initial point in our parameter space. This point is chosen without any information, but by making some “semi-educated guess” by using our knowledge of the nature of the data and parameters. In other words, we try to ensure that the initial solution is not too unlikely, in order to avoid falling in a region of the solution space corresponding to a very low likelihood.

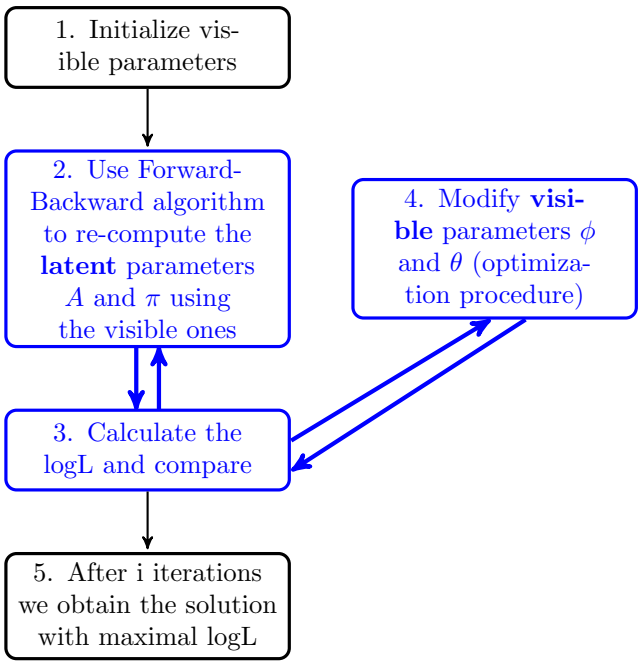


Figure 3.1: Steps in the estimation procedure.

3.3.1 Limits in the solution space

As the process describing the mean (and possibly the standard deviation) of the HMTD model is autoregressive (AR), we need to define the possible solution space for our parameters. We have to consider here only the parameters of the visible part of the model, since the latent parameters are re-estimated using elements calculated by the forward-backward procedure, before returning the log-likelihood of the model. For each component of the model, we have parameters $\phi_0 \dots \phi_p$ for the mean and $\theta_0 \dots \theta_q$ for the variance. Since the model could be used with any continuous variable, we cannot exclude the possibility to have a non-stationary AR process. Therefore, we cannot define any strict bounds on our parameter space. As an example, we can imagine a process in which one hidden state implies a constant increase in the observed variable. For what concerns the constants ϕ_{0i} and θ_{0i}, we also have little prior knowledge. However, we know that especially in the case of a constant standard deviation, θ_{0i} cannot be negative and should be "comparable" to the standard deviation of the data. One good strategy is to fix some initial arbitrary bounds around the empirical variance of the data, and further modify them whenever the current solution provided by the estimation algorithm approaches these limits. This will be beneficial especially at the beginning of

the estimation procedure because it will allow us to start from a most probable region of the solution space and prevent the algorithm from immediately spending time in exploring highly unlikely areas. In other terms, we can start in a narrow parameter space and broaden it gradually when the current solution approaches the limits. Thus we introduce "floating" limits to the solution space.

In practice, after having set limits, we can calculate the log-likelihood of a number of randomly chosen potential solutions. The initial values for the whole estimation procedure can then be chosen either as the parameter values giving the best log-likelihood, or as a centroid of the best solutions.

Another important point to be considered is the likely presence of dependence between (some of) the parameters. Noting that

$$\phi_{i0} + \phi_{i1} x_{t-1} \cdots + \phi_{ip} x_{t-p} \approx \mu_i$$

and

$$\theta_{i0} + \theta_{i1} x_{t-1}^2 \cdots + \theta_{iq} x_{t-q}^2 \approx \sigma_i^2$$

we observe that increasing all the ϕ_i parameters simultaneously leads to a larger mean for the component i (if the past observations are positive), which at certain points may diminish its probability. Therefore, all ϕ are interdependent. This information could also be incorporated in the estimation procedure in order to improve its efficiency. The same finding is valid for the variance of each component.

3.3.2 Searching the optimal solution

We describe here in detail the heuristic procedure used to re-estimate the visible parameters of the model (box 4 in Figure 3.1). Notice that after each use of this heuristic, the estimation procedure has to also re-estimate the latent parameters (box 2). Our heuristic is related to the algorithm implemented by Berchtold [14] in the case of the discrete MTD. An important feature is that we do not make use of any derivatives of the underlying log-likelihood objective function. First, we need to introduce an initial vector of parameters. As discussed above, a good guess for the ϕ and θ parameters is one that gives us a value close to respectively the mean and variance of our data. From this initial vector, we evaluate the change in the log-likelihood when each of the parameters is modified. Therefore, we consecutively increase and diminish each parameter before measuring the ΔlogL corresponding to the change. After computing all these

changes, we modify the entire vector of parameters in the direction that is optimal for each parameter separately.

We considered different versions of this procedure. The first one consisted in modifying only the parameter that enhances the most the log-likelihood and then re-estimate again the influence of the other parameters. Such a procedure make sense if our parameters are strongly dependent, but according to our experience, this dependence does not generally play a central role. The assumption of such a dependence costs too much in terms of computational time, and it appeared to be not sufficiently useful during our experiments.

If we modify all parameters simultaneously, in the direction that is optimal for them independently of all other parameters, in most of the cases we obtained a faster convergence towards the local maximum. However, this approach ignores any dependence between the parameters, which could be problematic. For instance, if one component with larger mean could fit the data better, we would see that an increase in ϕ_{i0} would improve our log-likelihood, and an increase in ϕ_{i1} or ϕ_{i2} would also be beneficial. According to the aforementioned approach, we should then increase all of these parameters simultaneously. Although we make several independently beneficial steps simultaneously, this could lead to a decrease of the log-likelihood, because we amplify the effect of increasing the mean, making it too large.

To fix this problem, we can include the modification of the previous parameters before testing the influence of the next one on the log-likelihood (a method that we name "**S**" in our outputs). For instance, we test the effect upon the log-likelihood of a change in parameter ϕ_{i2} with respect to the solution obtained after saving the modifications made on the previously tested parameters (i.e., ϕ_{i0} and ϕ_{i1} if the order of optimization is not permuted). By doing this, we account for the dependence between the parameters without introducing any additional computational costs.

One potential problem that remains after this change is that we may improve the log-likelihood by modifying not the "most important" parameter (in terms of log-likelihood increase) at first place, and therefore compel the most important one to adjust to the last modification of less important parameters (increase ϕ_{i1} to increase the mean and adjust ϕ_{i0} to it, instead of proceeding in the opposite order when the latter coefficient increases the most the fitness). In other words, we make a step in the less important dimension of the solution space before making a step in the most important one. This could slow down the speed of convergence of our approach. Introducing a permutation in the order of update of our parameters can solve this problem. Such a permutation is indicated by "**P**." Changing the order of modification after each iteration, according

to the absolute improvement of the log-likelihood (ΔlogL) may lead us faster to the optimal solution.

We can rarely improve the log-likelihood by both increasing and decreasing the same parameter. This would be possible only if the current value of this parameter corresponded to a (local) minima of the solution space, a very rare situation. Introducing a one-direction improvement check ("**E**") can help us spare calculations: if an increase of a parameter increases the fitness, we update the parameter without checking what happens if we decrease it. However, if the fitness value decreases in the first case, we also need to see what happens when we decrease the value of the parameter.

Finally, we also need to sacrifice some precision by introducing a minimal change step for each parameter according to its absolute value and initial limits ("**M**"). This would prevent us from spending too many computations unnecessarily, considering infinitesimal updates of the parameters. Moreover, it seems logical to allow the step to vary during estimation. Noting that all parameters need to evolve in a different manner, we also need to allow an independent variation of their step size.

The logic of this procedure is simple: assuming that our initial guess is arbitrary, in many cases, it will be very far from the optimum. That means that once the good direction for re-estimating a parameter is found, we need to accelerate the convergence by increasing the relative change of the given parameter. In order to keep the procedure stable, we introduce a limit to the change rate. When we approach the optimal value of one parameter (i.e., when a further big leap worsens our log-likelihood), we shrink its relative change considerably (up to a limit) in order to improve its estimation accuracy. If the modification of the other parameters changes the optimal value for this parameter (interdependence), we increase its amplitude of change once again, and so on until convergence. The different amplitudes of the change may play a role in determining the order of parameter re-estimation.

We implemented two types of limits for the step of each parameter: in relative (min and max limits) and in absolute (only min limits) values. The relative value limits are measured by fractions of the amplitude of the parameter limited by its initial constraints, and these fractions are the same for all parameters (e.g., between 0.5% and 30%). However, for the autoregressive parameters, a precision of more than 0.01 (in absolute value) does not seem necessary. Consequently, we fixed at 0.005 times the initial amplitude for the mean and the variance (normally distributed around the mean and variance of the data) as the lower absolute limits for the corresponding parameters.

As said before, the heuristic procedure used to re-estimate the visible part of the model alternates with the forward-backward algorithm used for the latent part of the

model. In practice, there are two main possibilities for the visible parameters: either we try to reestimate each of them once before going to the latent parameters, or we allocate a fixed number of function calls to the heuristic, and we go to the latent part when this number is reached. This second possibility means that some parameters could be reestimated several times by the heuristic before going back to the reestimation of the latent parameters. When some visible parameters are far from the optimum, this method may speed up the convergence.

3.3.3 Stopping criterion

The estimation procedure continues until a local optimum is reached, that is, the value of the log-likelihood remains the same for two consecutive iterations, which implies that no further change of the parameters improves the fitness. However, this situation is often reached too quickly and does not necessarily imply an optimal solution. Therefore, we need to "jitter" the solution in order to continue the procedure. This is done by adding random noise to the solution that we scale to 0.1 of the value of the corresponding parameter (making sure that the variance parameters remain positive). Then, we repeat the procedure until the maximal number of iterations is reached.

For most versions of the algorithm, the optimal solution is one of those achieved immediately before one of the jitters, and therefore, we need to store the parameter values only at this moment.

3.3.4 Pseudo-code

An illustration of the whole procedure including the four above-discussed improvements (S, E, M, and P) is given by the following pseudo-code:

Pseudo-code for SEMP procedure:

```
WHILE number of function calls < max function calls
   FOR each parameter i of the solution V in the given Order
      Increase V(i) and calculate logL
      IF new logL is higher than the saved value
         save the new V(i) and increase its future change step Change(i)
      ELSE
         decrease V(i) and calculate logL
```

IF new logL is higher
save and increase Change(i)
ELSE
decrease Change(i)
END
END
END
Change the Order of the parameters according to the logL increase
IF last logL = previous logL (no change of any parameter improves the fitness)
save the parameters and logL
jitter the vector of parameters to escape local optimum
END
END

Notice that before using the heuristic procedure for the first time, the maximal and minimal percentage change and minimal absolute change for the AR parameters have to be chosen. Refer to Section 3.5 for examples.

3.4 Alternative procedures

This heuristic approach can be compared to other existing search methods described in the literature. We will consider the following alternatives: Particle Swarm Optimization (PSO) Elbeltagi, Hegazy & Grierson [48], Kennedy & Eberhart [77], Shi & Eberhart [138], Simulated Annealing (SA) Cerny [33], Kirkpatrick, Gelatt Jr & Vecchi [80], Genetic Algorithm (GA) Holland [67], Srinivas & Patnaik [144], Differential Evolution (DE) Storn & Price [148], Nelder-Mead simplex algorithm (NM) Nelder and Mead [106], Singer & Nelder [140], and the Limited memory version of the Broyden--Fletcher-Goldfarb-Shanno (BFGS) algorithm (L-BFGSB) Byrd et al. [27].

Simulated annealing

SA This method, named after a procedure of heating and cooling of metals, is based on the thermodynamics theory that matter becomes more solid upon cooling. It was independently implemented by Kirkpatrick, Gelatt and Vecchi [80] and Cerny [33]. The algorithm starts from a random point in the solution space and continues with a random guess. The resulting position may be accepted with a given probability even if the

evaluated fitness at that point is less good. Note that this is a minimization algorithm, and therefore, higher fitness is associated with lower "energy." The probability of acceptance of a new solution is determined by an acceptance probability function. It depends on the temperature t (time-dependent) and on the energies of the old and new solutions: e and e_1. The probability of each point always remains positive in order to be able to escape from local optima. As t tends to zero, the acceptance probability $P(ee_1, t)$ of a solution with higher energy ($e < e_1$) also approaches zero, which makes the procedure greedy. In other words, as time advances, the algorithm becomes less tolerant to solutions with smaller fitness.

The parameters of this procedure are the neighbor selection function, the acceptance function, the temperature function, and the initial temperature. We need to highlight that the SA relies on finding "lucky jumps" that improve the position.

Particle swarm optimization

The *PSO* was first developed by Kennedy and Eberhart [77] and Shi [138]. It was also used as a simulation of the behavior of bird flocks and fish schools. It consists in introducing a group of candidate solutions (particles) that move into the search space. They are guided by some random process but also have a velocity v influenced by their personal best solution and the global best solutions. The current position of each particle is computed for each dimension separately. The constants and random functions, which determine the influence of the personal and the global best solutions, as well as the number of swarms, are parameters of the PSO.

After each iteration, the velocity of each particle is updated. As time advances, the particles tend to group near the best solution found. Therefore, convergence can be reached either when the swarms come together (even if it is a local optimum), or when the global optimum is reached by one of them. The efficacy of PSO has been proved in several studies ([48]).

Genetic algorithm

The genetic algorithm (GA) is a (meta)heuristic search that is inspired by natural selection [67]. In this well-known approach, the candidate solutions, represented by a binary coded vector of the parameters, are transformed and combined in order to obtain better solutions. During each iteration of the algorithm, a given proportion of the population is selected through a fitness-based procedure to create a new generation of offspring.

From this selection, we choose two or more solutions that would be the "parents" of an offspring. From these parents, the new solutions are most often obtained via two operations: crossover (mixing part of the two parents solutions) and mutation (randomly changing some parameter values of a solution); however, various other techniques also exist such as elitism (the best overall current solution is kept unchanged). The fittest solutions survive until the next iteration of the algorithm.

Apart from the selection procedure, it is important to tune the mutation probability, crossover probability, and the population size. We must note that a very high mutation rate could be a reason for the loss of good solutions and a very high crossover probability may lead to a premature convergence of the algorithm toward a less than optimal solution. However, the former problem is attenuated by elitist selection.

There are many different versions of GA. One of them is an adaptive version of GA in which the crossover and mutation probabilities adapt in each generation in order to preserve the diversity of the population and to sustain the convergence capacity Srinivas & Patnaik [144].

Differential evolution

Differential evolution (DE) is an evolutionary method introduced by Storn & Price [148] that uses a group of candidate solutions (agents) spread in the search space. The agent positions, represented by d-dimensional vectors, are combined to create new ones and only the new positions with higher fitness are accepted. While DE is very similar to GA, the major difference is that DE uses vectors of real numbers instead of binary representations, which has an influence on the crossover and mutation procedures.

New parameter vectors are generated by adding one vector to the weighted difference of two other vectors (mutation procedure). Those parameters are mixed with another determined "target" vector (crossover procedure) to obtain a "trial" vector. If the trial vector has a better fitness, it replaces the target vector (selection procedure). During one iteration, each agent serves once as a target vector.

Nelder-Mead simplex algorithm

The Nelder-Mead (NM) method was introduced by Nelder and Mead [106]. The algorithm is calculated using $N + 1$ points x (vertices of the polytope), where N is the number of dimensions of our solution space. It consists in three main steps: ordering (of each point according to its fitness), centroid calculation (x_0), and transformation. The latter step may include three different transformations:

1. Reflection: We compute the reflected point $x_r = x_0 + \alpha(x_0 - x_{n+1})$. If this point is between the best and the second worst point in terms of fitness, we replace the worst with it and return to the beginning;

2. If the reflected point is the fittest one, we compute the expanded point $x_e = x_0 + \gamma(x_0 - x_{n+1}))$. Then, we replace the worst point with the best point chosen between the expanded and the reflected points, and we return to the beginning.

3. If the reflected point is the worst or second worst one, we compute the contracted point $x_c = x_0 + \rho(x_0 - x_{n+1}))$. If the contracted point is not the new worst, we replace the worst with it. If it is the worst, we replace all but the best point by $x_i = x_1 + \sigma(x_i - x_1)$ (shrinkage).

The method requires to select values for parameters α, γ, ρ, and σ. The lack of possibility to introduce constraints is a major issue of the Nelder-Mead algorithm for our problem. For instance, solutions with very high auto-regressive parameter values are not acceptable for our problem. Another issue mentioned by Singer & Nelder [140] is that the method can take a big amount of iterations with negligible improvement in the function while being far from the optimum, which results in premature termination of iterations.

L-BFGS-B

We also considered a limited memory version of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm Byrd et al. [27], which is a quasi-Newton method for optimization with constraints. It uses the approximation to the Hessian matrix to perform a search through the parameter space. It finds a direction of search and then determines the optimal step size. The L-BFGS-B version of this algorithm adapts the method to the use of bound constraints.

Comments

Although there also exist many hybrid procedures combining several of the previous algorithms that are reported to work well (for instance, PSO-SA Fang, Chen & Liu [49], PSO-GA Premalatha & Natarajan [112] etc.), we have no evidence of any advantages in our particular case, and hence, we do not consider them further.

The study of these different optimization methods, including our own heuristic, shows that each method is particularly adapted to a particular kind of problem, of

objective function, and of solution space. No search procedure seems to be able to find the solution quickly in all problems. Therefore, determining the best procedure for our specific problem is related to the question of how complex the objective function is. However, if we intend to apply the HMTD model to any dataset, we cannot expect to always have the same kind of solution space. This make us surmise that no absolute "best" procedure can be found. The difficulty then lies in finding a procedure able to find an acceptable solution for the largest possible set of situations. Note that not all of these methods increase the likelihood at each iteration. Some explore the parameter space before, aiming to discover a higher likelihood region, others accept some lower likelihood solutions occasionally with a defined probability.

3.5 Numerical experiments

We describe in this section the different numerical experiments employed to evaluate the performance of our heuristic when used with the HMTD model. All computations were made in the open source R language R Core Team [116]. The Mersenne Twister pseudorandom number generator Matsumoto & Nishimura [91] was used for generating random values when required. A personal computer under Windows 10, with an Intel Xeon E5-2650 running at 2.00 GH with 8 physical cores was used for the simulations. All syntaxes are available on GitHub: https://github.com/ztau/5352

3.5.1 Comparison between several versions of the heuristic

We performed a first numerical experiment to compare different versions of our heuristic procedure. We used the following HMTD model specification: a hidden Markov chain of order two, two hidden states with constant variance, and autoregressive mean with one lag. Therefore, the visible parameter vector has the following content:

$$(\theta_1, \theta_2, \phi_{1,0}, \phi_{2,0}, \phi_{1,1}, \phi_{2,1})$$

All datasets are available and documented in R R Core Team [116]. For each dataset, we compare 6 different specifications of the model, starting with a standard implementation without S, E, M, and P options, and going up to a specification including these 4 refinements (SEMP). The initial solution is the same for all procedures, because the same seed was used for the random generator. We used the first stopping criterion, that is, until a local optimum is reached (two consecutive iterations with same log-likelihood), without jittering. Results are provided in Table 3.1. We observe that

after including all aforementioned improvements, our approach becomes more efficient and precise in most cases, however more evidence is needed. Even if we cannot clearly identify from our experiments one variant of the algorithm that would always be better than the others, the methods with most of the above-mentioned features generally have better performances. Our choice would then be the "SEMP" method, because it gives the most consistent results. However more examples will be provided to investigate this claim.

Table 3.1: Comparison between different variants of the heuristic. Use of the first convergence to a local optimum as stopping criterion, without jittering. For each computation, we provide the log-likelihood and the required number of iterations. Source of the datasets: R.

	Different variants of the heuristic					
R dataset	**Standard**	**S**	**SE**	**SEP**	**SEM**	**SEMP**
UKDriverDeaths	-141.60	-138.95	-127.95	-128.05	-128.08	-128.70
	52	65	162	177	123	111
sunspot.month	-127.19	-127.19	-120.21	-117.85	-120.20	-113.12
	52	52	108	165	108	202
faithful	-89.16	-91.23	-83.79	-85.46	-83.31	-87.90
	104	52	189	144	231	95
JohnsonJohnson	-41.54	-47.33	-39.64	-39.39	-39.62	-39.68
	143	52	141	160	142	96
sunspots	-205.70	-217.57	-203.39	-207.08	-203.53	-204.01
	156	52	185	96	198	144
Seatbelts	-86.66	-84.29	-78.52	-79.10	-78.55	-78.05
	52	65	153	128	153	176

3.5.2 Comparison between the new heuristic and the standard optimization procedures

In order to compare the performances of our heuristic with other methods, we ran a second set of simulations using the same HMTD model as in Section 3.5.1. Again, we fitted the model on several time series available in R. The different tested procedure are four versions of our heuristic (S, SE, SEM, SEMP) as well as the SA, GA, NM, DE and PSO procedures.

Most of the optimization methods are not too difficult to implement. However, in order to avoid any influence of the coding upon our results, we performed our comparisons by using the most common package available in R for each method: "SA" (Xiang et al. [172]), "GA" (Scrucca [136]), "PSO" (Bendtsen [10]), "optim" (package "stats" included in the base distribution of R), and "DEoptim" (Mullen et al. [101]). The number of iterations of most algorithms was limited in order to obtain comparable results. As we used different datasets with different characteristics, it was difficult to calibrate the constants and parameters of each optimization procedure (for instance, the velocity constants for PSO, the α, γ, ρ, and σ parameters for the Nelder-Mead algorithm, etc.). Therefore, we chose to leave all these parameters to their default values as chosen by the conceptors of the R implementation of each algorithm. The only parameters for which we chose the initial values were the limits of the parameter space, the initial solution for the HMTD parameters, and the maximal number of calls of the objective log-likelihood function.

The variants of our heuristic differed from each other by the presence of the different options previously described. They all include the modification of the previous parameters (S), the second variant adds the one-directional check (E) allowing us to spare function calls. The third one includes the minimal absolute parameter change (M) defined as 1/300 of the initial θ parameters, 1/200 of the initial $\phi_{i,0}$ parameters and 0.01 for the autoregressive ϕ parameters. The last variant adds the permutation of the parameters during the procedure (P).

Maximization of the log-likelihood

For each procedure, we measured the time to convergence, the maximal log-likelihood reached and the number of calls to the function computing the log-likelihood of the HMTD model. The latter measure gives us the best indicator of efficiency of each procedure because the effectiveness of the implemented code is not considered and the time of evaluation of the log-likelihood is much more important than the one of the creation of a new parameter vector (or solution) by each method. Globally, the execution time appears to be proportional to the number of function calls.

Table 3.2 provides for each dataset and each estimation algorithm the maximum log-likelihood, the number of calls of the objective function (or the corresponding number of iterations for some algorithms) required to achieve this maximum, and the optimization time in seconds. We can see that among the existing methods, PSO has a good overall performance in terms of both speed and achieved maximal log-likelihood values. Its

Table 3.2: Comparison between the different versions of the hill-climbing heuristic, PSO, SA, GA, L-BFGS-B, Nelder-Mead, and DE: For each computation, we provide the log-likelihood, the number of function calls, and the running time in seconds (between brackets). NA means that the algorithm was unable to converge to a usable solution. The best solution found for each dataset is in bold.

R dataset	**S**	**SE**	**SEM**	**SEMP**	**PSO**	**SA**	**GA**	**LBFGSB**	**NM**	**DE**
Seatbelts	-176.93	**-171.48**	-171.70	-172.46	-174.87	-174.98	-178.69	-179.82	NA	-175.05
	507(35.2)	503(37.9)	503(38.1)	500(38.0)	490(36.9)	768(58.0)	1000(63.2)	35(34.3)		42(94.7)
UKDriverD.	-130.22	**-127.79**	-128.36	-128.27	-132.66	-132.66	-135.47	-139.44	NA	-132.74
	507(17.9)	506(19.1)	502(18.8)	508(19.8)	490(18.7)	716(26.7)	1000(30.9)	10(4.9)		42(47.9)
sunspot.m.	-119.45	-122.70	**-112.18**	-118.51	-117.86	-117.86	-122.56	-122.40	NA	-117.28
	507(26.4)	509(28.6)	510(28.7)	511(28.8)	490(28.2)	547(32.1)	1000(49.2)	36(26.2)		42(69.7)
faithful	-69.03	-67.82	**-57.34**	-60.99	-71.40	-85.10	-73.07	-60.07	-58.18	-71.42
	507(44.8)	500(48.4)	501(48.1)	511(49.2)	490(51.3)	287(29.2)	1000(77.1)	40(49.5)	501(47.8)	42(126.8)
JohnsonJ.	**-39.60**	-41.20	-39.63	-39.75	-41.19	-41.17	-44.33	-40.29	-39.94	-41.35
	507(17.4)	504(18.7)	508(18.9)	501(18.8)	490(18.0)	300(11.0)	1000(30.8)	42(20.1)	501(18.4)	42(46.3)
lh	-29.53	-33.40	-32.71	-32.40	-29.08	-29.06	-34.50	-21.46	**-20.88**	-29.91
	507(51.9)	508(56.3)	510(56.5)	509(56.8)	490(59.2)	500(56.5)	1000(94.1)	36(51.4)	501(55.2)	42(139.9)
ldeaths	-535.03	-534.47	-533.39	-537.41	**-523.81**	-592.85	-591.14	-554.66	NA	NA
	507(78.4)	509(85.7)	504(83.3)	507(83.7)	490(80.6)	500(78.1)	1000(137.0)	7(14.9)		
nottem[1:10]	-26.25	-22.59	-22.60	-22.59	-22.91	-22.89	-26.78	-23.22	**-22.51**	-23.49
	507(7.9)	509(8.8)	508(8.6)	519(8.9)	490(8.4)	500(8.4)	1000(14.0)	41(8.9)	501(8.4)	42(21.2)
nottem[1:30]	-92.75	-85.17	**-85.15**	-85.25	-88.77	-88.77	-97.13	-88.89	-95.01	-89.35
	507(25.4)	506(27.7)	507(27.9)	509(28.5)	490(26.8)	500(27.3)	1000(45.8)	42(29.8)	501(27.0)	42(69.6)
lynx	-157.05	-157.34	-157.72	**-156.65**	NA	NA	NA	-166.79	NA	NA
	507(18.0)	504(19.2)	505(19.3)	509(19.9)				31(15.3)		

closest concurrent is the Nelder-Mead simplex optimization algorithm, whose main drawback is the lack of possibility to introduce constraints. Unfortunately, this is a major issue in our case, because some infeasible solutions may spuriously achieve higher log-likelihood values (for instance: autoregressive coefficients above 100'000). It happened several times during our experiments, therefore we had to ban the Nelder-Mead method despite its overall good performance.

The genetic algorithm was too slow in our experimets, and it was surpassed by all other methods. A possible reason is that GA appears to perform well in very difficult problems, and not well enough in simple ones, as suggested by Pukkala & Kurttila [114]. The simulated annealing, L-BFGS-B, and differential evolution performed rather well, but not better than PSO and NM. These results suggest that PSO remains the biggest competitor of our hill-climbing heuristic.

Among the different versions of our heuristic, it is difficult to determine an overall best method. It appears that the "one-directional" check enhances the procedure, allowing us to use the function calls elsewhere instead of wasting them to check both directions. Introducing a minimal parameter change is definitely better even if we have less precision in the answers. However, the permutation of the parameters according to their importance for the log-likelihood improvement does not seem to improve the performance. Therefore, our choice tends to the third method (SEM).

Acceptability of the solutions

A higher log-likelihood of a given solution does not necessarily imply its superiority over another solution. It is also very important to discuss the usefulness of a potential solution in terms of its interpretability before we accept it. Therefore, we need to examine the values of the hidden parameters (A and Pi) that one solution implies. For instance, if we test a model with two hidden components but the hidden transition matrix A of our solution suggests that one of the states is improbable, we may reject that solution because a simpler model could be more appropriate, the improbable state being probably associated to only a very few number of (maybe extreme) observations.

As an example, we chose our experiment with the "JohnsonJohnson" data. This choice was made because of the availability of answers from all procedures, and because of the proximity of the best log-likelihood values achieved. These data represent the quarterly earnings in US dollars per Johnson & Johnson share during the period 1960-1980 (Figure 3.2). The Figure suggests a different behavior before and after 1970, with a higher variability in the second part of the series. A two-component HMTD model

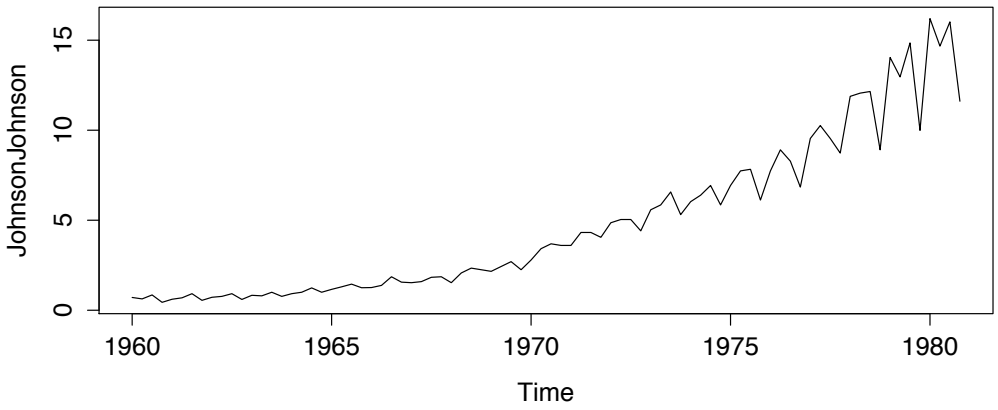


Figure 3.2: The "JohnsonJohnson" dataset.

could then prove to be appropriate.

Let us first explore the visible parameters of the solutions of each procedure (Table 3.3). If we take a close look at it, we can see that some of them appear to be relatively close to each other. For instance, the solutions of our four hill-climbing heuristic all assume one component with a low standard deviation ($\theta_{0,1}$ around 0.5 except for "S"), and a slightly negative autocorrelation for the mean ($\phi_{1,1}$ between [-0.107;-0.069]). The other component has a relatively large standard deviation and positive autocorrelation for the mean.

If we look at the solutions (Table 3.3), we can also observe a common part in the solutions of PSO, SA, GA, and DE. However, in this case, only one component appears similar in their solutions: the second one for SA and the first for PSO, GA, and DE, with θ_0, ϕ_0, and ϕ_1 parameters in the intervals, respectively [3.59;6.25], [3.95;8.05], and [0.16;0.61]. Note that the order of the two components is random and that we chose not to change it in presenting the results. This logic is fully confirmed if we look at the transition matrix A and the initial probabilities Pi (Table 3.4). Values of zero and one indicate that only one component of the model is used for the entire sequence of observations (probably more iterations are necessary), whereas the hill-climbing heuristic, NM, and L-BFGS-B algorithms use both components equally to model the data sequence. To go one step further and to decide whether a simpler one-component model should be used to model this particular dataset, we could compute different HMTD models, and rely on the Bayesian information criterion (BIC).

Observing the log-likelihood values, we also see that even if they are all indeed very

Table 3.3: Optimal solutions: visible parameters for each method on the Johnson and Johnson dataset.

	$\theta_{0,1}$	$\theta_{0,2}$	$\phi_{0,1}$	$\phi_{0,2}$	$\phi_{1,1}$	$\phi_{1,2}$
S	12.20	8.079	8.871	9.194	-0.069	0.181
SE	0.506	4.538	8.151	9.358	-0.107	0.154
SEM	0.512	4.082	7.823	8.626	-0.069	0.192
SEMP	0.562	4.086	8.153	8.581	-0.099	0.182
PSO	3.674	9.629	3.953	60.26	0.611	0.033
SA	5.148	3.594	55.02	4.175	0.225	0.585
GA	6.250	6.216	8.056	41.30	0.165	0.265
L-BFGS-B	5.113	1.970	7.782	4.190	0.437	0.482
NM	3.118	0.238	3.042	8.702	0.754	0.051
DE	3.747	10.27	4.892	51.24	0.497	0.408

Table 3.4: Optimal solutions: hidden parameters for each method on the Johnson and Johnson dataset. Since each row of both the transition matrix A and the probability distribution of the first two hidden states Pi is a probability distribution of two elements summing to one, we only provide the first one of each of them. $A_{11,1}$ is the transition probability from the state defined by $X_{t-2} = 1$, $X_{t-1} = 1$ to $X_t = 1$; $A_{21,1}$ is the transition probability from the state defined by $X_{t-2} = 2$, $X_{t-1} = 1$ to $X_t = 1$, Pi_1 is the unconditional probability of the first hidden state to be $X_1 = 1$, and so on.

