How to explore within-person and between-person measurement model differences in intensive longitudinal data with the R package *lmfa*

Vogelsmeier, Leonie V.D.E, Vermunt, Jeroen K., De Roover, Kim

Tilburg University, The Netherlands

Draft version 1, 11/06/2021. This manuscript is a preprint and has not been peer reviewed.

The research leading to the results reported in this paper was sponsored by the Netherlands Organization for Scientific Research (NWO) [Research Talent grant 406.17.517; Veni grant 451.16.004]. Correspondence concerning this paper should be addressed to Leonie V. D. E. Vogelsmeier, Department of Methodology and Statistics, PO Box 90153 5000 LE Tilburg. E-mail: l.v.d.e.vogelsmeier@tilburguniversity.edu

Abstract

Intensive longitudinal data (ILD) have become popular for studying within-person dynamics in psychological constructs (or between-person differences therein). Prior to investigating what the dynamics look like, it is important to examine whether the measurement model (MM) is the same across subjects and time and, thus, whether the measured constructs have the same meaning. If the MM differs (e.g., because of changes in item interpretation or response styles), observations cannot be validly compared. Exploring differences in the MM for ILD can be done with latent Markov factor analysis (LMFA), which classifies observations based on the underlying MM (for many subjects and time-points simultaneously) and thus shows which observations are comparable. However, the complexity of the method or the fact that no open-source software for LMFA existed until now may have hindered researchers from applying the method in practice. In this article, we introduce the new user-friendly software package *lmfa*, which allows researchers to perform the analysis in the freely available software R. We provide a step-by-step tutorial for the *lmfa* package so that researchers can easily investigate MM differences in their own ILD.

Keywords: intensive longitudinal data, ESM, measurement invariance, factor analysis, latent Markov modeling, three-step approach, R, software package

1. Introduction

In recent years, researchers have shown an increased interest in intensive longitudinal data (ILD) for studying dynamics of psychological latent constructs (or "factors") such as depression or affective well-being, for many subjects over a longer period of time. The ILD are commonly obtained by means of Experience Sampling Methodology (ESM; Scollon, Kim-Prieto, & Diener, 2003), where many subjects repeatedly complete small questionnaires—containing items intended to measures the latent factors—at random (or event-based) time-points, several times a day for several days or weeks via a smartphone app. State-of-the-art analyses to model dynamics in psychological factors for many subjects over time range from basic random effect models (for studying individual differences in the dynamics or average levels of the measured factors; Hamaker, Ceulemans, Grasman, & Tuerlinckx, 2015; Myin-Germeys et al., 2018), over multilevel autoregressive models (for studying individual differences in lagged relationships between measured factors; e.g., Bringmann et al., 2013), to dynamic structural equation modeling (DSEM) that, in addition to the methods just mentioned, also allow for the estimation of more complex models (e.g., models containing multiple outcome variables; McNeish & Hamaker, 2020).

While the technology to gather ILD and approaches to analyze dynamics in the measured constructs are readily available, an important point of concern of many researchers before they start their analyses is whether the latent factors actually have the same meaning across subjects and time-points and, thus, whether observations are comparable. For this, the measurement model (MM) needs to be invariant across observations, that is, measurement invariance (MI) must hold. The MM indicates which factors are measured by which indicators and, for continuous item responses, is traditionally obtained with factor analysis (FA; Lawley & Maxwell, 1962). In the resulting MM (or "FA model"), factor loadings indicate the extent to which items measure the

factors and item intercepts indicate the expected item scores when scores on the factors are equal to zero. If the loadings, the intercepts, or the number of factors differ within or across subjects, MI is violated and factors cannot be meaningfully compared (Adolf, Schuurman, Borkenau, Borsboom, & Dolan, 2014). However, invariance within and between subjects is easily violated because of differences and changes in response styles (Moors, 2003; Paulhus, 1991) or item interpretations (Oort, Visser, & Sprangers, 2005). Thus, the MM may be different across subjects but also change within a person over time.

To clarify possible non-invariance of MMs, consider the following example. A research team conducts an ESM study to investigate between-subject differences regarding dynamics in affective well-being of employees in the travel sector during the Covid-19 pandemic. On the one hand, the underlying MM may differ across employees because people generally differ in their ability to label emotions in a granular way (Barrett, Gross, Christensen, & Benvenuto, 2001; Erbas, Kalokerinos, Kuppens, van Halem, & Ceulemans, 2020; Kashdan, Barrett, & McKnight, 2015). The "high differentiators" differentiate more between specific emotions such as feeling content or happy than the "low differentiators", who focus more on the valence of a feeling and, thus, whether an emotion is positive or negative (Barrett, 1998; Erbas, Ceulemans, Koval, & Kuppens, 2015). A result could be that several factors underlie the responses of the high differentiators (say, four factors pertaining to high- and low-arousal positive and negative affect) while only one factor underlies the responses of the low differentiators (say, a bipolar "valence of affect" factor). On the other hand, the MM may change within some employees over time. For instance, employees who are normally high differentiators may also switch to a MM with a single "valence of affect" factor when being exposed to a stressful situation (e.g., learning about government restrictions extending temporary part-time work) because stress triggers a valence focus (Erbas et al., 2018). Because the

low differentiators respond according to a single valence of affect factor, regardless of experienced stress, the same MM would be underlying their responses during the entire participation.

Undetected measurement non-invariance is a threat to valid inferences from ILD analyses. Therefore, detecting non-invariance is crucial. Until recently, researchers could only test whether the MM is invariant across (groups of) subjects and/or time-points (e.g., by means of traditional MI tests that are, for example, available in the R package *lavaan*; Rosseel, 2012). However, if the results indicate that invariance is untenable, researcher cannot automatically identify for which subjects or time-points the MMs differ and what the different MMs look like without conducting pairwise comparisons of subject- or time-point specific MM parameters. This quickly becomes infeasible for ILD that usually contain many observations from many subjects. Furthermore, it is only possible to investigate non-invariance across subjects (assuming invariance over time) or to investigate invariance over time (assuming invariance across subjects) and not to investigate both at the same time. These problems were solved by Vogelsmeier, Vermunt, van Roekel, and De Roover (2019), who developed latent Markov factor analysis (LMFA), which allows researchers to conveniently explore all kinds of MM differences, both across subjects and time. LMFA is a mixture modeling approach that combines a latent Markov model (LMM; Bartolucci, Farcomeni, & Pennoni, 2014; Collins & Lanza, 2010) with mixture FA (McLachlan & Peel, 2000; McNicholas, 2016): First, the LMM clusters observations according to their underlying MM into dynamic latent states. Note that the latent states are equivalent to latent classes in a latent class analysis or mixture model but are called states in a LMM because subjects can transition between latent classes over time. Second, for each state, FA reveals what the underlying MM looks like. Summarized, LMFA classifies observations into different states that pertain to different MMs and invariance holds for observations in the same state but is violated for observations in different

states. Researchers can then decide how to continue with their data analysis (e.g., retaining observations from one state or removing non-invariant items; see Section 4 for a more elaborate discussion on this). Researchers can also learn from subjects' transitions between MMs by including time-varying or time-constant covariates as predictors of the state memberships (e.g., "stress" could be included when analyzing the changes in the MM in our employee example).

Although more and more researchers are eager to explore MI in their ILD (Horstmann & Ziegler, 2020), many researchers are still unfamiliar with LMFA, how it is applied, or how results must be interpreted. Furthermore, until now, LMFA was only available in the commercial software Latent GOLD (Vermunt & Magidson, 2016) and, thus, not all researchers had access to the novel method. The latter has now changed with the release of the package *lmfa* (Vogelsmeier & De Roover, 2020) that allows researchers to perform all necessary steps in the open-source software R (R Core Team, 2020). The aim of this paper is to provide a tutorial for the *lmfa* package that guides users through the different steps of performing the analysis and interpreting the results with the goal to increase researchers' confidence and ease in using LMFA. This can also indirectly improve research of applied or substantive researchers. The tutorial is targeted at an audience that has a basic understanding of R but not necessarily of the LMFA model.

The remainder of this paper is organized as follows: First, in Section 2, we describe the data structure, introduce an example dataset, recap the LMFA method, and describe how it is estimated in *lmfa*. Then, in Section 3, we guide the reader through the different analysis steps by means of annotated R code. Next, in Section 4, we describe how to proceed with ILD analyses based on the results of LMFA and, finally, in Section 5, conclude with a discussion about current limitations and possible future extensions of *lmfa*.

2. Method

2.1. Data structure

We assume typical ILD and, thus, repeated measures data containing several continuous and normally distributed[1] variables that were assessed for multiple subjects at multiple measurement occasions and which measure one or more latent factors. In addition to these variables, the data may contain time-constant or time-varying explanatory variables (i.e., covariates), which may be dichotomous, ordinal, or continuous and which may explain transitions between the underlying MMs. For the mathematical notation of the data structure, see Supplementary Material S.1. To clarify the data structure, consider the following example dataset that will be used throughout this tutorial.

2.2. Constructed data example

The data is a synthetic dataset that was inspired by a real ESM dataset, which was used in Vogelsmeier, Vermunt, van Roekel, et al. (2019) to illustrate how to explore MM changes by means of a LMFA without covariates. Every evening for about three months, multiple subjects (suffering from anhedonia, one of the core symptoms of depression; Van Roekel et al., 2017) reported their affect and the unpleasantness of the most unpleasant event they experienced since the previous measurement occasion (in the following just "negative event"). Affect was measured by means of ten positive affect (PA) and eight negative affect (NA) items (see LMFA output box 1) and a single item was used to assess the negative event. All items were assessed on a Visual Analogue Scale ranging from 0 = "Not at all" to 100 = "Very much". Moreover, after the first

[11] Note that one may also investigate ordinal data as long as the data has more than a few response categories (say at least five; Dolan, 1994) and the distribution across the categories is approximately normal. However, the influence on the parameter estimation has not yet been investigated for LMFA in particular and results should be therefore interpreted with caution. Highly skewed responses with only a few response categories may lead to convergence problems and local optima (for a description of local optima, see Section 3.2.3). For such data, the LMFA extension to categorical data should be used, that is, latent Markov latent trait analysis (Vogelsmeier, Vermunt, Keijsers, & De Roover, 2020).

month, subjects were randomly assigned to receiving an intervention to reduce anhedonia or not.[2] The results of LMFA indicated that most subjects transitioned between three MMs that differed with regard to the number and nature of the factors. Descriptive statistics showed that there was a relation between the states and the two covariates “had an intervention” and “negative event”.

For the tutorial in this article, we created a dataset with MMs similar to the ones found in the real data application (yet, somewhat adjusted and simplified) and with the two time-varying covariates “had an intervention” (coded as 1 = “yes” and 0 = “no”) and “negative event” having an effect on the transitions between the states. The dataset contains data for 100 subjects with a mean of 47.76 observations and an SD of 6.56, resulting in a total number of observation equal to 4776. The intervals between measurement occasions differ within and across subjects with an average length of 1.22 days and an SD of 1.02. The negative event scores differ within and across subjects with a mean of 49.65 and an SD of 15.11. Of all subjects, 50 receive no intervention and 50 receive one intervention after approximately 1/3 of their total participation duration. The dataset will be used throughout the article to ease understanding of the method and the steps of the tutorial.

2.3. Latent Markov Factor Analysis (LMFA)

LMFA consists of two building blocks. The first one pertains to the state-specific MMs and, thus, a FA model for each state that indicates which constructs are measured by which items. Note that the state-specific FA assumes continuous item responses (or responses that can be treated as such, say, with five or more categories; Dolan, Oort, Stoel, & Wicherts, 2009). It is possible to use a latent variable model for items with categorical item responses (Vogelsmeier et al., 2020) but this option is not included in the *lmfa* package. The second building block is the LMM that models the transitions between MMs over time (Bartolucci, Farcomeni, & Pennoni, 2015; Zucchini,

[2] The intervention was either a personalized lifestyle advice or the advice in combination with a skydive. For simplicity in this tutorial, we do not distinguish between different types of interventions.

MacDonald, & Langrock, 2016). Note that there are two types of LMMs. First, the discrete-time (DT-) LMM (Bartolucci et al., 2015; Zucchini et al., 2016) assumes the intervals between time-points to be equal across subjects and time. In contrast, the continuous-time (CT-) LMM (Böckenholt, 2005; Jackson & Sharples, 2002) accommodates unequally-spaced observations, which is usually more realistic in ILD (e.g., due to random beeps or skipped measurement occasions). However, the CT-LMM also works for equal intervals. In fact, estimating a CT-LMM with equal intervals is similar to estimating a DT-LMM but the parameter interpretation differs (Vermunt & Magidson, 2016), which will be clarified in Section 2.3.2. The *lmfa* package uses CT-LMM because it is more generally applicable.

LMFA can be estimated with a full information maximum likelihood (FIML) estimation (Vogelsmeier, Vermunt, van Roekel, et al., 2019) or with a three-step (3S) estimation (Vogelsmeier, Vermunt, Bülow, & De Roover, 2019). The latter breaks down the estimation of LMFA into three steps, which makes the analysis faster and more convenient for the investigation of covariate effects. Therefore, *lmfa* uses the 3S estimation. For a detailed discussion about the estimation procedures, we refer to Vogelsmeier, Vermunt, Bülow, et al. (2019).

In the next section, we explain and illustrate the method using LMFA results for our example data. Note that the true number of states and factors (i.e., like in the data generating model) was applied. Of course, the number of states and factors and the relevant covariates are not known in empirical practice. The required model selection and covariate selection procedures are explained in Section 3.2. In the following, we first explain the state-specific MMs (i.e., the FA models; Section 2.3.1). Then, we describe the (covariate-specific) transition model (i.e., the CT-LMM; Section 2.3.2). Next, we summarize the finding of applying LMFA to our example data (Section 2.3.3). Thereafter, we explain the 3S estimation (Section 2.3.4).

2.3.1. The state-specific measurement models

In LMFA, the MMs are determined by state-specific FA models, which consist of three types of parameters. Depending on which parameters differ across states, different levels of MI are violated. In this section, we first explain the parameters of the FA model, the different levels of (non-)invariance, and the exploratory FA that is used to obtain the parameter estimates. Thereafter, we demonstrate potential non-invariance of the FA models across states with our example data.

State-specific FA models and (non-)invariance across states. The first type of parameters are the factor loadings, which determine the item-factor relations and, hence, the degree to which an item measures a factor or, stated differently, to what extent an item is predicted by the underlying factor. Thus, items with stronger loadings are better measures of a factor than items with lower loadings. Second, item intercepts are the expected scores for an item when the factor scores are equal to zero. Third, the items' unique variances indicate the variance of an item that is unique to the item and, hence, that is not explained by the factors (for the mathematical notation and the technical details, see Supplementary Material S.2.2). The three types of parameters can take on different values across states and inform us about violations of four different levels of MI (Meredith, 1993). These levels are configural invariance (invariance of the number of factors and the pattern of non-zero loadings), weak invariance (invariance of the non-zero loadings), strong invariance (invariance of the intercepts), and strict invariance (invariance of the unique variances). It is important to note that strict invariance is assumed to hold within each state, since the states capture differences in loadings, intercepts as well as unique variances.

For obtaining the state-specific MMs, LMFA uses exploratory FA (EFA) and not confirmatory FA (CFA). CFA is too restrictive because it imposes a priori assumptions about the presence or absence of item-factor relations by setting certain loadings equal to zero. Thus, CFA cannot detect

MM differences pertaining to the configural model, such as the number and nature of the underlying factors in our previous employee example. In contrast, EFA detects all types of loading differences, including configural non-invariances like different cross-loadings. Note, however, that the EFA model is not identified without setting constraints. Firstly, one needs to set the scale of the factors. To this end, *lmfa* sets the factor (co-)variances equal to an identity matrix (with dimensions equal to the state-specific number of factors), which means that factors are initially uncorrelated. This initial solution is usually not well interpretable because many items may have high loadings on more than one factor (i.e., there is no "simple structure"; Thurstone, 1947). In order to achieve a more interpretable solution, *lmfa* applies a rotation of the factors for each state. An oblique rotation (i.e., one that allows factors to be correlated) results in the best simple structure and is usually more valid for psychological constructs (Clarkson & Jennrich, 1988; De Roover & Vermunt, 2019; Kiers, 1997). Finally, the factor means are set equal to zero per state. This implies that the state-specific intercepts are in fact state-specific item means.

State-specific FA models for the example data. To illustrate possible measurement non-invariance across states, consider the state-specific MMs resulting from LMFA with three states and three, two, and three factors ("[3 2 3]") of the synthetic dataset described in Section 2.2. The results are displayed in LMFA output box 1.[3]

LMFA output box 1

```
## Estimation converged after 168.67 seconds and 40 iterations.
##
## LL = -353166.81
##
## Number of states: 3
##
## Number of factors: [3 2 3]
##
## ------------------------------------------------------------
##
```

[3] Note that, for clarity, loadings with absolute values larger than .3 are printed in boldface in this tutorial. However, the normal output in R does not include any boldface loadings.

```
## Obliquely rotated standardized loadings:
##
##               S1F1  S1F2  S1F3     S2F1  S2F2     S3F1  S3F2  S3F3
## Interested    0.66  0.04  0.00     0.68  0.01     0.57 -0.01  0.02
## Joyful        0.60  0.02  0.02     0.65 -0.01     0.88  0.01  0.06
## Determined    0.37  0.03 -0.55     0.61  0.00     0.84  0.02 -0.01
## Calm          0.37 -0.58 -0.01     0.59  0.00     0.18 -0.15  0.82
## Lively        0.63  0.03  0.03     0.65  0.00     0.88 -0.01  0.01
## Enthusiastic  0.65 -0.01  0.02     0.64  0.00     0.89  0.02  0.00
## Relaxed       0.64  0.02  0.00     0.64  0.01     0.16 -0.14  0.85
## Cheerful      0.63  0.07  0.01     0.63 -0.01     0.91  0.01  0.02
## Content       0.61  0.00  0.03     0.67  0.02     0.93  0.02  0.01
## Energetic     0.64 -0.01  0.00     0.63 -0.01     0.90  0.05 -0.01
## Upset         0.09  0.62 -0.01     0.00  0.53     0.03  0.83 -0.03
## Gloomy       -0.24  0.39  0.44    -0.01  0.53     0.02  0.82 -0.01
## Sluggish      0.07 -0.01  0.73    -0.01  0.50    -0.29  0.34  0.77
## Anxious       0.09  0.70 -0.02     0.00  0.52     0.05  0.79 -0.01
## Bored         0.07 -0.01  0.74    -0.01  0.52     0.04  0.47 -0.04
## Irritated     0.06  0.51 -0.05     0.01  0.58     0.04  0.85 -0.02
## Nervous       0.08  0.73 -0.04     0.00  0.51     0.03  0.74  0.01
## Listless      0.06 -0.05  0.73     0.01  0.54     0.02  0.46 -0.03
##
##
## ----------------------------------------------------------------
##
## Factor correlations after oblique rotation:
##
## S1
##       F1    F2    F3
## F1  1.00 -0.08 -0.24
## F2 -0.08  1.00  0.07
## F3 -0.24  0.07  1.00
##
## S2
##       F1    F2
## F1  1.00 -0.37
## F2 -0.37  1.00
##
## S3
##       F1    F2   F3
## F1  1.00 -0.04 0.23
## F2 -0.04  1.00 0.02
## F3  0.23  0.02 1.00
##
##
## ----------------------------------------------------------------
##
## Intercepts:
##
##                 S1    S2    S3
## Interested   49.24 61.46 51.98
## Joyful       48.92 61.12 49.95
## Determined   46.60 61.20 50.35
## Calm         46.25 61.14 54.76
## Lively       49.29 60.85 50.57
```

```
## Enthusiastic 48.99 61.16 50.24
## Relaxed      49.00 61.12 54.90
## Cheerful     49.03 61.02 50.42
## Content      49.39 60.84 49.98
## Energetic    49.35 60.90 50.41
## Upset        44.12 26.54 36.42
## Gloomy       45.88 27.09 35.93
## Sluggish     44.95 26.54 33.26
## Anxious      45.81 26.48 35.83
## Bored        44.98 26.75 29.94
## Irritated    43.48 26.69 35.66
## Nervous      46.39 26.50 35.94
## Listless     45.35 26.84 29.67
##
## ------------------------------------------------------------
##
## Unique variances:
##
##                  S1    S2     S3
## Interested   273.26 53.37  96.43
## Joyful       273.82 48.67  92.81
## Determined   261.88 49.93  92.70
## Calm         265.99 51.75  99.17
## Lively       286.09 48.20 104.04
## Enthusiastic 257.06 50.09 107.07
## Relaxed      270.54 49.55  99.75
## Cheerful     284.69 50.47  83.05
## Content      271.52 41.15  92.38
## Energetic    271.12 53.24  95.55
## Upset        278.71 46.13  92.89
## Gloomy       256.03 46.46  73.85
## Sluggish     245.57 51.70  82.24
## Anxious      276.61 45.65  87.14
## Bored        253.52 47.69 103.11
## Irritated    267.30 44.56  84.99
## Nervous      261.57 49.30  86.21
## Listless     269.07 47.29  92.10
```

Intercepts. We first look at the intercepts because, for this data example, it allows us to give labels to the states that can be used throughout the interpretation of the other parameters. Specifically, we see that, in all three states, the positive emotions (i.e., “interested”, “joyful”, “determined”, etc.) are larger than the negative emotions (i.e., “upset”, “gloomy”, “sluggish”, etc.). However, the states differ regarding their overall positive and negative emotions. More specifically, the positive emotions are lowest in state 1 and the negative emotions are highest in

state 1, followed by state 3 and then state 2. Therefore, in the following, we label the first state the "displeasure" state, the second one the "pleasure" state, and the third one the "neutral" state.

Loadings. Next, we inspect the loadings. Note that the default output displays standardized[4] obliquely rotated factor loadings.[5] The reason is that unstandardized values can be difficult to interpret as they often exceed an absolute value of 1 (especially when a large rating scale is used like in our example dataset; Section 2.2) and, hence, rules of thumb to evaluate which items have strong loadings on a factor cannot be applied. In contrast, for standardized loadings, rules of thumb are available (e.g., loadings with an absolute value larger than or equal to 0.3 can be seen as considerable, which is also the threshold used in our example).

Looking at the loadings, we see that, in all states, the first factors correspond to a positive affect (PA) factor containing loadings of most or all positive emotion items. However, in the displeasure state (i.e., state 1), the loadings of the items "determined" and "calm" are somewhat lower and, in the neutral state (i.e., state 3), the loadings "calm" and "relaxed" are even lower than the chosen threshold. Furthermore, the second factors are, broadly speaking, negative emotion factors but with even more prominent differences across states (especially between the displeasure and the pleasure state; i.e., state 1 and state 2). While the pleasure state has a clear negative affect (NA) factor with high loadings of all negative emotions, the displeasure state has a bipolar "distress" factor with loadings of the high arousal negative emotions and a reversed loading of the item "calm". The second factor in the neutral state (i.e., state 3) lies in between the factors of the pleasure and displeasure states in that it has considerable loadings of all the items but relatively low loadings of the low arousal emotions "sluggish", "bored", and "listless". The most striking difference is that the displeasure state (i.e., state 1) contains a third bipolar "drive" factor, whereas the neutral state

[4] More specifically, they are standardized by means of the standard deviations of the item scores across all states.
[5] If desired, however, the user can also request unstandardized and unrotated loadings.

(i.e., state 3) contains a third "serenity" factor. More specifically, the drive factor (or rather lack-of-drive factor) has high loadings of the low arousal negative emotions "gloomy", "sluggish", "bored" and "listless" and a reversed loading of the item "determined". The serenity factor has high loadings of the low arousal emotions "calm", "relaxed", and "sluggish". In conclusion, when subjects are in the displeasure or neutral state, they have a more differentiated representation of their emotions than when they are in the pleasure state. The drive factor in the displeasure state is especially interesting because it is in line with research showing that drive differs from general PA when persons are anhedonic (Berridge, Robinson, & Aldridge, 2009; Treadway & Zald, 2011). Moreover, it is noticeable that the loadings in the neutral state (i.e., state 3) are generally higher than the loadings in the other states. Thus, considering the overall item variances, the item-factor relations are strongest in the neutral state. Note that overall larger loadings may also be a result of larger factor variances as a consequence of constraining the factor variances to 1 (Section 2.3.1).

In addition to the loadings, it is interesting to also inspect the factor correlations that result from the oblique rotations – which are not part of the MM. First, in the displeasure state (i.e., state 1), we see a small negative correlation between PA and the lack-of-drive factor (i.e., factors 1 and 3). In the neutral state (i.e., state 3), we see a small positive correlation between PA and the serenity factor (i.e., factors 1 and 3). In the pleasure state (i.e., state 2), PA and NA (i.e., factors 1 and 2) are moderately negatively correlated. All other correlations are close to zero, indicating that the other factors are rather independent of each other.

Unique variances. Finally, looking at the unique variances, we see that they are largest in the displeasure state (i.e., state 1), followed by the neutral and the pleasure state (i.e., state 3 and 2). The large emotion-specific variability in the displeasure state is in line with findings that depression and emotional complexity are related (Grühn, Lumley, Diehl, & Labouvie-Vief, 2013).

2.3.2. The transition model

After examining the MMs, the next step is to investigate what the transitions between the MMs look like by means of the CT-LMM. As previously stated, the CT-LMM is a latent class model that allows subjects to transition between latent states over time. Specifically, we will inspect the probability to start in a state at the first time-point (i.e., "initial state probabilities") and the probabilities to transition to other states from one time point to the next (i.e., "transition probabilities"). In this section, we first explain the initial state parameters and show what they look like for our example data. Then we describe the transition parameters and show the corresponding results for our example.

Initial state parameters. The initial state probabilities pertain to the probability to start in a certain state at the first time-point. The probabilities sum to one and are stored in a vector with the number of elements being equal to the number of states. For example, the vector $\boldsymbol{\pi} = (.42 \quad .34 \quad .24)$ shows that the probability to start in state 1 is equal to $.42$, the probability to start in state 2 is $.34$, and the probability to start in state 3 is $.24$. In *lmfa*, logit models are used to model the initial state probabilities (similar as in logistic regression; Agresti, 1990). The inherent logit values (or "log-odds") indicate the relative chance to start in a state compared to a reference state (in *lmfa*, this is state 1). Note that a separate logit model is required for all states but the reference state. These logit values do not have to be interpreted because the initial state probabilities can be calculated from these logit models (see Supplementary Material S.2.1).[6]

Finally, the initial state parameters may be related to covariates, which could be, for instance, scores on a baseline questionnaire (e.g., a depression score or a score for the general ability to differentiate between emotions). Note, however, that including covariates on the initial state

[6] Note that *lmfa* users do not have to calculate any probabilities themselves as the package provides them.

parameters only makes sense if the dataset contains data of more than a few subjects.[7] Otherwise, there is not enough information to investigate the covariate effects. The covariates are related to the initial state parameters by means of regression and they affect the logits and not the probabilities directly (for details, see Supplementary Material S.2.1). However, in order to see what effect the covariates have on the initial state probabilities, one can convert logits into probabilities for different covariate values and compare them. For example, for a categorical covariate with two categories, one could compare the initial state probabilities for both categories. For continuous covariates, one could compare the initial state probabilities corresponding to the sample mean plus the standard deviation of the covariate to the probabilities corresponding to the sample mean minus the standard deviation (or compare probabilities for different quantiles of the covariate) while setting other covariates equal to their (sample) means.

Initial state parameters for the example data. To illustrate the interpretation, consider the results of a LMFA without covariates on the initial state parameters. The results are presented in LMFA output box 2.

LMFA output box 2

```
## Model estimation:
##
## Estimation converged after 1159.95 seconds.
##
## LL = -3712.4847
##
## ------------------------------------------------------------
##
## Wald tests:
##
##                   Wald df p-value
## intervention  213.3821  6       0
## negativeEvent  55.7629  6       0
##
## ------------------------------------------------------------
##
## Parameter estimates:
```

[7] For an estimate of the required sample size, users may consult guidelines for multinomial logistic regression (e.g., de Jong et al., 2019).

```
##
##                               coef   s.e. z-value p-value
## initial state parameters 2 -0.1864 0.2408 -0.7741  0.4389
## initial state parameters 3 -0.5479 0.2949 -1.8577  0.0632
## transition parameters 1|2  -1.1725 0.3207 -3.6555  0.0003
## transition parameters 1|3  -1.5951 0.4235 -3.7661  0.0002
## transition parameters 2|1  -1.6011 0.4260 -3.7588  0.0002
## transition parameters 2|3  -0.4188 0.4183 -1.0013  0.3167
## transition parameters 3|1  -0.5761 0.2975 -1.9368  0.0528
## transition parameters 3|2  -1.3186 0.7413 -1.7788  0.0753
## intervention 1|2            0.6000 0.1786  3.3604  0.0008
## intervention 1|3            0.3228 0.2565  1.2582  0.2083
## intervention 2|1           -0.9528 0.2624 -3.6308  0.0003
## intervention 2|3           -0.4081 0.2474 -1.6497  0.0990
## intervention 3|1           -1.0119 0.1809 -5.5946  0.0000
## intervention 3|2            0.4767 0.4931  0.9668  0.3336
## negativeEvent 1|2          -0.0194 0.0057 -3.3903  0.0007
## negativeEvent 1|3          -0.0096 0.0081 -1.1931  0.2328
## negativeEvent 2|1           0.0153 0.0091  1.6812  0.0927
## negativeEvent 2|3          -0.0071 0.0079 -0.9001  0.3681
## negativeEvent 3|1           0.0104 0.0060  1.7364  0.0825
## negativeEvent 3|2          -0.0142 0.0132 -1.0753  0.2823
##
## Note: For the initial state parameters, state 1 is the
## reference category. The transition intensity parameters
## are sorted by rows of the transition matrix and the
## staying rates serve as references.
##
## ------------------------------------------------------------
##
## Probabilities:
##
## 1. Initial state probabilities:
##
## (no covariates defined)
##
##     S1     S2     S3
## 0.4153 0.3446 0.2401
##
## 2. Transition probabilities:
##
## interval length: 1
## intervention score: 0.4139
## negativeEvent score: 49.6505
##
##        S1     S2     S3
## S1 0.7923 0.1032 0.1046
## S2 0.2542 0.5388 0.2070
## S3 0.3909 0.1073 0.5019
##
##
## Note: The probabilities are calculated for covariate scores
## equal to the sample means (and a unit time interval). Use
## the function probabilities() to calculate initial state and
## transition probabilities for any covariate score (and
```

```
## interval) of interest.
##
## ------------------------------------------------------------
##
## State proportions:
##
##     S1     S2     S3
## 0.5517 0.2567 0.1916
```

For now, we focus only on the parts "`Parameter estimates`" and "`Initial state probabilities`", starting with the former. The "`coef`" and "`s.e.`" columns indicate the point estimates and standard errors, respectively. The "`z-value`" and "`p-value`" columns show the corresponding z-statistic and p-values. Since no covariates were included for the initial state parameters, there are only two "`initial state parameters`". These parameters always correspond to the logit values for covariate scores being equal to zero. In case of covariate effects, they would be shown below the initial state parameters. As previously described, it is more convenient to interpret the corresponding initial state probabilities. More specifically, in order to obtain a good impression of what the probabilities look like for the average person, it makes most sense to inspect the initial state probabilities for covariates being equal to the sample means. These probabilities can be found in the "`Initial state probabilities`" part further below in LMFA output box 2. Of course, if no covariates are defined (as in our model), the probabilities do not depend on the values of a covariate. The probabilities indicate that starting in the displeasure state was most likely, followed by the pleasure state and the neutral state.