	$A_{11,1}$	$A_{21,1}$	$A_{12,1}$	$A_{22,1}$	Pi_1	$Pi_{1,1}$	$Pi_{2,1}$
S	0.38	0.60	0.40	0.37	0.59	0.58	0.59
SE	0.53	0.69	0.41	0.06	0.85	0.84	0.88
SEM	0.49	0.66	0.39	0.07	0.83	0.82	0.86
SEMP	0.50	0.65	0.38	0.07	0.79	0.78	0.82
PSO	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SA	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GA	1.00	1.00	1.00	1.00	1.00	1.00	1.00
L-BFGS-B	0.40	0.53	0.44	0.33	0.18	0.22	0.17
NM	0.56	0.84	0.70	0.71	1.00	1.00	1.00
DE	1.00	1.00	1.00	1.00	1.00	1.00	1.00

close (which is the reason why we chose this dataset), all methods from the first group achieved values similar to each other and slightly higher than those from the second group. That lead us to two suggestions. The first one is that the second group was probably trapped in a local optimum using only one component and it was not able to escape from it. The second is that the model specification with two components is slightly more appropriate for this dataset compared to another one using only one component (in which case the latent part of the model would not have been necessary). In other words, using two components, and thus using the hidden layer of the HMTD model, improved the modelling.

3.5.3 Sequence length and speed of convergence

We performed a last set of numerical simulations to evaluate the influence of sample size on the different heuristics, and to compare their speed of convergence. We generated data sequences according to a two-component HMTD model. Both components followed a Gaussian distribution with variance $\sigma^2 = 0.5^2$ and mean $\mu_t = 1 + 0.2 * x_{t-1}$ for the first one, and $\sigma^2 = 2^2$ and $\mu_t = 3 + 0.6 * x_{t-1}$ for the second one. The probability to start with the first component was set to 0.75, and the hidden transition matrix between components was

$$A = \begin{bmatrix} 0.75 & 0.25 \\ 0.40 & 0.60 \end{bmatrix}$$

We considered 5 sequence lengths (15, 25, 50, 100, 200 and 300 data points), and we generated 200 sequences of each length.

Number of function calls and convergence

The various optimization methods follow very different procedures, and rely on very different convergence and stopping criteria. Therefore, they also require very different amounts of time to reach an optimum, what is crucial in comparing them, because a procedure that requires more calls of the objective function has higher chances of reaching a better solution, but at the cost of more computing time. For instance, even a slight change in the stopping criterion of one specific method could result in a slightly better performance in exchange of a higher execution time. The speed of convergence to the optimum is therefore an important criterion when choosing an optimization method.

We decided to compare the different heuristic by allocating them a fixed maximal number of log-likelihood function calls (500). Often, the different methods managed to

converge with this number of function calls (perhaps because of the bounded parameter space), but some did not succeed. This raises the question of the presence of error due to the non-convergence of some methods during some iterations. A possible solution is to only analyse the solutions that have converged. However by doing this we would omit the fact that for a given dataset, some methods did simply not succeed to find an acceptable solution, whereas others did. As we would inevitably obtain different proportions of solutions, we would arrive in a situation where we compare only the successful solutions, what could benefit the method with the less tolerant stopping criterion. Therefore, we chose to include all solutions obtained with a given number of function calls. This probably introduces a non-convergence error in our analysis, but results allow us to fulfil the main objective of the simulation, that is to find the fastest procedure that offers acceptable solutions by treating equally all the methods.

Results

Table 3.5 summarizes our results by providing for each sequence length and each optimization method the mean and standard deviation of the log-likelihood of the 200 datasets.

Table 3.5: Results of the simulations with 500 log-likelihood function calls. We generated 200 series of each data length, and we provide the mean and standard deviation of the 200 log-likelihood.

Data length		SEM	SEMP	NM	PSO	L-BFGS-B	GA	DE
15	μ	-25.42	-25.87	-24.80	-25.36	-26.13	-27.97	-27.74
	σ	6.05	6.75	6.17	6.03	6.06	10.85	6.97
25	μ	-45.83	-46.85	-44.93	-45.67	-46.96	-53.84	-49.74
	σ	8.05	10.13	8.05	8.09	7.74	86.21	9.84
50	μ	-97.62	-100.28	-96.84	-98.41	-100.25	-110.43	-107.60
	σ	10.59	13.99	10.44	11.25	10.12	85.15	20.42
100	μ	-200.62	-204.38	-199.18	-202.21	-205.23	-234.00	-219.43
	σ	15.45	19.21	14.60	15.64	14.50	388.30	29.92
200	μ	-408.95	-416.68	-406.25	-414.84	-417.99	-450.80	-450.95
	σ	21.64	31.83	20.27	25.23	22.31	270.48	78.72
300	μ	-615.56	-626.65	-612.02	-625.61	-630.34	-668.43	-678.95
	σ	26.44	40.70	25.13	31.97	30.60	151.42	138.86

Results show that Nelder-Mead appears to be the best method to maximize the log-

likelihood of the model, right before the SEM and PSO. However, even though it is very popular in numerical optimization, as we mentioned, this method suffers from a major issue: the impossibility to fix constraints. This may often be problematic, especially if we analyse small datasets where spurious optima exist. Among examples that we experienced during the optimization are negative standard deviations, and exceedingly high autocorrelation values, leading to non-interpretable solutions, even if they may be better from a strict mathematical point of view. For these reasons NM should be used only if the datasets are large enough and there are no hard constraints imposed by the model or the nature of the data.

The other methods compared here do not suffer from such a limitation, and among them the SEM and PSO appear as the best choices. The advantage in favour of SEM increases with the length of the data sequence. On the other hand, it appears that permuting the re-estimation order of the coefficients is not useful (SEM leads to better results than SEMP). Overall, our new heuristic behaves well against its competitors allowing to reach good and interpretable results in all situations.

3.5.4 Simulated data experiment

We run a simulation experiment to be able to compare the results obtained using each optimization procedure with the parameters that generated the data. After a burn-in period of 2000 data points, we simulated one sequence of 500 data points by using a model with two hidden states, first order dependence for the latent level, constant variance and one lag for the mean at the visible level. The true visible-level parameters were: $\theta_{0,1}$=8, $\theta_{0,2}$=3, $\phi_{0,1}$=4, $\phi_{0,2}$ =1, $\phi_{1,1}$ =0.3, $\phi_{1,2}$ =-0.2. The hidden transition matrix was defined as:

$$A = \begin{pmatrix} 0.3 & 0.7 \\ 0.2 & 0.8 \end{pmatrix}$$

and both hidden states had the same 50% initial probability. The visible parameters and the log-likelihood achieved by each procedure are given in Table 3.6. We observe that all methods reached a very similar value for the log-likelihood. However, the parameters are not very similar even if they approach their true value. The largest difference is observed for the standard deviations ($\theta_{0,1}$ and $\theta_{0,2}$). We note that in this case L-BFGS-B and SEM give us slightly higher log-likelihoods, but their solutions are not the closest ones to the parameters. In both cases, the constant of the mean of the first component ($\phi_{0,1}$ = 15.01 and 15.44) compensates for the positive auto-regressive coefficient $\phi_{1,1}$, which tends to vanish. Similar problems are encountered with the Nelder-Mead and

SE algorithms, suggesting that an important local optimum is present. We hypothesize that by increasing the length of the sequence, the auto-regressive coefficients will capture better the time dependence (if such exists) and therefore they will become more important to discriminate the sequences, what is likely to solve this problem.

	$\theta_{0,1}$	$\theta_{0,2}$	$\phi_{0,1}$	$\phi_{0,2}$	$\phi_{1,1}$	$\phi_{1,2}$	LL	f.calls
True param.	**8**	**3**	**4**	**1**	**0.3**	**-0.2**		
SE	11.83	3.853	14.56	1.224	0.058	-0.166	-1496.71	5000
SEM	18.12	5.034	15.44	1.230	-0.044	-0.171	-1496.17	5006
SEMP	44.52	1.406	3.536	1.123	0.516	-0.218	-1496.27	5008
PSO	43.80	14.60	3.583	1.129	0.527	-0.222	-1496.26	4662
GA	36.49	14.94	5.546	0.708	0.476	-0.090	-1504.15	5000
L-BFGS-B	17.93	14.60	15.01	1.207	0.000	-0.166	-1496.07	49
NM	8.511	6.343	14.80	1.208	0.028	-0.160	-1496.64	345

Table 3.6: Optimal solutions for the simulated data: Visible parameters obtained using each method after 5000 iterations, log-likelihood, and number of calls to the function evaluating the log-likelihood.

On the other hand, the three other methods (SEMP, PSO, GA) have detected the positive parameter $\phi_{1,1}$, what resulted in a more accurate estimation of $\theta_{0,1}$ as well. In general, the estimation of the standard deviation parameters remain quite inaccurate, apart from the fact that all the procedures estimated correctly a smaller standard deviation for the second component. By observing these results, we may think that as the number of data points increases, the estimation procedures increase their accuracy. This intuition is confirmed after replicating the same experiment with four of the optimization algorithms on a sequence of 1000 observations (Table 3.7). However, that seems to apply to the parameters related to the mean of each component, but not to the standard deviation.

Properties of the proposed heuristic optimization and the GEM algorithm

If we explore the properties of an EM algorithm, we know that the M step ensures the increase of the marginal log-Likelihood and thus (most often) the convergence to an optimum that may be global or just local.

	$\theta_{0,1}$	$\theta_{0,2}$	$\phi_{0,1}$	$\phi_{0,2}$	$\phi_{1,1}$	$\phi_{1,2}$
True parameters	**8**	**3**	**4**	**1**	**0.3**	**-0.2**
SEM	47.32	0.767	3.069	0.938	0.419	-0.205
SEMP	47.11	0.186	3.175	0.947	0.402	-0.186
PSO	39.73	13.24	2.967	0.968	0.408	-0.212
GA	19.90	15.61	16.26	1.037	-0.069	-0.123

Table 3.7: Optimal solutions performed with a sequence of 1000 observations: visible parameters obtained using each method after 5000 iterations.

Recall that the "expectation" step computes the fixed parameters of the latent part and the maximization step is then used to re-estimate the visible-level parameters by maximization of the model's fit to the data. As discussed, because of the non-existence of unique closed form solutions of the derivative equations of the M-step in case of non-constant component variance, it is replaced by a heuristic optimization. The convergence of the resulting Generalized EM (GEM) algorithm depends on this maximization procedure.

In general, heuristic procedures rely on systematic behaviour of random evolution and therefore their convergence cannot be proved mathematically. This is also the case of the used heuristic. However, the way that this procedure is implemented also simulates to an extent the behaviour of the M step: only parameter values that increase the Likelihood are accepted and if this is impossible after exploring all directions in every dimension of the parameter space, the step size is reduced before the next iteration. If the smallest modification of the parameters is still not beneficial, then one assumes that a local optimum is reached and the procedure is restarted. Of course, this procedure is computationally expensive and this must be taken into account when the complexity of the solution space increases.

By construction at every iteration i of the modified M-step only visible-level parameters that increase the likelihood are accepted, which indicates:

$$Q(\Theta|\Theta^{i+1}) \geq Q(\Theta|\Theta^{i})$$

As showed by Little and Rubin [87] (chapter 8.4.1), the aim of each M-step of EM (and GEM) is to find a Θ^{i+1} that improves the complete-data likelihood $Q(\Theta|\Theta^{i+1})$, but this also leads to increasing the likelihood function of the observed data at each step.

Therefore the monotonicity property of the ordinary EM algorithm is preserved in

the GEM (see also section 3.2 and 3.3 of McLachlan and Krishnan [93] for more details). However, the rate of convergence with the heuristic optimizations is generally slower because of the lack of information obtained from the likelihood function derivation and the need of exploration of the neighbourhood instead.

In the procedure that we implemented, some of the parameters can be constrained. Note that if this is done, the convergence of both the EM and the GEM algorithm is not granted: McLachlan and Krishnan [93] cite several examples in the literature, in which the algorithm is guaranteed to converge and a few where it is not. Specific proofs of convergence are therefore required for each problem. The convergence properties of a GEM algorithm have been showed in Dempster, Laird & Rubin [41] and Wu [170]. However, the latter also mentions some exceptions in which the EM algorithm converges to a saddle point instead of to a local optimum. GEM also does not necessarily converge to a single point (see McLachlan and Krishnan [93] and the references therein).

3.6 Discussion

In this chapter we proposed a new heuristic approach for the M-step of the GEM algorithm optimization of the HMTD model. Our motivation was that the HMTD model is very complex, with many constraints on the solution space, and hence, standard available algorithms may have difficulties in finding acceptable solutions. Of particular importance is the fact that the log-likelihood function of the HMTD is difficult to differentiate. Thus, our approach does not use any derivatives.

Since our method can be qualified as a hill-climbing method, it can be compared to other neighbourhood search or hill-climbing methods (stochastic hill climbing, random restart hill climbing, etc.). Therefore, it also shares some of the issues of these methods. Among them is the ascension of *ridges* (or descend of *alleys*): All of these methods update each dimension separately, and therefore, if the direction of the ridge (or alley) is not aligned to the axis of one dimension, the algorithm has to progress in zig-zag, spending more time. Such problems may be solved, but we need to be able to detect the problem first in order to eliminate it. A typical situation of ridge is manifested when a change in one dimension introduces a possibility to improvement in another dimension that was not possible until then. If this relation between two variables continues on the same sense for more than two consecutive iterations, we can include the *simultaneous change* feature that we developed before. This feature should spare a lot of unnecessary steps. However, when the number of dimensions increases, the detection of such situations becomes more difficult. We need to highlight that the possible inclusion of

simultaneous change distinguishes such procedure from standard hill-climbing methods, because it allows the solution to evolve in more than one dimension simultaneously.

Similar to the majority of optimization algorithms, our heuristic procedure requires to start from a good set of initial values. Therefore, it is a good idea to draw a number of randomly chosen points in the parameter space, evaluate them, and select either the fittest one or the centroid (mean or median point) of the best five solutions. This solution may be beneficial for our procedure despite the additional computations. Our approach is also better adapted to smooth functions, and its drawbacks are obvious if we test it on more rough functions with many local optima. In this case, our approach brings us only to the nearest optimum. The solution then is to add some noise to the parameters after reaching an optimum in order to escape this optimum in the case it is a local one, what we called "jittering".

During a test with the Rastrigin function with 40 parameters, we also observed another interesting particularity: the re-estimation step needed to be higher when we attempted to optimize high-dimensional problems. The reason was probably the diminishing sensibility of the function to the modification of a single parameter as dimensionality increases. When the outcome remained unmodified after changing one parameter, the algorithm was trapped into the current solution. A possible remedy to this situation could be to introduce higher initial values for the re-estimation step and to rise the maximum possible value of this step as the number of dimensions increases.

Another issue is that for some solutions, the computed log-likelihood decreases even below the minimal number that the machine can consider (especially if we calculate it using highly unlikely values for the parameters). As it is easier to work with log-likelihood instead of the likelihood itself, we need to ensure that the returned value of the likelihood is not rounded to 0 ($\log(0) = -\infty$). To fix this problem, we impose the likelihood to be greater or equal to the minimal number for the machine that we use. This problem also shows the importance of the initial solution to our approach in order to avoid areas where the objective function is flat (i.e., the log-likelihood around remains null despite the changes in the parameters). These flat areas are also an issue for the local search methods, and they are one more reason to include bounds.

In addition to the different approaches used throughout this paper, another alternative would be the meta-optimization, implemented first by Mercer & Sampson [98], which applies one optimization method to tune the parameters of another one. This procedure has been previously applied to many situations, but in our case, it does not guarantee an improvement of performance, especially if we modify the configuration of the model (number of lags, latent states, etc.). The notion of hyper-heuristics is also

worth mentioning. These are techniques that are applied to heuristic methods in order to determine the most appropriate of these for a given problem or, alternatively, to generate a new heuristic method by combining existing heuristics. The aim is to find a more general optimization procedure. However, again, we do not know whether the objective function remains similar when we change the specification of the model, which is necessary when we choose the number of components for instance. Moreover, as observed, the best optimization method changes in function of the chosen specification of the model, even when applied on the same dataset.

Even if our heuristic shows good performance, it could be improved in different ways. First of all, and as already mentioned, the performance of our hill-climbing approach is influenced by the initial solution. As the speed of convergence to a local optimum is rather fast, in simple problems we may want to introduce parallel computing starting from different points instead of using the "jitter" procedure. That would allow the algorithm to explore the presence of any local optima, without consuming additional time in practice, provided that nowadays most computers have multi-core processors. The performances of most of the procedures tested here are also influenced by the limits of the parameter space. Since the choice of these limits is very arbitrary, an introduction of "floating" limits (limits that serve as orientation, but are broadened as soon as the current best solution approaches them) may be a solution. Another possibility, directly related to the structure of the HMTD model, would be to estimate simultaneously, instead of sequentially, the visible and latent parameters. In this case we would need to introduce constraints on the hidden transition matrix A in order to ensure that it will remain a proper transition probability matrix. Such estimation procedure appears to be computationally demanding taking into account the increasing number of dimensions.

The complexity of the solution space arising from statistical models such as the HMTD (especially when the number of components, lags and/or covariates increases) and the additional specificities associated with each particular dataset imply that no one optimization algorithm can be demonstrated to be always the best, and that even with a good algorithm, the fine tuning of its parameters can have a very high impact on the final result. That being said, the contingencies of applied research imply that finding the best overall solution in terms of fit to the data, hence of log-likelihood, is not always required. In most situations, increasing the log-likelihood by one or two points is useless, since it will not imply a dramatic change in the parameters, hence in the interpretation that will be made of the model. The focus then has to be put on the speed of convergence (what disqualifies GA) and on the probability to find an acceptable solution, that is one that is not influenced by the boundaries of the solution

space and that is sounds in regard of the dataset. Regarding these requirements, the new heuristic presented in this paper for the HMTD model works well by minimizing the number of situations with useless results, and it could be applicable to many similar statistical models.

Chapter 4

Clustering uncertainty

In this chapter we will explore important concepts that are necessary to understand a complete mixture models clustering procedure from the beginning to the end. We focus on frequentist estimation and our major objective is to assess how uncertain the parameters of the clustering solution are and how to obtain confidence intervals for the parameters. To understand the problematic, we need to discuss the estimation procedure, the important ways to choose an optimal clustering solution, the problems that arise during the construction of the intervals based on this solution (label-switching among others). Because bootstrap is the method found most appropriate to obtain these intervals, we will explore how is it possible to overcome the label-switching and other problems when bootstrapping the data. At the end, an important overview which summarizes the entire procedure will be provided.

4.1 Introduction

Cluster validation is a complex task with multiple aspects on which there is rarely consensus. Evaluating the quality of clustering is important for every researcher. This is true for not only the choice of number of clusters and clustering method but also the reliability of clustering. Different aims of clustering require different features: for example, pattern recognition requires the separation of clusters, social network analysis requires that small within-cluster distances be researched, and the clustering for information reduction requires that the resulting components both represent the data well enough and neutralize the outliers present in the original variables Hennig et al. [66].

In all the domains of application of clustering, however, a major concern is uncertainty of the obtained solution; that is, how sure can one be that the obtained clustering

reflects the true structure of the population.

Several distinct types of uncertainty generally exist in computer data modeling, according to Kennedy & O'Hagan [78]. The authors classify them as follows:

- *Parameter* uncertainty is one of the parameter values used in computer data modeling. These values are unknown and often correspond to some features of the data.

- *Model* uncertainty arises from the model?s adequacy for the data. All models have underlying assumptions supposed to correspond to the true underlying data-generating process. However, because the models are only approximations of the reality, there always exists a difference between estimated and real data distribution.

- *Residual variability* indicates the difference between the outputs of a particular model on two different occasions (points in time, for instance) even when all the conditions are exactly the same.

- *Parametric variability* arises when some parameters cannot be specified and are left to vary according to some approximate distribution.

- Uncertainty can arise also from a *measurement* (observation) error. It is inevitable especially when the data are not directly and objectively observed (for example, in social sciences, the observations of individuals are often reported by themselves).

- Uncertainty about coding is another problem that must be considered.

Numerical uncertainty is also important, especially in complex models. Numerical optimization or approximation may often introduce numerical errors. This is also true for the heteroscedastic mixture transition distribution (HMTD) models that use the generalized expectation maximization (EM) algorithm with a forward-backward algorithm to maximize (or estimate) log-likelihood. Interpolation and extrapolation due to the lack of data in some regions are also a source of errors. The unequal spacing between time series observations is a simple example of a cause for such uncertainty.

Uncertainty can broadly be categorized as aleatory and epistemic types, according to Der Kiureghian & Ditlevsen [42]). The former type encompasses all uncertainty that cannot be reduced by the researcher. On the contrary, epistemic uncertainty may be reduced by collecting more data or enhancing the model, for instance. The poor distinction between these two uncertainty types may lead to the prediction of a very

small part of data variance (with a bias toward aleatory uncertainty), or to over-fitting and the modeling of spurious relations between the variables when the epistemic part is overestimated. Although the distinction between these two broad uncertainty types is not obvious in most situations, epistemic uncertainty may be a good topic for general discussion.

Here, we distinguish between the search for optimal clustering in terms of model choice and number of clusters, and inference on model parameters; that is, their estimation, significance, and variability. Although we focus on the latter, because the distinction between the two parts is not obvious, we need to discuss both parts.

In the frequentist framework for mixture models, parameter inference may be performed using bootstrap in three different ways, that we will be detailed later in this chapter. One method is to resample from the entire sample before fitting the clustering model. By doing this, we mix the parameter and model uncertainties, but this can make our task much more complex. In another approach, we can isolate both uncertainty types. A third approach is a hybrid solution by which we assume that the clustering is correct and draw "stratified" samples, but still we mix both types of uncertainties. This approach may be appropriate if a small-size cluster is discovered in the optimal clustering.

In general, once we choose a model and approve the clustering of a given number of clusters, several questions arise:

- To what extent are the parameters of the chosen solution meaningful?
- How much would they vary if another sample is drawn from the same population? and
- Of the optimal parameters, which are the ones truly important to define the clusters, and which are the ones spuriously estimated?

Let us now take a small example on the importance of parameters. Assume that we have a two-component Gaussian Mixture Model (GMM) for the optimal clustering of a two-dimensional dataset and that we estimate a given covariance matrix. We find the covariance between the variables ($\hat{\sigma^2}_{1,2}$), but how can we be sure that this covariance (i.e., an elliptical shape of the cluster) really exists? If no difference can be found between the variances and covariances for each cluster, then the much simpler k-means clustering variance minimization model would be as appropriate for the problem as the GMM. Such questions are important because the aim of the researcher is often not only to find a good clustering for the given sample, but also to understand the nature

and specificity of the clusters in the underlying population. Therefore, the clusters? stability and significance are critical.

In this chapter, we first introduce the use of bootstrap in mixture models for clustering and then briefly explore the widely used alternative of Bayesian analysis for such models. Section 2 introduces the well-known label-switching issue and some solutions to it. Section 3 focuses on the parameter inference approaches and presents the proposed bootstrap procedures in frequentist estimation. Finally, we explore some methods to choose, compare, and validate an initial stable clustering solution, on which the proposed bootstrap methods rely. In the following chapter, we will provide an example that illustrate these points.

4.1.1 Mixture models for clustering

Mixture models are frequently used for data clustering. However, most of the studies consider only univariate or multivariate transversal data. Therefore, we first briefly overview some of the important characteristics and issues of the mixture models most often discussed and used with transversal data, although most of them can also be applied to longitudinal data.

In a basic variant of the GMM, mixture models are similar to nonprobabilistic K-means clustering, except that the clustering method is "softer." In other words, the observations cannot be defined as arising from one particular mixture, but instead have a probability of arising from any of the clusters. The GMM is also more flexible, allowing the variances and covariances to change, thereby introducing elliptical distributions. To illustrate the similarity between the two clustering models, note that we can obtain the same answers from GMM and K-means clustering by modifying just two features of the GMM. The first modification is to fix the variances of each GMM component (ex: $\theta_c = 1$) and the covariances between them, and the second is to make the prior distribution of the GMM uniform on each iteration of the EM algorithm (see Bishop [23] for more details on GMM)

Mixture models belong to the so-called *ill-posed problems* because they do not satisfy all the three properties of a well-posed mathematical problem (i.e., a solution exists, the solution is unique, and the behavior of the solution changes continuously with the initial conditions). More precisely, one can have multiple solutions, and, most importantly, small modifications in the data have large impacts on the results. Two completely different parameterizations can lead to a similar joint distribution, meaning that a unique solution does not necessarily exist. This is also an *inversed problem* because the

data provide information on the parameters indirectly; that is, we extract information on the data-generating process from the data itself. Thus, there actually exists a non-null probability that one of the model?s components is empty and the sample does not provide any information about its parameters; this possibly explains why the likelihood function can become unbounded Marin, Mengersen and Robert [89].

The optimal solution in mixture models can typically be found by maximizing the likelihood. Depending on the model complexity, it could be difficult to find an optimal solution, especially because the likelihood function can contain multiple local optima. A major reason for this is the identifiability problem (or the genuine multimodality problem). Furthermore, for some models (such as the HMTD), deriving the likelihood function could be difficult, requiring the use of a heuristic procedure and thus more computational time.

Another important issue in parameter inference arises from the fact that a likelihood is invariant to a permutation of components. This implies that all other things being equal, every likelihood optimization solution can result in a different order of components. This is the so-called *label-switching problem*, which is common for mixture models and especially important when assessing the parameter uncertainty of a mixture model. This issue arises every time we try to infer the parameters specific to each component. The problem is often aggravated by the identifiability (genuine multimodality) problem, which we will discuss in more detail shortly.

Finally, we also find a *singularity problem*, which we will discuss in detail in the following section. When clustering data, assessing the uncertainty of the parameters is crucial. We need to find the significant parameters for every class and hence have to discover the particular features of each class (including covariates) in order to distinguish it from the others. We will discuss the classification issues in more detail later in this chapter.

4.1.2 Use of bootstrap in clustering

When discussing parameter significance and uncertainty in frequentist estimation, one should consider the bootstrap method. The bootstrap was introduced by Efron [46] in 1979 and represents a major development of the jackknife. One objective in clustering is to eliminate as much as possible the sampling error that we obtain when clustering a small sample in place of the true population. If the sample size n is very large, one may divide it and cluster the different independent parts, but since it is often limited in practice, the only way to approximate the true underlying distribution F of the data

is through the empirical distribution $\hat{F}_n$ of the sample. In bootstrap, we draw samples of size n with replacement from the original sample.

The bootstrap is widely used to empirically estimate the variability of a parameter estimate (mainly in its nonparametric form) in various classification and clustering techniques. For instance, in random forests, one forms bootstrap samples from the original sample to generate multiple training sets (bagging). This procedure has also been found useful in clustering, such as in averaging with k-means to reduce the computational time in large data samples Davidson and Satyanarayana [40], assessing cluster stability, and selecting the number of clusters Fang and Wang [50].

4.1.3 Estimation procedure: Frequentist or Bayesian

In this section, we discuss two methods commonly used to estimate the parameters of a mixture model: the frequentist and Bayesian approaches. The two approaches have different concepts for the parameters: in the frequentist perspective, the parameters are fixed and an error term may be computed around the estimates; this indicates how close the real parameters can be to the estimation. These error terms are often estimated through bootstrap resampling. From the Bayesian point of view, the parameters are not fixed but random variables with a given distribution. Both concepts have positive and negative sides, neither concept being superior to the other. However, one needs to discuss the advantages and disadvantages of both concepts before choosing one.

According to McLachlan and Peel [94], the Bayesian approach for mixture models entails some difficulties. Improper prior distribution, for instance, results in improper posteriors. The label switching due to lack of prior information on how to distinguish the components results in multimodality in the likelihood function (with $k!$ different modes, where k is the number of components). Label switching is a problem not in the maximum-likelihood (ML) estimation of the mixture, but in the Markov Chain Monte Carlo (MCMC), because the labels may switch between different iterations when sampling parameter realization from the posterior distribution. Although the authors stress that ML estimation is invariant on label switching and therefore not problematic, we must note that to assess the stability (or standard error) of the ML estimates, one needs to address this problem because after the re-estimation of each bootstrap sample, one should identify the components to which the parameter estimates of each iteration belong (their order is random).

Rydén [131] reviews the advantages and drawbacks of both the frequentist (EM) and Bayesian (MCMC) estimations for Hidden Markov Models (HMMs). He particu-

larly examines three cases: selection of the model order, continuous-time HMM, and the HMMs where several latent variables influence the observed data in an overlapping way. Bootstrap is used with the Expectation-Maximization (EM) algorithm to estimate the intervals for the parameter estimates. The comparison is purely from a computational perspective. The conclusion is that no approach is clearly superior to the other, but this depends on the specific problem. In his examples, the Bayesian analysis appears faster. On the other hand, in the frequentist approach, "i.i.d. bootstrap replicates require no analyses of correlations, etc., in order to assess the precision of the results." Therefore, the authors state that the serial dependence of the sampler imposes additional preparatory analyses; note that a single long sequence is estimated by the authors. Since the sequences of parameter samples are not independent from one another in MCMC, this dependence multiplies their variances by a constant C_α, which should be calculated. Therefore, one must check the consecutive MCMC iterations for whether the cases in which a parameter exceeds its CI upper bound, for instance, are autocorrelated, and to what extent. For instance, several of the most extreme parameter values can be obtained in just a few consecutive iterations. Using this information, one might estimate the extent of inflation of the parameter quantile variance and correct the problem (see the appendix in Ryden's article for details).

Note that their bootstrap procedure was parametric (we will discuss parametric bootstrap later in this chapter), indicating that the sequences were generated from the estimated model. Moreover, the label-switching problem was simple and could be resolved by just one identifiability constraint for the means. Of course, when we need only a point estimate, the EM approach is simpler and faster.

Dias & Wedel [45] compare the performance of the EM, Stochastic EM, and MCMC (see also Celeux, Chauveau & Diebolt [28] for a comparison between the EM and SEM) when estimating a Gaussian mixture, particularly when there are sharp ridges and a saddle point in the likelihood. The authors point out the slow convergence of the ordinary EM and the label-switching vulnerability of the SEM and MCMC, and attempt different methods to solve the latter issue. In the concrete example of Dias & Wedel [45], the SEM and MCMC appear to be faster, while the MCMC suffers less from degenerate solutions due to components with single observations (although this speed of convergence claim is not unanimously approved in the literature). The authors finally show the superiority of the MCMC for their examples with problematic surface of the likelihood function and warn about the costly implementation, especially the label-switching problem that attenuates the advantages of MCMC. In our opinion, the latter issue can be more dangerous when considering more complex mixture models.

The large possible number of parameters and lack of a common principle to distinguish the component parameters in the HMTD model can cause problems when solving the label-switching issue. This is the main reason why we prefer the EM approach instead of the Bayesian approach for the model estimation. Although the MCMC can be faster and accurate in simple HMTD specifications, in more complex cases we may not be able to estimate it owing to the latter problem.

Frequentist analysis of mixture models

The frequentist estimation of mixture models usually uses the EM algorithm or some of its variants (see Celeux, Chauveau & Diebolt [28]). EM estimation starts by choosing k points (number of clusters) randomly in the parameter space. This represents the means of the distributions. In the first step, we estimate the probabilities that each observation is generated from one of the k components, and the procedure continuous as detailed in Chapter 2.

One problem arises when we attempt to estimate a GMM using the ML framework, the *singularity problem* (see Bishop [23]). This issue arises when one component is stuck to a single observation of the data. Because it contains only one observation, the marginal distribution of the component becomes spiky and the variance becomes null. The resulting covariance matrix is singular (its determinant is null and therefore it is not invertible). This could happen with a single extreme value, for instance, and also when the variance of a component is very small.

When we consider the likelihood of the GMM

$$P(X|\Theta,\pi) = \sum_{i=1}^{K} \pi_i P(X|\Theta)$$

$$lnP(X|\pi,\mu,\Sigma^2) = \sum_{i=1}^{N} ln\left(\sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k)\right), \tag{4.1}$$

a null variance of the component k results in an infinite value of the likelihood of the data point j on which the component is stuck:

$$N(x_j|\mu_k = x_j, \Sigma_k = 0) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_t-\mu_k)^2}{2\sigma_k^2}\right) \to \infty, \quad \text{as} \quad \sigma_k \to 0. \tag{4.2}$$

As Bishop explained, the singularity problem does not occur in Bayesian analysis or when we have a single component model because the variance cannot be null if

we cannot have a single observation in the component. In the latter case, even if the components collapse at one point, the multiplicative likelihood due to the other data points will be null and the total likelihood will become 0. This is also another advantage of the separate bootstrap inference approach using one component, which we will discuss later, because the number of observations per component do not change throughout the iterations. For Bishop, this problem illustrates an over-fitting issue when the ML approach is used. A remedy for this would be to employ a suitable heuristic technique that detects the collapsing of a Gaussian component, resets the mean, and increases the variance.

Bayesian analysis for mixtures

As with the frequentist estimation, we consider it necessary to briefly introduce the Bayesian inference. The Bayesian mixture model has become very popular since the 1990s following the development of MCMC methods. One reason is that this method allows complicated structures of the model to be decomposed into simpler structures through the use of latent variables. Consider the Bayesian formula

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

. The quantity of interest in Bayesian inference is the distribution of parameters, given the data $P(\theta|x)$. In order to estimate this, we need to suggest a prior distribution for the parameters ($P(\theta)$). We also need to compute the likelihood which represents our thought (based on a model) of the data distribution, given the parameters ($P(x|\theta)$). $P(x)$ is the evidence that the data are generated by the model and represents the most complex part of the equation. To calculate it, we need to integrate over all the possible values of the parameters $P(x) = \int_\theta P(x, \theta)d\theta$. However, because we do not know their distribution, we often have to sample them using MCMC. Furthermore, because we use a Bayesian framework, we need to choose a prior distribution for every parameter we want to infer.

In sampling using MCMC, we first decide an initial arbitrary parameter value θ_0 (of dimension p). The Markovian part arrives when we choose to move to another proposed value θ_1 (or stay with the current one). This proposal point in the parameter space could be made in different ways—at random or by using a more complex approach with some assumptions. The Metropolis sampler, for instance, simply draws from a Gaussian distribution around the current point within a determined range (standard deviation).

In order to decide whether the new point θ_1 is better and we should accept it, we calculate the probability of obtaining the data from the model with this parameter value; that is, we compute the log-likelihood ($P(x|\theta_1)$. However, we do not have to simply find the optimal value (and therefore be stuck with it), but rather get a posterior distribution of the parameters; that is, we obtain a sample of parameter values with more points in the regions with higher probability. Some acceptance probability is often attributed to the new point even when it does not increase the log-likelihood. However, this probability usually increases with the likelihood of the new point, since the aim is to obtain a posterior distribution covering the more probable regions of the parameter space. Therefore, each region could be visited according to its probability approximated by the log-likelihood using the acceptance ratio.

If we draw a new point in the neighborhood with a higher probability, we usually retain it, but if its probability is lower, we can still accept it. Therefore, the sampler moves around the high-probability regions and does not stay for long in the lower ones.

To sample from the correct region of the joint distribution $\theta = \{\theta_1, \ldots, \theta_k\} \sim p(\theta_1, \ldots, \theta_k)$, we have various types of samplers. The most well-known samplers are the Metropolis-Hastings (MH) and Gibbs samplers. Therefore, we briefly introduce them.

In random-walk Metropolis algorithms, generally, the acceptance probability is important and should not be too small or too large.

An advantage of the MH sampler is that we do not have to know the posterior distribution beforehand. If we have an intuition of the posterior distribution, however, and we have a multidimensional problem but are not able to directly sample from the joint distribution, another approach may be more appropriate.

The Gibbs sampler is applied in such situations because it does not need to sample from the joint distribution, but sample alternately from the conditional distributions. Gibbs may be seen as a special case of MH when we have the full conditional distributions (i.e., the conditional distributions of every θ, given all other variables $P(\theta_i|\theta_1, ..., \theta_i - 1, \theta_i + 1, ..., \theta_k)$). The Gibbs sampler has been used for the estimation of mixtures even before Tanner & Wong [151] it became popular through Gelfand & Smith [56] (see Marin, Mengersen and Robert [89]). In short, the procedure is as follows (if $\theta = \{\theta_1, \theta_2\}$):

1. Initialize the parameter values $(\theta_1^{(0)}, \theta_2^{(0)})$.

2. Then, sample from $\theta_1^{(j)} \sim p(\theta_1|\theta_2^{(j-1)})$ (for instance, the unobserved variables indicating the component generating the observation, conditional on the means and variances of the components).

Contrary to the MH sampler, Gibbs sampling always accepts the sample.

3. Now, using the estimation obtained above, sample from $\theta_2^{(j)} \sim p(\theta_2|\theta_1^{(j)})$.

4. Iterate the last two steps until convergence.

In general, each θ_1 or θ_2 may be multivariate for the Gibbs sampler.

Note that the sequence of parameter estimations $\theta^{1,\ldots,J}$ follows a Markov chain and is therefore not independent. The relaxation of the i.i.d. assumptions generally applies to all the MCMC methods. The accept-reject methods provide i.i.d. sampling but have several inconveniences, especially with multivatriate problems such as ours, and therefore are not discussed here.

When applying the Gibbs sampling to mixture models, the steps consist of the successive estimation of z (the variable indicating the component from which the observation is generated), $\hat{p}$ (the mixing probabilities), and $\hat{\theta}$ (the parameters of each component). The latter parameters may imply several additional steps based on the underlying distribution. For a Gaussian mixture, for instance, the means $\hat{\mu}$ may be sampled first, and then the variances $\hat{\sigma^2}$ generated.

According to Jasra, Holmes & Stephens [69], the Gibbs sampler is not the most appropriate sampler since it does not explore many regions of the posterior and thus "cannot visit all of the modes of a mixture target." They adopt the MH updates of the parameters and discard the latent variables. For full convergence, the sampler needs to visit all the $n!$ modes, that is, all labeling combinations. This illustrates one major concern in Bayesian statistics: Can the MCMC sampler manage to visit sufficiently the support of the underlying parameter distribution?

Another important issue in Bayesian analysis is the decisive role of the choice of priors for the parameters. This problem is illustrated in Aitkin [3] with an example of the Galaxy data used by various authors to find the optimal number of components in a mixture model. Likelihood and Bayesian analyses show that the various Bayesian implementations reach contradictory conclusions on the number of components because of the different opinions on the appropriate prior distribution and hyperparameter choice. In this context, Aitkin concludes that the complexity of the prior structures needed for Bayesian analysis leaves the user confused about "what the data say", and although likelihood analysis has some inconvenience, it is well understood and more straightforward to answer some questions using the bootstrap method.

4.2 The label-switching problem

4.2.1 The label-switching problem and multimodality

Label switching is a major issue in the Bayesian estimation of mixture models. As discussed, it is also an issue in frequentist estimation when (and only if) one needs to use a resampling procedure, such as bootstrap, to evaluate the validity of the solution. However, much more research has been dedicated to the Bayesian mixture models; for example, see Celeux, Hurn and Robert [30], Stephens [147], Sperrin, Jaki and Wit [143], and Jasra, Holmes & Stephens [69].

Label switching arises from the invariance in the permutation of component labels. It causes the need for methods that can identify the components obtained after each iteration of an MCMC sampler (in the Bayesian framework) or solution of a Bootstrap sample. In the Bayesian estimation of mixture models, label switching arises when one attempts to use marginal distributions to summarize joint posterior distributions and provide estimations using the posterior mean of the latter.

We must note that in EM procedure, if one is interested, for instance, only in the density of the mixture, label switching is not an issue because it is invariant to labeling permutations. This is the case, for instance, when one uses the model for data prediction.

An issue that arises with label switching is the multimodality problem; this means the presence of multiple modes in the distribution. In fact, if we have k components in a mixture model, the number of modes is of order $O(k!)$, as we have all possible permutations of the indices. Moreover, if we use an exchangeable prior on the parameters, all the marginal distributions, and therefore all the posterior parameter expectations, are identical Marin, Mengersen and Robert [89]. Many authors (Rodriguez & Walker [126] Celeux, Hurn and Robert [30]) stress that label switching is necessary for the convergence of MCMC. If no label switching occurs, it means that the sampler is not exploring all the mixture model?s $k!$ possible posterior distribution modes that should be explored. This is then seen as a symptom of poor sampler mixing, and various modified samplers have been proposed to remedy this problem (see Rodriguez & Walker [126] (2014) and the references therein).

However, there is another type of multimodality that is more problematic, especially in frequentist estimation, where the solutions obtained with each bootstrap sample are completely independent of one another. As mentioned, the mixture models are ill-posed problems, and therefore two completely different mixtures may result in very similar

total density distributions. This results in multiple "genuine" modes (or multiple local maxima) in the likelihood function, because several different parameterizations allow for very close likelihood values, and thus equivalent fitting, to the data. This issue aggravates the label-switching problem, and we cannot find the corresponding unique labeling of the parameter estimates because of incompatibility of the two parameterizations (genuine multimodality). Particularly, in such cases, relabeling algorithms fail because they attempt to label the parameters as if they correspond to a simple permutation of the same parameters. Thus, the fact that similar mixture distributions may be obtained with completely different mixtures is ignored.

Several papers deal with the label-switching problem under genuine multimodality (see Grün and Leisch [61], Stephens [147]). The goals are typically not only to relabel the parameters, but also to separate the solutions in different modes by including more clusters than the components of the mixture, for instance.

However, hereafter we will focus only on the label-switching issue; we will also briefly overview some of the existing strategies to solve the issue. The first strategy is to introduce *identifiability constraints* to remove the symmetry of the likelihood; this strategy has been shown to fail in some cases (Stephens [147]). Various *relabeling approaches* Stephens [147], Celeux, Hurn and Robert [30] have also been presented. We also mention methods that could minimize a label-invariant *loss function*.

4.2.2 Solutions to the label-switching problem

Several papers during the last 20 years have proposed various solutions to the label-switching problem. Sperrin, Jaki and Wit [143] divide them into three general types: identifiability constraints, deterministic relabeling algorithms, and probabilistic relabeling algorithms.

Identifiability Constraints

The first (and oldest) type of solution involves the use of *identifiability constraints*. This consists of simply constraining the parameter space of the components Richardson & Green [124]. For example, a simple three-component Gaussian mixture can result in the following constraints: $\mu_1 < \mu_2 < \mu_3$. The parameter constraints are chosen such that only one label permutation can satisfy all of them. These constraints are called artificial because they do not originate from any knowledge about the data, but rather reflect the model or the researchers thoughts on how many groups exist and what characteristics differentiate them.

In Bayesian analysis, the introduction of such constraints means truncating the prior distributions, which can be done by adding the indicator function $\pi(\theta, p)I_{\mu_1 \leq \ldots \leq \mu_k}$. This means imposing identifiability constraints on the components from the initial stage.

Although imposing constraints can be effective in simple models, it can be difficult to implement in more complex models or when two or more types of parameters account for the same component in the model. As illustrated in Celeux, Hurn and Robert [30], a three-component mixture ordered by either means, variances, or mixing probabilities give completely different and incompatible results. Therefore, the choice of constraints, especially for multivariate problems, is not obvious and can have a crucial impact on the final results Yao [175]. Furthermore, McLachlan and Peel [94] argue that the differences between the parameters of each component are overestimated in this approach: they are "pushed apart" by the constraints, truncating the parameter space of the components. Richardson & Green [124] provide an example with a two-component mixture by mixing the proportions close to 0.5 and those with relatively close means. When the overlapping components are relabeled by ordering the means, the estimates are biased with clearly overestimated differences. The truncated distributions (obtained with the constraints) do not necessarily respect the modes in the prior and the likelihood. Instead of considering one single mode, there is a risk of including different modes in the same truncated distribution. The resulting posterior distribution can fall between two or several modes in a low-probability region. Such cases are illustrated in Marin, Mengersen and Robert [89] and Celeux, Hurn and Robert [30].

Deterministic approach

In a *deterministic relabeling* algorithm, one permutation matches another if they are "close" according to a given criterion.

First, characteristic C is defined, and then the distance between the iterations of the optimization process is measured using the chosen loss function $L(C^{it1}; C^{it2})$. L is large when the discrepancy between the characteristics of two iterations is large. This characteristic may be defined in various ways. For Cron and West [39], it is a classification vector $\hat{Z}$ with elements $\hat{z}_i = argmax_{j \in 1:k} \pi_j(x_i)$, which assigns each observation to its component using the last iteration classification probability ($\pi_j(x_i)$). The loss function here is the misclassification that $\hat{Z}$ implies compared to the classification vector $\hat{Z}^R$ of some reference solution. A specific algorithm is then used to find the optimal label permutation.

Another example of a loss function for relabeling is introduced in Celeux, Hurn and

Robert [30]. A collection of reference points (noted as $t_1, ..., t_n$) is placed in the parameter space. The distance $d(t_i, \theta)$ between t_i and the closest parameter θ_i is measured according to a given metric, Baddeley metric in this case. Each θ_i is a d-dimensional parameterization (the MCMC vector sample). For instance, it has the dimension 2*3 when we model a two-component Gaussian mixture with μ, σ^2 and p. The loss function L then becomes

$$L(\theta, \hat{\theta}) = \sum_{i=1}^{n} (d(t_i, \theta) - d(t_i, \hat{\theta}))^2$$

.

The loss is higher when the distance between t_i and the nearest $\hat{\theta}_j$ differs more than that between t_i and the nearest θ_j. The objective is to have zero loss between the two point configurations when they coincide: $L(\theta, \hat{\theta}) = 0$ if and only if $\hat{\theta} = \theta$. The choice of reference points is therefore very important, with a part of the simulation effort dedicated to find an appropriate choice of points t_i and the rest allocated to estimate the expectation of the difference in distances (to mainly estimate $E(d(t_i, \theta))$ by sampling from the posterior (MCMC sample) and averaging at the end. Since only the point closest to the given fixed point t_i is taken, there is no labeling problem in this case.

Jasra, Holmes & Stephens [69] stress that these methods are an "automatic way to apply or induce an identifiability constraint," and not "a fully decision theoretic method," because one cannot derive a loss function for every quantity of interest. They mention another common problem for identifiability constraints and relabeling algorithms: if the data are not well separated between components, one component may overweight and dominate the others Gruet, Philippe & Robert [59]. Therefore, one should be careful when using this method if there are several similar components.

Finally, another possibility is to first properly estimate the parameters without imposing ordering identifiability constraints on the parameter space. Then, one may apply ex-post reordering constraints after the simulations have been performed. A loss function depending on labeling can then be used (Marin, Mengersen and Robert [89]). In this method, simulations are performed normally and the components are "reordered" ex post; the posterior mean is then the simple average of the parameters after the reordering. This method uses a Monte Carlo approximation of the Maximum A Posteriori (MAP) estimator. In general, the MAP estimator corresponds to the maximum of the posterior distribution of the parameters Θ (considered as random variables here),

$$\hat{\Theta}_{MAP}(x) = \arg\max_{\Theta} f(x|\Theta) g(\Theta)$$

, and equals the MLE in case of a uniform (constant) prior.

For each of the simulated samples, we choose the parameter permutation closest to the approximate MAP estimator permutation. Rodriguez & Walker [126] (2014) briefly review the deterministic methods with loss functions based on the classification probabilities on each iteration of the MCMC, which should be matched to the estimated "true" classification probabilities; this is equivalent to using the allocations of the observations rather than classification probabilities. The authors also present an interesting alternative of using the data themselves to undo the label?switching problem. They use the center and the dispersion of the clusters within the data. This method can be effective in case of distribution with a small number of parameters. However, it could be difficult to apply in the HMTD model because several parameters of the model are responsible for the mean and variance of the distribution of the sequences within each component. It is also a problem for longitudinal data because different clusters with different properties or sequence paths may display an identical center and dispersion.

Probabilistic approach

Probabilistic relabeling algorithms are another family of tools for the label-switching issue. This approach does not consider the permutation of each solution as certain, but has a probability distribution that needs to be estimated. Sperrin, Jaki and Wit [143] describe it as an application of the EM algorithm, where the missing data is the order of the components at each MCMC iteration. The advantage of this method over the two previous ones is that it quantifies the uncertainty of the chosen permutation and calculates the probability of the accepted one to be the "true" permutation. Furthermore, it recovers the tails of the posterior distributions using such methods Sperrin, Jaki and Wit [143], Yao [174]. The vector of parameters needs to be known in advance and the discrete density of the permutations must be estimated. The latter is estimated via an EM algorithm, conditioning on the data the last estimates of the parameters and the last allocation vector z. The missing data in the EM algorithm are the permutations applied at each stage, and the available data are the output of the MCMC sampler.

Sperrin, Jaki and Wit [143] suggest that quantifying the uncertainty is informative on the number of components since a high uncertainty suggests an ambiguity between the components and therefore a too high number of components.

Label switching and frequentist estimation

All the above methods were developed for the Bayesian estimation of standard mixture models without hidden layers and for transversal data. However, because of the fast development of the Bayesian methods, much attention was not paid to frequentist mixture model estimation. To the best of our knowledge, only one recent paper by Yao [175] in 2015 has considered label switching in the frequentist approach. The author proposes two solutions, one based on the complete likelihood that is not invariant to label permutation, and the other based on the idea of minimizing the Euclidean distance between the classification probabilities and latent labels. In both propositions, however, the latent labels (or group membership) of each observation must be known a priori. This is not problematic when using the parametric bootstrap, but such bootstrap methods are highly criticized and rarely suitable for real-world data. When using the nonparametric bootstrap, the latent labels are not known and have to be estimated, but this would introduce another source of bias in the procedure. Furthermore, if data clustering is the goal of mixture modeling, this would mean that we need to pre-cluster the data before clustering them, which appears to be illogical.

Some critics to these approaches

All the methods presented above can be criticized. In complex models with more than one parameter for each state (as with the HMTD model), or models with multiple states, most of these methods are difficult to implement. Introducing constraints for several parameters simultaneously is highly likely to bias the results. Jasra, Holmes & Stephens [69] affirm that label switching can easily be solved in frequentist estimation by applying simple inequality constraints. In our opinion, this solution cannot be generalized over all kinds of mixture models, and more attention needs to be paid when using constraints even in simple mixture model with unequal variances (especially when some components are close) or in more complex mixtures when the parameters are not independent of each other. For instance, in the HMTD model, the mean consists of a constant and an autoregressive (AR) part, and, as mentioned before, by increasing the AR part and decreasing the constant (or vice versa), we can obtain exactly the same mean for a given cluster. Thus, imposing separate constraints for every AR coefficient and constant would hardly be feasible even for only two components. Furthermore, if, for example, the first or second AR coefficient is not significant even for one single component, the constraints on this level could be misleading.

Dias & Wedel [45] also find that identifiability constraints tend to deteriorate the solutions compared to other approaches based on loss functions and clustering, for example. Furthermore, their examples were only based on simple two- and three-component Gaussian mixtures.

Therefore, the use of identifiability constraints is generally not a good answer to the label-switching problem (especially for less simple models), whether the constraints are imposed before or after optimization. Moreover, relabeling and other methods are computationally costly and some are applicable only to the Bayesian framework (especially those that use successive iterations).

The efficiency of the approaches for large datasets, parameter space, or number of clusters represents a major issue for all the approaches, according to Zhu and Fan [177], who proposes alternatives to tackle this particular problem. In fact, some of the relabeling methods may need more computational time than the MCMC itself, and for the algorithms that attempt to match each observation to the previous one, the high correlations in MCMC may be problematic Cron and West [39].

The biggest problem in all the approaches, however, is the probability of mislabeling, especially when the components of the mixture overlap or are more complex (in our case, several parameters account for each component). This may also be a problem when the number of clusters increases.

From our experience, a major issue that aggravates the above-mentioned problems, especially in EM estimation, is again the problem of "genuine multimodality." A similar likelihood value can sometimes be obtained even with an empty component, although a solution obtained with less components is more difficult to interpret and does not provide an acceptable partition. This problem becomes even more important when using an EM-type algorithm that is known to converge to the nearest optimum and often fails to explore the entire parameter space with a more "complex" model with multiple components and parameters. In other words, the effectiveness of label-switching strategies is less preoccupying in the EM estimation of HMTD than the occurrence of incompatible solutions (for which a true relabeling does not exist).

4.3 Parameter inference

4.3.1 Inference and standard error approximation in mixture models

In real-world clustering problems, one often has no a priori knowledge of the different classes. However, through inference on mixture models (or model-based clustering, in general), one may try to reconstitute the latent group membership or provide an estimation of the parameters (indicating the characteristics of different groups), or even find the optimal number of groups. One needs to not only have an optimal clustering solution (in terms of Bayesian Information Criterion (BIC), for instance), but also explore the variability of the estimated parameters and assess the stability of the underlying clusters. This is important when estimating the role of the covariates in clustering. These different objectives reflect the various sources of uncertainty that we will discuss further in the next sections, but now we overlook them and focus on measuring the coefficients' significance and dispersion.

Usually, every solution includes nonzero values for all parameters. However, it is important to verify whether this parameter estimate is significant, or whether it is due to a local optimum or just randomness. Therefore, we need to compute the confidence intervals (CI) for the parameter estimates of the optimal clustering solution in order to validate them. One traditional way to do this is to approximate the standard deviation of each parameter, but this can be done using other methods as well, which we will discuss.

Several approaches to compute the CIs exist in mixture models, but from the type of the models and their complexity, not all of these methods are feasible. We enumerate below some well-known methods:

- **Finite-difference approximation of the Hessian matrix** is the most logical choice for estimating parameters and their standard errors since the Maximum likelihood estimate (MLE) is asymptotically normal. That is, as the sample size increases, we have $\hat{\theta}_{ML} \xrightarrow[n\to\infty]{} \mathcal{N}(\theta, \frac{\mathcal{I}^{-1}}{n})$, where $\mathcal{I}$ is the information matrix at the true parameter θ. $\mathcal{I}$ can be estimated from the data as the Hessian of the log-likelihood at the $\hat{\theta}_{ML}$. Thus, one can easily estimate a two-sided CI for any parameter:

$$\hat{\theta}_{i,j}^{MLE} \pm \frac{z_{1-\frac{\alpha}{2}}(\hat{\mathcal{I}}^{-1})_{ii}}{\sqrt{n}}$$

. As Rydén [131] has stated, from the covariances and the fact that the product of

Gaussians results in a $\chi^2(q)$ distribution, one may build a confidence "elipsoid" for the parameters corresponding to the region defined by $(\theta-\theta_{MLE})^T\hat{\mathcal{I}}(\theta-\theta_{MLE}) \leq \chi^2_{1-\alpha}(q)$. Note that the asymptotic normality assumption is very important for such interval computation. This becomes especially problematic when the parameters are clearly non-Gaussian, like the variances that are truncated ($\theta_k > 0$).

Moreover, one may encounter problems when estimating the Hessian One example is the HMTD model we used here, whose log-likelihood derivatives are very complex because of the complexity of the log-likelihood equation for the variance parameters and the different number of parameters and components (and therefore the large number of distinct possible log-likelihood equations).

Even for the simple hidden Markov models, calculation of the Hessian may become infeasible, like when we deal with time series or longitudinal data of large length, as discussed in Visser, Raijmakers & Molenaar [164]. The difficulty in obtaining the second derivatives of the likelihood function prevents one from obtaining the Fisher information.

- **Likelihood profiles** are presented as expansions of the likelihood function around the ML estimate of each parameter separately (Visser, Raijmakers & Molenaar [164]). Assume that we have a two-parameter model (ϕ and θ). If we are interested in ϕ, then θ is treated as a nuisance parameter in order to obtain the profile likelihood

$$L_p(\phi) = \max_{\theta} L(\phi, \theta)$$

. The value of ϕ that maximizes this function is denoted by ϕ_M. Then, one moves ϕ away from this maximum, and using the new value of ϕ_0, re-estimates the ratio

$$R_p = -2(log(L_p(\phi_M)) - log(L_p(\phi_0))) = -2\ln\frac{L_p(\phi_M)}{L_p(\phi_0)}.$$

The procedure is repeated until we find the value of ϕ_0, when $R_p = 3.841$. The latter value is a threshold corresponding to the Type-I error $\alpha = 5\%$ for a $\chi^2(1)$ distribution with one degree of freedom. This represents finding the limits ϕ_0 of the region beyond which the likelihood ratio test becomes significant (i.e., the null model becomes significantly different from the optimal one).

By repeating the procedure on both sides of the ML estimate and for each parameter separately, we obtain the CIs.

This procedure is an alternative to obtaining the intervals by estimating the standard error, in which case the normality assumption is again crucial. The separate computation on both sides of the estimate allows us to obtain unsymmetric intervals.

However, in case one does not want to rely on asymptotic normality, there are several alternatives. Three of them are given below:

- **Nonparametric bootstrap** is the most frequently used bootstrap method. It consists of building samples by drawing at random observations with replacement from the original sample. The model is fitted to each of the resulting samples, to obtain a corresponding number of parameter estimations. The 95% CIs are then estimated by simply using the percentile method: the CI intervals are given by the 2.5% and the 97.5% percentiles of the distribution. For the frequentist mixture model, the bootstrap is the main tool used to obtain parameter inference. Friedman, Hastie and Tibshirani [54] relate it also to the Bayesian methods, suggesting that bootstrap may be considered to provide a noninformative, nonparametric posterior distribution for the parameters: it "approximates the Bayesian effect of perturbing the parameters, and is typically much simpler to carry out."

- The **jackknife** method has also been used to estimate parameter variability for hidden Markov models. As an old predecessor of the popular nonparametric bootstrap, this method is similar to it, except that the subsamples are formed by removing one observation at a time from the original sample. As regards a misclassified Markov model, the jackknife has not shown better results than the bootstrap, as demonstrated by Rosychuk, Sheng & Stuber [129]. This is not surprising and explains why the method has often been replaced by the nonparametric bootstrap since its discovery.

- The **parametric bootstrap** uses an original dataset to obtain the ML estimate of a given model. The estimated parameters are then used to generate a new dataset of the same size. The model is fitted to the artificial data and the optimal parameters are estimated again. The procedure is repeated n times, to result in an empirical distribution around the ML estimate of the parameters. We then compute the standard errors, to obtain the desired CI for each parameter.

  The most obvious problem with the parametric bootstrap is the assumption that the model can perfectly explain all the features of the data. Since this assumption

is almost never verified in the real world, the possibility of obtaining incorrect results is considerable. Using the parametric bootstrap with more complex models could result in higher specification errors of the estimations. This type of bootstrap is useful only when we know the form of the underlying distribution Efron and Tibshirani [47]. Therefore, this method seems to be the least appropriate for our parameter variability estimation problem.

A critic of nonparametric bootstrap is that a data pattern not observed in the sample has a zero probability to appear in the bootstrap samples. However, as Dias and Vermunt have shown, the same problem can occur with parametric bootstrap if one parameter estimate is on the limit of the parameter space Dias & Vermunt [44].

In a mixture model, the most straightforward of these methods to approximate the covariance matrix of $\hat{\Theta}$ is to compute the inverse of the information matrix. Another approach (of Dietz and Böhning) is mentioned by McLachlan and Peel [94]. This is based on the log-likelihood change resulting from omitting a given variable in the MLE. However, as the authors state, "the estimates of the covariance matrix of MLE based on the expected or observed information matrices are valid inferentially only asymptotically," and "for mixture models (...) the sample size has to be very large before the asymptotic theory of maximum likelihood applies." The standard errors calculated from the information matrix are also found to be too unstable, and a bootstrap resampling approach is recommended instead.

Another motivation to use resampling approaches for our particular problem is as follows. As already mentioned, the complexity of the likelihood function might make it more difficult to obtain the Fisher information, and therefore (for the frequentist models) we usually need to bootstrap the data.

Visser, Raijmakers & Molenaar [164] compared three methods to compute the CIs of the parameters of an HMM that is applied on single long sequences. The first method attempted finite-difference approximations of the Hessian. The three piecewise linear approximation methods, quadratic and cubic polynomial approximations, resulted in erratic and often small CIs. Bootstrap and the likelihood profile methods, on the other hand, provided similar and better results.

All these points make us focus on the nonparametric bootstrap to compute the CIs.

4.3.2 Alternative bootstrapping procedures in clustering

Nonparametric bootstrapping does not guarantee that the underlying clusters will be represented equally in bootstrap iterations. Combined with the genuine multimodality issue, this may sometimes lead to incompatible clustering solutions, necessarily introducing a bias in the CIs and therefore in the conclusions on their significance. The problem is even more important when small samples are considered or one of the clusters is small. In addition, the label-switching issue can sometimes be very difficult to solve when working with rather complex mixture models (that have multiple components or parameters per component).

Therefore, the best solution would be to avoid, rather than deal with, these issues. This is our motivation for using different bootstrap methods for parameter validation. We propose two possible alternatives to the straightforward nonparametric bootstrap method in the following section and discuss their advantages and disadvantages.

Separate bootstrap

Nonparametric bootstrap has been used previously to compute the intervals of parameters in mixture models. In combination with label-switching solutions, this method has shown positive results. Grün and Leisch [60] showed that bootstrap is useful to evaluate the stability of parameters when estimating finite mixture models, and recommend it in addition to multiple initialization of EM estimation. Using simulated data, the authors apply both parametric and nonparametric bootstrap, to find that the parametric bootstrap is also a useful tool to analyze the stability of parameter estimates. However, the examples provided are based on very simple models (and only simulated data) with two-component mixtures, for which the application of simple identifiability constraints is sufficient. The use of bootstrap on more complex models appears more difficult and may bias the estimations from misidentification of components. Our proposal for such problems is to apply bootstrap estimation only after a reliable valid optimal solution (cluster partition) is found. This is equivalent to saying that once an acceptable solution has been found, each cluster can be considered a mutually independent populations. Then, the bootstrap procedure is performed separately on each cluster, with the CIs computed using a single-component model for each generated sample.

Indeed, this could imply that we consider the chosen solution the best that could be found, and neglect the fact that another may exist, possibly yielding to a lower BIC. However, in clustering, different solutions can be obtained with different models and a lower BIC does not necessarily imply a better, more useful solution. Therefore, one

needs to find the solution most suitable to the data based on our knowledge of the data. In this case, by applying the bootstrap to the chosen clustering partition, we do not measure the stability of the solution, but rather isolate the variability and significance of the parameters for this particular clustering partition. This makes sense especially when two iterations (or final solutions) of a given complex model may be completely incompatible (due to cluster instability, for instance), resulting in the relabeling strategy proving wrong.

Another advantage of this approach is that we avoid the typical *singularity problem*, that we discussed, in likelihood maximization, because we use a single-component model for constructing the intervals.

This procedure may be effective even outside the "hard" clustering problem, such as in a transition allowed between hidden states in time series modeling, or in "soft" clustering (recall the different forms for the matrix A in chapter 1). In this case, one would need a long time-series for estimation, and only the same-state sequence is used to estimate the CI of parameters. By bootstrapping the part of the time-series associated to each hidden state separately, we would be able to interpret the different components of the model more accurately and with more certainty.

However, note that the parameters for the components weights are indeed not inferable through separate bootstraps, but must be assessed beforehand while choosing the optimal model solution (they are considered a part of it).

On the other hand, a separate bootstrap has the advantage of completely eliminating the issue of genuine multimodality that we discussed earlier. Single-component models imply that we cannot deal with another genuine mixture solution that gives similar joint density results. Furthermore, the label-switching and singularity problems are relaxed.

Stratified bootstrap

One reason for finding it difficult to correctly identify the clusters obtained from a bootstrap sample is the difference in proportion of data in each cluster between the original sample and the bootstrap samples. This is especially so when the proportion of data belonging to each cluster is very different from one clust another. One possible solution is to include the proportion of cluster proportions in the bootstrap procedure and ensure equal presence of all the presumed clusters in each sample. Now, assume that our original sample of raw unlabeled data $x_1...x_n$ is treated by a clustering procedure with c classes. The obtained solution attributes class labels ω_j $j \in [1, ..., c]$ to each

observation. These class labels are then used for a bootstrap on the original sample, but by using a sampling procedure that respects the initial clusters proportions at each iteration. In other words, we create the bootstrap samples by selecting a quantum of data from each cluster proportional to its presence in the chosen clustering partition. The full clustering model is then applied on the bootstrap sample. This approach may be seen as a kind of "stratified" bootstrap, where the "strata" are defined by a model solution (partition) obtained from the original sample before performing the bootstrap. It is important to note that these proportions are at the center of this procedure because they insure that any further solution will preserve the features of the classes of the initial approved solution.

The advantage of this procedure is that it maintains the proportions of the already validated clustering obtained from the original sample. This may result in a more stable bootstrap estimate for the parameters compared to the ordinary nonparametric bootstrap sampling of the original data. The effect is particularly important in small samples (or larger samples with small but clearly distinct clusters), because in a basic nonparametric bootstrap on a small size sample, the representativity of each possible underlying class in the data is not respected, probably leading the clustering model to find a completely different solution compared to that of the original sample. In other words, very small but distinct classes may "vanish" in resampling (for example, voters of small political parties in a survey, or low-probability components in a mixture). In this case, all the relabeling methods will prove useless, because the components of each solution would simply be incompatible to each other. Of course, our approach will not suppress the label-switching issue, but can reduce it by increasing the probability of finding bootstrap solutions close to the original solution.

Compared to the "separate bootstrap" presented above, the initial solution in the stratified bootstrap has less influence on the final results because the elements of each class are still represented and can still be drawn during the resampling procedure (model uncertainty is still present).

Simulation experiment

In this experiment, we compare the behavior of three bootstrap procedures to evaluate the CIs of the HMTD parameters used for clustering. The three bootstrap procedures (ordinary nonparametric, "separate," and "stratified" bootstraps) are applied to simulated data. The data consist of 150 sequences of length 25 generated (after a burn-in period) through either of the following second-order or first-order AR processes. One

hundred sequences were generated from

$$x_t = 2.5 + 0.4 \times x_{t-1} + 0.3 \times x_{t-2} + \epsilon_1; \quad \epsilon_1 \sim \mathcal{N}(0, 2^2)$$

, and 50 sequences were generated from

$$x_t = 0 + 0.9 \times x_{t-1} + 0 \times x_{t-2} + \epsilon_2; \qquad \epsilon_2 \sim \mathcal{N}(0, 2^2)$$

.