Transition parameters. The transition probabilities are stored in a matrix with dimensions equal to the number of states and the elements within a row of the transition probability matrix sum to one (Bartolucci et al., 2015; Zucchini et al., 2016). In order to clarify how to read a transition probability matrix, consider the following matrix:

$$\mathbf{P} = \begin{pmatrix} p_{11} = .66 & p_{12} = .18 & p_{13} = .16 \\ p_{21} = .20 & p_{22} = .49 & p_{23} = .31 \\ p_{31} = .32 & p_{32} = .17 & p_{33} = .51 \end{pmatrix}. \quad (1)$$

The rows indicate the state memberships at the previous time-point and the columns indicate the state memberships at the current time-point. This implies that the values on the diagonal specify the probabilities to stay in a state and the off-diagonal elements refer to the probabilities to transition to another state. For example, the first row of the matrix shows that the probability to stay in state 1 is equal to $.66$ and the probabilities to transition from state 1 to state 2 and from state 1 to state 3 are equal to $.18$ and $.16$, respectively.

As described before, the transition probabilities depend on the interval between two consecutive measurement occasions. The larger the interval, the larger the probabilities to transition to another state. To accommodate the interval length, LMFA (using CT-LMM) does not estimate the transition probabilities directly. Instead, transition intensities (or "rates"; i.e., transition probabilities per very small time unit)[8] are estimated and the transition probabilities are computed based on the transition intensities and the intervals (Böckenholt, 2005; Jackson & Sharples, 2002).[9] The transition intensities are also captured in a matrix with dimensions equal to the number of states. However, intensities are only estimated for the transitions away from the origin state and, hence, for the off-diagonal entries. The diagonal entries are equal to the sum of the off-diagonal transition intensities, which implies that rows sum to zero (Cox & Miller, 1965). For example, consider the matrix that corresponds to the transition probabilities in Equation (1):

$$\mathbf{Q} = \begin{pmatrix} -q_{12} - q_{13} = -.51 & q_{12} = .31 & q_{13} = .20 \\ q_{21} = .20 & -q_{21} - q_{23} = -.86 & q_{23} = .66 \\ q_{31} = .56 & q_{32} = .28 & -q_{31} - q_{32} = -.84 \end{pmatrix}. \quad (2)$$

The rate to transition from state 1 to state 2 is $q_{12} = .31$ and the rate to transition from state 1 to state 3 is $q_{13} = .20$. Larger rates are related to larger transition probabilities away from a state.

[8] For readers familiar with survival models, note that the intensities are actually equivalent to hazard rates (Cox & Miller, 1965; Kalbfleisch & Lawless, 1985; Kleinbaum & Klein, 2012).

[9] More specifically, the probabilities are equal to the matrix exponential of the product of the intensities and the interval (see Supplementary Material S.2.1).

The transition intensities are modeled by means of a log-linear model such that the parameters are not intensities but log intensities (thus, the parameterization differs from the logit parameterization of the initial state parameters). For example, the estimates for the log intensities corresponding to intensities for the first row in Equation (2) are $\log(q_{12} = .31) = -1.17$ and $\log(q_{13} = .20) = -1.60$. Intensities can be obtained from the log intensities by exponentiation (e.g., $e^{-1.60} = .20$).

Finally, like the initial state parameters, the transition parameters may be related to covariates, which may be either time-constant, such as scores from baseline questionnaires, or time-varying[10], such as the negative event scores and the intervention that some subjects receive during their participation in our example data. The covariates are related to the transition parameters by means of regression (as is the case for the initial state parameters; for details, see Supplementary Material S.2.1). Because the parameters of the transition model are log intensities, the regression effects have to be exponentiated to obtain the effects of the covariates on the transition intensities. However, it is more convenient to interpret the covariate effects on the transition probabilities. To this end, one can convert the intensities into probabilities for a certain interval length and different covariate values and compare them (as for the initial state probabilities).

Transition parameters for the example data. To illustrate the interpretation of the transition parameters, consider again the LMFA results for our data, which are presented in LMFA output box 2. Note that we included the two covariates "had an intervention" and "negative event" on the transition parameters. This time, we focus on the parts "`Parameter estimates`" and

[10] Note that, for time-varying covariates, the score at time-point t is used to predict the transition probabilities from time-point $t-1$ to time-point t. This makes most sense for LMFA because MMs are typically triggered by momentary circumstances (e.g., social interactions). Even when assuming temporal precedence (e.g., the effect of perceiving a negative event prior to time-point t), questionnaires are usually designed in a way that they ask subjects about such covariates at the current time-point (e.g., "Please rate the unpleasantness of the most unpleasant event you have experienced since the previous measurement occasion").

"`Transition probabilities`". The "`transition parameters`" in the "`Parameter estimates`" part correspond to the log intensities for covariate scores being equal to zero. However, for a better interpretability, we inspect the corresponding transition probabilities for a unit time interval for the average person in the sample, and thus, for covariate scores being equal to their sample means. These probabilities are displayed in the "`Transition probabilities`" part in LMFA output box 2. We can see that the sample mean for "had an intervention" is equal to .41 and the sample mean for "negative event" is equal to 49.65. The probabilities indicate that the probabilities to transition to another state are generally lower than to stay in a state, especially when staying in the displeasure state (i.e., state 1). The transition probabilities from the displeasure state (i.e., state 1) to the pleasure and the neutral state (i.e., state 2 and 3) are approximately equal. The transition probability from the pleasure state (i.e., state 2) to the neutral state (i.e. state 3) is smaller than from the pleasure to the displeasure state (i.e., state 1). Finally, the transition probabilities from the neutral state (i.e., state 3) to the displeasure state (i.e., state 1) are larger than to the pleasure state (i.e., state 2).

Next, we investigate the covariate effects on the transition probability matrix (for a unit interval). To this end, we make two comparisons. First, we keep the "negative event" score equal to the sample mean and compare the probabilities for both categories of "had an intervention". The probabilities pertaining to the "no intervention observations" are shown in LMFA output box 3.

LMFA output box 3

```
## 1. Initial state probabilities:
##
## (no covariates defined)
##
##   S1   S2   S3
## 0.42 0.34 0.24
##
## 2. Transition probabilities:
##
## interval length: 1
## intervention score: 0
```

```
## negativeEvent score: 49.65
##
##      S1   S2   S3
## S1 0.84 0.07 0.08
## S2 0.37 0.44 0.19
## S3 0.54 0.08 0.39
```

The probabilities for the "intervention observations" are displayed in LMFA output box 4.

LMFA output box 4

```
## 1. Initial state probabilities:
##
## (no covariates defined)
##
##   S1   S2   S3
## 0.42 0.34 0.24
##
## 2. Transition probabilities:
##
## interval length: 1
## intervention score: 1
## negativeEvent score: 49.65
##
##      S1   S2   S3
## S1 0.71 0.16 0.13
## S2 0.14 0.66 0.20
## S3 0.23 0.15 0.61
```

Comparing the transition probabilities, we see that having had an intervention is related to relatively smaller probabilities to transitioning to and staying in the displeasure state (i.e., state 1).

Second, we compare the transition probabilities for the sample mean of "negative event" minus the standard deviation (i.e., 34.54) to the transition probabilities for the sample mean of the covariate plus the standard deviation (i.e., 64.76), thereby keeping the "had an intervention" score equal to the sample mean. The probabilities for a "negative event" score of 34.54 are displayed in LMFA output box 5.

LMFA output box 5

```
## 1. Initial state probabilities:
##
## (no covariates defined)
##
##   S1   S2   S3
## 0.42 0.34 0.24
##
## 2. Transition probabilities:
```

```
##
## interval length: 1
## intervention score: 0.41
## negativeEvent score: 34.54
##
##      S1   S2   S3
## S1 0.74 0.14 0.13
## S2 0.21 0.55 0.24
## S3 0.33 0.13 0.53
```

The probabilities for a “negative event” score of 64.76 are shown in LMFA output box 6.

LMFA output box 6

```
## 1. Initial state probabilities:
##
## (no covariates defined)
##
##   S1   S2   S3
## 0.42 0.34 0.24
##
## 2. Transition probabilities:
##
## interval length: 1
## intervention score: 0.41
## negativeEvent score: 64.76
##
##      S1   S2   S3
## S1 0.84 0.08 0.09
## S2 0.31 0.51 0.18
## S3 0.45 0.08 0.46
```

Comparing the probabilities, we see that higher scores on “negative event” are related to larger probabilities of transitioning to and staying in the displeasure state (i.e., state 1).

2.3.3. Summary of the LMFA findings for our example data

Summarized, based on the finding of LMFA, we conclude the following. First, the number and nature of the factors changed, which implies that configural invariance is violated for our example data. Second, subjects transitioned rather frequently between the states. However, transitioning to and staying in the displeasure state (i.e., state 1) was most likely, especially when experiencing negative events. However, the probabilities for transitioning to and staying in the neutral or pleasure state (i.e., state 3 and state 2) increased after receiving an intervention.

2.3.4. Estimation

In *lmfa*, the maximum likelihood (ML) parameter estimates are obtained by means of the 3S estimation (Vogelsmeier, Vermunt, Bülow, et al., 2019), which builds on Vermunt (2010)'s ML method and its extension for DT-LMM by Di Mari, Oberski, and Vermunt (2016). The 3S estimation separates the estimation of the state-specific MMs and the CT-LMM as follows:

1. The state-specific MMs are estimated while disregarding the transitions between the latent states at consecutive measurement occasions and the covariate effects on these transitions.
2. Each observation is assigned to the state with the highest state-membership probability, that is, "modal state assignment" is applied.[11] Furthermore, the inherent classification uncertainty is calculated. Note that there is always uncertainty unless all observations are assigned to a state with a probability of 1.
3. The MMs (i.e., the factor parameters) are kept fixed and the state assignments from step 2 are used as single indicators for the estimation of the CT-LMM (with covariates), while correcting for step 2's assignment uncertainty. This correction is necessary to prevent underestimation of the relations between the states (i.e., the transition probabilities) and the covariate effects. Also note that the final state assignments will slightly differ from the step 2 state assignments (for details, see Supplementary Material S.4 and S.5). Usually, the assignments improve because the step 3 estimation benefits from additional information from the transition model (with covariates) to classify the observations (Vogelsmeier, Vermunt, Bülow, et al., 2019).

For technical details about the steps, their likelihood functions, and the algorithms to maximize them, see Supplementary Material S.3–S.5.

[11] Note that it is theoretically also possible to use a "proportional assignment", which assigns the state memberships according to the posterior state-membership probabilities. However, the proportional assignment is unfeasible for data that contain a large number of measurement occasions for many subjects (Di Mari et al., 2016) and, thus, for ILD.

3. How to Conduct LMFA with the *lmfa* Package

In the following, we guide the readers through the different steps of conducting LMFA in the package *lmfa*. These steps are based on the three estimation steps described in Section 2.3.4: Step 1 is investigating the MMs, step 2 is obtaining the state assignments and classification errors, and step 3 is investigating the transition model. Note that we introduce an additional step 0, which pertains to checking the data requirements prior to performing LMFA. Furthermore, as mentioned in Section 2.3, the best model complexity in terms of the number of states and factors is not known in advance and has to be evaluated in step 1. Additionally, depending on the subsequent analyses to investigate dynamics in psychological construct, researchers require factor scores corresponding to the state-specific MMs. Therefore, step 1 is divided into selecting the number of states and factors (step 1a), interpreting the MMs (step 1b), and attaching factor scores to the dataset (step 1c). Moreover, one has to decide which covariates should be included in the final transition model. Therefore, step 3 is divided into selecting covariates (step 3a) and interpreting the transition model and updating the final state assignments (step 3b). Figure 1 provides a summary of the steps with references to the required *lmfa* functions.[12] In the following, we describe the steps and functions by means of our example data introduced in Section 2.2. To follow the steps of this tutorial, the *lmfa* package and the example data have to be loaded into R. Before using the package for the first time, it has to be installed once, which can be done using the following command:

```
install.packages("devtools")
library("devtools")
install_github("LeonieVm/lmfa@0.1.0")
library("lmfa")
```

[Insert Figure 1 about here]

[12] Note that the package consists of only six functions in total (next to the general `summary()` and `plot()` functions). An overview of these functions is provided in the Appendix.

Note that the package *devtools* is required to install packages from the GitHub repository. The dataset used in this tutorial can be loaded into the R environment with the command:

```
data("ESM")
```

3.1. Step 0: Checking data requirements

The first step is to check the data requirements with regard to the format (Section 3.1.1) and missing values (Section 3.1.2).

3.1.1. Data format

In line with the assumed data structure, the data has to be in long format, that is, with rows equal to the number of total observations. Furthermore, next to the columns with the indicators of the latent factors (in our case, `"Interested", "Joyful", "Determined",...`) and possibly covariates (in our case, `"intervention"`, `"negativeEvent"`), the data must contain a column with the subject identification numbers (in our case, `"id"`). Moreover, if observations should not be treated as equidistant, a column must be specified with the time intervals between two consecutive observations (in our case, `"deltaT"`). Regarding the latter, a proper unit should be used. For instance, if there is approximately only one observation per day, the unit "days" is appropriate (e.g., with an interval of 1.42 days representing an interval of one day and 10 hours). If there are several observations per day, say nine, "hours" is an appropriate unit. With "minutes" or "seconds" as unit, the intervals for these examples would take large values that likely lead to numerical problems when estimating the model.[13] Furthermore, measurement occasions within subjects must be ordered by time (i.e., intervals must not be negative). Additionally, for obtaining valid results, intervals for consecutive observations within a subject must not be equal to zero.

[13] The reason is that the value of the transition intensities (for covariate scores being equal to zero) are directly related to the size of the unit (note that the model fit is not influenced by the unit, however). More specifically, the larger the size of the unit, the smaller the intensities and the more likely numerical problems occur.

Zero and negative intervals may occur from technical errors during data collection and should be removed (otherwise, an error message is displayed).

3.1.2. Missing data

The data should only include the records for the measurement occasions at which the subjects completed the questionnaires because the CT-LMM automatically accounts for differences in the intervals, including skipped measurement occasions. Note that, depending on the data collection software, it may happen that a subject started a questionnaire but did not finish it such that some indicators or covariates contain missing values. These cases must be imputed (e.g., by means of the *mice* package in R; van Buuren & Groothuis-Oudshoorn, 2011) or removed before running *lmfa* (otherwise an error message is displayed).

3.2. Step 1a: Selecting the number of states and factors

When estimating a LMFA model, the number of underlying states and factors per state has to be specified. However, in an exploratory approach like LMFA, the best model complexity is not known in advance and has to be determined by estimating a number of plausible models and comparing their results in terms of fit and parsimony. To this end, one can use criteria that balance the loglikelihood and number of parameters such as the Bayesian information criterion (BIC; Schwarz, 1978) and the convex hull (CHull; Ceulemans & Kiers, 2006) method (Bulteel, Wilderjans, Tuerlinckx, & Ceulemans, 2013; Vogelsmeier, Vermunt, van Roekel, et al., 2019). In the following, we first describe the two criteria (Section 3.2.1). Then, we explain how to decide what range of states and factors to include in the model selection procedure (Section 3.2.2). Thereafter, we describe how to increase the chance of finding the "global" maximum and how to assess convergence of the estimation procedure (Section 3.2.3). Finally, we show how to perform the model selection with *lmfa* (Section 3.2.4).

3.2.1. BIC and CHull

First, the BIC considers model fit and model complexity of the model by penalizing models with more parameters (see Supplementary Material S.6.2). Second, the CHull is a generalized scree test (Bulteel et al., 2013; Ceulemans & Kiers, 2006) that automatically identifies models at the higher boundary of the "convex hull" (or CHull) in a "loglikelihood vs. number of parameters" plot (Cattell, 1966) and that chooses the best model by finding the point (or "elbow") in this scree plot (or CHull plot) at which improvement in fit levels off when adding additional parameters to the model. Detecting this elbow is done by means of comparing "scree ratios" (see Supplementary Material S.6.3) for all models on the upper boundary and the model with the largest ratio is chosen. In this way, the CHull also balances complexity and parsimony.

Both the BIC and the CHull offer valuable information about which model should be selected. However, for many real datasets, it is possible that the BIC keeps increasing when adding additional states and/or factors to the model (Bauer, 2007; McNeish & Harring, 2017). Then, investigating the relative improvement in the loglikelihood value by means of the CHull is especially important. Additionally, the CHull does not make distributional assumptions and may therefore perform better for many empirical datasets. The CHull method, however, has two drawbacks that should be accounted for. First, the least and the most complex models at the higher boundary of the CHull cannot be chosen because no scree ratios can be computed (see Bulteel et al., 2013). Therefore, it is always advisable to also inspect the CHull plot visually (e.g., the most complex model might still fit considerably better than the preceding model on the hull). The *lmfa* package will remind the user of this by displaying a note. Second, for some cases, it is possible that the scree ratio is artificially inflated, even though the more complex model does not add much in terms of the fit. Specifically, when adding additional parameters hardly increases the fit

anymore, both the numerator and denominator of the scree test ratio (Supplementary Material S.6.3) approach zero which results in a very large scree test ratio, whereas the hull is pretty much a straight and horizontal line at that point (for a detailed explanation, see Wilderjans, Ceulemans, & Meers, 2013). The *lmfa* package displays a note if there are signs of artificial inflation. When the note is displayed, the user should inspect the CHull plot visually and also consider the next best model(s). Finally, it is best practice to look at the results of competing models and take the interpretability into account.

3.2.2. Range of states and factors

For the model selection, one must decide on the range of states and factors to be considered. Regarding the former, one may start with a few states (say, 1–3). If models with three states barely improve model fit (i.e., according to the BIC and CHull) or if the estimation of three states already causes estimation problems, there is no point in adding more states. Otherwise, one may increase the number of states. Moreover, the maximum number of states is restricted by the number of observations (i.e., one should have at least 1000 observations for each state; Vogelsmeier, Vermunt, van Roekel, et al., 2019). For instance, for our example dataset (with 4776 observations), we should not include more than four states. In order to decide on the number of factors, one should think about theoretically plausible factor structures and consider that each factor should ideally be measured by at least three items. Otherwise, the factors may not be well measured or "determined", which may cause convergence problems, Heywood cases (Van Driel, 1978) or less reliable parameter estimates. For example, if the data consist of six indicators of which three are intended to measure PA and three are intended to measure NA, no more than two factors should be included. Additionally, similarly to the number of states, one should begin with a small number of factors and examine the increase in fit and convergence problems for the most complex factor structure.

3.2.3. Increasing the chance to find the global maximum and assessing convergence

For estimating the state-specific FA models, the algorithm searches for the maximum of the loglikelihood function (Supplementary Material S.3), that is, the solution with the largest loglikelihood value. However, it is possible that the solution is not a "global" maximum but a "local" one. To clarify this, consider the loglikelihood function as a landscape with multiple hills. Each hill has its own local maximum (i.e., the top) but only one hill is the highest and thus has the global maximum. To start searching for a global or local maximum, the algorithm requires initial parameter values. Different start values may lead to finding different (local) maxima (comparable to searching for the highest hill starting from different locations in the landscape). Therefore, it is essential for the algorithm to use multiple start sets with different initial values and, in the end, provide the solution with the best loglikelihood value (Supplementary Material S.3.5). Users should choose at least 25 start sets but the larger the number of start sets, the more likely it is to obtain the solution pertaining to the global maximum.

Moreover, it is possible that the model estimation does not converge at all. This means that the algorithm did not find a (local or global) maximum in a pre-specified number of maximum iterations (Supplementary Material S.3). Especially for more complex models, it is possible that the algorithm requires more iterations to achieve convergence. However, it could also be a sign that the model is not suited for the data at hand (e.g., too many factors). The user may decide to re-estimate corresponding models once, allowing for more iterations, before continuing with the model selection procedure. The *lmfa* package displays this advice as a reminder.

3.2.4. Model selection with *lmfa*

In order to select the "best" model among the models with different numbers of states and factors, we have to use *lmfa*'s `step1()` function. In the following, we compare models with one

to four states and one to three factors per state (i.e., 14 models in total). Note that the order of factors does not matter because they result in the same fit and estimates. For instance, model [3 3 2] is the same as model [3 2 3] and model [2 3 3]. In the *lmfa* package, the permutation of the states of the estimated models is always determined based on the size of the states, starting with the largest. The function can be used as follows (because the estimations start from random state-membership assignments (see Supplementary Material S.3.5), we set a seed for reproducibility):

```
set.seed(1000)
modelselection <- step1(data = ESM,
                        indicators = c(
                          "Interested",
                          "Joyful",
                          "Determined",
                          "Calm",
                          "Lively",
                          "Enthusiastic",
                          "Relaxed",
                          "Cheerful",
                          "Content",
                          "Energetic",
                          "Upset",
                          "Gloomy",
                          "Sluggish",
                          "Anxious",
                          "Bored",
                          "Irritated",
                          "Nervous",
                          "Listless"),
                        modelselection = TRUE,
                        n_state_range = 1:4,
                        n_fact_range = 2:3,
                        n_starts = 25,
                        max_iterations = 1000)
```

There are five mandatory arguments that we have to specify. First, we have to provide the data via the `data` argument (in our case, `ESM`). Second, via the `indicators` argument, we specify the variable names of the indicators in the same order as they appear in the data. These are `c("Interested", "Joyful", "Determined",...)`. Third, we indicate that we want to perform model selection via the argument `modelselection` (i.e., `modelselection = TRUE`). Fourth and

fifth, we determine the range of states and factors that should be included in the model selection with `n_state_range = 1:4` and `n_fact_range = 2:3`. Additionally, we could change the default values for the number of start sets and the number of maximum iterations after which the estimation terminates regardless of whether convergence has been reached or not, but we simply use the default values `n_starts = 25` and `max_iterations = 1000`.[14]

When the estimation is terminated, we obtain the model-selection results as follows:

```
summary(modelselection)
```

Note that the model selection for our example data took about 3 hours. In order to follow the next tutorial steps in R, readers can simply load the model selection object with the command: `data("modelselection")`. The output is displayed in LMFA output box 7.

LMFA output box 7

```
##                LL      BIC convergence n_par
## [323]  -353166.8 708485.3           1   254
## [333]  -353149.0 708602.3           1   272
## [3322] -353071.7 708913.6           1   327
## [3233] -353065.6 709053.8           1   345
## [3333] -353016.6 709108.4           0   363
## [3222] -353316.0 709249.7           1   309
## [322]  -353855.3 709709.8           1   236
## [33]   -354421.0 710375.3           1   181
## [2222] -353962.6 710390.5           1   291
## [32]   -355010.3 711401.4           1   163
## [222]  -354986.3 711819.4           1   218
## [22]   -356377.4 713983.1           1   145
## [3]    -361759.6 724281.6           1    90
## [2]    -363744.0 728098.0           1    72
##
## Note: When re-estimating models that did not converge, the
##    number of maximum iterations should be increased.
```

The first column (i.e., "`LL`") pertains to the loglikelihood value. The second column (i.e., "`BIC`") shows the value of the BIC. The third column (i.e., "`convergence`") indicates whether the model estimation converged (with 1 = "convergence" and 0 = "non-convergence"). The fourth column

[14] Note that the function contains additional arguments related to the estimation procedure for which default values are provided. These values may be changed by the user if desired. For an explanation of the additional arguments, see Section A.1 and the function documentation, which can be called with `?step1`.

(i.e., "n_par") shows the total number of parameters.[15] The models are ordered by the value of the BIC, starting with the lowest value and thus the model with the best fit according to this criterion.

Before continuing with the model selection, we check if models have to be re-estimated due to non-convergence. Indeed, the estimation of model [3 3 3 3] did not converge. For estimating single models, we use the `step1()` function but without model selection (i.e., with `modelselection = FALSE`). The code to estimate model [3 3 3 3] is:

```
set.seed(1000)
model3333 <- step1(data = ESM,
                   indicators = c(
                     "Interested",
                     "Joyful",
                     "Determined",
                     "Calm",
                     "Lively",
                     "Enthusiastic",
                     "Relaxed",
                     "Cheerful",
                     "Content",
                     "Energetic",
                     "Upset",
                     "Gloomy",
                     "Sluggish",
                     "Anxious",
                     "Bored",
                     "Irritated",
                     "Nervous",
                     "Listless"),
                   modelselection = FALSE,
                   n_state = 4,
                   n_fact = c(3,3,3,3),
                   n_starts = 25,
                   max_iterations = 2000)
```

When `modelselection = FALSE`, it is mandatory to provide a single number of states via the argument `n_state` (i.e., `n_state = 4`) and a vector with state-specific numbers of factors via the argument `n_fact` (i.e., `n_fact = c(3,3,3,3)`). As previously described, when re-estimating

[15] Note that the number of parameters are equal to the sum of the state-specific intercepts, unique variances, and loadings and the state proportions minus 1 (minus 1 because one state is treated as a reference state; see Supplementary Material S.6.1).

models that did initially not converge, it is wise to increase the number of maximum iterations. Therefore, we set `max_iterations = 2000`. In order to replace the old by the new models, the following command can be used:

```
modelselection$`[3333]` <- model3333
```

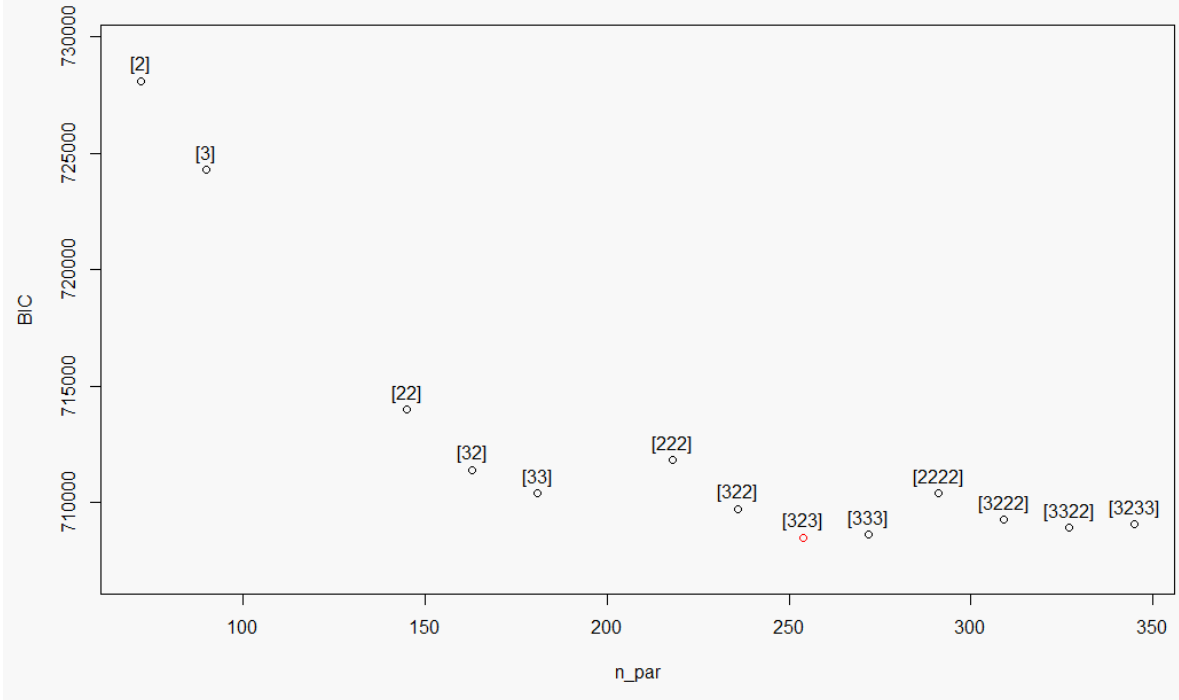
However, the model did not converge (it might simply not be suitable for the data) and, therefore, we continue with the original model selection object.