Thus, the parameters that we attempt to estimate are $\phi_{01} = 2.5, \phi_{02} = 0, \phi_{11} = 0.4, \phi_{12} = 0, \phi_{21} = 0.3, \phi_{22} = 0$[1], $\theta_1 = 2$ and $\theta_2 = 2$. Table 4.1 summarizes the results obtained from 300 bootstrap replications. In each case, we provide the 95% and 99% CIs obtained for the parameters, with the top part of the table corresponding to the first component, and the bottom part corresponding to the second component.

In this experiment, the initial "true" model is the first two-component solution that we obtained, and using it, we performed 300 bootstrap iterations for each method. The parameters were initialized, as usual, randomly around the values corresponding to the observed mean and standard deviations *of the entire dataset.* In this initial clustering solution, a small part of the sequences were misidentified: nine sequences were wrongly assigned to the second group, while three were misclassified in the first group. This small misclassification in the initial separation on clusters could introduce some bias into the CIs for the proposed methods that use the initial solution ("separate" and "stratified"). However, this is similar to what can be expected in real situations. Moreover, since the proportion of misidentification is small, the CIs should still be able to recover the true parameter values. As observed in Table 4.1, this is indeed the case for all parameters.

As expected, the label-switching issue arose when we attempted to relabel the ordinary and stratified bootstrap solutions. However, since our example contains simulated data consisting of only two components, although the sequences from the two generating processes were overlapping, we could easily solve the problem by crossing the group memberships of the initial solution and the bootstrap solutions, and selecting the label that had the best match. However, the complexity of the likelihood function caused some bootstrap solutions to get stuck in local optima and exhibit irregularities, the most common one being convergence to a one-cluster solution. We need to identify such degenerated solutions and remove them from the CI calculation in order to avoid

[1]Notice that the coefficient corresponding to the 2-nd lag in component 2 is not required, since its value is zero, but we choose to keep it in order to check that it is correctly estimated to zero by the optimising algorithm and by the bootstrap procedure.

Figure 4.1: 95% and 99% CIs obtained for the parameters using three types of bootstrapping.

First component		σ_1^2	ϕ_{01}	ϕ_{11}	ϕ_{21}
True values		**4**	**2.5**	**0.4**	**0.3**
Ordinary	95 %	(3.633; 4.215)	(2.076; 2.915)	(0.339; 0.422)	(0.276; 0.362)
bootstrap	99 %	(3.561; 4.353)	(1.905; 3.024)	(0.328; 0.478)	(0.254; 0.369)
Separate	95 %	(3.645; 4.100)	(2.261; 2.830)	(0.341; 0.422)	(0.275; 0.357)
bootstrap	99 %	(3.555; 4.119)	(2.189; 2.922)	(0.332; 0.427)	(0.267; 0.368)
Stratified	95 %	(3.640; 4.206)	(2.084; 2.870)	(0.331; 0.422)	(0.276; 0.375)
bootstrap	99 %	(3.572; 4.457)	(2.000; 3.016)	(0.313; 0.431)	(0.237; 0.386)
Second component		σ_2^2	ϕ_{02}	ϕ_{12}	ϕ_{22}
True values		**4**	**0**	**0.9**	**0**
Ordinary	95 %	(3.746; 4.406)	(-0.022; 0.272)	(0.839; 0.983)	(-0.060; 0.123)
bootstrap	99 %	(3.683; 4.522)	(-0.055; 0.414)	(0.640; 1.003)	(-0.074; 0.312)
Separate	95 %	(3.808; 4.380)	(-0.023; 0.178)	(0.895; 0.997)	(-0.062; 0.053)
bootstrap	99 %	(3.737; 4.452)	(-0.052; 0.200)	(0.882; 1.004)	(-0.075; 0.062)
Stratified	95 %	(3.721; 4.461)	(-0.011; 0.232)	(0.841; 0.989)	(-0.060; 0.113)
bootstrap	99 %	(3.639; 4.558)	(-0.059; 0.347)	(0.663; 1.006)	(-0.082; 0.296)

additional bias. One obvious way to discover these solutions is to check the presence of all components. Irregularities are, however, not limited to the absence of one component. A local optimum can consist of a solution incompatible with the initial solution even when all components are used. For instance, highly influential or extreme solutions may be drawn several times in the same subsample (especially when n is small), creating their own component. A numerical likelihood optimization problem can also lead to aberrant solutions. Therefore, we need to check for the presence of extreme values in the parameters of each component after the relabeling procedure. In our experiment, 19 out of 300 solutions were found to be irregular for the stratified bootstrap, and 20 were found irregular for the ordinary bootstrap procedure. The separate version was not involved, because all calculations were made separately for both the components in the initial solution.

As mentioned earlier, all the CIs effectively recover the true values of the parameters. The separate bootstrap provided a systematically narrower CI compared to the

two other approaches, but this was as expected, because the uncertainty is lower when we estimate the initial solution correctly (despite a small number of misclassified sequences). On the other hand, the ordinary and stratified bootstrap yielded similar results, certainly because the model was simple enough to be easily estimated with the ordinary procedure. In more complicated real-world situations (and with more unequal size clusters), a difference could appear, with a larger CI for the ordinary bootstrap compared to the stratified procedure.

4.4 Validation, comparison, and stability

When discussing the uncertainty in clustering, we cannot focus only on the dispersion of the parameter estimates in each cluster. Putting the sample representativity aside, the actual cluster membership of the sample units is unknown, just as the number of clusters, and, in some cases, even the presence of any "real" separation between the data units. One may generalize the problem as

$$\text{Clustering uncertainty} = \begin{cases} \text{Presence and type of heterogeneity} \\ \text{Number of clusters} \\ \text{Stability of partition} \\ \text{Significance of the parameters (characteristics of the clusters)} \end{cases}$$

The presence of several levels of uncertainty makes the independent assessment of the estimates' variability impossible. Dealing with one of these problems without considering the others may lead to unstable and unreliable results. For instance, if the number of clusters is incorrect, the membership of a data unit will also be incorrect, in turn making the parameter estimates incorrect as well. On the other hand, incorrect parameter estimates may lead to a different optimal number of clusters and erroneous cluster memberships, since close clusters may merge together and erratic parameter estimates result often in unlikely empty clusters.

Therefore, a model-based clustering would generally make it very difficult to cope with all the above problems simultaneously, and one may better isolate and solve these issues separately and in a given order. Here, we describe our point of view on how a complete model-based clustering analysis should be performed, proposing a given order for solving the issues as follows:

1. Before clustering, one may ask oneself as to what the properties of the data that are supposed to be captured in the clusters are. Then, a clustering method with

compatible assumptions must be chosen. If one models phenomena with values that are constantly increasing linearly, one may privilege linear instead of AR paths as a basis for the clustering. The opposite may be true when no specific shape of the path is relevant, but rather the behavior over time, which is typically the case for longer but less stable sequences, for instance.

One might also use data visualization or other methods to confirm the presence of heterogeneity in the data, although visualization might be complex when dealing with longitudinal or multidimensional data, especially when the sample size is large.

2. Once compatible methods are chosen, one can find the correct number of clusters. Numerous criteria have been proposed for this purpose, the most usual one being the Bayesian Information Criterion (BIC).

3. The stability of the obtained solutions must also be assessed. If every bootstrap sample results in a completely incompatible clustering with the same model, it is difficult to justify the selected solution. For instance, the same distribution may often be approximated by different Gaussian mixtures, and it can be problematic if this problem persists for the bootstrap samples.

4. Assessing the variability of parameters in a model-based clustering is the final step in solving a clustering issue. At this point, the clusters should have already been decided and validated. If this is not the case, it would be difficult to distinguish one kind of variability from another.

   Our main interest here is to obtain CIs for the parameters, because it is crucial in social sciences (as well as in other fields) to understand the covariates or characteristics of the population that are significant sources of dissimilarity between the individuals.

All the above-mentioned sources of variability are often confounded. Hereafter, we discuss some methods to cope with the different uncertainties.

4.4.1 Clustering and distance between clusters

The researcher must first decide the characteristics of the data that make them belong to the same cluster. This is decisive for both selecting the most appropriate clustering method and choosing an appropriate validation method for the obtained clusters. An

expert view on the data and clustering process can reveal important factors that would improve the usability of the final clustering, but this would apply only if it is well-founded and does not reflect the expert a priori ideas or bias (e.g. Berchtold et al. [16]).

Various suitable scaling and projection methods are available for multidimensional data on how to visualize the data (Ratio MDS). For instance, principal component analysis (PCA) is the most traditional projection tool.

As mentioned earlier, a large variety of distance (dissimilarity) measures exists in the literature, such as optimal matching and its variations for a discrete longitudinal case. Euclidean, Mahalanobis, Metropolis, and Chord distances can deal with continuous data, but only transversal ones. Their use in longitudinal data is meaningless because the sum of the distances over all the periods does not allow us to measure the distance between two sequences. Computing such distances leads to representing the sequences as single points in a multivariate space, neglecting the time dependence between the observations. Model-based approaches also have their assumptions on distance. One must choose the correct approach from the conclusions based on visualization of the data and the a priori knowledge about their nature.

4.4.2 Choice of number of clusters and model

Bayesian and Akaike information criteria Different criteria exist in the literature aiming to select the most optimal among a group of models, including the choice of optimal number of clusters.

Probably the oldest commonly used criterion is the Akaike Information Criterion (AIC) first presented by Akaike at a symposium in 1971 (Armenia, former USSR). Its first term includes the log-likelihood computed at the maximum likelihood estimate (MLE) of the parameters, thus it privileges the solutions that generate a higher probability of observing the given sample from the model (i.e., higher log-likelihood $logL = \ln(P(X|K, \hat{\Theta}_k^{MLE}))$). At the same time, the second term penalizes for the complexity of the model in order to avoid overfitting. The penalty increases with the number of independent parameters p to estimate in the model:

$$AIC = -2logL + 2p$$

Another very common general-purpose criterion is the Bayesian Information Criterion (BIC) from Schwarz [134]. It represents an asymptotic result obtained under the assumption that the data are distributed from an exponential family distribution.

Inspired from the AIC, BIC increases the penalty term proportional to the size of the data (n):

$$BIC = -2logL + \ln(n)p$$

Minimization of AIC or BIC is the simplest way to choose the optimal number of clusters in model-based clustering.

However, these criteria are not necessarily the most appropriate for all types of models and purposes. One of the disadvantages of BIC (and AIC) as a criterion for the number of clusters is that by privileging the fit to the data, a single non-Gaussian cluster may be represented by two or more Gaussian clusters which provide better fitting. Therefore BIC may sometimes over-estimate the number of clusters.

Note that for models containing several parameters for each component (cluster), by increasing the number of parameters one may penalize the criterion more severely compared to the case of having only one parameter per component. Given the HMTD model we are using, if one adds two lags for the mean of each cluster ($\Theta_k = \{\varphi_{k,0}, \varphi_{k,1}, \varphi_{k,2}, \theta_{k,0}\}$), the optimal solution may contain less clusters than when we use only one lag for the mean ($\Theta_k = \{\varphi_{k,0}, \varphi_{k,1}, \theta_{k,0}\}$) because of the penalty term.

Integrated Complete Likelihood After the paper of Biernacki, Celeux and Govaert [21] in 2000, another criterion has gained popularity in mixture models particularly for clustering use - the **Integrated Complete Likelihood** (ICL).

In order to understand this method we must recall the notion of *Integrated (or marginal) Likelihood (IL)*. Referred as the evidence of the model, it is important concept in Bayesian statistics. In general, its computation consists in marginalising out (integrating) the parameters in the likelihood function. The sampled values are used for this purpose. The aim is to obtain a remaining variable that represents the particularity of the model itself: for instance in mixture models often the selection of optimal number of components (variable k) is a major issue. Therefore one needs a likelihood function that indicates the probability that the data come from a mixture with k clusters, without assuming particular values for any other parameters (function of k only). In this case the marginal likelihood of interest is integrated over all other parameters (noted Θ), but K:

$$P(x|K) = \int_{\Theta} p(x|\Theta, K)p(\Theta|K)d\Theta$$

The objective is to compute the model evidence of one model with k_1 components,

against another model with k_2 components. The posterior odds ratio is computed by multiplying the prior odds ratio by the ratio of the marginal likelihoods (called Bayes factor): $\frac{p(k_1|x)}{p(k_2|x)} = \frac{p_M(K_1)}{p_M(K_2)} \frac{p(x|K_1)}{p(x|K_2)}$

In cases of clustering where the mixture components are not well separated an alternative version of IL using the complete data is often recommended (Biernacki, Celeux and Govaert [21], Celeux [32]). The Integrated *Complete* Likelihood (ICL) makes use of the true missing data z (i.e. the allocation of observations to clusters) in addition to the observations x in the computation of the log-Likelihood of the model. It is however, not easy to estimate and several approximation methods have been proposed by Celeux [32].

The first and most straightforward computation of ICL proposed by Biernacki et al. is the BIC-like approximation denoted ICL_{BIC}. This approximation of the ICL uses the value of the BIC, penalized by the mean entropy of the solution:

$$ICL_{BIC}(K) = BIC(K) - \sum_{g=1}^{K} \sum_{i=1}^{n} p(z_i|x, \hat{\theta}_k, \hat{\phi}_k, K) \log p(z_i|x, \hat{\theta}_k, \hat{\phi}_k, K)$$

where $p(z_i|x, \hat{\theta}_k, \hat{\phi}_k, K)$ is the probability that the observation is generated by the i-th component.

By taking into account the entropy, the ICL privileges the partition that provides more separated clusters compared to the classic BIC criterion. The latter is well suited to evaluate the fit of the model to the data and select the optimal data generating model, but ICL is more adapted to clustering where the discrepancy between groups matters, because it eases the interpretability of each cluster.

Various different computations of the ICL also exist in the literature (see Celeux [32]). Biernacki, Celeux and Govaert [22] and Bertoletti, Friel and Rastelli [20] discuss methods of exact computation of ICL using for instance different prior distributions. However, we must note that these papers, like the majority of the publications, focus on Bayesian estimation of the mixtures and are not adapted to the frequentist case.

Therefore for the examples in the next chapters we will implement the computation of the ICL_{BIC} approximation.

Other criteria Other approaches to choose the number of clusters, besides the above-mentioned, also exist. Some of them are based on bootstrap re-sampling. While such strategies are often used to measure the stability of clustering, another goal may be to choose the optimal number of clusters k for a given dataset and clustering method.

Fang and Wang [50], for instance, aim to find the number of clusters for which the average dissimilarity between the partitions (instability S) is minimal. More concretely, the bootstrap stability assessment follows four steps:

1. Generate B pairs of bootstrap subsamples with size n (number of observations) (X_b, X'_b), $b = 1, \ldots, B$.

2. Using the same method, calculate the clustering partitions P_{bk}, P'_{bk} for each subsample on k clusters.

3. To calculate the clustering dissimilarity s_{bk} between the pairs of subsample clustering partitions, check whether or not every pair of observation falls within the same group in both partitions,

$$s_{bk} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |1\{P_{bk}(x_i) = P_{bk}(x_j)\} - 1\{P'_{bk}(x_i) = P'_{bk}(x_j)\}|$$

   Then, define the clustering instability as the average of all dissimilarities between the b pairs of samples,

$$\bar{s}_{Bk} = \frac{1}{B} \sum_{b=1}^{B} s_{bk}$$

4. Repeat these calculations over all the possible number of clusters k. Now, the optimal number of clusters is the one for which the instability is minimal.

$$\hat{k} = argmin_{k \in [2 \ldots K]} \bar{s}_{Bk}$$

Although this procedure is designed to find the optimal number of clusters for the same clustering method, it could also be applied to compare the stability of the solutions of two different methods for a given dataset, provided the same number of clusters is chosen. The authors also propose a similar procedure to estimate the standard error of the estimated clustering instability.

Other methods are based on the between- and within-cluster sum of squared distances. The gap statistic, for instance, is a very popular method for k-means (Tibshirani, Walther, and Hastie, 2001). It evaluates the goodness of clustering based on average dispersion within the clusters as compared to a reference distribution. It is calculated with different number of clusters in order to choose the optimal number. Note that the indices based on between and within sum of squares and those based on dissimilarity are not adapted to continuous longitudinal data.

4.4.3 Validation of clustering, comparison, and stability assessment

Validating a clustering and comparing two clustering solutions

The two large categories of clustering validation indices are the *internal* and the *external* indices. The former are mostly based on distances (Euclidean, Mahalanobis etc.) or dissimilarities between observations. Often they compare the within and between sum of squares of the clusters and aim to evaluate the separation of the data that the models provide. The internal indices are described in details by Hennig [64]. One could imagine to apply distance-based indices on transformations of the data, such as AR coefficients fitted on each sequence, variances and average values, but such approaches can be highly criticized. This kind of measures are easily applicable to transversal multivariate data for instance, but they are not adapted directly to continuous longitudinal data. Therefore they will not be discussed in this thesis.

The *external* validation indices however can be independent from the nature of the data when applied to the membership of the observations. Often they are used to approve a model by comparing the resulting partition to the true class membership. However, in unsupervised learning, one may use a reference solution instead of the true classes (which are unknown). This is why we will more focus now on the external measures.

A recent review of these measures is provided by Meila [97] who distinguishes between four types of clustering comparison criteria. The first one is the comparison *by set matching*. An example is the *misclassification error* distance; this approximates the probability of the cluster labels disagreeing on an observation "under the best possible label correspondence." First, the clusters of the two compared partitions P^1 and P^2 are matched to find their corresponding equivalents. Then, one estimates the "unmatched probability mass" using the formula

$$\mathcal{H}(P^1, P^2) = 1 - \frac{1}{n} \max_{\mathcal{M}} \sum_{k=1}^{K} n_k, \mathcal{M}(k)$$

where $\mathcal{M}$ is the mapping of the clusters of P^1 into P^2 (both partitions do not necessarily have the same number of clusters k), and n_k is the number of observations in the cluster k.

The second type of comparison is *by information theoretic criteria*. It calculates the joint entropy of two clusterings, their marginal entropy, and their mutual information (representing the average (over all points) clustering information that we obtain about

one of the clustering partitions if we know the other). These quantities allow us to obtain the variation of information between two clusterings.

The last two categories of measures comparing the clustering partitions (obtained by any method) are *by counting pairs* and *by adjusted indices*. Several indices that we mention below (Rand, Jaccard, etc.) are part of these categories along with their adjusted versions. Basically, they count the number of data point pairs on which both partitions agree or disagree. Since they never reach 0, they are sometimes adjusted to correct this problem.

One can measure the proportion of sequences assigned to the same cluster in two distinct clusterings using, for instance, the adjusted Rand index, the Jaccard index, or the Fowlkes and Mallows index Steinley [146]. These indices can also be used to validate a clustering. For instance, the Rand index allows the comparison of two clustering partitions even when the number of clusters are different. Considering a set of observed sequences $S = s_1 \dots s_n$ and two different clustering partitions $V = \{v_1 \dots v_r\}$ and $W = \{w_1 \dots w_r\}$ with subsets (clusters) r and m respectively, the index takes into account

- a - the number of pairs of elements belonging to the same cluster in V and the same cluster in W;
- b - the number of pairs of elements belonging to the same cluster in V and a different cluster in W;
- c - the number of pairs of elements belonging to different clusters in V and the same cluster in W;
- d - the number of pairs of elements belonging to different clusters in V and W.

We can compute the Rand index by creating a table from the two clustering partitions:

V cluster /W cluster	1	...	r	Total
1	n_{11}	...	n_{1r}	$n_{1.}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
m	n_{m1}	...	n_{mr}	$n_{m.}$
Total	$n_{.1}$	...	$n_{.r}$	$n_{..}$

where $n_{m,r}$ is the number of observations clustered at the same time in the m-th and r-th groups in partitions V and W, respectively. Then,

$$a = \frac{\sum_{m=1}^{M}\sum_{r=1}^{R} n_{mr}^2 - n_{..}}{2}$$

$$b = \frac{\sum_{m=1}^{M} n_{m.}^2 - \sum_{m=1}^{M}\sum_{r=1}^{R} n_{mr}^2}{2}$$

$$c = \frac{\sum_{r=1}^{R} n_{.r}^2 - \sum_{m=1}^{M}\sum_{r=1}^{R} n_{mr}^2}{2}$$

$$d = \frac{\sum_{m=1}^{M}\sum_{r=1}^{R} n_{mr}^2 + n_{..}^2 - \sum_{m=1}^{M} n_{m.}^2 - \sum_{r=1}^{R} n_{.r}^2}{2}$$

and the Rand index is written as

$$R(V,W) = \frac{a+d}{a+b+c+d} = \frac{a+d}{\binom{n}{2}} \tag{4.3}$$

A transformation of the Rand index adjusted for chance is the Adjusted Rand Index (ARI), proposed by Rand [122]. ARI measures the similarity between two data clusterings, but instead of varying between 0 and 1, it takes negative values when its value is smaller than the expected index value. Acording to Hennig [65], ARI may also be used to compare two clustering methods on the same dataset.

The Jaccard similarity measure is also based on membership of the observations to a given cluster. It is calculated as

$$J(V,W) = \frac{\{V \cap W\}}{\{V \cup W\}} = \frac{\{V \cap W\}}{\{V\} + \{W\} - \{V \cap W\}}$$

where

$$V, W \subseteq x_n, \quad J(V,W) \in [0;1]$$

This measure represents the part of the observations belonging to both sets divided by the total number of observations in either of sets V and W (partitions of observations x_n) Bank and Cole [6]. Since the labels are not always identifiable when V and W are clustering partitions, the *Jaccard index* Meila [97] is computed as

$$J(V,W) = \frac{a}{a+b+c}.$$

Another example is the silhouette statistic proposed by Rousseeuw [130], which can be calculated with any distance metric.

Assessing the stability of clustering

The stability of clusters needs to be assessed because the methods of clustering usually partition the data even when there is no true clustering inside, or when the clustering model is only partially adapted or not adapted at all to the data. Several criteria are often used for a "good clustering". According to Jin [71]:

- *Compactness* involves reducing the within-cluster variation. A specific method to achieve this is the K-means algorithm that performs well when the data are clearly separated, but does not perform well when the cluster structure is more complex.

- *Connectedness* implies that neighboring data should belong to the same cluster. Density-based methods implement this principle. According to Jin, they are adapted to detect irregular shape clusters, but "lack robustness when there is little spatial separation between clusters."

- *Spatial separation* is a criterion that "gives little guidance during the clustering process." As Jin underlines, it should be combined with other objectives, "most notably [the] measures of compactness or balance of cluster sizes." Spatial separation is actually the opposite of connectedness: the observations must be connected within a group, but there must be spatial separation between the different groups.

Instability can result not only from the clustering method, but also from fromfeatures of the data or a mismatch of the data to the model or bad preprocessing choices" Hennig [65]. On the other hand, clustering stability is defined in many different ways (see Steinley [145] and the references therein), such as the stability of data with regard to the choice of the clustering method, measuring clustering robustness against the randomness of the sample Fang and Wang [50], and the ability of the cluster solution to be recognized in different random samples of the population.

Leisch [84] details the existing resampling methods to assess the stability of a clustering. In this perspective, he highlights two sources of randomness in cluster partitioning: sampling and the algorithm. Clustering every bootstrap sample separately is one way to assess the influence of sampling on the stability of a clustering. A cluster is said to be stable if it is still present (identifiable) when the data suffer some nonessential modifications. Such modifications can be:

- bootstrapping (the empirical as well as parametric version);

- jittering, which consists in adding noise to the original data sample (note that jittering alone was not recomended by Hennig [64] for stability assessment);
- subsetting, which is a kind of cross-validation wherein one draws a training set from the original sample of random size πn (where $\pi \in [0, 1]$) and uses the rest of the observations as a validation set;
- replacing points by noise.

We might add that the clustering should be present when a new (or unused) sample is received from the same population.

Leisch also presents a general approach to evaluate the cluster stability in several steps as follows:

1. From the original sample, draw two training samples A^i, B^i and one evaluation sample Γ^i, where i is the iteration ($1 \leq i \leq I$).
2. Cluster A^i, B^i using the same number of clusters k. The resulting partitions are Π^{A^i} and Π^{B^i}.
3. Predict the membership of the evaluation set Γ^i from the partitions obtained on the training samples.

   The author provides examples with a k-means-type approach with transversal data, that is, assigns the Γ^i points to the nearest centroids from A^i and B^i. However, given the model-based clustering approach that we promote in this thesis, we could use the estimated sets of parameters $\hat{\Theta}_{A^i}$, $\hat{\Theta}_{B^i}$ obtained during both partitions, to obtain the following partitions on Γ^i: $\Pi^{\Gamma A^i}$ and $\Pi^{\Gamma B^i}$.
4. The resulting two clusterings are compared using the given statistic s^i.
5. The above procedure is repeated I times ($1 \leq i \leq I$).
6. s^i is summarized.

Some examples of the statistics and particular procedures following this general scheme are provided. Among them is the procedure implemented by Hennig [64] (2007), which measures the local stability (instead of the global one) of each cluster within a partition. At first, only one sample is drawn using bootstrap (the other data modifications mentioned above could also be used). Instead of the evaluation set Γ^i, Hennig uses the intersection between the original sample X and the bootstrap sample A^i ($\Gamma^i = X \cap A^i$,

which is equivalent to the bootstrap sample without the repeating observations). The Jaccard similarity measure is computed between the k-th cluster of the original sample (A_k) and each of the bootstrap clusters $\Pi_{k*}^{\Gamma A,i}$. The maximum Jaccard agreement corresponds to the maximum of these measures, and represents the stability of the cluster k:

$$j_k^i = \max_{1 \leq k* \leq K} \frac{\{A_k \cap \Pi_{k*}^{\Gamma A,i}\}}{\{A_k \cup \Pi_{k*}^{\Gamma A,i}\}}, where\ A, B \subseteq x_n$$

By averaging the stability (j_k^i) over all iterations i, one obtains an indicator of the stability of each particular cluster A_k from the original sample.

When we attempt to use a simple nonparametric bootstrap to provide parameter estimate CIs for clusters on longitudinal data, we need to determine whether all the resulting clustering partitions are mutually compatible. For instance, if we obtain a group of monotone decreasing sequences in one clustering partition, a group with the same property should also be identifiable in a clustering partition obtained from another bootstrap sample using the same method. Hence, the stability measure is an important indicator that one should take into account. However, following Hennig et al. [66], a large stability is not sufficient to validate a clustering solution. A low or high stability does not depend only on the applied method, because methods that are less influenced by single observations or small modifications of the data can give meaningless but stable clusters. Clustering is useful only when the data really exhibit several clusters, and the model assumptions need to be adapted to the data. However, a low stability can be an indicator of potential problems. In order to accept a clustering solution, data visualization is recommended, along with a thorough knowledge of the data and their source.

Dias & Vermunt [44] used bootstrap methods to measure the classification uncertainty in latent class models that represent mixtures of conditionally independent multinomial distributions. They measured the uncertainty at both individual and aggregate levels, to provide examples with longitudinal data. Another use of the procedure is to identify observations characterized by higher classification uncertainty. However, the authors show that the number of clusters is fixed, indicating that it should have been optimized beforehand. This method is not affected by label switching as the measures do not depend on the labels.

The aim is to maximize the estimated probability $\hat{\alpha}_{ik}$ that observation i belongs to the given cluster k:

$$\hat{\alpha}_{ik} = \frac{\hat{\pi}_k f_k(x_i; \hat{\Theta}_k)}{\sum_{c=1}^{K} \hat{\pi}_c f_c(x_i; \hat{\Theta}_c)}$$

where $\hat{\Theta}_k$ and π_k are respectively the parameters and the probability of component k, and its distribution is $f_k(x_i; \hat{\Theta}_k)$. The "soft" partitioning is transformed into a "hard" partitioning by assigning the observation to the class k that maximizes its probability of membership (a.k.a. the optimal Bayes rule):

$$\hat{k}_i = \arg\max_k \hat{\alpha}_{ik}$$

At the individual level, the uncertainty is measured by the misclassification risk for each observation i:

$$e_i = 1 - \max_k \alpha_{ik}$$

and at the aggregate level, the entropy is calculated as

$$E(\alpha) = -\sum_{i=1}^{n}\sum_{c=1}^{K} \alpha_{ik} \log \alpha_{ik}$$

4.5 Conclusion

In this chapter, we explored the particularities and issues that arise when assessing the uncertainty of the parameter estimates of mixture models in the presence of label switching and other problems. We also described some methods to validate an initial clustering solution, and we proposed two bootstrap methods to achieve this goal. They rely on the first clustering partition "approved" by the researcher to different extents. We also listed some methods to approve (or validate) such partition before using it for that purpose. The use of these two alternative procedures, especially the *separated* bootstrap method, implies a way of considering the clustering process by separating the model choice and validation from parameter uncertainty. The use of *ordinary* nonparametric bootstrap implies that the tasks of discovering the optimal clustering model, validating it, and calculating the parameter intervals are performed simultaneously. On the other hand, the *stratified* bootstrap is a hybrid solution that uses the initial solution, but that leaves the possibility of a model choice error. However, it ensures that the initial clusters are well represented throughout the bootstrap sampling. In the *separated* version, the current model obtained from the original sample is hypothesized to be correct, such that we only need to validate its parameters, for instance when we try to simplify the model (e.g., when some lag parameters are not significant), or to merge two clusters into a single one, should their parameters be close enough to do that.

When the chosen initial clustering solution corresponds to a local optimum in the estimation procedure, or when the number of components is not appropriate, the initial solution may contain more misidentified sequences that could introduce bias into the CI calculation. This is especially true for the separated bootstrap method, and this highlights the importance of the initial solution for this method. However, when the sequences are well identified from the beginning, and the chosen number of components is appropriate, this method can prove advantageous over its alternatives. Furthermore, model uncertainty is not always a primary objective in practice, because the researcher often attempts to find not only the stability of a clustering solution, compared to all other possible solutions, but also an adequate clustering based on the knowledge of the data behavior and the aim of the study. In this case, one may ask the question, "If this clustering appears correct and meaningful considering our knowledge about the data, how much could its characteristics vary?" rather than, "Is this really the one and only clustering that is appropriate for those data?". While a separated bootstrap is clearly more appropriate to answer to the first question, the ordinary and stratified bootstraps are better suited to answer to the second one. Therefore, each procedure finds its utility, depending on the problem to be solved.

In addition to the above considerations, the stratified version may also be useful for small-size samples, or when at least one of the groups appears evident but small, and it is not certain to be represented in the bootstrap samples.

Before introducing an example in the following chapter, hereafter we summarize our point of view on how a full clustering procedure in the frequentist mixture model estimation should be performed:

procedure FREQUENTIST MIXTURE CLUSTERING

1. Apply a model to find the appropriate solution(s)
2. Evaluate the model uncertainty and compute the stability of the clustering partitions
3. Use a method to compare the possible candidate solutions
4. Evaluate the interpretability of the solutions

if A stable, interpretable solution is validated **then**

Use **separate** bootstraps

Pros: Eliminates the label-switching problem;
Eliminates the genuine multimodality problem as well as the singularity issue

Cons: The selected model must be tested and accepted beforehand

else

if The solution does not correspond to all criteria or is not entirely satisfactory **then**

Stratified bootstrap
combined with label-switching and multimodality solutions.

Pros: Small important samples do not "vanish";
Multimodality issues should be a bit less important than in ordinary bootstraps

Cons: Label switching is present;
Multimodality and singularity problems cannot be ignored

else No stable or interpretable clustering is found

Ordinary bootstrap
combined with label-switching and multimodality solutions.

Pros: No assumption for the model (→ more flexibility)

Cons: Label switching is present; singularity issue is possible
Multimodality is present; small clusters may vanish.

end if

If label switching or multimodality are too difficult to solve → Separate bootstrap

end if

end procedure

Chapter 5

Coping with clustering uncertainty: example

In this Chapter, we provide an example of some of the concepts developed in the previous chapter. We illustrate these concepts through the somatic complaints data.[1] The specific aim of this chapter is not only to study the concrete problem of the somatic complaints data (which was the objective of the published paper), but also to re-consider the example by focussing on the concepts that have been developed in the previous chapter. Therefore the chapter includes additional parts also for illustration purposes.

5.1 Introduction

Somatic complaints such as headaches, stomach aches or sleep disturbances are very common at all ages. They are a leading reason for seeking medical care, accounting for up to 50% of new medical outpatient visits Mohapatra, Deo, Satapathy & Rath [100]. These symptoms often appear during childhood and then increase through adolescence and adulthood. In Switzerland, a study showed an increase in the number and importance of these symptoms among 11-15 year olds between 1996 and 2004 Dey, Jorm & Mackinnon [43]. In addition to lowering the quality of everyday life, the presence of somatic complaints is often a clue for more serious problems, either already present or likely to grow rapidly. Understanding the causes of somatic complaints is therefore crucial to prevent and/or identify and treat more important health problems.

[1]Some parts of this Chapter, and especially the final model clustering the trajectories of somatic troubles into five groups, are taken from a paper accepted for publication in the Swiss Journal of Sociology Berchtold A., Surís J.-C., Meyer T. and Taushanov Z. [19].