From the summary in LMFA output box 7, we can already see that the best model according to the BIC is model [3 2 3] and, thus, the true model. For an easier inspection of the results, we also plot the BIC of the converged models against the number of free parameters. The plot is readily available by running the following command:

```
plot(modelselection)
```

The output is shown in LMFA output box 8.

LMFA output box 8

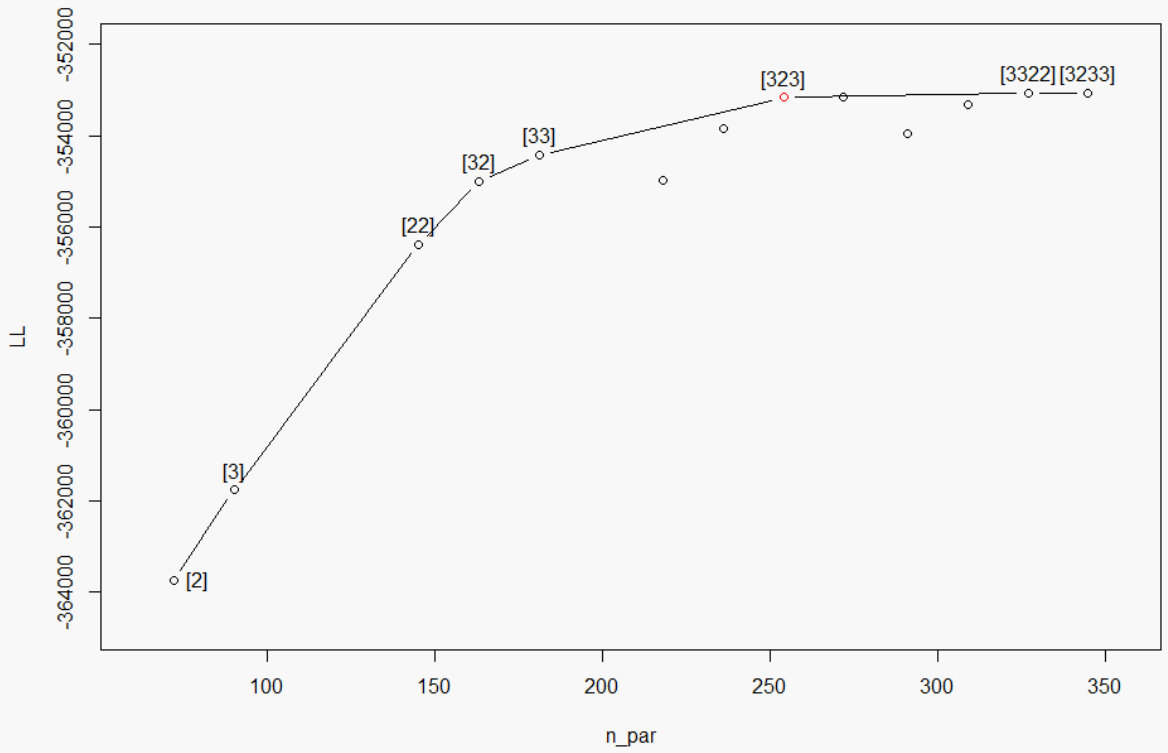


The model corresponding to the lowest BIC value is indicated by a red dot. Note that, for our example, the BIC does not keep increasing for more complex models. Therefore, we would consider it relatively save to choose the model with the lowest BIC value. However, in order to support our choice, we also investigate the results of the CHull method for the converged models, which can be obtained with the `chull_lmfa()` function as follows:

```
chull_lmfa(x = modelselection)
```

We only have to specify argument `x`, which pertains to the model-selection object (in our case, `modelselection`).[16] The output is shown in LMFA output box 9.

LMFA output box 9




```
## Models on the upper boundary of the CHull:
##
##        n_par        LL        st
## [2]       72 -363744.0        NA
## [3]       90 -361759.6  1.126582
## [22]     145 -356377.4  1.288490
```

[16] For details about the function, see Section A.4 or call the documentation file with `?chull_lmfa`.

```
## [32]     163 -355010.3  2.319779
## [33]     181 -354421.0  1.905623
## [323]    254 -353166.8 13.191076
## [3322]   327 -353071.7  3.812062
## [3233]   345 -353065.6        NA
##
## Selected model(s):
##
##       n_par        LL
## [323]   254 -353166.8
##
## Note 1: The least and most complex models cannot be selected.
##   Therefore, it is advisable to also visually inspect the CHull plot.
##
## Note 2: The st value(s) of the best model(s) might be artificially
##   inflated. Therefore, it is advisable to also consider the next best model(s).
```

The output consists of three parts, the CHull plot, the summary of the models on the upper boundary of the CHull (including their scree-test values "`st`"), and the selected model(s). We can see that the model [3 2 3] was selected. However, we got the note that the scree value might be artificially inflated. Therefore, in practice, one should also consider the results of the next best model(s). Because we know the true model, we continue with model [3 2 3].[17] In order to inspect the model, we have to extract it from the model-selection object `modelselection` and store it as follows:

```
measurementmodel323 <- modelselection$`[323]`
```

The parameters can be displayed with the command:[18, 19]

```
summary(measurementmodel323)
```

3.3. Step 1b: Interpreting the measurement models

Once the best model in terms of the number of states and factors is selected, we can interpret

[17] The second best model according to the scree value is model [3 3 2 2]. However, visual inspection of the CHull shows that adding a fourth state does not considerably improve the fit. The next best model is model [3 2]. If we would inspect model [3 2], we would see that the smallest state (i.e., the neutral state) would be divided into the pleasure and the displeasure state.

[18] Note that, in the `summary()` function, the user can specify an additional argument to change the number of decimals to which the parameters should be rounded. The default for the summary of the MM parameters is `rounding = 2`.

[19] Note that the model object (i.e., `measurementmodel323`) contains additional information that is not directly relevant for the interpretation but that may be interesting for some users (e.g., unrotated and unstandardized loadings). For an overview of all available output, see Section A.1.

the state-specific MMs. The output (obtained with `summary(measurementmodel323)`) was already displayed in LMFA output box 1 and the results were already interpreted in Section 2.3.1. To briefly summarize them, we found three states (a displeasure, a neutral, and a pleasure state) that all contained a positive affect and a negative affect (or distress) factor but the displeasure state was additionally characterized by a drive factor and the neutral state by a serenity factor.

3.4. Step 1c: Attach factor scores to the dataset

Before proceeding with step 2, we can attach state-specific factor scores to our dataset for each observation in the dataset.[20] The factor scores are estimates of the latent constructs and can be used for subsequent analyses to investigate dynamics in psychological constructs (for suggestions on how to proceed in the presence of non-invariance, see Section 4). A copy of the dataset with the factor scores attached can be obtained with:[21]

```
ESM_fs <- factorscores_lmfa(data = ESM, model = measurementmodel323)
```

In this function, two arguments are required. First, via the argument `data`, we have to provide the data that was used for the `step1()` estimation (in our case, `ESM`). Second, via the argument `model`, we have to specify the `step1()` object with the state-specific MMs (in our case, `measurementmodel323`). In the resulting dataset (i.e., `ESM_fs`), the columns are called "`S1F1`", "`S1F2`", etc., where the "`S`" refers to the state and "`F`" to the factor.

3.5. Step 2: Obtaining state assignments & classification errors

[20] In *lmfa*, the factor scores are calculated by means of the regression method (Thomson, 1934; Thurstone, 1935), which is one of the most commonly used approaches. For the exact computation, see Supplementary Material S.6.8. Note that the calculation of factor score estimates is generally considered controversial because different methods can result in (very) different scores (which is referred to as the problem of factor score indeterminacy; for discussions on this and possible solutions to account for biases in subsequent analyses, see, e.g., Devlieger, Mayer, & Rosseel, 2016; Green, 1976; Grice, 2001).

[21] Note that, in the `factorscores_lmfa()` function, the user can specify two additional arguments. The first one indicates whether the factor score estimates should be obtained for the obliquely rotated loadings. The default is `oblique = TRUE`. Otherwise, the factor score estimates corresponding to the unrotated factor loadings are obtained. The second argument pertains to the number of decimals to which the factor score estimates should be rounded. The default is `rounding = 4`. For details about the function, see Section A.5 or call the documentation file with `?factorscores_lmfa`.

The next step is to obtain information about the classification and the (modal) state assignments. In this section, we first describe how to obtain the results for our example data with *lmfa* and then, based on the output, we explain the different classification statistics.

In order to obtain the classification information, we use the `step2()` function as follows:

```
classification <- step2(data = ESM_fs, model = measurementmodel323)
```

The function contains two arguments that we have to specify. First, we have to provide the data that was used for the `step1()` estimation via the argument `data`. It is most conveniently to use the version including the factor scores estimates (in our case, `ESM_fs`) because we will add additional columns later on and this allows us to obtain a complete dataset for further analyses. Second, we need to specify the `step1()` object with the state-specific MMs via the argument `model` (in our case, `measurementmodel323`).[22] The following code prints the results:[23]

```
summary(classification)
```

The output is shown in LMFA output box 10.

LMFA output box 10

```
## R2_entropy: 0.86
##
## Total classification error: 0.05
##
## Classification errors:
##
##         S1      S2     S3
## S1 2568.11    4.12  59.50
## S2    2.71 1146.53  44.37
## S3   50.19   89.35 811.13
##
## Classification-error probabilities:
##
##      S1   S2   S3
## S1 0.98 0.00 0.02
## S2 0.00 0.96 0.04
## S3 0.05 0.09 0.85
##
## State proportions:
```

[22] For an additional explanation of the function arguments, see Section A.2 or call the documentation file with `?step2`.
[23] Again, the user can adjust the number of decimals to which the parameters should be rounded. The default for the summary of the classification outputs is `rounding = 2`.

```
##
##   S1   S2   S3
## 0.55 0.26 0.19
```

First, the R-square measure $R^2_{entropy}$ (called "`R2_entropy`" in the output) indicates how well the states are separated (and thus how much the MMs differ) with values ranging from 0 (bad separation) to 1 (good separation; Lukočienė, Varriale, & Vermunt, 2010). Note that a larger state separation implies less classification error. It is important to inspect the $R^2_{entropy}$ value because a bad state separation (with $R^2_{entropy} < .5$) can lead to an incorrect classification-error correction[24] and, in turn, to an underestimation of transition probabilities and the covariate effects (Vermunt, 2010). When observing a bad state separation, which is rather unlikely in practice, it is advised to use the FIML estimation, which is currently only available in Latent GOLD. The $R^2_{entropy}$value for our example data indicates that the states are well separated, which explains the small total classification error (called "`Total classification error`" in the output).

Second, information about the classification errors can be obtained from the classification error matrix (called "`Classification errors`" in the output), which cross-classifies the modal state assignments by the "true" state assignments and which is used to correct for the error in step 3 of the analysis (for details, see Supplementary Material S.4). Higher values on the diagonal and lower values on the off-diagonal are indicative of less classification error. For an easier interpretation, the counts can be translated into proportions (called "`Classification-error probabilities`" in the output). Inspecting the classification error matrices, we see that the classification error is lowest in the displeasure state (i.e., state 1), followed by the pleasure state (i.e., state 2) and the neutral state (i.e., state 3). Thus, the classification into the neutral state was accompanied by the

[24] The reason is that the $R^2_{entropy}$ tends to be overestimated for bad state separations (e.g., if we find a value of .4, the real state separation is probably even lower). In turn, the classification errors are underestimated, leading to an incorrect correction in the final step of the analysis (Vermunt, 2010).

most uncertainty, which is not surprising, given that the neutral state lies somewhat in between the displeasure and pleasure state.

Third, the state proportions (also called like that in the output; i.e., "`State proportions`") pertain to the overall state sizes. Looking at the state proportions for our data, we see that the displeasure state is the largest, followed by the pleasure and the neutral state.

Finally, the state assignments are not displayed in the output because R cannot display all assignments simultaneously. However, we can simply obtain a copy of our dataset with additional columns corresponding to the state assignments with the following command:

```
ESM_fs_cl <- classification$data
```

Specifically, the columns with the posterior state probabilities (called "`State1`", "`State2`" etc. in the dataset) indicate the probabilities for an observation to belong to a certain state and thus, that the state-specific MM underlies the responses for this observation. As explained in Section 2.3.4, the modal state assignments (called "`Modal`" in the dataset) correspond to the state with the largest probability and, hence, to the most likely state membership.

3.6. Step 3a: Selecting the covariates for the transition model

When the state-specific MMs are obtained and the observations are assigned to the states, we can continue with investigating the transitions between the states and what may cause them by means of estimating a LMFA with covariates on the initial state and/or transition parameters. In order to test if a covariate is significantly related to the transition model parameters (and, thus, whether it should be included in the model) can be evaluated with Wald tests (Agresti, 1990). In the following, we first explain the covariate selection with the Wald tests (Section 3.6.1) and then show how to perform the covariate selection for our example data with *lmfa* (Section 3.6.2).

3.6.1. Covariate selection procedure using Wald tests

Every covariate in the model is accompanied by separate covariate effects on the initial state or transition parameters (e.g., the covariate "had an intervention" has six effects, one on each of the transition parameters). The Wald tests in *lmfa* are omnibus tests that show whether including a covariate is significant overall (i.e., across the initial and transition parameters). Thus, for every covariate, there is one Wald-test statistic. In order to select which covariates to include, one can start with a LMFA with all covariate candidates in it. Then, the least significant covariate is removed and the model gets re-estimated. This "backward selection" continues until only significant covariates are left (say, according to an alpha level of .05). When only significant covariates are left in the model, one can continue with the interpretation of covariate effects on the transition probabilities (as we did for our example in Section 2.3.2). Note that, instead of using such a data-driven approach, a more theory-driven approach is also possible (e.g., investigating the significance and effects of a set of covariates that was previously found to be significantly related to the transition model parameters).

3.6.2. Covariate selection with *lmfa*

In the following, we estimate a transition model with covariate effects of "had an intervention" and "negative event" on the transition parameters. In order to estimate the transition model, we use the `step3()` function as follows (because the estimation starts from random values for the transition parameters (see Supplementary Material S.5.4), we set a seed for reproducibility):

```
set.seed(1000)

transitionmodel <- step3(data = ESM_fs,
                         identifier = "id",
                         n_state = 3,
                         postprobs =
                           classification$classification_posteriors[,-1],
                         timeintervals  = "deltaT",
                         initialCovariates = NULL,
```

```
                        transitionCovariates =
                          c("intervention", "negativeEvent"),
                        n_starts = 25,
                        max_iterations = 1000)
```

There are four mandatory arguments that we have to specify. First, we provide the data via the `data` argument. We use the dataset `ESM_fs` and, thus, the data including the factor scores but without the state assignments from step 2 because they are updated in step 3. Second, we specify the name of the column with the subject identification numbers via the argument `identifier` (in our case, `"id"`). Third, we define the number of states with `n_state = 3`. Fourth, we specify the posterior state probabilities by means of the argument `postprobs`. The probabilities can be extracted from the `step2()` classification output with the command `classification$classification_posteriors[,-1]`, where `[,-1]` indicates that we leave out the column with the modal state assignments.

The following three arguments are not required, but have to be specified if the model should account for differences in intervals and if covariate effects on the transition model parameters should be included. Both applies to our example. Thus, first, via the argument `timeintervals`, we provide the function with the name of the column in the dataset that contains the time intervals. In our case, this is `"deltaT"` (if no such column name is provided, observations would be assumed to be equidistant). Next, via the arguments `transitionCovariates` and `initialCovariates`, we can provide (a vector of) column names that contain the covariate scores (the default for both arguments is `NULL`, i.e., no covariates are used). Thus, for our analysis, we provide the vector `c("intervention", "negativeEvent")` as input for `transitionCovariates`.

Finally, similarly to the `step1()` function, the users may decide to change the default values

pertaining to the number of start sets[25] via the argument `n_starts` and the number of maximum iterations via the argument `max_iterations`[26]. However, for our analysis, we simply use the default values, that is, `n_starts = 25` and `max_iterations = 1000`.[27] After termination of the estimation, the results are obtained as follows:[28]

```
summary(transitionmodel)
```

The estimation for our example data took about 20 minutes. Again, readers who want to follow the rest of the tutorial can also load the results with the command: `data("transitionmodel")`. The results are shown in the "`Wald tests`" part in LMFA output box 2. For each covariate, we get a significance test with the corresponding Wald-test statistic (i.e., "`Wald`"), the degrees of freedom (i.e., "`df`"), and the p-value (i.e., "`p-value`"). We see that both covariates have significant effects on the transition parameters. Thus, we keep both covariates in the model.

3.7. Step 3 b: Interpreting the transition model & updating the state assignments

After selecting the covariates for the transition model, we can interpret the effects on the probabilities, investigate changes in the final state proportions, and obtain the final state assignments. In the following, we show how to obtain the relevant output in *lmfa* and how to interpret the results for our example data. First, as previously shown, the regression parameters and the probabilities for covariates being equal to their sample means (and a unit interval) can be obtained with `summary(transitionmodel)`. In order to obtain the initial state and transition

[25] Note that the results of the `step3()` function are sensitive to start values for the transition intensities (see Supplementary Material S.5.4). Therefore, one should use at least 25 start sets.
[26] If the maximum number of iterations is reached without convergence, *lmfa* displays a note with the advice to repeat the estimation with an increased number of `max_iterations`.
[27] Note that the user may change the defaults of additional arguments pertaining to the estimation procedure. For an explanation of these arguments, see Section A.3 and the function documentation that can be called with `?step3`.
[28] Again, the user can change the number of decimals to which the parameters should be rounded. The default for the summary of the transition model is `rounding = 4`. Thus, by default, there are two more decimals than for the other representations. This is because some parameters can get very small such that differences would vanish too easily when using less decimal points.

probabilities for any covariate score and interval of interest, we can use the `probabilities()` function. For example, in order to obtain the probabilities for a "had an intervention" score equal to zero, a "negative event" score equal to the sample mean of 49.65, and a unit interval, we use the following command:

```
probabilities(model = transitionmodel,
              deltaT = 1,
              initialCovariateScores = NULL,
              transitionCovariateScores = c(0, 49.65))
```

Only the first argument is mandatory, that is, we have to provide the output of the `step3()` function (in our case, `transitionmodel`) via the `model` argument. By default, the function prints the probabilities for a unit interval and covariate scores equal to the sample means (i.e., `deltaT = 1`, `initialCovariateScores = NULL` and `transitionCovariateScores = NULL`). In order to print the probabilities for specific covariate scores, we have to provide a vector with these scores in the same order as we included the covariates in the estimation of the transition model with the `step3()` function. In our case, we include `transitionCovariateScores = c(0,49.65)`.[29, 30] The results for the transition model with covariates were interpreted in Section 2.3.2. To briefly summarize them, first, most subjects started in the displeasure state. Second, the probabilities to stay in a state were generally higher than to transition to another state (especially for subjects in the displeasure state). Third, the intervention was related to transitions to a more positive state and experiencing negative events was related to transitioning to the more negative state.

Next, the final state proportions are shown under "`State proportions`" in the same output as the transition model parameters (see LMFA output box 2). Comparing the results to the state

[29] An additional explanation of the arguments can be found in Section A.6 and in the function documentation, which can be called with `?probabilities`.
[30] The user can determine an additional argument in the `probabilities()` function to change the number of decimals to which the parameters should be rounded. The default for the probabilities is `rounding = 2`.

proportions in LMFA output box 10 (and thus to the state proportions resulting from the modal state assignment in step 2), we see that the there was no change. This is not surprising considering the small classification errors from step 2.

Third, we can again obtain a copy of our dataset with the posterior state-membership probabilities and the modal state assignment attached. To this end, we use the following command:

```
ESM_fs_cl <- transitionmodel$data
```

These are the state assignments that should be considered for subsequent data analyses because, as described in Section 2.3.4, they may be more accurate than the step 2 assignments. Therefore, we simply overwrite the previous dataset with the step 2 assignments `ESM_fs_cl`.

4. Proceeding Based on the Results of LMFA

Once the MM differences and possible explanations are known, the question is of course: Based on the LMFA results, how to proceed with (originally planned) analyses to investigate the dynamics in psychological constructs? The answer to this question largely depends on the findings. It is important to note that a comparison of the state-specific MMs may indicate violations of different levels of invariance and that the required level of invariance depends on the type of comparisons one wants to make. When comparing the state-specific loadings, one may find that the MMs differ a lot across states – specifically, in the number and/or nature (the zero-loading pattern) of measured constructs – which indicates a violation of configural invariance. It may also be that the pattern of (near-)zero loadings appears to be equal across states, but that the non-zero loadings differ in size. This suggests that configural invariance holds but weak invariance fails. When configural or weak non-invariance is indicated, continuing with analyses that assume invariance is not possible for the full dataset because factor scores are not validly comparable. Differences in the means of the constructs or relations between constructs could be due to underlying differences in the MMs. Finding such differences in MMs is interesting in its own right

(e.g., the additional drive factor for anhedonic subjects in our data example), however. In any case, it is possible to proceed with factor scores from one specific state (e.g., the largest state or the state that best corresponds to an a priori assumed MM) and, thus, with observations for which strict invariance holds.[31]

If weak invariance holds across the states – that is, if the (near-)zero and non-zero loadings are highly similar across states – users may examine whether covariances (e.g., regression coefficients or autocorrelations) between latent constructs (e.g., positive affect and negative affect) differ across subjects and/or change across time, because factor covariances are not affected by intercept differences (Oberski, 2017; Steenkamp & Baumgartner, 1998). However, examining whether mean construct scores differ across subjects and time-points calls for strong invariance to avoid mixing up differences in latent means and intercept differences (Meredith & Teresi, 2006). This implies that strict invariance is not necessary for meaningfully comparing latent covariances or means (Putnick & Bornstein, 2016; Vandenberg & Lance, 2000). Thus, finding states that differ in the unique variances only does not preclude latent variable comparisons. Note that it is best to allow for non-invariances of intercepts or unique variances (as indicated by LMFA) as much as possible in your follow-up analysis, ideally by including states.[32] Otherwise, the latent means and/or covariances may be estimated incorrectly, especially in case of large non-invariances (Chen, 2008; Guenole & Brown, 2014). Alternatively, one could perform one analysis per state (using the state-specific factor scores) and weight the observations according to the posterior state-

[31] When proceeding with factor scores from one state, it is important to investigate the certainty of the final state assignments. For instance, the largest posterior state probabilities for some observations might be rather low (say, below .6 or .7), indicating that it is less clear which of the state-specific MMs best fits the scores of the observation. Therefore, it is advisable to remove these observations before conducting further analyses.

[32] Note that this is possible with more advanced analyses like dynamic latent class analysis (DLCA; Asparouhov, Hamaker, & Muthén, 2016). Moreover, in some analyses, it is at least possible to allow for MM differences across subjects like in dynamic structural equation modeling (DSEM; McNeish & Hamaker, 2020; McNeish, Mackinnon, Marsch, & Poldrack, 2021).

membership probabilities such that observations with larger probabilities receive more weight than observations with lower probabilities. Another option could be to conduct a weighted multilevel analysis, in which the states would be considered as observed groups. Furthermore, if "partial" metric or strong invariance holds (i.e., if only a few loadings or intercepts differ; Byrne, Shavelson, & Muthén, 1989), one may exclude non-invariant items or, again, capture the differences by letting parameters differ across states in subsequent analyses or dealing with it by conducting separate analyses with weighted data. Moreover, in order to avoid non-invariance in future studies, one could consider removing the problematic items from the questionnaire or rephrase them. To conclude, LMFA can be viewed as a primary analysis step that indicates which observations are actually comparable, what the MMs look like, and that, in turn, facilitates decisions about how to further analyze the ILD.

5. Discussion

When studying dynamics in psychological constructs in intensive longitudinal data (ILD), it is crucial to investigate whether the measurement models (MMs) underlying the responses are invariant across subjects and time, which is easily violated due to between-person differences and/or situation-specific changes in item interpretation and response styles. Undetected measurement non-invariance poses a threat to valid inference from state-of-the-art ILD analyses. In this tutorial, we showed how to explore which MMs underlie the data, what transitions between these MMs look like, and how to investigate if covariates are related to such transitions by means of latent Markov factor analysis (LMFA). Most importantly, the method indicates which observations are actually comparable by classifying them into the same MM-state, which helps to safeguard valid inferences. Moreover, researchers gain substantive insights into the dynamics of the underlying MM in their ILD.

The package *lmfa* was implemented in the open source software R to provide researchers with a freely available software option to perform LMFA. Even though this is a huge advantage, it is important to stress that some features are currently not (yet) available. In the following, we will elaborate on current limitations of the package and ideas for future extensions.[33] Firstly, the state-specific MMs in step 1 of the analysis are currently obtained by means of exploratory factor analysis (EFA). As previously explained, EFA is less restrictive than confirmatory factor analysis (CFA), which implies that it allows to detect all types of non-invariance in the loadings. However, for some datasets, it is certainly interesting to use a CFA model – thus, with fixed patterns of zero factor loadings. For example, researchers may want to rely on LMFA results or other results from previous research showing that the configural factor structure is rather stable across subjects and time. Therefore, in the future, *lmfa* will include an option to perform CFA within the states.

Secondly, the factor analysis models in step 1 assume continuous item responses. If items are measured with only a few categories or if the item responses are heavily skewed, state-specific "latent trait" (or "item response theory") models should be employed in step 1 of the analysis to adequately deal with categorical data, as is done in the extension called latent Markov latent trait analysis (LMLTA; Vogelsmeier et al., 2020). Performing LMLTA is currently only possible in Latent GOLD, but advanced R users could theoretically specify their own state-specific models for step 1 (for instance, by using mixture models for categorical data from other packages) and use the posterior state-membership probabilities as input[34] for the step 3 analysis with *lmfa*. However, suitable packages are, to the best of our knowledge, currently not available in R. If a package would become available, possibilities to include it in the *lmfa* package (in order to perform

[33] Note that the commercial software Latent GOLD offers most of the features discussed below.
[34] As described in Section 3.6.1, this would be done by providing a data frame with posterior state-membership probabilities per state via the argument `postprobs`.

LMLTA) will be examined.

Thirdly, prior to performing step 1 in *lmfa*, users have to remove or impute records that contain missing values on some of the indicators, that is, for measurement occasions that were not completely skipped (note that completely omitted measurement occasions are dealt with by the continuous-time approach). Generally, technological advances in many experience sampling methodology apps prevent subjects from submitting incomplete responses. However, sometimes researchers rather have incomplete data than to lose the measurement occasion entirely. Furthermore, missing data may be a result of the increasingly employed "planned missing-data designs", in which researchers deliberately assess only selected items at each measurement occasions while omitting others in order to reduce the burden on the subjects, which, in turn, tends to increase the quality of the responses (Silvia, Kwapil, Walsh, & Myin-Germeys, 2014; van Roekel, Keijsers, & Chung, 2019). In the future, *lmfa* will be extended to be applicable for ILD collected with such innovative missing-data designs and missing data in general.

Fourthly, the inclusion of covariates in step 3 of the analysis helps researchers understand why certain subjects transition between MMs over time, but some researchers might be more (or also) interested in individual transition patterns, especially in case of only a few subjects. Estimating subject-specific transition parameters is currently not possible in *lmfa*. However, one can estimate one transition model per subject. More specifically, step 1 and 2 (i.e., estimating the MMs and obtaining the state assignments and classification errors) would still be conducted for all subjects[35] but step 3 would be performed for each subject individually. Additionally, instead of inspecting subject-specific transition parameters, it might be interesting to investigate whether unobserved subgroups of subjects have similar transition patterns, especially in case of many subjects.

[35] One may also perform the entire LMFA for data of a single subject if the number of observations is large enough. For guidelines about the required number of observations, see Vogelsmeier, Vermunt, van Roekel, et al. (2019).

Theoretically, it is possible to cluster subjects based on their transition behavior by adding a latent grouping variable to the LMFA in step 3 (e.g., see Crayen, Eid, Lischetzke, & Vermunt, 2017; Vogelsmeier et al., 2020). This is not possible with *lmfa* but advanced R users may consider using the *depmix* package in step 3 of LMFA by passing the modal state assignments and classification error probabilities as fixed parameters to the `depmix()` function. This package allows for a latent grouping variable in the transition model but uses a discrete-time latent Markov model and, hence, does not account for differences in intervals, which may impair the estimation of the transition model parameters when intervals are unequal (for details about the syntax and about how to fix parameters, see the package documentation; Visser, 2007).

Lastly, *lmfa* users currently have to draw conclusions about (non-)invariance by visually comparing the state-specific MMs. If the number and nature of the factors appear to be the same across states, determining which parameters differ substantially becomes a daunting task, especially when comparing parameters for models with many states and factors. Furthermore, one will always find small parameter differences across states due to sampling fluctuations and error fitting. Deciding which differences are practically or statistically significant is not a trivial problem. On top of that, the states also capture differences in the factor variances (in the loadings) and factor means (in the intercepts) due to the model identification constraints (see Section 2.3.1). To obtain loadings that are optimally comparable across states and to enable hypothesis testing for these loadings (using Wald tests), multigroup factor rotation (MGFR; De Roover & Vermunt, 2019) should be applied. MGFR solves rotational freedom by rotating the loadings towards a simple structure within the states and towards agreement across states while unraveling differences in the loadings from differences in the factor variances. MGFR is currently only available in Latent GOLD but possibilities to include it in *lmfa* will be investigated in the future. Similarly, a solution

for optimally comparing intercepts (with hypothesis tests) could be to employ multiple group factor alignment (MGFA; Asparouhov & Muthén, 2014), in which the factors are rescaled and shifted (or "aligned") with respect to their means, thereby disentangling differences in the intercepts from differences in the factor means. However, currently, MGFA is only applicable to CFA models without cross-loadings. If an MGFA extension for EFA would become available, possibilities to include the method in *lmfa* will be examined. Until MGFR and MGFA are implemented, users can inspect whether there appears to be a difference in the scaling of all loadings of a factor and/or a "shift" in all intercepts of items that correspond to the same factor (as indicated by loadings that are not close to zero). If separate loadings or intercepts differ across states, it is unlikely that these differences are caused by differences in the underlying factor variances or factor means, respectively.