Accordingly, a Dutch study showed that young adults with severe somatic disabilities since childhood achieved less life milestones than their healthy peers, or achieved them with delay, implying a lower probability of full social and professional integration Verhoof, Maurice-Stam, Heymans & Grootenhuis [162]. A higher level of somatic issues at mid-age has also been associated with reduced accumulation of social capital from adolescence throughout the life course Jonsson, Hammarström & Gustafsson [73].

The main purpose of this study was to assess the presence and development of somatic complaints among adolescents and young adults living in Switzerland during a life period during which crucial transitions occur, such as labour market entrance and the foundation of a family. With regard to the development of individuals from mid-adolescence to young adulthood, we searched to identify specific subgroup trajectories of somatic complaint development and link these trajectories both to personal and socio-economic factors prone to shape the overall trajectory, as well as to critical life events. Given the scarcity of previous longitudinal studies analysing somatic complaints among adolescents and young adults, it was difficult to predict which would be the most likely shapes of these trajectories, excepted maybe a slight trend to an increase of the number of complaints during early adolescence associated with a high variability between individuals. Moreover, it was reasonable to postulate than some adolescents experience only a very small number of complaints, which corresponds to a quite flat trajectory with a low average value.

5.2 Data and modeling

We used data from the Transitions from Education to Employment study [TREE]. TREE is a follow-up survey of the Swiss sample tested by the Programme for International Student Assessment (PISA) survey in 2000, collecting longitudinal data among more than 6,000 school leavers from 2001 (mean age 16 years) to 2014. Data available to date include PISA 2000 (baseline survey) and nine follow-up panel waves carried out between 2001 and 2014 (at annual intervals between 2001 and 2007), but the study is still ongoing and a further wave is planned for 2019. The presence of eight somatic troubles (stomach ache, lack of appetite, lack of concentration, vertigo, sleeping disorder, nervousness, fatigue, headaches) was regularly surveyed on each TREE panel wave, drawing on the Berner Fragebogen zum Wohlbefinden Jugendlicher (Grob et al. [58]). There were five possible answers for each somatic trouble ranging from never to everyday. These answers were recoded from 0 to 4 and then summed in order to obtain an overall somatic trouble score ranging from 0 to 32. This score was then used as the

dependent variable in our analyses.

Several covariates were included in the model, either fixed or time-varying. The fixed covariates were gender (female/male), country of birth (Switzerland/other), academic track attended at mandatory school (high/extended/basic), residence (rural area/urban area), PISA reading literacy (6 levels from 0 = very low to 5 = very high; treated as a numerical scale hereafter), highest parental socio-economic status (ISEI scale) and family wealth (scale representing the possessions of the family such as cars and TV sets). Residence was included based on the hypothesis that living in an urban area can be more stressful than living in a rural area, which could in turn favour the development of somatic troubles. All variables were measured in 2000 as part of the PISA survey. On the contrary, critical life events were measured at each subsequent wave of the TREE survey. The number of surveyed critical life events varied between 12 and 16 across panel waves, including an open text option from the second wave onward. Reported events comprised relocation of parental family; moving out of the parental home; parental and own separation or divorce; death, serious accident/illness or unemployment of relevant others; trouble with the police; unhappy love; serious conflicts in the family, at school or at work; pregnancy and parenthood. Two time-varying covariates were computed from these life events. The first one was the number of critical life events reported each year. The second one was a dichotomous factor indicating whether at least one critical event was reported or not. Finally, the consumption level of four types of substances (alcohol, tobacco, cannabis, tranquilizers/sleeping pills) was assessed at each TREE wave with five possible answers ranging from "never" to "every day".

In the first step, the HMTD model was used to identify the required number of groups for classifying the data sequences, and the order of dependence for the autoregressive modelling of the mean value of the somatic troubles scale. No covariates were used at this point. Models were compared on the basis of their log-likelihood, their Bayesian Information Criterion (BIC) values (Raftery [120]), and the number of sequences assigned to each group, thereby discarding solutions with very few sequences in some groups. Then, time-invariant covariates were introduced one by one at the hidden level. All significant covariates were then introduced simultaneously in the model to improve the clustering. Finally, the time-varying covariate representing the occurrence of critical life events was added at the visible level to improve the modelling of the mean of the somatic trouble score.

Critical life events were mainly shocks occurring at a precise time, but whose effects could be felt for a long period. Specific examples were the death of a family member, an unhappy love or a sudden hospitalization. Their impact on somatic troubles could

therefore be easily conceptualized. On the other hand, substance use was mostly a continuous behaviour that was difficult to break into specific events. Even the beginning of consumption of a specific substance was difficult to assess, because 1) someone could begin (and cease) to use a substance several times, and 2) our dataset could not be used to determine whether a specific substance was used before the first wave. Moreover, a sudden change in the level of consumption could not always be clearly identified in our data, because questions about substance use only referred to the month preceding the survey panel. Therefore, we chose to integrate critical life events and substance use in two different ways in our analyses. Critical events were used as a time-varying covariate influencing the average level of somatic troubles into each group of the clustering, when the association between substance use and trajectories was established a posteriori on the final clustering using a chi-square test.

Continuous covariates were standardized in order to ease the convergence of the optimization algorithm. The Type I error was fixed to 5%.

5.3 Results

Data from N = 1161 respondents continuously observed from 2001 to 2014 were included in all analyses. These individuals represent only 18% of the total TREE sample, but we preferred not to impute missing data, based on further analyses. However it is important to note this information. Table 5.1 summarizes the main information about the sum score for somatic complaints. This scale showed good psychometric properties with a Cronbach's alpha value ranging from 0.78 (T1) to 0.82 (T6). Whatever the wave, the score was highly variable from one respondent to another, but the central tendency measured by the mean and the median did not vary much. Most scores were below 20, but each year a small number of larger values were observed.

Table 5.2 and Table 5.3 describe the covariates considered in this study. When comparing our data with the full TREE sample, females and students in the pre-gymnasial school track were over-represented, while German speaking youths were slightly under-represented. This will be discussed later, but it did not affect our results. The number of critical life events increased year to year, which is to be expected given the probability of experiencing some of the events surveyed, such as getting married or becoming a parent, increases with age. Moreover, the variability between respondents also increased with age, even though few individuals reported many events in a given survey wave.

Table 5.1: Main characteristics of the somatic complaint score.

Year	2001	2002	2003	2004	2005	2006	2007	2010	2014
Minimum	0	0	0	0	0	0	0	0	0
Maximum	32	29	28	28	31	32	27	25	28
Median	6	6	6	6	5	6	6	6	5
Mean	7.14	6.91	7.07	6.75	6.27	7.14	6.85	6.26	6.05
Standard deviation	4.83	4.81	4.79	4.76	4.58	4.76	4.59	4.31	4.24
Cronbach's alpha	0.78	0.80	0.80	0.80	0.80	0.82	0.80	0.79	0.79

Table 5.2: Main characteristics of the time invariant covariates measured in year 2000. We provide the prevalence of each category and the corresponding percentage for categorical variables, and the mean and standard deviation for numerical variables.

Variable	Categories	Distribution
Gender	Female	749 (64.51%)
	Male	412 (35.49%)
Country of birth	Switzerland	1097 (94.49%)
	Other	64 (5.51%)
Academic track attended at lower secondary education level	High	574 (49.44%)
	Extended	432 (37.21%)
	Basic	155 (13.35%)
Residence	Rural area	433 (37.30%)
	Urban area	728 (62.70%)
(PISA) Reading literacy	0 - 5 [2]	3.50 (1.00)
Highest parental ISEI	16 - 90 [2]	53.18 (15.45)
Family wealth	-2.93 - 3.38 [2]	0.05 (0.76)

Table 5.3: Main characteristics of the critical life events score.

Year	2001	2002	2003	2004	2005	2006	2007	2010	2014
Minimum	0	0	0	0	0	0	0	0	0
Maximum	6	5	7	12	6	8	8	14	9
Median	0	0	0	0	1	1	1	2	2
Mean	0.69	0.69	0.70	0.74	0.91	1.09	1.09	2.26	2.57
Standard deviation	0.97	0.93	0.94	1.06	1.11	1.26	1.23	1.50	1.50
Number of respondents reporting > 0 events	518	526	536	547	605	664	680	1041	1085

As a first step, we considered HMTD models with 2 to 8 hidden groups and a first- or second-order dependence for the mean of the somatic complaint score. Based on the BIC, the preferred model was the second-order model with 6 groups. Using the previous two observations of the dependent variable to explain the current observations yielded better results than using only the immediately preceding observation. We subsequently added the fixed covariates one by one at the hidden level. Five covariates contributed to improve the fit of the model: gender, residence, reading literacy, socio-economic status and family wealth. These covariates were then introduced together at the hidden level, and we added the time-varying critical life events covariate at the visible level, either in its continuous or dichotomous form. Both versions of this latter covariate proved useful in improving the clustering of the somatic complaint score trajectories, but the best results were obtained with the continuous covariate. Finally, since two of the six groups were very close in terms of trajectories and parameters, we computed the same model with only five groups. This model was chosen as the final solution.

Table 5.4 displays the parameters of the final HMTD model, and Figure 5.1 shows the clustering of the somatic complaint trajectories into the five groups identified by the model. Trajectories must be analysed in terms of average value and of variability, both during a particular sequence or between sequences. Accordingly, the figure and the model parameters indicated that 1) the groups differed both in terms of average level of somatic complaints and variability; 2) inter-subject variability remained high,

[2]Minimal and maximal values observed in the sample.

even within the same cluster, indicating that almost all individuals followed their own trajectory; and 3) intra-subject variability (that is across time for a specific individual) was high for the trajectories classified in groups 1 to 3, and much lower for trajectories classified in groups 4 and 5. Group 5, which comprises about half the sample (n = 528), was used as the reference group for the analysis. Regarding the covariates used at the hidden level, gender was the most important one for distinguishing between the groups, with more males in group 4 and fewer in groups 1 to 3 as compared to group 5 (see Table 5.5). The proportion of females classified in each of the five groups were 80%, 90%, 76%, 49%, and 68% respectively. The only other significant covariate was reading literacy, which was lower in group 2. When considering other groups as the reference (data not shown), it appeared that groups 1, 2 and 4 significantly differed in terms of gender, while groups 4 and 5 differed from group 2 regarding the reading literacy level, with a significantly lower reading literacy level in group 2. On the other hand, the residence, socio-economic and family wealth covariates were never significant, even though some coefficients were very close to significance, especially the coefficient of the critical life events covariate in the case of group 5.

Group 4 comprised the respondents with lower overall somatic complaint scores and with relatively low changes between periods, that is the individuals with the overall lowest level of somatic complaints. Both lags of the dependent variable and the critical life events covariate were significant at the visible level. Compared to group 4, group 5 comprised respondents with a slightly higher variability of scores over time, while average scores varied more and were significantly higher in a number of cases. On the other hand, groups 1 to 3 comprised respondents with substantially more complex trajectories of somatic complaints: both their variability and their overall level was higher, especially in group 1, and no influence of the critical life events covariate was observed. Moreover, the individuals classified into these three groups had generally one or several periods with a high level of somatic complaints. Both lags of the dependent variable were significant in group 2, while only the first lag was significant in groups 1 and 3, indicating that in these two latter groups, the past levels of somatic complaints had less effect on the current level.

Table 5.4: Parameters of the final HMTD clustering model. At the hidden level, the last group served as reference for the computation of the multinomial regression used to add the fixed covariates to the model. We provide for each parameter the point estimation and the 95% confidence interval. Parameters significant at the 95% level are printed in bold.

Hidden level: Clustering of somatic complaints trajectories into 5 groups					
Groups	Residence (urban area)	Gender (male)	Reading literacy	Hisei	Family wealth
1	0.42	**-1.00**	-0.13	0.06	0.16
	[-0.08;0.93]	[-1.59;-0.41]	[-0.39;0.12]	[-0.18;0.31]	[-0.16;0.47]
2	0.43	**-2.10**	**-0.47**	-0.03	0.13
	[-0.25;1.12]	[-3.29;-0.91]	[-0.81;-0.12]	[-0.36;0.30]	[-0.29;0.56]
3	0.02	-0.30	-0.14	0.03	0.04
	[-0.32;0.35]	[-0.65;0.05]	[-0.32;0.04]	[-0.14;0.21]	[-0.19;0.26]
4	0.03	**0.44**	-0.02	0.03	0.18
	[-0.26;0.31]	[0.15;0.72]	[-0.17;0.13]	[-0.12;0.18]	[-0.01;0.37]
5	Ref	Ref	Ref	Ref	Ref

Visible level: Observed levels of somatic complaints					
		Factors explaining the mean level of somatic complaints			
Groups	Variance	Constant	Lag 1	Lag 2	Critical life events
1	**26.99**	**11.68**	**0.20**	-0.09	-0.08
(n=46)	[21.31;32.56]	[10.15;13.62]	[0.09;0.33]	[-0.22;0.01]	[-0.47;0.39]
2	**20.55**	**9.55**	**0.16**	**0.29**	0.17
(n=30)	[15.48;25.34]	[7.90;11.99]	[0.04:0.26]	[0.15:0.41]	[-0.39;0.68]
3	**18.01**	**6.29**	**0.28**	0.01	0.14
(n=204)	[16.92;19.06]	[5.79;6.92]	[0.24;0.33]	[-0.05;0.06]	[-0.10;0.36]
4	**3.39**	**2.25**	**0.23**	**0.11**	**0.14**
(n=353)	[3.18;3.57]	[2.06;2.45]	[0.20;0.27]	[0.08;0.15]	[0.04;0.23]
5	**7.21**	**1.71**	**0.42**	**0.31**	0.05
(n=528)	[6.80;7.62]	[1.48;1.96]	[0.38;0.46]	[0.27;0.34]	[-0.05;0.16]

Table 5.5: Main characteristics of the respondents classified in the five groups of the final model. We display, separately for each group, the percentage of each category for categorical variables, and the mean and standard deviation for numerical variables.

		Groups				
Variables	Categories	1	2	3	4	5
Gender	Female	80.4%	90.0%	75.5%	49.0%	67.8%
	Male	19.6%	10.0%	24.5%	51.0%	37.2%
Country of birth	Switzerland	87.0%	86.7%	92.6%	95.2%	95.8%
	Other	13.0%	13.3%	7.4%	4.8%	4.2%
Academic track attended	High	58.7%	40.0%	52.5%	45.6%	50.6%
at lower secondary	Extended	26.1%	30.0%	28.9%	36.0%	31.1%
education level	Basic	15.2%	30.0%	18.7%	18.4%	18.4%
Residence	Rural area	17.4%	33.3%	38.7%	38.0%	38.3%
	Urban area	82.6%	66.7%	61.3%	62.0%	61.7%
(PISA) Reading	0 - 5	3.35	3.10	3.46	3.53	3.53
literacy		(0.92)	(1.09)	(0.96)	(0.98)	(1.02)
Highest parental	16 - 90	54.70	48.90	52.84	54.15	52.78
ISEI		(15.71)	(16.67)	(15.53)	(15.70)	(15.15)
Family wealth	-2.93 - 3.38	0.20	-0.03	-0.01	0.14	0.00
		(0.74)	(0.82)	(0.75)	(0.73)	(0.77)

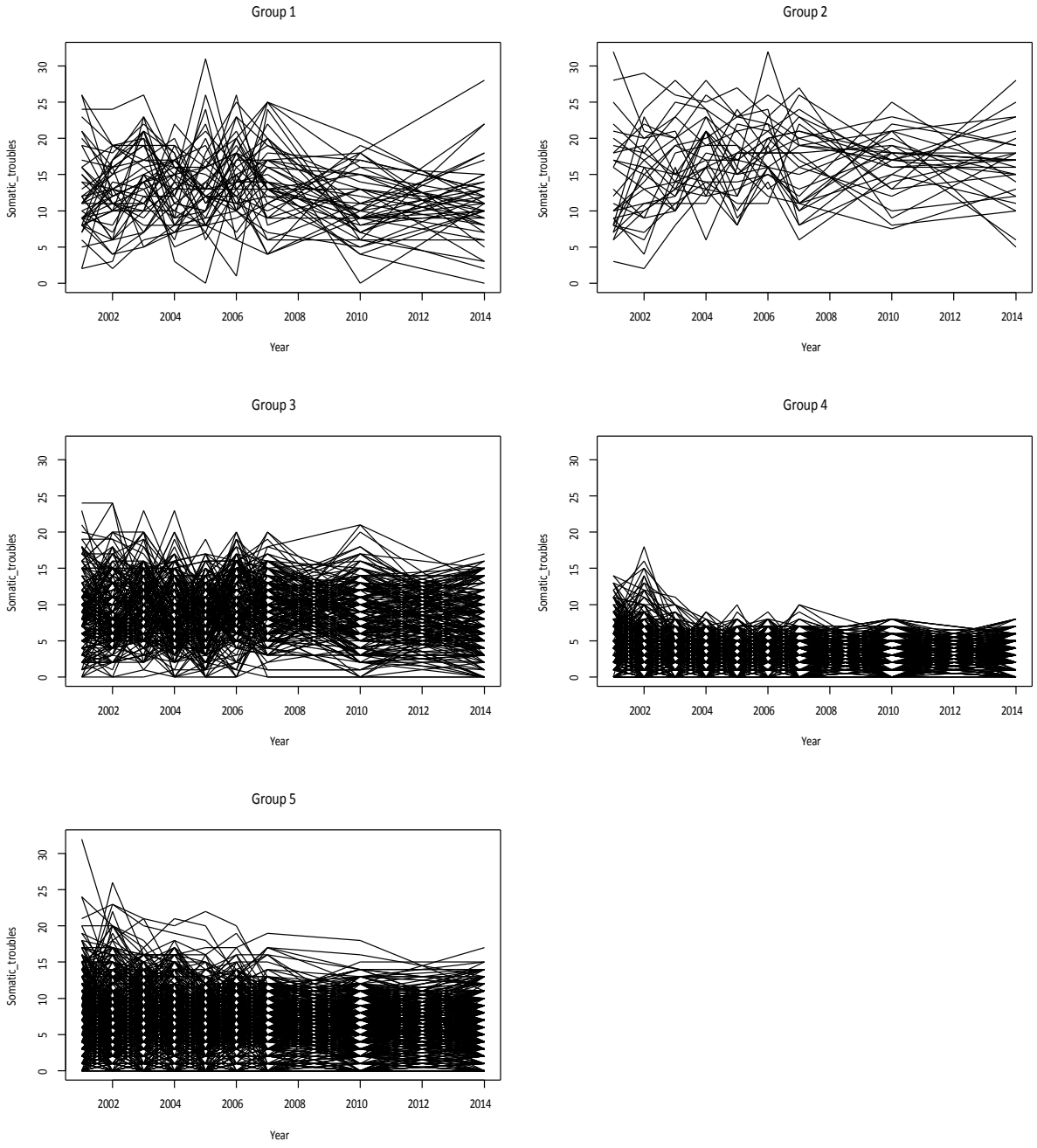


Figure 5.1: Somatic complaint trajectories of the final five groups identified by the model.

Table 5.6: Relationships between the five groups of the final model and substance use. For each wave of the study, we provide the Cramer?s V measure giving the level of association between groups and substance use, and the corresponding p-value.

		2001	2002	2003	2004	2005	2006	2007	2010	2014
Alcohol	V	0.05	0.06	0.08	0.05	0.08	0.07	0.07	0.08	0.09
	p	0.706	0.441	0.018	0.758	0.011	0.057	0.200	0.035	0.002
Tobacco	V	0.09	0.08	0.09	0.07	0.08	0.09	0.09	0.07	0.08
	p	0.001	0.039	0.002	0.107	0.016	0.001	0.004	0.194	0.010
Cannabis	V	0.06	0.08	0.08	0.08	0.07	0.06	0.09	0.06	0.08
	p	0.466	0.045	0.035	0.018	0.121	0.483	0.001	0.447	0.007
Tranquilizers	V	0.13	0.10	0.16	0.10	0.13	0.10	0.16	0.13	0.09
& sleep. pills	p	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Lastly, we compared the final clustering with the variables measuring the level of substance consumption (Table 5.6). Overall, the level of association was not very strong, but it was nevertheless highly significant in many cases. The association with alcohol consumption tended to change rapidly from one period to the next, but with no discernible trend. Similar findings were observed for cannabis consumption, where significant and non-significant associations alternated. The pattern of tobacco use was more distinct: the association with the five groups of the clustering was always significant except for 2004 and 2010. Compared to groups 4 and 5, daily smokers were represented at a substantially higher proportion in group 1, and to a lesser extent in groups 2 and 3 (data not shown). Finally, the association between the five groups and the use of tranquilizers and sleeping pills was highly significant for each period. Even if the consumption level was rarely higher than 1-3 times per month in all groups, and if most respondents did not consume at all, the average consumption level was significantly higher among respondents clustered into group 2 (data not shown).

5.4 Optimal clustering and validation

The model described before represents our final solution for the clustering of the somatic trouble trajectories. However, as noted before, the HMTD model often needs to be optimized several times before a solution is chosen. The presence of covariates at the visible and/or hidden level makes the parameter space even more complex and the maximization of the log-likelihood even more difficult.

Number of clusters In this study, we needed first to choose the number of clusters. An optimal solution for every number of clusters between two and eight was computed. The solutions were compared both in terms of interpretability and in terms of BIC and ICL. A five component solution was found to be the most appropriate and interpretable for this dataset.

However, as stated before, the solution space includes often several local optima. In order to insure that our model choice was not biased because of this issue, a bootstrap computation approach was used between the neighboring (and most plausible) number of components. In this case, we compared clusterings with four, five and six components. For all three models without covariates we drew 50 bootstrap samples and for each of them the BIC was computed. Sometimes, due to the complexity of the solution space, some solutions contained one or more empty clusters. Only the bootstrap iterations that had no empty clusters in any of the three model specifications were considered, in order to be able to compare them. We computed the average AIC and BIC for each number of clusters, but since the parameter number penalty term is the only difference between AIC and BIC, $AIC \propto BIC$. In order not to be influenced by single extreme BIC values, we also ordered each group of three solutions from the smallest to the largest BIC, and we computed the average rank over the iterations. Table 5.7 summarizes our results.

Table 5.7: Comparison of bootstrap solutions with 4 to 6 clusters.

	4 clusters	5 clusters	6 clusters
average AIC	41440.43	41387.38	41481.26
average BIC	41521.34	41488.52	41602.63
average BIC rank	2.01	1.71	2.28

Even though we cannot claim that these results are statistically significant, we observe that the solution with five groups seems better than the one with four groups, and the six-group solution takes the last place. We must note that, in order to take into account the increase in complexity of the parameter space as the number of clusters increases, we allowed a larger number (linearly) of iterations to optimize the likelihood of the more complex models. However, this choice seems not to have biased our results, since even with more iterations allowed, the 6 clusters solutions ranked last, and with less iterations allowed, the 4 clusters solution ranked second.

However, as explained in the previous chapter, AIC and BIC are not the only criteria for the choice of number of components. The ICL for instance, has the important advantage to take into account the separation within the data by including the entropy of each separation. In Table 5.8 we present the optimal solutions for each number of clusters, their entropy, log-likelihood and ICL values. For simplicity and parsimony the models are again computed without covariates.

Table 5.8: Log-likelihood, entropy, and ICL approximation for each number of clusters

number of clusters	2	3	4	5	6
logL	-20636.88	-20615.08	-20597.64	-20540.22	-20518.66
Entropy	-452.13	-658.12	-855.40	-1027.85	-1241.60
ICL	41743.47	41914.66	42085.85	42152.24	42331.65

The results here indicate that the two-clusters solution appears better. The main reason is that it is less penalized by the entropy of the solutions. Actually, in our case the gain in terms of log-likelihood from increasing the number of clusters is clearly inferior to the loss in terms of entropy and this trend appears to be rather linear.

One explanation of this rather unexpected result is that the data are not getting much better separation compared to the increasing penalty for the number of components implied in the entropy. To illustrate this, let us take an example with the maximal entropy (uncertainty of clustering) for a single data point in two models: with two and four clusters. With two clusters the maximum entropy is $2 \times 0.5 \times log(0.5) = -0.693$, whereas with four clusters we have: $4 \times 0.25 \times log(0.25) = -1.386$.

An important reason is also the fact that we cope with longitudinal data. With several periods for the same individual, we tend to have general means that are less distinct between the individuals, for instance the sequences (1,2,3,4) and (4,3,2,1) have exactly the same means and variances, although they are very different. Indeed, this difference is supposed to be captured by the other parameters (AR part for the mean) and they should still be discerned by the components, but perhaps the small number of periods is not allowing the clear separation based on the AR terms of the mean.

Moreover the fact that we were able to compute only the BIC approximation and not the true value of ICL, could also play a role in our results. However, the suggested ICL solution with only 2 large clusters was not as interesting in terms of interpretation, because few particular features were distinguishable between both groups.

Choice of solution and stability Once the number of components was chosen, we explored different solutions corresponding to different optima. In order to choose and validate a clustering, we also evaluated the stability of these candidate solutions. From the different procedures and indices previously mentioned, we needed first to choose the one that seems most appropriate to our data and particular problem. Since the indices based on dissimilarity or Euclidean distances are not adapted to continuous sequences, the only possibility was to assess the stability *by set matching*, i.e. to compare the membership of the observations after resampling. The procedure we adopted is based on the one proposed by Hennig [64], but with some modifications. Instead of evaluating the stability of every cluster separately, we tried to measure the stability of the entire clustering partitions in order to compare their global stability and choose one of them. For each iteration i:

1. A bootstrap sample x_n^i with n observations is randomly chosen.

2. A clustering on the resulting sample is performed. According to Hennig, the repeating observations may introduce a bias and need to be taken out of the sample Hennig [64].

3. The obtained partitions are compared to the original ones on the basis of the Rand or Jaccard indices.

4. The above steps are repeated m times and the indices are averaged and compared.

The choice of this procedure is motivated by the invariance to the labels when using the Rand or Jaccard indices. Since only the membership to either the same or a different class is taken into account, switching the labels does not have an impact on the index. For instance the two following sets match perfectly:

$$X = \{1, 1, 2, 3, 1, 3\}, \quad Y = \{2, 2, 3, 1, 2, 1\} \quad \rightarrow \quad Rand(X, Y) = 1$$

When using these indices, we need to also take into account some issues related to this index. The value of the Rand index does heavily depend on the cluster sizes within the partitions. That means that for the same number of divergences between two partitions, the values of the index are very different depending on the size of the clusters containing misclassified data. For instance, both of the following pairs of sets contain only two differences, but they do not achieve the same Rand and Jaccard values (using the R packages *fossil* and *clusteval*):

$X = \{1,1,1,1,1,1,1,2,2,3,4,4\}$,$Y = \{1,1,1,1,1,1,1,2,\mathbf{3},3,\mathbf{3},4\}$
$Rand(X,Y) = 0.924 \quad Jaccard(X,Y) = 0.808$
$X = \{1,1,1,1,1,1,1,2,2,3,4,4\}$,$Y = \{1,1,1,1,1,1,1,2,\mathbf{1},3,\mathbf{1},4\}$
$Rand(X,Y) = 0.742 \quad Jaccard(X,Y) = 0.553$

The same is true for the three following situations:
$X = \{1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3\}$,$Y = \{1,1,1,1,1,1,2,2,2,2,2,\mathbf{3},\mathbf{2},3,3,3,3,3\}$
$Rand(X,Y) = 0.869 \quad Jaccard(X,Y) = 0.636$
$X = \{1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3\}$,$Y = \{1,1,1,1,1,1,1,1,1,1,1,1,\mathbf{2},\mathbf{1},2,3,3,3\}$
$Rand(X,Y) = 0.830 \quad Jaccard(X,Y) = 0.726$
$X = \{1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,3\}$,$Y = \{1,1,1,1,1,1,1,1,1,1,1,1,1,2,\mathbf{3},\mathbf{2},3,3\}$
$Rand(X,Y) = 0.961 \quad Jaccard(X,Y) = 0.929$

In this case, the indices strongly privilege the errors that increase the smaller clusters. In the last two examples, we see that by exchanging the labels of two observations, both indices penalize more the errors made in the largest cluster, even though the proportions of the final clusters remain the same. The fact that the last example results in much higher stability indices is not convenient to achieve our aim of measuring cluster stability. The values of both indices are explained by the proportion and the size of the clusters within each partition, rather than by the misclassification in each iteration.

In my opinion, the sizes of the clusters should not be neglected, but the interpretability of the clusters is the most important argument for the choice of a clustering solution. All above mentioned approaches and indices should then only be considered as a complementary validation means.

We provide now an illustration with three alternative solutions (compared to the one fully described in the previous subsections) obtained with five clusters on the same dataset of somatic trouble trajectories. These alternative solutions are represented on Figures 5.2 to 5.4. These solutions correspond to local optima. From the three Figures, we observe that they do not seem to be optimal for this problem, because each solution contains at least one very small cluster. These small clusters lack interpretability and they could represent a situation of overfitting of the model, preventing generalizability. Their BIC values are also fairly close to the BIC of the chosen solution, but the latter remains slightly higher.

However, it is interesting that these solutions remain more stable according to the results of the Rand and Jaccard indices shown on Table 5.9. These results are obtained after 50 bootstrap iterations of the above-mentioned stability evaluation procedure and

they show a considerable advantage of all the alternative solutions over the chosen one. Note that the Jaccard values are systematically smaller than the Rand values. The main difference is that the Jaccard formula omits the true negatives, that is the number of pairs of observations that are not clustered together in both partitions (noted d in equation 4.3). These results are surprising and may be due to the difference in the cluster sizes between solutions that we illustrated above.

Table 5.9: Comparison of alternative solutions: average Rand and Jaccard indices.

	Rand	Jaccard	BIC
original solution	0.628	0.450	-20668.58
alternative solution 1	0.905	0.875	-20726.44
alternative solution 2	0.930	0.906	-20673.59
alternative solution 3	0.893	0.854	-20836.22

After observing the figures of the alternative solutions, we might conclude that evaluating the stability of the solutions via Rand and Jaccard indices is probably not the most reliable criterion to chose a clustering solution. If the interpretability of the solution is superior, the stability indices should not be taken into account, especially if the BIC confirms this choice. For these reasons, we still prefer the chosen solution.