6. Figures

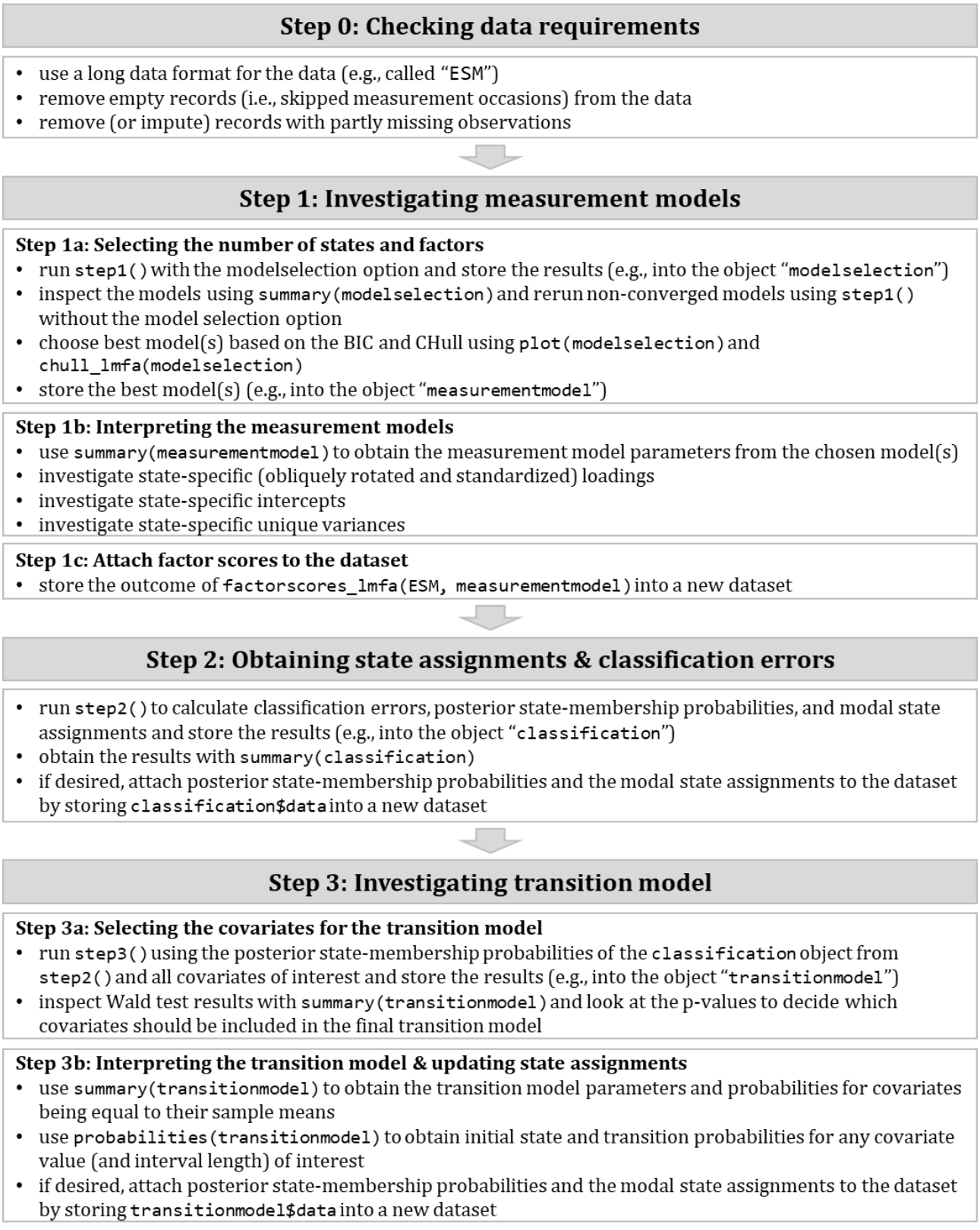


Figure 1. Summary of the three steps to conduct latent Markov factor analysis with *lmfa*.

References

Adolf, J., Schuurman, N. K., Borkenau, P., Borsboom, D., & Dolan, C. V. (2014). Measurement invariance within and between individuals: a distinct problem in testing the equivalence of intra- and inter-individual model structures. *Frontiers in Psychology, 5*, 1–14. doi:10.3389/fpsyg.2014.00883

Agresti, A. (1990). *Categorical Data Analysis*. New York: John Wiley & Sons.

Asparouhov, T., Hamaker, E. L., & Muthén, B. (2016). Dynamic latent class analysis. *Structural Equation Modeling: A Multidisciplinary Journal, 24*, 257–269. doi:10.1080/10705511.2016.1253479

Asparouhov, T., & Muthén, B. (2014). Multiple-group factor analysis alignment. *Structural Equation Modeling: A Multidisciplinary Journal, 21*, 495–508. doi:10.1080/10705511.2014.919210

Barrett, L. F. (1998). Discrete emotions or dimensions? The role of valence focus and arousal focus. *Cognition & Emotion, 12*, 579–599. doi:10.1080/026999398379574

Barrett, L. F., Gross, J., Christensen, T. C., & Benvenuto, M. (2001). Knowing what you're feeling and knowing what to do about it: Mapping the relation between emotion differentiation and emotion regulation. *Cognition & Emotion, 15*, 713–724. doi:10.1080/02699930143000239

Bartolucci, F., Farcomeni, A., & Pennoni, F. (2014). Comments on: Latent Markov models: a review of a general framework for the analysis of longitudinal data with covariates. *Test, 23*, 473–477. doi:10.1007/s11749-014-0387-1

Bartolucci, F., Farcomeni, A., & Pennoni, F. (2015). *Latent Markov Models for Longitudinal Data*. Hoboken, NJ: CRC Press.

Bauer, D. J. (2007). Observations on the use of growth mixture models in psychological research. *Multivariate Behavioral Research, 42*, 757–786. doi:10.1080/00273170701710338

Berridge, K. C., Robinson, T. E., & Aldridge, J. W. (2009). Dissecting components of reward: 'liking', 'wanting', and learning. *Current Opinion in Pharmacology, 9*, 65–73. doi:10.1016/j.coph.2008.12.014

Böckenholt, U. (2005). A latent Markov model for the analysis of longitudinal data collected in continuous time: States, durations, and transitions. *Psychological Methods, 10*, 65–83. doi:10.1037/1082-989X.10.1.65

Bringmann, L. F., Vissers, N., Wichers, M., Geschwind, N., Kuppens, P., Peeters, F., . . . Tuerlinckx, F. (2013). A network approach to psychopathology: new insights into clinical longitudinal data. *PLoS One, 8*, e60188. doi:10.1371/journal.pone.0060188

Bulteel, K., Wilderjans, T. F., Tuerlinckx, F., & Ceulemans, E. (2013). CHull as an alternative to AIC and BIC in the context of mixtures of factor analyzers. *Behavior Research Methods, 45*, 782–791. doi:10.3758/s13428-012-0293-y

Byrne, B. M., Shavelson, R. J., & Muthén, B. (1989). Testing for the equivalence of factor covariance and mean structures: The issue of partial measurement invariance. *Psychological Bulletin, 105*, 456–466. doi:10.1037/0033-2909.105.3.456

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research, 1*, 245–276. doi:10.1207/s15327906mbr0102_10

Ceulemans, E., & Kiers, H. A. (2006). Selecting among three-mode principal component models of different types and complexities: a numerical convex hull based method. *British*

Journal of Mathematical and Statistical Psychology, 59, 133–150. doi:10.1348/000711005X64817

Chen, F. F. (2008). What happens if we compare chopsticks with forks? The impact of making inappropriate comparisons in cross-cultural research. *Journal of Personality and Social Psychology, 95*, 1005–1018. doi:10.1037/a0013193

Clarkson, D. B., & Jennrich, R. I. (1988). Quartic rotation criteria and algorithms. *Psychometrika, 53*, 251–259. doi:10.1007/BF02294136

Collins, L. M., & Lanza, S. T. (2010). *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences*. New York: Wiley.

Cox, D. R., & Miller, H. D. (1965). *The Theory of Stochastic Process*. London: Chapman & Hall.

Crayen, C., Eid, M., Lischetzke, T., & Vermunt, J. K. (2017). A continuous-time mixture latent-state-trait Markov model for experience sampling data. *European Journal of Psychological Assessment, 33*, 296–311. doi:10.1027/1015-5759/a000418

de Jong, V. M. T., Eijkemans, M. J. C., van Calster, B., Timmerman, D., Moons, K. G. M., Steyerberg, E. W., & van Smeden, M. (2019). Sample size considerations and predictive performance of multinomial logistic prediction models. *Statistics in Medicine, 38*, 1601–1619. doi:10.1002/sim.8063

De Roover, K., & Vermunt, J. K. (2019). On the exploratory road to unraveling factor loading non-invariance: A new multigroup rotation approach. *Structural Equation Modeling: A Multidisciplinary Journal, 26*, 905–923. doi:10.1080/10705511.2019.1590778

Devlieger, I., Mayer, A., & Rosseel, Y. (2016). Hypothesis testing using factor score regression: A comparison of four methods. *Educational and Psychological Measurement, 76*, 741–770. doi:10.1177/0013164415607618

Di Mari, R., Oberski, D. L., & Vermunt, J. K. (2016). Bias-adjusted three-step latent Markov modeling with covariates. *Structural Equation Modeling: A Multidisciplinary Journal, 23*, 649–660. doi:10.1080/10705511.2016.1191015

Dolan, C. V. (1994). Factor analysis of variables with 2, 3, 5 and 7 response categories: A comparison of categorical variable estimators using simulated data. *British Journal of Mathematical and Statistical Psychology, 47*, 309–326.

Dolan, C. V., Oort, F. J., Stoel, R. D., & Wicherts, J. M. (2009). Testing measurement invariance in the target rotated multigroup exploratory factor model. *Structural Equation Modeling: A Multidisciplinary Journal, 16*, 295–314. doi:10.1080/10705510902751416

Erbas, Y., Ceulemans, E., Kalokerinos, E., Houben, M., Koval, P., Pe, M. L., & Kuppens, P. (2018). Why I don't always know what I'm feeling: The role of stress in within-person fluctuations in emotion differentiation. *Journal of Personality and Social Psychology, 115*, 179–191.

Erbas, Y., Ceulemans, E., Koval, P., & Kuppens, P. (2015). The role of valence focus and appraisal overlap in emotion differentiation. *Emotion, 15*, 373–382. doi:10.1037/emo0000039

Erbas, Y., Kalokerinos, E., Kuppens, P., van Halem, S., & Ceulemans, E. (2020). Momentary Emotion Differentiation: The derivation and validation of a framework to study within-person fluctuations in emotion differentiation. *Assessment*. doi:10.31234/osf.io/v6u7t

Green, B. F. (1976). On the factor score controversy. *Psychometrica, 41*, 263–266. doi:10.1007/BF02291843

Grice, J. W. (2001). Computing and evaluating factor scores. *Psychological Methods, 6*, 430–450. doi:10.1037/1082-989X.6.4.430

Grühn, D., Lumley, M. A., Diehl, M., & Labouvie-Vief, G. (2013). Time-based indicators of emotional complexity: interrelations and correlates. *Emotion, 13*, 226–237. doi:10.1037/a0030363

Guenole, N., & Brown, A. (2014). The consequences of ignoring measurement invariance for path coefficients in structural equation models. *Frontiers in Psychology, 5*, 1–16. doi:10.3389/fpsyg.2014.00980

Hamaker, E. L., Ceulemans, E., Grasman, R. P. P. P., & Tuerlinckx, F. (2015). Modeling affect dynamics: State of the art and future challenges. *Emotion Review, 7*, 316–322. doi:10.1177/1754073915590619

Horstmann, K. T., & Ziegler, M. (2020). Assessing Personality States: What to Consider when Constructing Personality State Measures. *European Journal of Personality*. doi:10.1002/per.2266

Jackson, C. H. (2011). Multi-state models for panel data: The msm package for R. *Journal of Statistical Software, 38*, 1–29.

Jackson, C. H., & Sharples, L. D. (2002). Hidden Markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients. *Statistics in Medicine, 21*, 113–128. doi:10.1002/sim.886

Kalbfleisch, J. D., & Lawless, J. F. (1985). The analysis of panel data under a Markov assumption. *Journal of the American Statistical Association, 80*, 863–871. doi:10.2307/2288545

Kashdan, T. B., Barrett, L. F., & McKnight, P. E. (2015). Unpacking emotion differentiation. *Current Directions in Psychological Science, 24*, 10–16. doi:10.1177/0963721414550708

Kiers, H. A. (1997). Techniques for rotating two or more loading matrices to optimal agreement and simple structure: A comparison and some technical details. *Psychometrika, 62*, 545–568. doi:10.1007/BF02294642

Kleinbaum, D. G., & Klein, M. (2012). *Survival analysis* (3 ed.). New York, NY: Springer.

Lawley, D. N., & Maxwell, A. E. (1962). *Factor analysis as a statistical method*. London, UK: Butterworth.

Lukočienė, O., Varriale, R., & Vermunt, J. K. (2010). The simultaneous decision(s) about the number of lower- and higher-level classes in multilevel latent class analysis. *Sociological Methodology, 40*, 247–283. doi:10.1111/j.1467-9531.2010.01231.x

McLachlan, G. J., & Peel, D. (2000). Mixtures of factor analyzers. In P. Langley (Ed.), *Proceedings of the Seventeenth International Conference on Machine Learning.* (pp. 599–606). San Francisco: Morgan Kaufmann.

McNeish, D., & Hamaker, E. L. (2020). A primer on two-level dynamic structural equation models for intensive longitudinal data in Mplus. *Psychological Methods, 25*, 610–635. doi:10.1037/met0000250

McNeish, D., & Harring, J. R. (2017). The effect of model misspecification on growth mixture model class enumeration. *Journal of Classification, 34*, 223–248. doi:10.1007/s00357-017-92

McNeish, D., Mackinnon, D. P., Marsch, L. A., & Poldrack, R. A. (2021). Measurement in Intensive Longitudinal Data. *Structural Equation Modeling: A Multidisciplinary Journal*, 1–16. doi:10.1080/10705511.2021.1915788

McNicholas, P. D. (2016). *Mixture Model-Based Classification*. Boca Raton: Chapman and Hall/CRC Press.