5.5 Discussion

The main finding of this study was the identification of several apparently distinct groups of somatic complaints trajectories based on a scale representing the sum of eight different complaints. These trajectories remained distinct throughout the entire observation period covered by the TREE data, which is from age 16 to age 30. As these trajectories already differed at age 16, we can hypothesize that factors already present during childhood, and thus beyond the control of youths themselves, may be the cause of such a differentiation. Since it is known that a higher level of somatic complaints is associated with subsequent health issues, we can conclude that 1) possibly some groups of adolescents in Switzerland were experiencing a situation of vulnerability beginning before adolescence, and that 2) this condition may be likely to persist even beyond the period covered by our study. A second conclusion is that if critical life events may be related to somatic complaints, this relationship was visible only among individuals

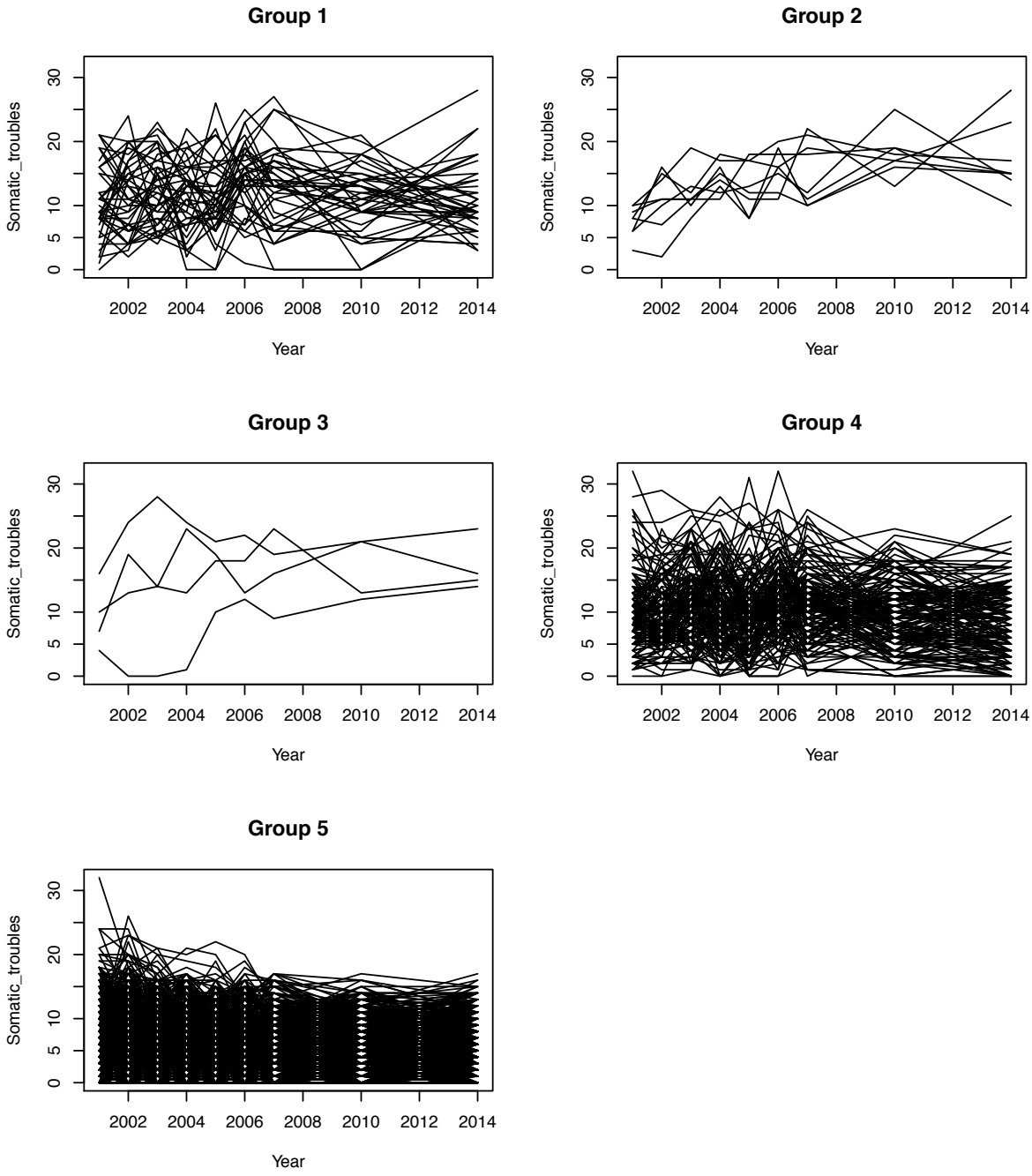


Figure 5.2: Alternative five-component solutions obtained for comparison: solution 1

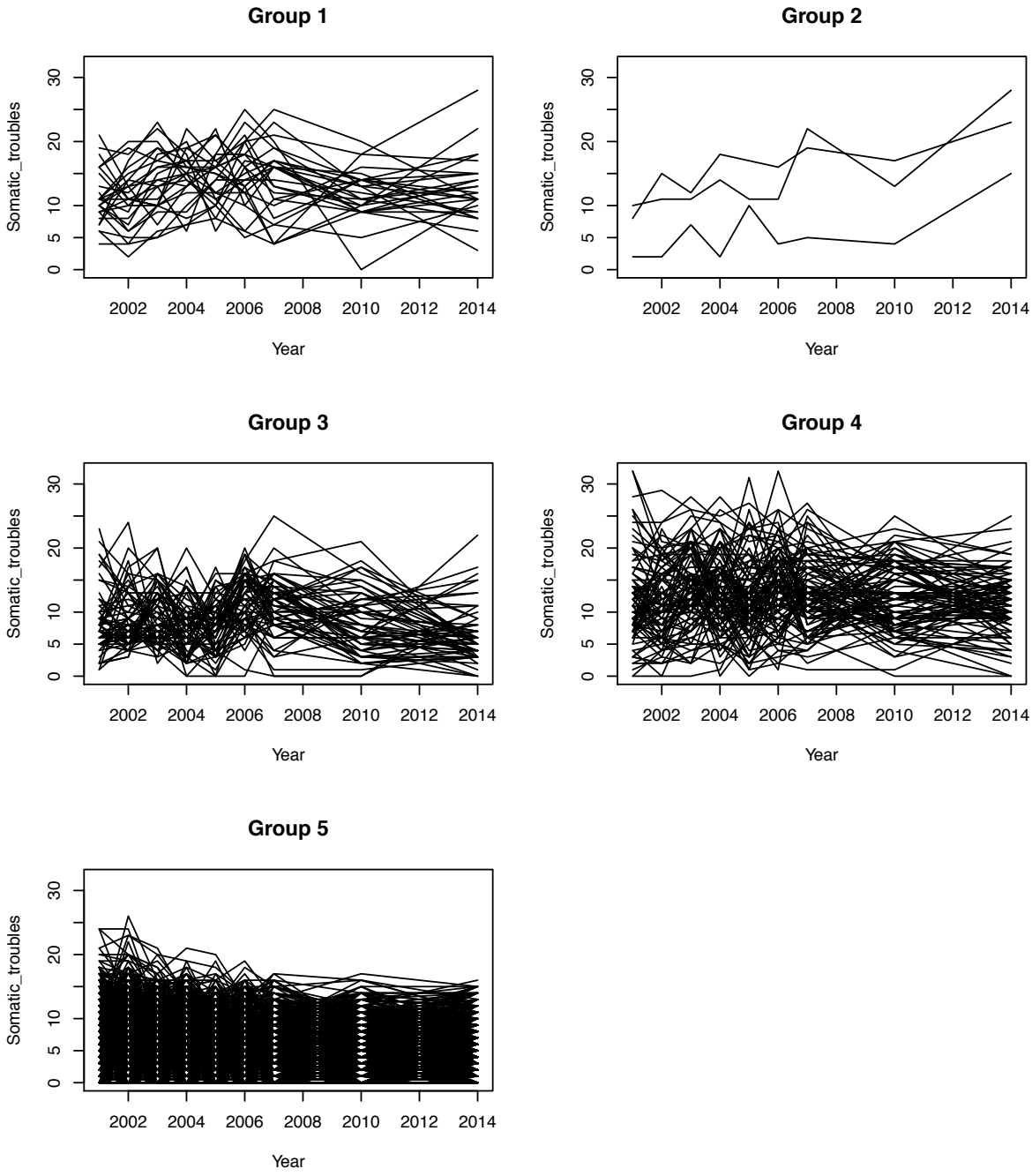


Figure 5.3: Alternative five-component solutions for comparison: solution 2

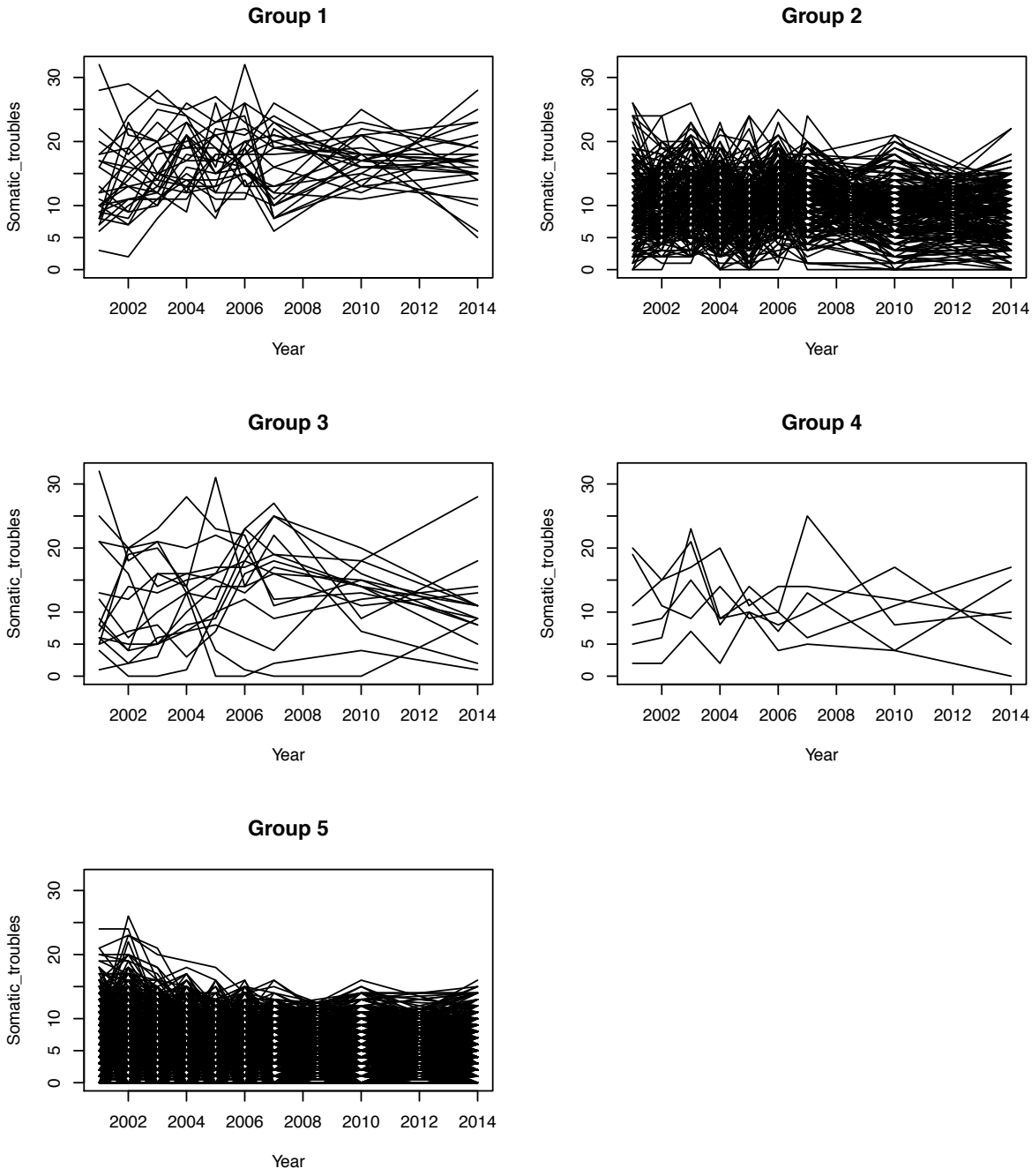


Figure 5.4: Alternative five-component solutions for comparison: solution 3

with low levels of somatic complaints. For their counterparts with high levels of somatic complaints, the impact of critical life events could have been masked by the inherent variability observed in somatic complaint scores. This brings us back to the fact that even if critical life events experienced during the transition period of adolescence and young adulthood could have an impact on somatic complaints, this impact remains limited, and the most important causes of a high level of somatic complaints are to be found elsewhere. This also leads us to the assumption that if somatic complaints are triggered by the occurrence of critical life events, the influence of such events appears to be mainly of short duration, as the somatic complaint score often decreased one period later. Thirdly, the consumption of tobacco and tranquilizers and sleeping pills was significantly associated with the typology of somatic complaint trajectories: higher substance consumption appears to be associated with the groups reporting the highest overall somatic scores. As most substance use began during the period covered by this study rather than before, it should not be considered as a cause of somatic complaints but rather as a consequence, especially in the case of tranquilizers and sleeping pills, which can be used to relieve some of these complaints.

The typology of somatic complaint trajectories identified in this study illustrates both the importance and the long lasting aspects of somatic complaints: important differences were observed between respondents classified into each group, with clearly differentiated overall somatic complaint levels. Moreover, even though both inter- and intra-subject variability may be high, many respondents classified in the first three groups stayed at a high level of reported complaints during the entire period of observation, i.e. from 16 to 30 years. In terms of life course, that means that the presence of somatic complaints early in life may be susceptible to deploy effects during the whole adolescence and (at least) the beginning of adulthood. Since many important determinants for the entire life (such as the entrance into the labour market, and the beginning of a steady relationship) are also taking ground during the same period, somatic complaints could be a very important indicator for the success or not of the entire life of an individual.

Finally, regarding the stability of the retained solution, we note that the choice of a five-cluster solution appears to be superior in terms of BIC to alternative four- and six-cluster solutions. Moreover, even if the stability of the final clusters can be questionned on the basis of the Jaccard and Rand measures, the retained solution presents the advantage of being more interpretable, hence more useful from a medical point of view.

Chapter 6

Clustering of IAT trajectories

In this chapter, I propose a full application of the HMTD model to the clustering of Internet Addiction Test (IAT) trajectories.[1]

6.1 Introduction

The clustering of trajectories has gained much interest in recent years from the scientific community, especially in the social sciences, because the number of longitudinal studies, as compared to cross-sectional ones, has been constantly increasing. As regards categorical data, the most common approach relies on the Optimal Matching (OM) to compute a distance between each pair of trajectories before clustering them, whereas the Growth Mixture Model (GMM) can be applied for continuous data. However, these two approaches suffer from some shortcomings, calling for the need to develop and apply alternative approaches. For instance, OM requires the choice of a substitution cost measure and other parameters. GMM gives a lot of importance to the shape of sequences. Therefore, there is a risk to overfit the data when nonlinear trajectories are considered on quite short sequences. The other issues of GMM include computational load, presence of local optima, missing data treatment, model selection criteria, the need for large sample size, and unclear Type I error rates Wang and Bodner [168].

In this paper, we study the use of a specific class of Markovian Models called the Hidden Mixture Transition Distribution (HMTD) model Bolano & Berchtold [24] for clustering purpose. Even if this model-based approach was developed as a tool for the analysis of continuous trajectories, it also allows for their clustering without a priori

[1] A slightly different version of this chapter has been accepted for publication in the post-proceedings volume of the LaCosa II conference Taushanov and Berchtold [153].

knowledge of cluster membership. Moreover, covariates can be easily included in the model.

The HMTD and GMM clustering approaches are applied and compared on a dataset of trajectories of the Internet Addiction Test (IAT). Excessive Internet use, especially among youths, is an emerging health issue in the medical literature, with studies showing contrasting results. Surís, Akre, Piguet, Ambresin, Zimmermann and Berchtold [150] show a significant association between problematic Internet use and several somatic disorders, including back, weight, musculoskeletal, and sleep problems. Moreover, several chronic conditions are also significantly associated with problematic Internet use. In contrast, another study finds no significant Internet use effect on the development of overweight among youths Barrense-Dias, Berchtold, Akre and Surís [7].

While several alternative approaches have been introduced over the years Skarupova, Olafsson and Blinka [141], the tool most often used to quantify the degree of addiction to Internet is still the Internet Addiction Test (IAT), developed by Young [176]. However, since the test's scale is based on 20 items and is quite long, its psychometric properties are matters of controversy Faraci, Craparo, Messina and Severino [51] and the test is not considered suitable for the successive measurement of the same subjects (test-retest). Its use in longitudinal contexts remains problematic because of the difficulty to distinguish between the real evolution of subjects and changes due to behavior of the IAT itself.

To gain information on the behavior of the IAT in longitudinal studies, we need to compare the typical trajectories of the repeated IAT measurements with other characteristics of the subjects under study. Thus, we first cluster the IAT trajectories into a finite set of meaningful groups and then compare these groups with the known characteristics of subjects that are either time-invariant or evolve over time. Specifically, the goals of this study are (1) to separate the Internet addiction trajectories into an optimal number of meaningful categories using HMTD, (2) to explore how does the introduction of the covariates influence the previous optimal partition, and (3) to compare the HMTD clustering with an equivalent GMM clustering in order to gain information on the respective strengths of both approaches. We hypothesize that (1) the IAT scores computed for the same person can vary considerably over time, implying that the trajectories are difficult to classify; (2) a classification using covariates is easier to interpret than a classification without any additional information on the clustered variable itself; and (3) the HMTD approach can lead to more sound and easier-to-use solutions as compared to the solutions obtained using GMM. However, we must stress that it is impossible to conclude that one method is superior to another, especially using real

data, without knowing the true cluster membership. So this work must be considered as a first step in the comparison of HMTD and GMM as clustering tools.

6.2 Data and methods

6.2.1 Data

The data we considered are from *ado@Internet.ch* Surís JC, Akre, Berchtold, Fleury-Schubert, Michaud and Zimmermann [149], a longitudinal study on the use of Internet among youths in the Swiss canton of Vaud (the largest canton in the French-speaking part of Switzerland). The data were collected five times with six months of interval, between Spring 2012 (T0, baseline) and Spring 2014 (T4) using an online questionnaire. The data for the first time were collected from schools during the computerlab periods. Then, the students who agreed to participate in the study were contacted again by email from T1 to T4 to answer follow-up questionnaires on their home computer. A convenience sample of n=185 adolescents who answered all five questionnaires is used for the present study (67% females; mean age at T0: 14.1 years). For more details on the overall design of the study and data collection, see Piguet, Berchtold, Zimmermann and Surís [111], Surís JC, Akre, Berchtold, Fleury-Schubert, Michaud and Zimmermann [149].

The main outcome is the IAT score measured at each wave for each subject. The IAT developed by Young [176] and validated in French by Khazaal and colleagues Khazaal et al. [79] is a scale ranging from 0 to 100, based on the answers to 20 items whose possible answers range from Never (coded 0) to Always (5). Examples of items are, *How often do you find yourself staying online longer than you intended?* and *How often do you fear that life without the Internet would be boring, empty, and joyless?*

In addition to the IAT, we also considered several important characteristics of the subjects, either fixed in time [gender, age at baseline, and education track at baseline (extended requirements vs. basic requirements)] or evolving over time [emotional well-being (measured by the WHO-5 index) and Body Mass Index (BMI, computed from auto-reported measures of height and weight)]. Note that the WHO-5 index was not evaluated on the third wave of the study, and so for the present paper, we imputed values as the simple mean between the values of the second and fourth waves. Similarly, we imputed the BMI for the second wave of the study as the mean between the values of the first and third waves.

6.2.2 Clustering using the HMTD model

We used a specific class of Markovian Models, the HMTD model, to cluster the longitudinal sequences of continuous data. This model combines a latent and an observed level Bolano & Berchtold [24]. The visible level is a Mixture Transition Distribution (MTD) model that was first introduced by Raftery in 1985 as an approximation of high-order Markov chains Raftery [119] and then developed by Berchtold [14], Berchtold and Sackett [18] and Berchtold & Raftery [17]. Here, we used a Gaussian version of the MTD model, where the mean of the Gaussian distribution is a function of past observations. Because of the small size of each sequence of the observed outcome (five data points, from T0 to T4), long dependencies between successive observations could not be considered, and therefore we fix the dependence order for the mean of the Gaussian distributions of each component to one:

$$\mu_{g,t} = \phi_{g,0} + \phi_{g,1}\, x_{t-1}$$

where $\phi_{g,0}$ is the constant for the mean for component g and $\phi_{g,1}$ is the autoregressive parameter indicating the dependence from the previous observation x_{t-1}. Similarly the variance of each component can be written as a function of the past periods variability: $\sigma^2_{g,t} = \theta_{g,0} + \sum_{s=1}^{S} \theta_{g,s}\, x^2_{t-s}$. However given the small number of time periods in our dataset, and for the sake of simplicity, we decided to treat the variance as a constant: $\sigma^2_{g,t} = \theta_{g,0}$.

In addition to the clustering based on the IAT variable only, we performed a second clustering adding information from five covariates (gender, age at T0, education track at T0, WHO-5, and BMI). These covariates are introduced as additional terms in the specification of the mean of each visible component of the model, and the categorical variables are introduced as dummy variables. We then rewrite the mean of the g-th component as

$$\begin{aligned} \mu_{g,t} \;=\; & \phi_{g,0} + \phi_{g,1}\, x_{t-1} + \phi_{g,2}\, Gender(male) + \phi_{g,3}\, Age \\ & + \phi_{g,4}\, Education(extended) + \phi_{g,5}\, WHO-5 + \phi_{g,6}\, BMI \end{aligned}$$

with *female* and *basic requirements* used as reference modalities for Gender and Education, respectively.

In practice, continuous covariates are centered around the sample mean before computing the clustering model in order to allow for a better convergence of the estimation

algorithm. A comparison of the two specifications of the mean, with and without covariates, illustrates whether the inclusion of covariates in the model helps to improve the clustering process. It must be mentioned that, in addition to these two HMTD models, many other specifications were tried, following a hierarchical approach Bolano & Berchtold [24], but none of these alternative specifications seemed to give a more useful clustering of IAT trajectories.

We used a bootstrap procedure to obtain confidence intervals for each parameter, but since our goal here was to validate not the initial classification itself, but the parameters associated with the model describing each visible component of the model, we adopted the following approach: Instead of performing the bootstrap on the whole original sample, we divided the original sample into as many groups as can be retained in the final classification. We then applied a single-component version of the HMTD model to each sub-sample separately in order to estimate the coefficients using bootstrap. By applying the model on the sub-samples separately, instead of on the initial sample, we avoided the so-called label-switching problem that is very common in latent variable clustering. The inconvenient of separate bootstrapping is that since we rely on the validated clustering solution, we ignore the model uncertainty including the weights of each cluster. We computed the confidence intervals using 1000 bootstrap samples, and we used the results to evaluate the significance of the estimated parameters.

All computations were done using R, and a specific package should be released soon. In the meantime, a first version of the R syntaxes is available on:

https://github.com/ztau/5352.

6.2.3 GMM as a gold standard alternative

To evaluate the HMTD approach as a tool for clustering sequences of continuous data, we need a gold standard alternative. We choose the Growth Mixture Model (GMM) approach for that purpose, since it is the only true longitudinal clustering tool used in the social sciences. A description of the GMM is provided in Section 2.3.6.

6.2.4 Statistical analyses

To start with, we used the HMTD model to identify the best clustering of the IAT dataset without covariates, considering solutions from two to five groups. The best solution was selected on the basis of the Bayesian Information Criterion (BIC) Raftery [120]. We then added covariates to this first model and analyzed the two resulting

models, with and without covariates, particularly focusing on the IAT trajectories that did change group when covariates were added to the initial model. In order to isolate the impact of the covariates from any other computational issue or local optimum, we used the optimal solution obtained without covariates as a starting point for the full model. Therefore, we observe how this new model escapes the previous optimum.

We then computed the GMM models using the same dataset, and we compared the classifications obtained with the HMTD and GMM approaches. The usefulness of each covariate for discriminating between groups was evaluated using either a chi-square test for categorical covariates, or a single factor ANOVA for continuous ones. Notice that since it is not easy to compare two solutions with different number of clusters, we chose to compute a four-cluster GMM solution with all covariates instead of finding its own optimal number of clusters.

Our results are presented as figures displaying the IAT trajectories, and as tables describing the characteristics of subjects classified into groups and giving the HMTD model parameters.

6.3 Results

We provide here the results of the various clustering performed using the HMTD and GMM approaches, and we compare the resulting classifications. Notice however that given the iterative nature of the optimization algorithms, it is never possible to be sure that the final models are the best possible ones. Therefore, results should never be overinterpreted.

6.3.1 HMTD clustering

Without covariates, the best model identified by the BIC is a four-component model (model 1). Figure 6.1 shows the IAT trajectories in each group. We clearly differentiate a group with average volatility and IAT level (group 1), a group with relatively low scores and variability (group 2), a group with very low variability and a low and constantly diminishing IAT score (group 3), and a group with more complex trajectories and hence variability (group 4).

The ICL criteria has also been explored for the choice of number of clusters (Table 6.1). For this example (without including any visible or latent-level covariates), the ICL suggest the choice of three clusters, even though the difference with the two cluster solution is small. The likelihood increase from two to three clusters is then barely

Table 6.1: Log-likelihood, entropy, and ICL approximation for each number of clusters for the IAT dataset

number of clusters	2	3	4	5	6
logL	-2610.211	-2597.25	-2587.52	-2571.48	-2565.91
Entropy	-30.83412	-50.74	-95.89	-125.71	-149.29
ICL	5260.913	5259.75	5290.25	5292.81	5310.09

enough to overcome the penalty of the increasing entropy. Note that this solution however, defines two clusters of only 7 and 18 observations.

In conclusion if the choice was based on the ICL criterion, probably a different and more parsimonious clustering model would have been chosen in this chapter. However, the solution suggested by ICL contains clusters of very unequal size: one with almost all observation and two with very few, which was less interesting in terms of interpretation. Therefore we stay with the four cluster solution.

When we include the covariates in model 1 (Figure 6.2) and relabel the four groups of the solution in order to match the groups of model 1, we obtain a similar four-group structure (model 2). As a comparison of the two figures might show, the most important difference is with the first two groups: group 2 of model 2 lost its higher-valued trajectories and focused more on a low IAT-level and stable trajectories. This change will be explored in more details later.

Table 6.2 provides the parameter estimation for both models. In addition to the point estimates, we also provide the 95% bootstrap confidence intervals.

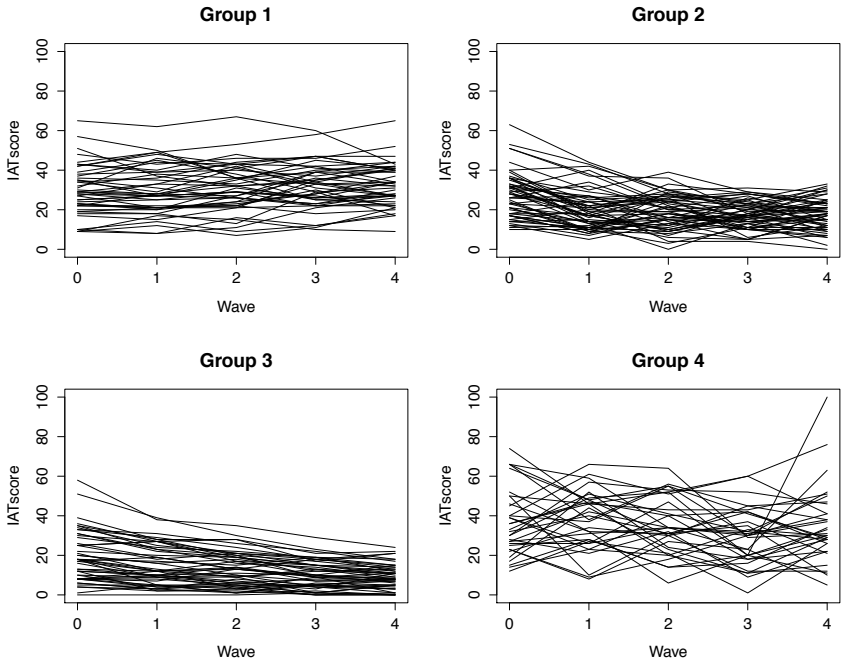


Figure 6.1: IAT trajectories associated with each group in the four-group HMTD solution without covariates (model 1).

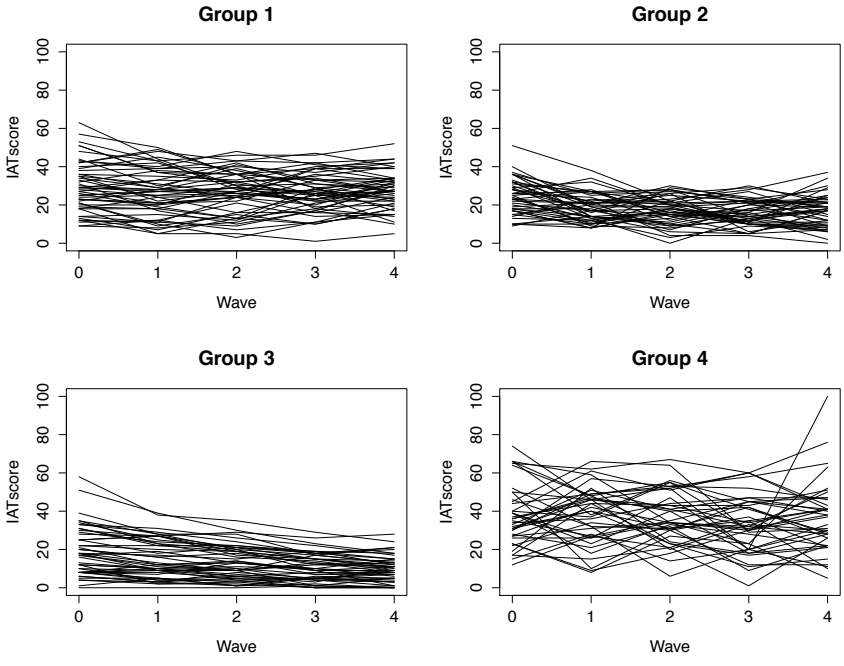


Figure 6.2: IAT trajectories associated with each group in the four-group HMTD solution with five covariates (model 2).

Table 6.2: Estimated coefficients for the two HMTD models. For each parameter, we also provide the minimum and maximum values, and the 95% bootstrap confidence intervals obtained from 1000 bootstrap samples. Significant parameters at the 95% level are indicated with an asterisk.

Model 1		θ_0	ϕ_0	ϕ_1					
group 1	coefficient	41.098*	2.798*	0.891*					
n=46	min, max	23.898, 108.133	1.881, 30.144	-0.070, 0.964					
	95% interval	(26.312; 38.317)	(3.417; 8.162)	(0.752; 0.919)					
group 2	coefficient	49.712*	7.228*	0.572*					
n=56	min, max	0.001 89.683	5.721 22.126	-0.036 0.615					
	95% interval	(36.871; 54.309)	(6.507; 10.735)	(0.375; 0.563)					
group 3	coefficient	11.570*	1.072*	0.753*					
n=50	min, max	7.535, 19.014	-0.102, 2.518	0.685, 0.814					
	95% interval	(8.715; 13.091)	(0.456; 1.737)	(0.715; 0.787)					
group 4	coefficient	186.582*	15.511*	0.514*					
n=33	min, max	0.000, 384.934	6.951, 39.902	-0.186, 0.743					
	95% interval	(0.002; 291.178)	(14.906; 28.189)	(0.226; 0.560)					
Model 2		θ_0	ϕ_0	ϕ_1	**WHO-5**	**BMI**	**Gender** (male)	**Age**	**Education** (extended req.)
group 1	coefficient	38.587*	-1.148	0.656*	-2.156*	1.331*	-2.663*	0.941	-0.468
n=52	min, max	24.369 39.084	-27.000 47.072	0.371 0.714	-3.503 -1.014	-0.079 3.269	-7.317 0.084	-2.232 2.883	-3.271 4.786
	95% interval	(27.009 37.035)	(-9.082 31.695)	(0.507 0.672)	(-2.962 -1.405)	(0.596 2.535)	(-5.433 -1.672)	(-1.116 1.675)	(-2.107 1.928)
group 2	coefficient	43.371*	-1.444	0.539*	-0.288	-1.025	0.549	0.638	-2.705
n=45	min, max	0.530 86.137	-15.527 55.269	0.100 0.601	-2.788 1.896	-3.692 0.427	-5.127 4.760	-3.290 1.838	-8.183 1.675
	95% interval	(34.931 52.838)	(-1.607 18.077)	(0.217 0.540)	(-1.266 0.723)	(-1.950 0.073)	(-0.319 2.851)	(-0.545 0.888)	(-4.887 0.237)
group 3	coefficient	9.162*	0.256	0.723*	-0.707*	-1.424*	-0.757	0.142	1.060*
n=48	min, max	5.987 11.771	-20.810 29.836	0.627 0.780	-1.955 0.161	-3.110 -0.036	-4.049 2.320	-1.995 1.870	-0.547 5.000
	95% interval	(7.222 10.013)	(-10.036 16.612)	(0.666 0.758)	(-1.487 -0.269)	(-2.069 -0.900)	(-2.151 0.458)	(-1.030 0.959)	(0.350 2.288)
group 4	coefficient	153.507*	22.274	0.307*	-2.771*	-4.014*	11.303*	-1.135	2.063
n=40	min, max	99.659 242.775	-48.500 96.927	-0.099 0.518	-6.676 2.444	-9.000 -0.593	4.427 20.000	-6.000 4.656	-6.000 8.000
	95% interval	(108.832 198.571)	(-48.500 54.129)	(0.011 0.378)	(-5.327 -0.650)	(-7.542 -2.135)	(7.541 18.251)	(-3.493 4.095)	(-3.450 7.002)

As regards the first model without covariates, the θ_0 parameters giving the variance of each component of the model confirm the first impression given by Figure 6.1: Group 4 is characterized by a much higher variability than the three other groups, and group 3 has the lowest variance, indicating less variation among the successive observations of a single individual. Parameters ϕ_0 corresponding to the constant in the modeling of the mean of each component also take expected values, with higher values associated with groups showing higher average IAT level. Finally, the autoregressive parameter ϕ_1 takes a value closer to one for the groups with trajectories showing smoother evolutions from one wave to the next, that is groups 1 and 3. All parameters of this first model are significant at the 95% level, as demonstrated by the confidence intervals.

As regards model 2, even if the first three parameters (θ_0, ϕ_0, and ϕ_1) take values different from those of model 1, θ_0 and ϕ_1 take values in the same range as of model 1. On the other hand, important differences are found for the constant parameter ϕ_0, and this parameter is no more significant in any group. Note that θ_0 and ϕ_1 tend to take smaller values in model 2. This can be interpreted as the first proof of interest of the covariates included in model 2: the groups are now more homogeneous (lower intra-group variance) and the explanation of a specific trajectory relies less on the immediately preceding observation. As regards the covariates, Age is never significant and could be eventually removed from the model. This could be due to the lack of a real age difference between participants (from 13 to 15 years old at baseline). Actually this non-significance is expected because generally in cohort data this difference is chosen to be small. However, the four other covariates remain significant for at least one of the groups.

When we consider each component of model 2 separately, the changes occurring in the trajectories associated with the first component are found related to the emotional well-being of the concerned adolescents: a higher emotional well-being such as measured by the WHO-5 index is significantly associated with a lower IAT-level. Males tend to have a lower IAT level than females, and a higher BMI is associated with higher IAT level. In group 3, a higher WHO-5 or BMI is associated with reduced IAT level, but being in the extended requirement school track is associated with a higher IAT level. Finally, in group 4, a higher WHO-5 or BMI is associated with reduced IAT level, and males tend to show a much higher IAT level than females.

Table 6.3 provides the main characteristics of the subjects classified into each group. For time-dependent variables, we considered the average value of each individual. A comparison is performed for each variable separately to test whether the groups are significantly different with regard to the variable. Considering only the two HMTD

models, we observe that in addition to the expected differences in IAT level, the only other variable with significantly different values across groups is the WHO-5 measure of emotional well-being. For both models, we observe two groups (2 and 3) with lower average IAT scores. The same two groups also display higher emotional well-being, as compared to the other groups, confirming previous results Surís, Akre, Piguet, Ambresin, Zimmermann and Berchtold [150]. No differences are observed for the other covariates, even if Gender comes close to significance in model 1. Even if not significant at the 95% level, probably because of the reduced sample size, we find a gender separation at the sample level; groups 2 and 4 contain a higher proportion of boys compared to the other two groups. The education track also shows a difference at the sample level: the first two groups contain more individuals following the highest education track as compared to groups 3 and 4. On the other hand, no notable difference is observed between the groups for Age and BMI, even if BMI, used as a covariate in model 2, is statistically significant in the modeling of the mean of each component.

6.3.2 Usefulness of the covariates

From the results of the previous section, we find that the inclusion of covariates in the first classification obtained with the HMTD model helped us better differentiate the four groups, but without entirely changing their interpretation. We would like to better understand the changes in trajectory classification that occurred between these two models. Table 6.4 indicates how many subjects changed groups between the initial model without covariates and model 2 with covariates. As noted earlier, most of these changes occurred between groups 1 and 2. In particular, 19 second-group subjects of model 1 were transferred to the first group in model 2, and the steady low Internet addiction profile of the second group became even more pronounced, with the higher Internet addiction subjects joining the first group. However, since some trajectories simultaneously left group 1 for the three other groups, the average IAT level of group 1 also decreased. Overall, the inclusion of covariates appears beneficial for the differentiation of trajectory features among groups.

The 19 individuals who switched from group 2 to group 1 represent the main difference between the two models, with all the other changes concerning at the most seven subjects. Thus, it is interesting to explore how these individuals differed from those who remained in the first or second group in both classifications. Table 6.5 summarizes our findings using t-tests and χ^2-tests to compare the different variables. The average IAT scores are quite different between the three considered sub-groups, and, as expected,

Table 6.3: Characteristics of subjects classified into groups for different clustering. The p-value gives the result of the test comparing the different groups for each variable. The number of sequences classified into each group is provided in brackets after the group number.

	IAT	**WHO-5**	**BMI**	**Gender**	**Age at T0**	**Educ. at T0**
	mean (sd)	mean (sd)	mean (sd)	% male	mean (sd)	% extended req.
HMTD model 1						
group 1 (46)	30.94 (11.7)	63.43 (15.6)	19.97 (2.35)	24	14.13 (0.499)	80.5
group 2 (56)	20.29 (9.78)	71.01 (15.6)	20.02 (3.30)	45	14.05 (0.585)	67.9
group 3 (50)	13.31 (9.88)	72.28 (13.6)	20.45 (2.57)	24	14.14 (0.670)	64.0
group 4 (33)	34.69 (16.1)	63.49 (16.8)	20.06 (3.03)	39	14.27 (0.452)	60.6
p	<0.001	<0.001	0.764	0.055	0.381	0.214
HMTD model 2						
group 1 (52)	27.43 (11.32)	67.35 (16.57)	20.12 (2.40)	31	14.19 (0.60)	71.2
group 2 (45)	18.57 (8.41)	70.85 (15.19)	19.96 (3.53)	40	14.02 (0.45)	73.3
group 3 (48)	13.62 (9.97)	70.64 (14.00)	20.46 (2.54)	21	14.10 (0.69)	64.6
group 4 (40)	36.36 (15.62)	63.06 (16.37)	19.96 (2.86)	43	14.22 (0.48)	65.0
p	<0.001	0.015	0.741	0.113	0.331	0.746
GMM 2						
group 1 (169)	22.08 (13.2)	68.79 (15.84)	20.20 (2.90)	32	14.15 (0.57)	0.68
group 2 (16)	39.90 (14.8)	61.13 (13.93)	19.40 (2.12)	43	14.00 (0.52)	0.75
p	<0.001	0.022	0.210	0.496	0.322	0.771
GMM 4						
group 1 (76)	13.35 (8.97)	73.32 (14.15)	20.69 (2.75)	32	14.09 (0.61)	0.63
group 2 (31)	38.98 (11.2)	58.48 (16.18)	20.15 (2.40)	29	14.16 (0.52)	0.74
group 3 (75)	26.46 (10.2)	67.09 (15.38)	19.62 (2.98)	33	14.17 (0.55)	0.73
group 4 (3)	54.06 (18,3)	62.40 (9.657)	18.78 (3.30)	100	14 (0)	2/3
p	<0.001	<0.001	0.043	0.094	0.802	0.593
GMM 4 cov						
group 1 (98)	18.79 (10.6)	69.64 (14.6)	20.06 (2.90)	29	13.91 (0.320)	77.9
group 2 (44)	18.58 (10.5)	68.88 (17.8)	20.85 (2.95)	24	15.16 (0.554)	44.0
group 3 (28)	39.38 (12.9)	64.95 (18.2)	20.14 (2.73)	48	14.24 (0.435)	48.3
group 4 (15)	41.98 (14.8)	60.00 (13.7)	19.36 (2.05)	54	14.00 (0.408)	76.9
p	<0.001	0.032	0.321	0.058	<0.001	<0.001

Table 6.4: Number of IAT trajectories associated with each group in HMTD models 1 (without covariates, rows) and 2 (including covariates, columns).

	Model 2			
Model 1	group 1	group 2	group 3	group 4
group 1	31	6	2	7
group 2	19	34	1	2
group 3	2	3	45	0
group 4	0	2	0	31

the "moving" sub-group shows an Internet dependence level between the two "stable" sub-groups. Thus, the moving individuals were among the most Internet-problematic members of the full second group of model 1, and even if the average IAT score is not the only indicator of group affiliation, a visualization of the trajectories would confirm the ambiguous nature of these individuals. The moving subgroup is also significantly different from the group of individual staying in group 1 as regards the WHO-5 index of emotional well-being and the gender ratio, with a higher emotional well-being and higher proportion of males among the moving subgroup. No other significant differences are observed.

Table 6.5: The characteristics of 19 subjects moving from group 2 to group 1 (group 2→1) as compared to subjects staying in the same group (either 1 or 2) in both HMTD classifications. The means (numerical variables) or proportions (categorical variables) are provided, and differences with the subjects remaining in the same group (either 1 or 2) are assessed using t-tests and χ^2-tests with continuity correction. ns: non-significant, * p <0.05, ** $p < 0.01$, *** $p < 0.001$.

	IAT	**WHO-5**	**BMI**
group 2→1	22.76	72.93	19.69
vs group 1	31.26 **	62.77 **	20.41 ns
vs group 2	18.72 ns	71.42 ns	20.30 ns
	Sex (% male)	**Age**	**Education**
group 2→1	57.9	14.26	52.6
vs group 1	9.7 ***	14.16 ns	80.6 ns
vs group 2	38.2 ns	13.97 ns	73.5 ns

6.3.3 GMM clustering

Without covariates, the best GMM solution in terms of BIC is a two-group solution (Figure 6.3), but given the high difference in number of trajectories associated to each group (169 vs 16), this solution in not really interpretable and hence less useful than the four-group solution given by the HMTD approach. Therefore, we also estimated a four-group GMM (Figure 6.4).

In the two-group solution, a large majority of trajectories are associated with group 1, and only 16 sequences are associated with group 2. The average IAT level is higher in group 2, but both groups exhibit an important variability, as indicated in Table 6.3. Moreover, in terms of interpretation, one can only say that IAT sequences with a clear increasing trend are separated from the other sequences. In the four-group solution, even if the number of groups is the same as in the HMTD models, there is no a priori correspondence between the HMTD and GMM groups. In the four-group GMM solution, the number of subjects per group shows much more variability than that observed with the HMTD group, with the majority of individuals classified in groups 1 or 3, and only three subjects in group 4.

Finally, as with the HMTD approach, we enhanced the four-group solution by adding covariates. Four of the five covariates used in the HMTD approach appeared useful in the GMM solution as well. Figure 6.5 displays the resulting groups obtained after adding Gender and Education as predictors for group membership (multinomial regression on c_i), and WHO5 and BMI as fixed effect. On the other hand, Age was not included in the final model because the estimation process would then lead to a one-group solution. Another important issue with the GMM approach is the results' sensitivity to the order in which the covariates are included in the model. Various covariate combinations were tested before we chose the above-mentioned combination as the best one in terms of clustering results. For instance, *classmb = gender + education track* does not give the same results as *classmb = education track + gender*. This surprising result may be due to a bug in the *lcmm* R package, but in our opinion the reason could rather be the optimization procedure. It is well known that EM-type algorithms converge to the nearest local optimum, and that this optimum is not always the global one. Therefore, the solution depends on the initial values of the parameters, especially when the solution space is complex, which is the case here.

As Figure 6.5 shows, the number of trajectories associated with each group is quite variable, with the large majority assigned to group 1. The first two groups are characterized by low variability and an overall low IAT level. The trajectories in these two

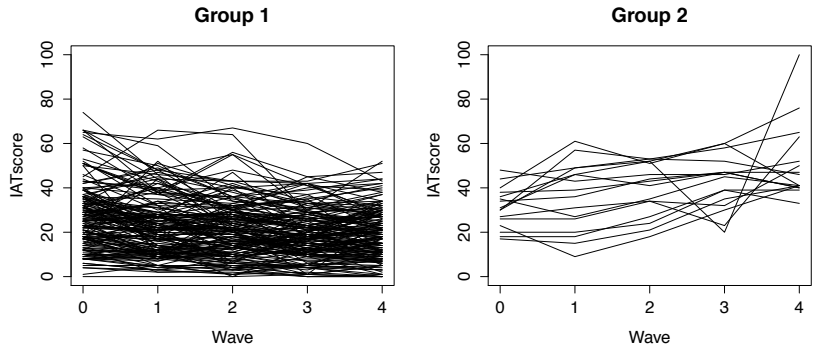


Figure 6.3: IAT sequences associated with each group in the two-group GMM solution without covariates.

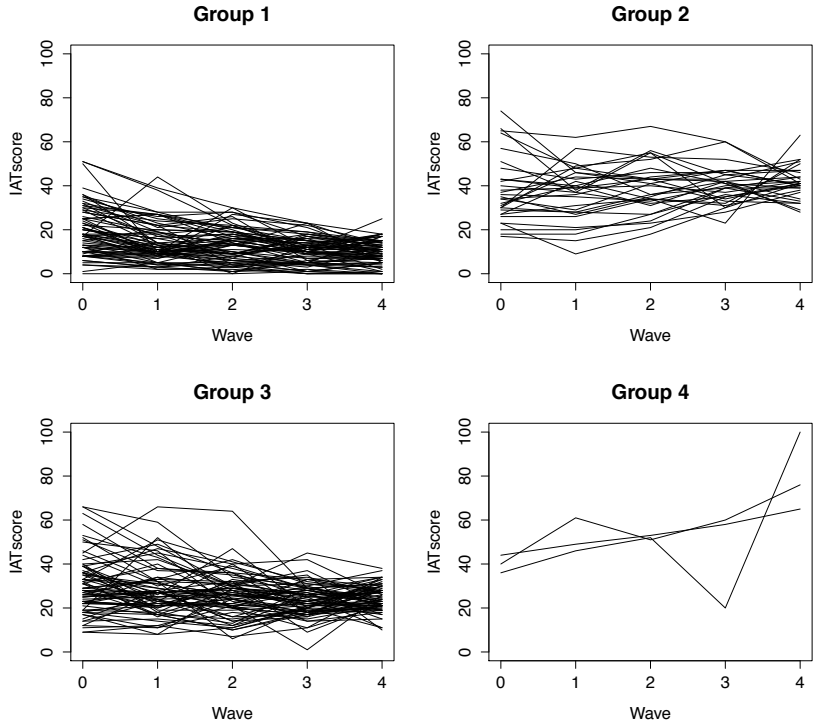


Figure 6.4: IAT sequences associated with each group in the four-group GMM solution without covariates.

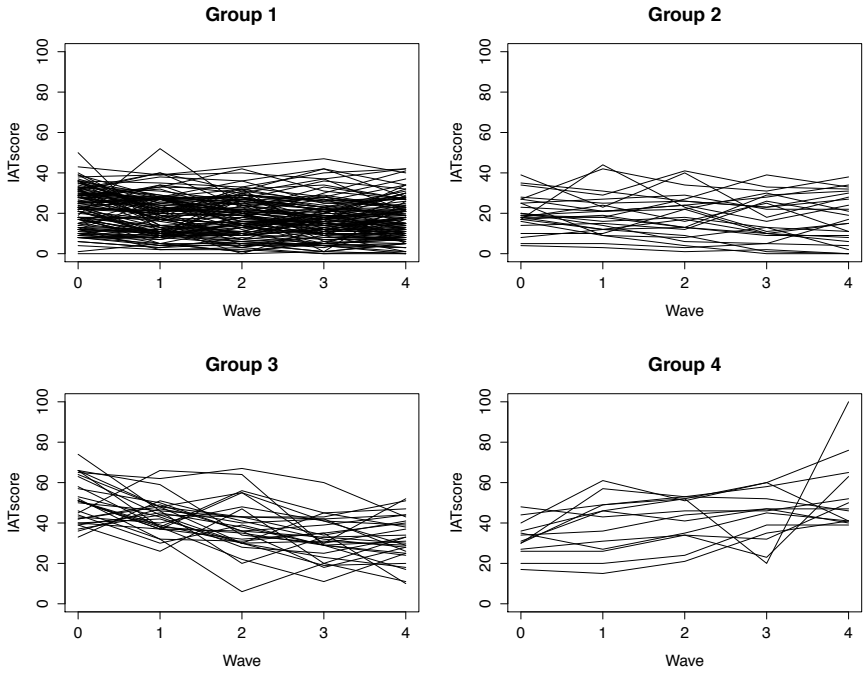


Figure 6.5: IAT sequences associated with each group in the four-group GMM solution with four covariates.

groups seem very similar, but since this four-group solution might be suboptimal and is computed only for the purpose of comparison with HMTD models 1 and 2, a three-group solution could merge these two groups into one group. The last two groups have a higher average IAT level, both exhibiting a general linear trend over time, decreasing in group 3 and increasing in group 4.

Table 6.3 gives the characteristics of individuals classified in each group of the GMM models and compares the groups for each variable. Note that given the large differences in group size, the test results for the GMM models should be interpreted with caution. As observed earlier in the HMTD case, significant differences exist between groups for both the IAT and WHO-5 variables. A significant difference exists also for BMI in the four-group GMM model without covariates. More interestingly, the Age and Education track at baseline also show significantly different values across groups, with one of the variables (Education track) being included in the model as covariable, but not the other. This difference between the HMTD and GMM clustering points to the fact that the solutions provided by both approaches are not identical or interchangeable, and that the two models used information in a different manner to provide usable data sequence clusterings.

6.4 Comparison of HMTD and GMM

When used for clustering purposes, the HMTD and GMM models share some characteristics: They both represent a kind of mixture model, they can include covariates of any type at the visible level, and they can also include covariates at the latent level and use them to estimate the initial probability of each cluster. However, HMTD and GMM also have several differences. First of all, since GMM is a mixture of mixed models, it is able to accept both fixed and random effects. Another difference is the possibility of HMTD to include an autoregressive specification for the variance and thus to allow for the clustering of longitudinal sequences whose variance evolves in time. For instance, sequences becoming more instable over time can more easily be grouped together. However, to exploit this feature, it is necessary to work with long data sequences, what was not the case here with the IAT example.

Another feature of HMTD that is worth stressing is the possibility of using it to perform different kind of clustering Bolano & Berchtold [24]. The transition between components is driven by the hidden transition matrix A. In this paper, A was constrained to be a diagonal identity matrix, implying that each sequence was assigned to one and only one group, and all sequences assigned to the same group were described

by the same visible model. However, there are several alternatives. For instance, different latent states may be required to alternate over time in order to find the optimal modeling of a given sequence. If A is constrained to have the following structure:

$$A = \begin{pmatrix} a_1 & 1-a_1 & 0 & 0 \\ a_2 & 1-a_2 & 0 & 0 \\ 0 & 0 & a_3 & 1-a_3 \\ 0 & 0 & a_4 & 1-a_4 \end{pmatrix}$$

where a_1, a_2, a_3 and a_4 are transition probabilities, then one performs at the same time a modeling and a clustering of the data sequences. The first two states are used to model the first cluster, and states 3 and 4 are used to model the second cluster. In other words, data sequences are clustered into two groups, but inside each group there are two different visible models allowing for a better representation of these sequences when their behavior evolves over time.

Another specification of A would allow some sequences to remain always in the same cluster, whereas other ones could transit at some point in time from the first to the second cluster:

$$A = \begin{pmatrix} a_1 & 1-a_1 & 0 & 0 \\ a_{21} & a_{22} & 1-a_{21}-a_{22} & 0 \\ 0 & 0 & a_3 & 1-a_3 \\ 0 & 0 & a_4 & 1-a_4 \end{pmatrix}$$

6.5 Conclusion

Hidden Markovian models are known to be valuable tools to analyze the dynamics in longitudinal continuous data and in life course data (eg. Helske S, Helske J and Eerola [63]). The present study demonstrates that the sequences of continuous longitudinal data can also be classified into as many groups as required, and that the HMTD model can be used as a valid alternative to GMM. The inclusion of covariates has beneficial effects on clustering, because the resulting groups have lower intra-variability compared to the solution without covariates.

In a comparative study involving the use of GMM for clustering, our first finding is that the HMTD approach is a good alternative to GMM, because in terms of interpretability its results are at least as interesting as the results given by GMM. However, on the basis of just one practical example, we obviously cannot conclude that one approach is better than the other; moreover, this is not the purpose of this study. What we

can conclude is that the HMTD approach is not only theoretically, but also practically useful to classify sequences of continuous data in mutually excluding groups.

In the literature, excessive Internet use has been found to be highly related to several somatic conditions, sleep disturbance in particular. However, in this paper, our main objective is not to explain IAT trajectories, but to find ways to classify such trajectories into meaningful groups. Moreover, there is still an ongoing debate on the direction of the relationship between Internet use and sleep disturbance, not to speak of causality. Therefore, we chose not to consider sleep disturbance in this analysis, but to concentrate on other covariates that are more neutral to IAT scores. Nevertheless, even with this restriction, the results obtained with the HMTD model are highly significant and allow for a sound interpretation. The four resulting groups differ in terms of average value and variability. The relationship observed between IAT and the emotional well-being of subjects suggests that both concepts are linked and that a higher risk of Internet addiction is related to poorer emotional well-being. Gender is also a discriminating factor between groups, with a lower proportion of males in the first and third groups, but, given the small sample size, the differences are not significant at the population level.

The main strength of this study is the demonstration of the usefulness of the HMTD approach as a valuable alternative to the GMM approach for clustering continuous data sequences. Researchers would be advised to consider both approaches to take full advantage of the information in their data. However, some weaknesses of this study are to be mentioned. At the theoretical level, we include covariates in the HMTD model only at the visible level, but it is also possible to include them at the latent level as well in order to enhance the prior probabilities of each cluster. As regards the application of the model to IAT trajectories, we used a rather small convenience sample; this is not representative of the population of adolescents living in the canton of Vaud. More analyses need to be conducted with larger databases to define a real typology of IAT trajectories.

Measurement invariance of the IAT score One may be interested in the measurement invariance as an important indicator of the quality of a construct. It represents the possibility of a score to measure the same concepts trough different groups. Taking an example with the IAT score, we may be interested in to what extent all items in the score are equally relevant or have similar impact on the addiction to internet, when comparing different groups or different measurement times and whether the obtained scores are comparable among the periods.

Invariance is present if the following equality is respected:

$$f(Y \mid \boldsymbol{s}, \mathbf{z}) = f(Y \mid \boldsymbol{s})$$

meaning that the group membership indicators z do not influence the observed IAT scores Y given the scores s.

This concept is often tested using factor analysis. In a matrix form the factor analysis equation at some measure time T is:

$$X^T = \alpha^T + \Lambda^T F^T + \epsilon^T$$

where, in our case X^T has dimensions $[20 \times 161]$ and denotes a matrix of the 20 items composing the tested measure as columns, α is the vector of intercepts for each item, Λ^T $[20 \times f]$ and F^T $[f \times 161]$ are matrices of factor loadings and factors, ϵ^T are the error terms and f is the number of factors.

Note that $E(F) = 0$ and $Cov(F) = I$ which indicates independence between the factors.

The covariance structure is denoted:

$$\Sigma^T = Cov(X^T) = Cov(\alpha^T + \Lambda^T F^T + \epsilon^T) = \Lambda^T I \Lambda^{T'} + cov(\epsilon^T) = \Lambda^T \Lambda^{T'} + \Theta^T$$

Putnick and Bornstein [115] (2016) summarize the measurement invariance in four most important steps. The first one is testing the equivalence of the model form, which tests if the factor loadings have the same structure across the compared groups. In this example we are interested if each item "loads" on the same factor (component) in every group. If one of the items is related to another or to more than one factor, the construct is not invariant. Testing the "metric" invariance consist in assessing the difference in the factor loadings, i.e. test if the items influence the constructs similarly. The two final steps consist in measuring the difference of the item intercepts and those of the item's residuals or unique variances or means. At each step the model is estimated with constrained parameters (factor loadings for instance) and compared to the previous unconstrained model.

In the majority of the papers the measurement invariance is tested between different groups or populations. The Internet Addiction Test has also been subject of various measurement invariance studies. For instance Jelenchick, Becker and Moreno [70] found two components reflecting dependent and excessive internet use for US college students. Lai et al. [83] compared IAT invariance between Hong Kong, Japanese and Malaysian adolescents and showed that the score is stable and reliable. However, in most papers

IAT was observed at a single point in time, whereas in our case we have longitudinal data and therefore one might be interested in whether IAT remains invariant trough the repeated measurements.

Ideally we would also like to explore the measurement invariance between the different clusters in order to be able to compare them. However, the limited size of our sample (161 individuals in total) does not allow us to obtain reliable results from Principal Component or Confirmatory Factor Analysis. Instead we can perform two analyses.

At first, in order to better understand the underlying structure of the IAT accros time, we can estimate the true number of principal components within the data by performing a bootstrap sampling. For each sample the number of components with eigenvalues>1 is observed. Because we dispose with longitudinal data, we proceed in two different ways: initially only the first wave is considered (161 observations) and then all observations from each wave are included together (925 observations from 161 individuals). The problem with the latter approach is the lack of independence between observations because every individual is represented 5 times, whereas in the former case only t=1 is included.

Table 6.6: Results of PCA from 5000 bootstrap iterations: average eigenvalues for the first 9 components and distribution of the number of components

Panel A: Average eigenvalues for the first components

Mean of first 9 eigenvalues	1	2	3	4	5	6	7	8	9
Only first wave (161 obs.)	6.47	2.00	1.44	1.23	1.09	0.97	0.88	0.80	0.72
All 5 waves (925 obs.)	7.12	1.98	1.19	1.03	0.90	0.83	0.77	0.72	0.67

Panel B: Distribution of the number of eigenvalues higher than 1

Number of eigenvalues >1	3	4	5	6	7	8	total	average
Only first wave (161 obs.)	3	394	2912	1615	75	1	5000	5.27
All 5 waves (925 obs.)	1292	3615	93				5000	3.76

The results from 5000 bootstrap samples are presented in Table 6.6: the average eigenvalues of the first PCA components are computed in Panel A and the distribution of the number of eigenvalues >1 is displayed in Panel B. The latter was greater than one

in all iterations. In the first wave (t_1) samples, their average is significantly smaller than in the complete-data samples (3.76 vs 5.27). These results hint at lack of invariance of the IAT score over time for our data.

A second analysis was performed using a factor analysis procedure implemented in the *semTools* package in R to test all relevant dimensions of measurement invariance used in the literature (see the procedure described by Vandenberg and Lance [160]). A longitudinal measurement invariance across the 5 waves was tested. In this test we explore the invariance across time periods (instead of clusters). Each of the four tested models includes an additional constraint and is compared to the less restrictive model. If significant decrease of its quality (fit) is detected, the constraint is not respected and therefore the corresponding type of invariance is not respected.

From the *semTools* package we have the following hypotheses:

Model 1: "configural" invariance. The same factor structure is imposed on all measurements.

Model 2: "weak" (metric) invariance. The factor loadings are constrained to be equal across measurements. ($\Lambda_{t_1} = \Lambda_{t_2} = \Lambda_{t_3} = \Lambda_{t_4}$)

Model 3: "strong" (scalar) invariance. The factor loadings and intercepts are constrained to be equal across measurements ($\alpha_{t_1} = \alpha_{t_2} = \alpha_{t_3} = \alpha_{t_4}$)

Model 4: The factor loadings, intercepts and factor means are constrained to be equal across measurements.

From Table 6.7 [2], we can conclude that fixing the factor loadings across units does not seem to deteriorate the model fit considerably. However, the means and the intercepts appear to be different and therefore not all dimensions of measurement invariance are respected.

As expected the package does not allow us to test the measurement invariance also between the clusters that we obtained previously and indicates an error message because of the small group size.

One possible reason for the non-invariance is that the IAT questionnaire may be too long especially for young adolescents. Another hypothesis is that the 20 IAT-composing questions could be found rather complex and repetitive, which may sometimes lead to random answers from the participants. At the end, the presence of multiple underlying factors within the items of the IAT score, shown by the first analysis, adds complexity to the clustering of the trajectories and affects the obtained results.

Overall, in spite of some shortcomings, the HMTD model can be considered as a

[2] RMSEA - root mean squared error of approximation; CFI - comparative fit index

Table 6.7: Results of longitudinal measurement invariance tests

Differences between models

	Df	AIC	BIC	Chisq	Chisq diff	Df diff	Pr(>Chisq)
configural	4820	50585	51648	10687			
loadings	4896	50523	51341	10776	89.72	76	0.135
thresholds (intercepts)	4972	50581	51154	10986	210.02	76	1.7e-14 ***
means	4976	50618	51178	11032	45.22	4	3.5e-09 ***

Measures of fit for every model

Fit measures:	cfi	rmsea	cfi.delta	rmsea.delta
configural	0.574	0.081	NA	NA
loadings	0.573	0.081	0.001	0.001
thresholds (intercepts)	0.563	0.081	0.010	0.000
means	0.560	0.081	0.003	0.000

complete framework for the analysis of continuous data sequences. It is an explanatory tool as well as a clustering tool, and by adding covariates, constraints on the transition matrix, and autoregressive modeling of the mean and variance of each component, the model goes well beyond the traditional Markovian models such as homogeneous Markov chains or hidden Markov models.

Chapter 7

Conclusion and further researches

The main contributions of this thesis are related to the estimation procedure for the HMTD model, the inclusion of latent-level covariates of any type that reestimate directly the initial probabilities, the general procedure for clustering together with the proposed bootstrap procedures for the parameter inference, and finally the applications to the clustering of sequences.

Several points need to be mentioned to resume the utility of this thesis. A discussion on the versatility of the model is also important to summarise its possible future developments.

7.1 Latent and visible covariates

The inclusion of latent and visible covariates simultaneously appear to be a useful extension that enhances the results of the model, as seen in the examples. Particularly for clustering purposes with a diagonal matrix A, the influence of the latent covariates on the initial probabilities π appears to provide interpretable results.

Covariates on both levels have already been included in the model (see Berchtold [15]). In this thesis however, the latent covariates have a different impact on the hidden level. Instead of influencing the latent transition matrix A, they rather help us to estimate the initial probability matrix π. Two main reasons argument this modification. First, when clustering the transition matrix is fixed to its diagonal form and therefore it makes no sense re-estimate it. This makes the latent level covariates inapplicable when clustering. Secondly, this modification simplifies the model estimation. Furthermore, it allows us to use a logistic regression which accepts all types of covariates: categorical, discrete or continuous. The longitudinal covariates however can only be applied to the

visible level.

7.2 Estimation

While the hidden mixture transition distribution (HMTD) model is a powerful framework for the description, analysis, and classification of longitudinal sequences of continuous data, it is notoriously difficult to estimate. One of the reasons is the complexity of its solution space, but the biggest issue is the difficulty to derive the Likelihood function when the standard deviation of the components is not constant. Even though by fixing the model variance, one could obtain exact solutions, our aim is to provide an estimation procedure that is as general as possible, and adaptable to all specifications of the model, no matter if we attempt to model sequences (free form of the transition matrix A), cluster them (diagonal transition matrix indicating mixture model), or combine both (constrained A).

In chapter 3 we briefly described and discussed the main tools that we used in the estimation of HMTD, as well as some alternatives.

Then we explored how a new heuristic, specifically developed for the HMTD, performs compared to different popular optimization algorithms within the M-step of a GEM algorithm (Taushanov and Berchtold [152]). This specific heuristic can be classified as a hill-climbing method, and different variants are proposed, including a jittering procedure to escape local maxima and measures to speed up the convergence.

Different popular approaches were used for comparison, including PSO, SA, GA, NM, L-BFGS-B, and DE. The same HMTD model was optimized on different datasets and the results were compared in terms of fit to the data and estimated parameters. Even if the complexity of the problem implies that no one algorithm can be considered as an overall best, our heuristic performed well in all situations, leading to useful solutions in terms of both fit and interpretability.

The principles presented in this chapter can be easily applied to the optimization of other statistical models with complex solution spaces.

The estimation procedure using a GEM algorithm is intended to be as general as possible and to apply to all possible specifications of the HMTD model. The choice of the fastest maximization algorithm in the M-step is a priority. However it is very important to obtain only plausible solutions that respect some floating constraints. The latter are not strict constraints, but rather artificial limits of the solution space that could be broadened when a high likelihood region is found in proximity of those constraints, that is when the optimization algorithm approaches the limits. The ad-

vantages are the introduction of a smaller solution space that is easier to explore, but also that this initial space does not exclude the favorable regions that are left ouside of it. Such floating constraints may be especially useful when an expert has a rough intuition about the solution space for a problem, but we do not want to exclude the possibility taht it may prove wrong. Still this intuition may be a good starting point for the initial solution in a very broad solution space.

7.3 Clustering and inference procedure

When a mixture model is used, either to explain a dataset or to perform a clustering of the data, one often has to choose between Bayesian or frequentist model estimation. We briefly reviewed the major issues related to both estimation approaches, and mention some of the existing solutions. In the frequentist case, the first step consists generally in finding an optimal solution using an EM-type algorithm as optimization procedure. The typical approach is to evaluate different model solutions, and to keep the one that provides the best fit in terms of BIC. Then, in order to assess the significance of the parameters of this optimal solution, a standard bootstrap procedure is applied using the full original sample and a confidence interval is computed. The optimal model specification with multiple components (clusters) is computed at each iteration. That leads to solutions with different degrees of similarity with the optimal solution and the well-known label-switching problem may occur.

Two alternative procedures were proposed in this thesis. They rely (to a different extend) on the initial best solution, and therefore we discuss that the BIC criterion alone is not sufficient, and this solution must be also validated in several ways before it is accepted. We discussed different criteria and methods to approve a solution before relying on it for the two proposed procedures. The first procedure to estimate the parameters' confidence intervals consists in applying separate bootstraps on each sub-sample defined by the partition of the solution that was approved as optimal beforehand. In this case, a model with one single component is estimated on each bootstrap iteration and for each cluster separately. This method also provides a confidence interval for each parameter and most importantly, it avoids the label-switching problem. Another alternative includes full-sample bootstrap, but with re-sampling proportional to each class of the validated solution, i.e. stratified sampling where the strata are the clusters of the accepted solution. Then the full model is applied to each sub-sample. The pros and cons of each approach have been described with real-world data examples. The importance of the initial solution in these procedures requires a discussion on clustering

choice and validation.

As discussed, validation can have several different dimensions. One can look at the optimality of the solutions in terms of model fitting, using information criteria for instance (AIC, BIC). This indicates us the model parameters that adapt best the model assumptions to the data. Another possibility is to question the stability of a candidate solution and compare it to another concurrent solution. Different external or internal indices and various methods have been listed to do this, and some remarks were made. Internal indices are useful to check the coherence of the (mainly continuous) transversal data to the clustering solution. However, in longitudinal data, internal indices are difficult to apply because of the time variation of the data. Even though one does not dispose with a true clustering, we showed that one can compare modified solutions obtained with bootstrap resampling with the original candidate solution to measure its stability.

Despite all these validation possibilities, the most important property of an acceptable solution is its interpretability. The solution must be interpretable by the researcher using his knowledge of the data. However, it is important that this knowledge does not introduce the bias of the researcher's expectations on the results. Therefore the other validation methods can be useful.

An illustration of the clustering and parameter inference is provided using the real-life example of somatic troubles data. A final solution of five clusters was presented and interpreted.

7.4 HMTD as a versatile clustering tool

In summary, the HMTD model is proven to be a very useful tool not only for modelling continuous sequences, but also for clustering. As explained above, among the main strengths of this model is the possibility of flexible clustering, that allows the subjects to be assigned to clusters in a less strict way. They may transit between clusters or be part of more than one cluster in several different manners. A transition may occur or not depending on the specification of the model. Furthermore, more than one state may be used inside the same cluster, allowing to perform simultaneously modelling and clustering. The multitude of specifications of the latent transition matrix adds to the flexibility of the HMTD and make this model more attractive for social sciences, but also for diverse other domains.

Furthermore the addition of covariates on both hidden and visible levels have been shown to additionally enhance the results of the model. In chapter 5, the use of this

model in clustering problems and the importance of the inclusion of covariates, have been illustrated in the example of internet addiction test sequences among adolescents. Besides providing a solution for these data, we also explored how the inclusion of covariates improved the previously obtained solution. In this chapter, we also provided a small overview of the advantages of HMTD and we also compared it with its alternative, the Growth Mixture Model. The strengths of both approaches have been summarised, and we concluded that both of them can be useful in longitudinal data clustering.

7.5 Coding particularities and R package

The largest amount of time of my thesis was dedicated to the implementation of all the estimation functions of the model in R. Therefore it is important to briefly discuss some particularities of the current model implementation and developments.

The implementation of the GEM algorithm in its current form, including the development of its new optimization function, was among the major challenges in the R implementation. All other functions such as the Forward-Backward, Viterbi and other algorithms have also been implemented in different R functions. Another major difficulty in the coding was the inclusion of the covariates on latent and visible level. For both cases, the model had to be adapted to continuous, discrete and nominal type of covariates in order to respond to the various needs in social sciences. For the visible covariates, time varying covariates were also included, which is useful when dealing with longitudinal data. A major concern was to adapt the model to all the different specifications of the latent and visible level such as the number of lags for the mean and variance, various dimensions of the inputs and the covariates and different treatment depending on the type and level of covariates, the transition and the initial probability matrix. The nature of the outputs is also considered, aiming to provide comprehensible results of the model. They are released as an S4 class object with multiple slots. Several likelihood optimization methods can still be chosen depending on the users' preference. It is also possible to apply the model on data with different sequence lengths, provided that the possible covariates have the same length. Finally the computation time has been an important issue especially considering the nature of the likelihood optimization and the fact the R is a rather high-level programming language (which means that it is considerably slower, in computational time, than other languages such as C or even Matlab for instance).

It is important to mention the possibility to include initial values for the visible-level parameters. These values are important if the user has an insight of what the possible

clusters (or at least the features of the data) can be. For instance the presence and the order of auto-regressive part of the data, the positive or negative influence of the visible covariates on the different groups (mostly men or women in a given class for example) or simply the number of groups. These initial values are important because they can hint at the possible region of the solution space that needs more attention, which could significantly reduce the computational time and result in more optimal final solution.

All the described features will be regrouped in an R package that will be as user friendly as possible. This package will propose all the possible tuning that the user requires, but also include default values for non-specialists in latent Markovian modelling.

7.6 Further developments

The implementation of an universal R package able to treat and cluster all kind of longitudinal data using any specification of the HMTD model (including HMM, DCMM, MTD, Mixture models etc.) will be a very useful tool. As this thesis has shown, HMTD may be a very good alternative to GMM and the other clustering methods. The package for continuous data will be released soon.

On the basis of preliminary trials (that are not included in this thesis), the HMTD model seems to copes well with discrete data too. However, it would be interesting to compare more extensively the performance of the HMTD with DCMM (R package MARCH) when clustering discrete data. Even though DCMM is a model that is specifically designed for discrete data, chances are that HMTD could be as good as it for this type of data, because many discrete distributions can be approximated by continuous ones.

The estimation procedure remains demanding in terms of computational time. Accelerating the convergence of the model in its full form could open new possibilities for treating larger datasets with higher number of covariates.

Another interesting point is the further study of the different flexible clustering possibilities, as well as their application using latent level covariates together with the visible ones. The combination of simultaneous clustering and modelling may be very attractive for many social studies involving different life course trajectories as discussed before. The possibility to identify general groups of persons that evolve differently by simultaneously estimating and modelling differently their latent trajectories, may represent an innovative and useful tool in various domains.

Finally, it would be interesting to see how the HMTD model performs in various

other fields and for different purposes. An interesting application may be to identify sequences whose distribution may not be appropriate according to the nature of the phenomenon of interest. As a small illustration in social sciences, we can imagine a longitudinal study that concerns a sensible subject to which the respondents may be afraid to answer and prefer to conform to the norm. In this case, one cluster of the model may capture the trajectories whose auto-dependence structure deviates from the others (randomly or incorrectly answered questions). In other fields, one may similarly identify errors due to the person in charge of collecting survey data, or a faulty measurement tool for instance. Besides ordinary clustering, these are only a few of the many possible applications of the HMTD model.

Bibliography

[1] Abbott A, Tsay A. (2000) Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect *Sociological Methods Research*, August Vol. 29 No. 1, 3-33

[2] Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering: A decade review. Information Systems, 53, 16-38. Chicago

[3] Aitkin, M. (2001). Likelihood and Bayesian analysis of mixtures.*Statistical Modelling*, 1(4), 287-304.

[4] Altman, R. M. (2007). Mixed hidden Markov models: an extension of the hidden Markov model to the longitudinal data setting.*Journal of the American Statistical Association*, 102(477), 201-210.

[5] Antoniadis, A., Brossat, X., Cugliari, J., & Poggi, J. M. (2013). Clustering functional data using wavelets.*International Journal of Wavelets, Multiresolution and Information Processing*, 11(01), 1350003.

[6] Bank, J., & Cole, B. (2008). Calculating the Jaccard similarity coefficient with map reduce for entity pairs in wikipedia. *Wikipedia Similarity Team*, 1-18.

[7] Barrense-Dias Y, Berchtold A, Akre C, Surís JC (2015) The relation between Internet use and overweight among adolescents: a longitudinal study in Switzerland. *International Journal of Obesity* 40: 45-50.

[8] Bauer, D., and Curran, P. (2003) "Distributional assumptions of growth mixture models: implications for overextraction of latent trajectory classes." *Psychological methods* 8.3: 338.

[9] Begleiter R., El-Yaniv R., Yona G. (2004) *On Prediction Using Variable Order Markov Models Journal of Artificial Intelligence Research* 22 p.385-421

[10] Bendtsen, C. (2012) pso: Particle swarm optimization. R package version 1.0.3. Available on*https://cran.r-project.org/web/packages/pso/index.html*.

[11] Berchtold A. (1995) Autoregressive Modelling of Markov Chains. In Proceedings of the 10th International Workshop on Statistical Modelling,*Springer-Verlag*, New York pp. 19-26

[12] Berchtold A. (1998) Chaînes de Markov et Modèles de Transition: Applications aux Sciences Sociales.*Editions HERMES*,Paris

[13] Berchtold A. (1999) The Double Chain Markov Model. *Communications in Statistics: Theory and Methods*, 28(11), 2569-2589.

[14] Berchtold A. (2001) Estimation in the mixture transition distribution Model. *Journal of Time Series Analysis* 22(4): 379-397.

[15] Berchtold, A. (2003) Mixture transition distribution (MTD) modelling of heteroscedastic time series. *Computational statistics and data analysis*41(3): 399-411.

[16] Berchtold, A., Jeannin, A., Akre, C., Michaud, P.-A. & Surs, J.-C. (2010) First use of multiple substances: Identification of meaningful patterns. *Journal of Substance Use* 15: 118-130.

[17] Berchtold, A. and Raftery, A. (2002) The mixture transition distribution model for high-order Markov chains and non-Gaussian time series. *Statistical Science* 17(3): 328-356.

[18] Berchtold A., Sackett G. (2002) Markovian Models for the Developmental Study of Social Behavior. *American Journal of Primatology*, 58 (3), 149-167.

[19] Berchtold A., Surís J.-C., Meyer T. and Taushanov Z. (2017) Development of somatic complaints among adolescents and young adults in Switzerland. Accepted for publication in the *Swiss Journal of Sociology*.

[20] Bertoletti, M., Friel, N., & Rastelli, R. (2015). Choosing the number of clusters in a finite mixture model using an exact integrated completed likelihood criterion. *Metron*, 73(2), 177-199.

[21] Biernacki, C., Celeux, G., & Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7), 719-725.

[22] Biernacki, C., Celeux, G., & Govaert, G. (2010). Exact and Monte Carlo calculations of integrated likelihoods for the latent class model. *Journal of Statistical Planning and Inference*, 140(11), 2991-3002.

[23] Bishop, C. (2007). Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn.*Springer*, New York.

[24] Bolano D. and Berchtold A. (2016) General framework and model building in the class of Hidden Mixture Transition Distribution models. *Computational Statistics and Data Analysis* 93: 131-145.

[25] Boussau B., Gueguen L. et Gouy M. (2009)A Mixture Model and a Hidden Markov Model to Simultaneously Detect Recombination Breakpoints and Reconstruct Phylogenies, *Evolutionary Bioinformatics*, 5 p.67-79

[26] Bühlmann P., Wyner A.J. (1999) Variable Length Markov Chains. *The Annals of Statistics*, 27, 480-513.

[27] Byrd, R. H., Lu, P., Nocedal, J. and Zhu, C. (1995) A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16(5): 1190-1208.

[28] Celeux, G., Chauveau, D., & Diebolt, J. (1995). On Stochastic Versions of the EM Algorithm. [Research Report] RR-2514,*INRIA*. 1995. inria-00074164

[29] Celeux, G., & Govaert, G. (1995). Gaussian 9parsimonious clustering models *Pattern recognition*, 28(5), 781-793.

[30] Celeux, G., Hurn, M. and Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions.*Journal of the American Statistical Association*, 95(451), 957-970.

[31] Celeux, G., Martin, O., & Lavergne, C. (2005). Mixture of linear mixed models for clustering gene expression profiles from repeated microarray experiments.*Statistical Modelling*, 5(3), 243-267.

[32] Celeux, G. On the different ways to compute the Integrated Completed Likelihood Criterion, *http://convegni.unica.it/cladag2015/files/2015/10/Celeux.pdf*

[33] Cerny, V. (1985) Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45: 41-51.

[34] Chariatte V., Berchtold A., Akr C., Michaud P.A., Suris J.C. (2008) Missed Appointments in an Outpatient Clinic for Adolescents, an Approach to Predict the Risk of Missing. *Journal of Adolescent Health*, 43:38-45.

[35] Cherif, A., Cardot, H., & Bon, R. (2011). SOM time series clustering and prediction with recurrent neural networks.*Neurocomputing*, 74(11), 1936-1944.

[36] Chiou, J. M., & Li, P. L. (2007). Functional clustering and identifying substructures of longitudinal data.*Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4), 679-699.

[37] Ciampi, A., Campbell, H., Dyachenko, A., Rich, B., McCusker, J., & Cole, M. G. (2012). Model-based clustering of longitudinal data: Application to modeling disease course and gene expression trajectories.*Communications in Statistics-Simulation and Computation*, 41(7), 992-1005.

[38] Coffey, N., Hinde, J., & Holian, E. (2014). Clustering longitudinal profiles using P-splines and mixed effects models applied to time-course gene expression data.*Computational Statistics & Data Analysis*, 71, 14-29.

[39] Cron, A. J., and West, M. (2011). Efficient classification-based relabeling in mixture models.*The American Statistician*, 65(1), 16-20.

[40] Davidson, I., & Satyanarayana, A. (2003, November). Speeding up k-means clustering by bootstrap averaging.*In IEEE data mining workshop on clustering large data sets*.

[41] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm.*Journal of the royal statistical society. Series B (methodological)*, 1-38.

[42] Der Kiureghian, A., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter?. *Structural Safety*, 31(2), 105-112.

[43] Dey, M., Jorm, A. F., & Mackinnon, A. J. (2015). Cross-sectional time trends in psychological and somatic health complaints among adolescents: a structural

equation modelling analysis of Health Behaviour in School-aged Children: data from Switzerland.*Social Psychiatry and Psychiatric Epidemiology*, 50(8), 1189-1198. https://doi.org/10.1007/s00127-015-1040-3

[44] Dias, J. G. & Vermunt, J. K. (2006). Bootstrap methods for measuring classification uncertainty in latent class analysis. In*Compstat 2006-Proceedings in Computational Statistics* (pp. 31-41). Physica-Verlag HD.

[45] Dias, J. G., & Wedel, M. (2004). An empirical comparison of EM, SEM and MCMC performance for problematic Gaussian mixture likelihoods.*Statistics and Computing*, 14(4), 323-332.

[46] Efron, B. (1979). Bootstrap methods: another look at the jackknife.*The annals of Statistics*, 1-26.

[47] Efron, B. and Tibshirani, R. J. (1994). An introduction to the bootstrap.*CRC*

[48] Elbeltagi, E., Hegazy, T. and Grierson, D. (2005) Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics* 19: 43-53.

[49] Fang, L., Chen, P. and Liu S. (2007) Particle swarm optimization with simulated annealing for TSP. *Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases* (AIKED 07).

[50] Fang, Y. and Wang, J. (2012). Selection of the number of clusters via the bootstrap method. Computational Statistics & Data Analysis, 56(3), 468-477.

[51] Faraci P, Craparo G, Messina R, Severino S (2013) Internet Addiction Test (IAT): Which is the Best Factorial Solution?. *J Med Internet Res.* 15(10): e225.

[52] Felsenstein J. and Churchill G. (1996) A Hidden Markov Model Approach to Variation Among Sites in Rate of Evolution, *Mol. Biol. Evol.* 13(1) p.93-104

[53] Francis, B. and Liu, J. (2015) Modelling escalation in crime seriousness: a latent variable approach. *Metron* 73.2 : 277-297.

[54] Friedman, J., Hastie, T. and Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). *Springer, Berlin: Springer series in statistics.*

[55] Gabadinho, A., & Ritschard, G. (2016). Analyzing state sequences with probabilistic suffix trees: the PST R package. *Journal of Statistical Software*, 72(3), 1-39. doi:doi:10.18637/jss.v072.i03

[56] Gelfand, A. E., & Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410), 398-409.

[57] Genolini, C., & Falissard, B. (2010). KmL: k-means for longitudinal data.*Computational Statistics*, 25(2), 317-328.

[58] Grob, A., Lthi, R., Kaiser, F. G., Flammer, A., Mackinnon, A., & Wearing, A. J. (1991). Berner Fragebogen zum Wohlbefinden Jugendlicher (BFW).*Diagnostica*, 37(1), 66-75.

[59] Gruet, M. A., Philippe, A., & Robert, C. P. (1999). MCMC control spreadsheets for exponential mixture estimation.*Journal of Computational and graphical Statistics*, 8(2), 298-317.

[60] Grün, B., & Leisch, F. (2004). Bootstrapping finite mixture models.*COMPSTAT 2004 Symposium*

[61] Grün, B., & Leisch, F. (2009). Dealing with label switching in mixture models under genuine multimodality.*Journal of Multivariate Analysis*, 100(5), 851-861.

[62] Hajjem, A., Bellavance, F., & Larocque, D. (2014). Mixed-effects random forest for clustered data.*Journal of Statistical Computation and Simulation*, 84(6), 1313-1328.

[63] Helske S, Helske J, Eerola M (2018) Combining Sequence Analysis and Hidden Markov Models in the Analysis of Complex Life Sequence Data. In G Ritschard & M Studer (eds),Sequence Analysis and Related Approaches: Innovative Methods and Applications. *Berlin: Springer*.

[64] Hennig, C. (2007). Cluster-wise assessment of cluster stability.*Computational Statistics & Data Analysis*, 52(1), 258-271.

[65] Hennig, C. (2016) Practical decision making in cluster analysis: Choice of method and evaluation of quality.*Talk on the 22nd International Conference on Computational Statistic COMPSTAT 2016*, Oviedo, Spain, http://www.compstat2016.org/docs/compstatvalidation.pdf?20160821232000

[66] Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2016). Handbook of cluster analysis.*CRC Press*.

[67] Holland, J. H. (1992) Genetic algorithms. *Scientific American* 267(1): 66-72.

[68] Jacques, J., & Preda, C. (2014). Functional data clustering: a survey.*Advances in Data Analysis and Classification*, 8(3), 231-255.

[69] Jasra A., Holmes C., & Stephens D. A. (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling.*Statistical Science*, 50-67.

[70] Jelenchick, L. A., Becker, T., & Moreno, M. A. (2012). Assessing the psychometric properties of the Internet Addiction Test (IAT) in US college students. *Psychiatry research*, 196(2), 296-301.

[71] Jin, Y. (2006). Multi-objective machine learning (Vol. 16).*Springer Science & Business Media*.

[72] Jones, B. L., Nagin, D. S., & Roeder, K. (2001). A SAS procedure based on mixture models for estimating developmental trajectories.*Sociological methods & research*, 29(3), 374-393.

[73] Jonsson, F., Hammarström, A., & Gustafsson, P. E. (2014). Social capital across the life course and functional somatic symptoms in mid-adulthood.*Scandinavian Journal of Public Health*. https://doi.org/10.1177/1403494814548749

[74] Jung, T. and Wickrama K.A.S. (2008) An Introduction to Latent Class Growth Analysis and Growth Mixture Modeling *Social and Personality Psychology Compass*, Volume 2, Issue 1, pages 302?317

[75] Kemeny J.G. & Snell J.L. (1976) Finite Markov Chains. *Springer-Verlag*, New York

[76] Kemeny J.G., Snell J.L., Knapp A.W. (1976) Denumerable Markov Chains. *Springer-Verlag*, New York

[77] Kennedy, J. and Eberhart, R. (1995) Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks* IV: 1942-1948.

[78] Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models.*Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3), 425-464.

[79] Khazaal Y, Billieux J, Thorens G, Khan R, Scarlatti E, Theintz F, Lederrey J, Van Der Linden M, Zullino D (2008) French validation of the Internet addiction test. *Cyberpsychology Behavior* 11(6):703-706.

[80] Kirkpatrick, S., Gelatt Jr, C. D. and Vecchi, M. P. (1983) Optimization by simulated annealing. *Science* 220: 671-680.

[81] Kohonen, T. (1998). The self-organizing map.*Neurocomputing*, 21(1), 1-6.

[82] Krogh, A., Larsson, B., Von Heijne, G. and Sonnhammer, E. L. (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of molecular biology*, 305(3), 567-580.

[83] Lai, C. M., Mak, K. K., Cheng, C., Watanabe, H., Nomachi, S., Bahar, N., ... & Griffiths, M. D. (2015). Measurement invariance of the internet addiction test among Hong Kong, Japanese, and Malaysian adolescents. *Cyberpsychology, Behavior, and Social Networking*, 18(10), 609-617.

[84] Leisch, F. (2016) Resampling Methods for Exploring Cluster Stability, In Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.) Handbook of cluster analysis (Chapter 28).*CRC Press*.

[85] Leng, X., & Müller, H. G. (2006). Classification using functional data analysis for temporal gene expression data.*Bioinformatics*, 22(1), 68-76.

[86] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest.*R news*, 2(3), 18-22.

[87] Little, R. J., & Rubin, D. B. (2014). Statistical analysis with missing data (Vol. 333). *John Wiley & Sons*.

[88] MacDonald I. and Zucchini W. (1997) *Hidden Markov and Other Models for Discrete-valued Time Series* Chapman & Hall

[89] Marin, J. M., Mengersen, K., and Robert, C. P. (2005). Bayesian modelling and inference on mixtures of distributions.*Handbook of statistics*, 25, 459-507.

[90] Maruotti, A. (2011). Mixed hidden Markov models for longitudinal data: an overview.*International Statistical Review*, 79(3), 427-454.

[91] Matsumoto, M. and Nishimura, T. (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 8(1): 3-30.

[92] McArdle, J.J. and Epstein, D. (1987) Latent growth curves within developmental structural equation models *Child development* (1987): 110-133.

[93] McLachlan G. et Krishnan T. (1997) The EM Algorithm and Extentions *Wiley Series in Probability and Statistics* 1997

[94] McLachlan, G., and Peel, D. (2004). Finite mixture models.*John Wiley & Sons*.

[95] McNicholas, P. D., & Murphy, T. B. (2010). Model-based clustering of longitudinal data. *Canadian Journal of Statistics*, 38(1), 153-168.

[96] Mehran F. (1989) Analysis of Discrete Longitudinal Data: Infinite-Lag Markov Models. In *Statistical Data Analysis and Inference*, pp. 533-541. (Edited by Y. Dodge),*Elsevier Science Publishers*

[97] Meila, M. (2016) Criteria for Comparing Clusterings, In Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.) Handbook of cluster analysis (Chapter 27).*CRC Press*.

[98] Mercer, R.E. and Sampson, J.R. (1978) Adaptive search using a reproductive meta-plan. *Kybernetes* 7(3): 215-228.

[99] Meyn S.P. et Tweedie R.L. (1993) *Markov Chains and Stochastic Stability*, Springer-Verlag

[100] Mohapatra, S., Deo, S. J. K., Satapathy, A., & Rath, N. (2014). Somatoform Disorders in Children and Adolescents.*German Journal of Psychiatry*, 17(1), 19-24.

[101] Mullen, K., Ardia, D., Gil, D., Windover, D. and Cline, J. (2011) DEoptim: an R package for global optimization by differential evolution. *Journal of Statistical Software* 40(6): 1-26.

[102] Muthén, B. "Latent variable mixture modeling." *New developments and techniques in structural equation modeling* (2001): 1-33.

[103] Muthén, B. and Asparouhov T. Growth mixture modeling: Analysis with non-Gaussian random effects.*Longitudinal data analysis* (2008): 143-165.

[104] Muthén, B. and Shedden, K. (1999) Finite mixture modelling with mixture outcomes using the EM algorithm. *Biometrics* 55.2 : 463-469.

[105] Nagin, D. (1999) Analyzing developmental trajectories: a semiparametric, group-based approach. *Psychological methods* 4.2 139.

[106] Nelder, J.A. and Mead, R. (1965) A simplex method for function minimization.*Computer Journal* 7: 308-313.

[107] Ocone D. (2009) *Markov Chains and Applications to Population Genetics,http://www.math.rutgers.edu/courses/338/coursenotes/markovchains.pdf*

[108] Pardo B. et Birmingham W. (2005) *Modeling Form for On-line Following of Musical Performances*, *Proceedings of the Twentieth National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, July 9-13, 2005

[109] Paterlini, S. and Krink, T. (2006). Differential evolution and particle swarm optimization in partitional clustering. *Computational statistics & data analysis*, 50(5), 1220-1247.

[110] Aneiros-Pérez, G., Cao, R., & Vilar-Fernández, J. M. (2011). Functional methods for time series prediction: a nonparametric approach.*Journal of Forecasting*, 30(4), 377-392.

[111] Piguet C, Berchtold A, Zimmermann G and Surís JC (2016) Rapport final de l'étude longitudinale Ado@Internet.ch.*Lausanne: Institut universitaire de médecine sociale et préventive.* (Raisons de santé, 255).

[112] Premalatha, K. and Natarajan, A.M. (2009) Hybrid PSO and GA for global maximization. *International Journal of Open Problems in Computer Science and Mathematics* 2(4): 597-608.

[113] Proust-Lima, C., Philipps, V. and Liquet, B. (2015) Estimation of extended mixed models using latent classes and latent processes: the R package lcmm. *arXiv*:1503.00890.

[114] Pukkala, T. and Kurttila, M. (2005) Examining the performance of six heuristic optimization techniques in different forest planning problems. *Silva Fennica* 39(1): 67-80.

[115] Putnick, D. L., & Bornstein, M. H. (2016). Measurement invariance conventions and reporting: the state of the art and future directions for psychological research. *Developmental Review*, 41, 71-90.

[116] R Core Team (2015) R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. URL: https://www.R-project.org/.

[117] Rabiner, L. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2): 257-286.

[118] Rabiner, L. R. and Juang, B. H. (1986). An introduction to hidden Markov models *ASSP Magazine*, IEEE, 3(1), 4-16.

[119] Raftery, A. (1985) A model for high-order Markov chains. *Journal of the Royal Statistical Society, series B* 47(3): 528-539.

[120] Raftery A (1995) Bayesian model selection in social research. *Sociological Methodology* 25:111-163.

[121] 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.

[122] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods.*Journal of the American Statistical association*, 66(336), 846-850.

[123] Reinecke, J. and Seddig, D. (2011) Growth mixture models in longitudinal research. *AStA Advances in Statistical Analysis* 95.4: 415-434.

[124] Richardson S., & Green P. J. (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.

[125] Rissanen J. (1983) *A Universal Data Compression SystemIEEE Transactions on Information Theory* 29 (5) p.656-664

[126] Rodriguez, C. E., & Walker, S. G. (2014). Label switching in Bayesian mixture models: Deterministic relabeling strategies.*Journal of Computational and Graphical Statistics*, 23(1), 25-45.

[127] Rossi, F., Conan-Guez, B., & El Golli, A. (2004, April). Clustering functional data with the SOM algorithm. *In ESANN* (pp. 305-312).

[128] Peter E. Rossi (2014). Bayesian Non- and Semi-parametric Methods and Applications.*Princeton University Press*

[129] Rosychuk, R. J., Sheng, X., & Stuber, J. L. (2006). Comparison of variance estimation approaches in a two-state Markov model for longitudinal data with misclassification.*Statistics in medicine*, 25(11), 1906-1921.

[130] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.*Journal of computational and applied mathematics*, 20, 53-65.

[131] Rydén, T. (2008) EM versus Markov chain Monte Carlo for estimation of hidden Markov models: a computational perspective. *Bayesian Analysis* 3(4): 659-688.

[132] Ryden, T. (2008) EM versus Markov chain Monte Carlo for estimation of hidden Markov models: a computational perspective. *Bayesian Analysis* 3(4): 659-688.

[133] Sarlin, P. (2013). Self-organizing time map: An abstraction of temporal multivariate patterns.*Neurocomputing*, 99, 496-508.

[134] Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2), 461-464.

[135] Scott, S. (2002) Bayesian methods for hidden Markov models. *Journal of the American Statistical Association* 97: 337-351.

[136] Scrucca, L. (2013) GA: A package for genetic algorithms in R. *Journal of Statistical Software* 53(4).

[137] Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: a data mining approach for longitudinal and clustered data.*Machine learning*, 86(2), 169-207.

[138] Shi, Y. and Eberhart, R.C. (1998) A modified particle swarm optimizer. *Proceedings of IEEE International Conference on Evolutionary Computation* 69-73.

[139] Shmilovici A. et Ben-Gal I. (2007) *Using a VOM model for reconstructing potential coding regions in EST sequences* Computational Statistics 22 p.49-69

[140] Singer, S. and Nelder, J. (2009) Nelder-Mead algorithm. *Scholarpedia* 4(7): 2928.

[141] Skarupova K, Olafsson K, Blinka L (2015) Excessive Internet Use and its association with negative experiences: Quasi-validation of a short scale in 25 European countries. *Computers in Human Behavior* 53:118-123.

[142] Song, J. J., Lee, H. J., Morris, J. S., & Kang, S. (2007). Clustering of time-course gene expression data using functional data analysis.*Computational biology and chemistry*, 31(4), 265-274.

[143] Sperrin, M., Jaki, T. and Wit, E. (2010). Probabilistic relabelling strategies for the label switching problem in Bayesian mixture models.*Statistics and Computing*, 20(3), 357-366.

[144] Srinivas, M. and Patnaik, L. (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on System, Man and Cybernetics* 24(4): 656-667.

[145] Steinley, D. (2008). Stability analysis in K-means clustering.*British Journal of Mathematical and Statistical Psychology*, 61(2), 255-273.

[146] Steinley, D. (2004). Properties of the Hubert-Arable Adjusted Rand Index.*Psychological methods*, 9(3), 386.

[147] Stephens, M. (2000). Dealing with label switching in mixture models.*Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4), 795-809.

[148] Storn, R. and Price, K. (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11: 341-359.

[149] Surís JC, Akre C, Berchtold A, Fleury-Schubert A, Michaud PA and Zimmermann G (2012) Ado@Internet.ch: Usage d'Internet chez les adolescents vaudois. *Lausanne: Institut universitaire de médecine sociale et préventive.* (Raisons de santé, 208).

[150] Surís JC, Akre C, Piguet C, Ambresin AE, Zimmermann G and Berchtold A (2014) Is Internet use unhealthy? A cross-sectional study of adolescent Internet overuse. *Swiss Med Wkly* 2014;144:w14061

[151] Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398), 528-540.

[152] Taushanov, Z., & Berchtold, A. (2017). A Direct Local Search Method and its Application to a Markovian Model. *Statistics, Optimization & Information Computing*, 5(1), 19-34.

[153] Taushanov, Z., & Berchtold, A. (2017) Markovian-based Clustering of Internet Addiction Trajectories. In G Ritschard & M Studer (eds), *Sequence Analysis and Related Approaches: Innovative Methods and Applications*. Berlin: Springer.

[154] Teicher, H. (1961). Identifiability of mixtures. *The annals of Mathematical statistics*, 32(1), 244-248.

[155] Teicher, H. (1963). Identifiability of finite mixtures. *The annals of Mathematical statistics*, 1265-1269.

[156] Titterington, D. M., Smith, A. F., & Makov, U. E. (1985). Statistical analysis of finite mixture distributions. *Wiley*.

[157] Tomida, S., Hanai, T., Honda, H., & Kobayashi, T. (2002). Analysis of expression profile using fuzzy adaptive resonance theory.*Bioinformatics*, 18(8), 1073-1083.

[TREE] TREE. (2016). Documentation on the first TREE cohort (TREE1), 2000-2016. Bern: TREE. Retrieved from *http* : *//www.tree.unibe.ch/unibe/portal/fak_wiso/c_dep_sowi/micro_tree/content/e206328/e305140/e305154/files476810/TREE_2016_Project_documentation_TREE1_2000 − 2016_English_ger.pdf*

[159] Tse, E., & Anton, J. (1972). On the identifiability of parameters. *IEEE Transactions on Automatic Control*, 17(5), 637-646.

[160] Vandenberg, R. J., & Lance, C. E. (2000). A review and synthesis of the measurement invariance literature: Suggestions, practices, and recommendations for organizational research. *Organizational research methods*, 3(1), 4-70.

[161] Verbeke, G. and 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.

[162] Verhoof, E., Maurice-Stam, H., Heymans, H., & Grootenhuis, M. (2012). Growing into disability benefits? Psychosocial course of life of young adults with a chronic somatic disease or disability. *Acta Paediatrica*, 101(1), e19-e26. https://doi.org/10.1111/j.1651-2227.2011.02418.x

[163] Vermunt, J. K., & Magidson, J. (2003) Latent class models for classification. *Computational Statistics & Data Analysis*, 41(3), 531-537.

[164] Visser, I., Raijmakers, M. E., & Molenaar, P. (2000). Confidence intervals for hidden Markov model parameters.*British journal of mathematical and statistical psychology*, 53(2), 317-327.

[165] Viterbi A. (1967) "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm" *IEEE Transactions on Information Theory* 13(2) p.260-269

[166] Von Luxburg, U. (2007). A tutorial on spectral clustering *Statistics and computing*, 17(4), 395-416.

[167] Wang, X., Smith, K., & Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3), 335-364.

[168] Wang, M. and Bodner, T. E. (2007) Growth mixture modeling identifying and predicting unobserved subpopulations with longitudinal data. *Organizational Research Methods* 10(4), 635-656.

[169] Wong C.S., Li W.K. (2001) On a mixture autoregressive conditional heteroscedastic model. *Journal of the American Statistical Association*, 96, 982-995.

[170] Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of statistics*, 95-103.

[171] Wu L. (2000) Some Comments on "Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect" *Sociological Methods & Research* Vol.29 399-411

[172] Xiang, Y., Gubian, S., Suomela, B. and Hoeng J. (2013) Generalized simulated annealing for global optimization: the GenSA package. *The R Journal* 5(1).

[173] Yakowitz, S. J., & Spragins, J. D. (1968). On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 209-214.

[174] Yao, W. (2012). Model based labeling for mixture models. *Statistics and Computing*, 22(2), 337-347.

[175] Yao W. (2015) Label switching and its solutions for frequentist mixture models. *Journal of Statistical Computation and Simulation*, 85(5), 1000-1012.

[176] Young KS (1998) Internet Addiction: The Emergence of a New Clinical Disorder. *CyberPsychology & Behavior* 1:237-244

[177] Zhu, W., & Fan, Y. (2016). Relabelling algorithms for mixture models with applications for large data sets. *Journal of Statistical Computation and Simulation*, 86(2), 394-413.

Glossary

AIC Akaike Information Criterion. 108

BIC Bayesian Information Criterion. 108

CI Confidence Interval. 97

DE Differential Evolution. 61

EM Expectation-Maximization algorithm. 41

GA Genetic Algorithm. 60

GEM Generalized Expectation-Maximization algorithm. 46

GMM Gaussian Mixture Models. 9

GMM Growth Mixture Models. 34

HMM Hidden Markov Model. 13

$HMTD$ Hidden Mixture Transition Distributions. 15

ICL Integrated Complete Likelihood. 109

MLE Maximum Likelihood Estimate. 97

MTD Mixture Transition Distributions. 9

NM Nelder-Mead optimisation. 61

PSO Particle Swarm Optimization. 60

SA Simulated Annealing. 59