Meredith, W. (1993). Measurement invariance, factor analysis and factorial invariance. *Psychometrika, 58,* 525–543. doi:10.1007/BF02294825

Meredith, W., & Teresi, J. A. (2006). An essay on measurement and factorial invariance. *Medical Care, 44,* 69–77. doi:10.1097/01.mlr.0000245438.73837.89

Moors, G. (2003). Diagnosing response style behavior by means of a latent-class factor approach. Socio-demographic correlates of gender role attitudes and perceptions of ethnic discrimination reexamined. *Quality and Quantity, 37,* 277–302. doi:10.1023/A:1024472110002

Myin-Germeys, I., Kasanova, Z., Vaessen, T., Vachon, H., Kirtley, O., Viechtbauer, W., & Reininghaus, U. (2018). Experience sampling methodology in mental health research: new insights and technical developments. *World Psychiatry, 17,* 123–132. doi:10.1002/wps.20513

Oberski, D. L. (2017). Evaluating sensitivity of parameters of interest to measurement invariance in latent variable models. *Political Analysis, 22,* 45–60. doi:10.1093/pan/mpt014

Oort, F. J., Visser, M. R., & Sprangers, M. A. (2005). An application of structural equation modeling to detect response shifts and true change in quality of life data from cancer patients undergoing invasive surgery. *Quality of Life Research, 14,* 599–609. doi:10.1007/s11136-004-0831-x

Paulhus, D. L. (1991). Measures of personality and social psychological attitudes. In J. P. Robinson & R. P. Shaver (Eds.), *Measures of Social Psychological Attitudes Series* (Vol. 1, pp. 17–59). San Diego, CA: Academic Press.

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. doi:10.1016/j.dr.2016.06.004

R Core Team. (2020). A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software, 48,* 1–36.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6,* 461–464.

Scollon, C., Kim-Prieto, C., & Diener, E. (2003). Experience sampling: Promises and pitfalls, strengths and weaknesses. *Journal of Happiness Studies, 4,* 5–34. doi:10.1023/A:1023605205115

Silvia, P. J., Kwapil, T. R., Walsh, M. A., & Myin-Germeys, I. (2014). Planned missing-data designs in experience-sampling research: Monte Carlo simulations of efficient designs for assessing within-person constructs. *Behavior Research Methods, 46,* 41–54. doi:10.3758/s13428-013-0353-y

Steenkamp, J. B. E. M., & Baumgartner, H. (1998). Assessing measurement invariance in cross-national consumer research. *Journal of Consumer Research, 25,* 78–107.

Thomson, G. H. (1934). The meaning of i in the estimate of g. *British Journal of Psychology, 25,* 92–99. doi:10.1111/j.2044-8295.1934.tb00728.x

Thurstone, L. L. (1935). *The vectors of mind*. Chicago, IL: University of Chicago Press.

Thurstone, L. L. (1947). *Multiple factor analysis*. Chicago: The University of Chicago Press.

Treadway, M. T., & Zald, D. H. (2011). Reconsidering anhedonia in depression: lessons from translational neuroscience. *Neuroscience & Biobehavioral Reviews, 35*, 537–555. doi:10.1016/j.neubiorev.2010.06.006

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software, 45*, 1–67.

Van Driel, O. P. (1978). On various causes of improper solutions in maximum likelihood factor analysis. *Psychometrika, 43*, 225–243. doi:10.1007/BF02293865

van Roekel, E., Keijsers, L., & Chung, J. M. (2019). A review of current ambulatory assessment studies in adolescent samples and practical recommendations. *Journal of Research on Adolescence, 29*, 560–577. doi:10.1111/jora.12471

Van Roekel, E., Vrijen, C., Heininga, V. E., Masselink, M., Bos, E. H., & Oldehinkel, A. J. (2017). An exploratory randomized controlled trial of personalized lifestyle advice and tandem skydives as a means to reduce anhedonia. *Behavior Therapy, 48*, 76–96. doi:10.1016/j.beth.2016.09.009

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*, 4–70.

Vermunt, J. K. (2010). Latent Class Modeling with Covariates: Two Improved Three-Step Approaches. *Political Analysis, 18*, 450–469. doi:10.1093/pan/mpq025

Vermunt, J. K., & Magidson, J. (2016). *Technical Guide for Latent GOLD 5.1: Basic, Advanced, and Syntax*. Belmont, MA: Statistical Innovations.

Visser, I. (2007). depmix: An R-package for fitting mixture models on mixed multivariate data with Markov dependencies. *R-package manual*.

Vogelsmeier, L. V. D. E., & De Roover, K. (2020). 'lmfa': An R-package for exploring measurement invariance in intensive longitudinal data with continuous-time latent Markov factor analysis.

Vogelsmeier, L. V. D. E., Vermunt, J. K., Bülow, A., & De Roover, K. (2019). *Evaluating covariate effects on ESM measurement model changes with latent Markov factor analysis: A three-step approach*. Preprint. Retrieved from https://doi.org/10.31234/osf.io/6ufrc

Vogelsmeier, L. V. D. E., Vermunt, J. K., Keijsers, L., & De Roover, K. (2020). Latent Markov latent trait analysis for exploring measurement model changes in intensive longitudinal data. *Evaluation & the Health Professions, 44*, 61–76. doi:10.1177/0163278720976762

Vogelsmeier, L. V. D. E., Vermunt, J. K., van Roekel, E., & De Roover, K. (2019). Latent Markov factor analysis for exploring measurement model changes in time-intensive longitudinal studies. *Structural Equation Modeling: A Multidisciplinary Journal, 26*, 557–575. doi:10.1080/10705511.2018.1554445

Wilderjans, T. F., Ceulemans, E., & Meers, K. (2013). CHull: a generic convex-hull-based model selection method. *Behavior Research Methods*, 1–15. doi:10.3758/s13428-012-0238-5

Zucchini, W., MacDonald, I. L., & Langrock, R. (2016). *Hidden Markov Models for Time Series: An Introduction Using R (2nd Edition)*. Boca Raton: Chapman & Hall/CRC.

Appendix

In the following, we summarize the arguments and the output for each function. Note that additional documentation files are available for all functions. These can be called by typing a questionnaire mark followed by the function name (e.g., `?step1`).

A.1. `step1()` function

Arguments

`data`	The dataset with the indicators.
`indicators`	The variable names of the indicators.
`n_state`	The number of states that should be estimated when `modelselection = FALSE`
`n_fact`	The number of factors that should be estimated when `modelselection = FALSE`
`modelselection`	The indication whether model selection should be performed or not. The default is `FALSE`.
`n_state_range`	The range of states that should be estimated when `modelselection = TRUE`.
`n_fact_range`	The range of factors that should be estimated when `modelselection = TRUE`.
`n_starts`	The number of start sets. Multiple start sets are required in order to increase the chances of finding the global maximum (for details, see Supplementary Material S.3.5). The default is 25.
`n_initial_ite`	The number of initial iterations, that is, the number of iterations that is performed for each start set (for an explanation, see Supplementary Material S.3.5). The default is 15.
`n_m_step`	The number of maximization-step iterations inside the implemented expectation maximization algorithm (for details, see Supplementary Material S.3). The default is 10.
`em_tolerance`	The estimation convergence criterion (for details, see Supplementary Material S.3.4). The default is 1e-8.
`m_step_tolerance`	The criterion for stopping the maximization-step iterations. The default is 1e-3. Thus, the maximization-step iterations stop when either `m_step_tolerance` or `n_m_step` has been reached.
`max_iterations`	The maximum number of iterations after which the estimation terminates regardless of whether convergence has been reached or not. The default is 1000 iterations.
`n_mclust`	The number of mclust start sets (for details, see Supplementary Material S.3.5). The default is 5.

Output

`n_it`	The number of iterations.
`seconds`	The time in seconds that was required to reach convergence.
`convergence`	Indicates whether the model estimation converged prior to reaching the maximum number of iterations. A convergence of 1 indicates that the model converged.
`LL`	The value of the loglikelihood.
`BIC`	The value of the BIC.
`intercepts`	The state-specific intercepts.
`loadings_stand_obli`	The state-specific standardized obliquely rotated loadings. If the number of factors is equal to one, the loadings are equal to the ones in `loadings_stand_list`.
`unique_variances`	The state-specific unique variances.
`state_proportions`	The state proportions.
`n_obs`	The total number of observations across all subjects and time-points.
`n_par`	The total number of free parameters (for details, see Supplementary Material S.6.1).
`explained_variance`	The amount of explained variance weighted by the state sizes (for details, see Supplementary Material S.6.4).
`n_state`	The number of states.
`n_fact`	The state-specific number of factors.
`intercepts_list`	List of state-specific intercepts.
`loadings_list`	List of state-specific loadings.
`loadings_stand_list`	List of state-specific standardized loadings.
`loadings_obli_list`	List of state-specific obliquely rotated loadings. If the number of factors is equal to one, the loadings are equal to the ones in `loadings_list`.
`loadings_stand_obli_list`	List of state-specific standardized obliquely rotated loadings. If the number of factors is equal to one, the loadings are equal to the ones in `loadings_stand_list`.
`unique_variances_list`	List of state-specific unique variances.
`factor_correlations_stand_obli_list`	List of state-specific factor correlations resulting from the rotations of the standardized loadings.
`factor_correlations_obli_list`	List of state-specific factor correlations resulting from the rotations of the loadings.
`activated_contraints`	The number of activated constraints (for details, see Supplementary Material S.3.4).
`classification_posteriors`	The posterior state-membership probabilities and the modal state assignments.

`classification_errors`	The classification errors when using the modal state assignment (for details, see Supplementary Material S.4).
`classification_errors_prob`	The classification-error probabilities when using the modal state assignment (for details, see Supplementary Material S.4).
`R2_entropy`	The entropy-based R-squared measure (for details, see Supplementary Material S.6.6)
`warning_loadings`	A message indicating whether convergence for rotating the loadings was reached or not.
`warning_loadings_stand`	A message indicating whether convergence for rotating the standardized loadings was reached or not.
`raw_data`	The data corresponding to the indicator items that were used in the analysis.

A.2. `step2()` function

Arguments

`data`	The dataset used in `step1()`.
`model`	The model estimated with `step1()`.

Output

`classification_posteriors`	The posterior state-membership probabilities and the modal state assignments.
`classification_errors`	The classification errors when using the modal state assignment (for details, see Supplementary Material S.4).
`classification_errors_prob`	The classification-error probabilities when using the modal state assignment (for details, see Supplementary Material S.4).
`R2_entropy`	The entropy-based R-squared measure (for details, see Supplementary Material S.6.6).
`totoal_classification_error`	The total classification error.
`state_proportions`	The state proportions.
`data`	The data with the posterior state-membership probabilities and the modal state assignments attached.

A.3. `step3()` function

Arguments

`data`	The dataset (including the covariate values).
`timeintervals`	The name of the column containing the intervals between measurement occasions. The default is NULL, which means that the measurement occasions are assumed to be equidistant.
`identifier`	The name of the column containing the subject identifiers.
`n_state`	The number of states that was used for the estimation with `step1()`.
`postprobs`	The posterior state-membership probabilities of `step2()`.
`transitionCovariates`	The covariate(s) for the transition intensities. The default is `NULL`, which means that no covariate effects are estimated.
`initialCovariates`	The covariate(s) for the initial state probabilities. The default is `NULL`, which means that no covariate effects are estimated.
`n_starts`	The number of random start sets (for details, see Supplementary Material S.5.4). The default is 25.
`n_initial_ite`	The number of initial iterations that should be performed for each start set. The default is 10.
`method`	The estimation method. The default is `"BFGS"`, which is usually faster and more stable when including covariates. The alternative is `"CG"`.
`max_iterations`	The maximum number of iterations after which the estimation stops regardless of whether convergence has been reached or not. The default is 1000.
`tolerance`	The convergence tolerance (for details, see Supplementary Material S.5.3). The default is 1e-10. When the message occurs that the model has likely not converged because the Hessian is not positive definite, it is advisable to set the argument to a lower value and repeat the analysis (e.g., 1e-16; Jackson, 2011).
`scaling`	A scaling parameter for the loglikelihood that can prevent numerical problems from occurring, which is internally passed to the optimization function `optim()`. An appropriate scale value is close to -2 times the loglikelihood, but the loglikelihood is of course unknown prior to estimating the model. Therefore, by default, *lmfa* uses an approximation, which is based on the loglikelihood value of a CT-LMM without transitions. Next to this default (i.e., `scaling = "proxi"`), it is also possible to specify own scale values.

Output

`seconds`	The time in seconds that was required to reach convergence.
`convergence`	Indicates whether the model estimation converged prior to reaching the maximum number of iterations. A convergence of 1 indicates that the model converged. Note that it is not possible to obtain the number of iterations because this information is not returned by the `optim()` function.
`LL`	The value of the loglikelihood.
`WaldTests`	The Wald test output.
`estimates`	The parameter estimates of the transition model.
`classification_posteriors`	The posterior state-membership probabilities and the modal state assignments.
`state_proportions`	The state proportions.
`n_initialCovariates`	The number of covariates for the initial state probabilities.
`n_transitionCovariates`	The number of covariates for the transition intensities.
`n_initial_covariates`	The number of covariates specified for the initial state parameters.
`transition_covariate_means`	The number of covariates specified for the transition parameters.
`n_state`	The number of states.
`data`	The data with the posterior state-membership probabilities and the modal state assignments attached.

A.4. `chull_lmfa()` function

Arguments

`x`	The model-selection output of the function `step1()`.

Output

Prints the models on the upper boundary of the CHull, the corresponding scree-test values, and the selected model(s).

A.5. `factorscores_lmfa()` function

Arguments

`data`	The dataset used in `step1()`.
`model`	The model estimated with `step1()`.
`oblique`	The indication whether the factor scores should be obtained for the obliquely rotated loadings or unrotated loadings. The default is `TRUE`, indicating that the obliquely rotated loadings are considered.
`rounding`	The number of decimals to which the results should be rounded. The default is 4.

Output

Attached the state-specific factor scores to the dataset.

A.6. `probabilities()` function

Arguments

`model`	The transition-model output of the function `step3()`.
`deltaT`	The interval for which the transition probabilities should be calculated.
`initialCovariateScores`	The covariate scores for which the probabilities should be calculated. The default is NULL, which implies that any scores are set equal to the sample means.
`transitionCovariateScores`	The covariate scores for which the probabilities should be calculated. The default is NULL, which implies that any scores are set equal to the sample means.
`rounding`	The number of decimals to which the results should be rounded. The default is 2.

Output

Prints the initial state and transition probabilities for specified covariate values (and intervals).

Supplementary Material for the Manuscript:

How to explore within-person and between-person measurement model differences in intensive longitudinal data with the R package *lmfa*

Vogelsmeier, Leonie V.D.E, Vermunt, Jeroen K., De Roover, Kim

Tilburg University, The Netherlands

Draft version 1, 11/06/2021. This manuscript is a preprint and has not been peer reviewed. Please do not copy or cite without author's permission.

Online Supplement

In this online supplement, we provide all technical information about latent Markov factor analysis (LMFA) and the three-step (3S) estimation with the R package *lmfa*. In the following, we first introduce relevant data notation (S.1). Then, we explain LMFA (S.2). Thereafter, we describe the three estimation steps (as presented in the main body of the paper) and how they are implemented in *lmfa* (S.3–S.5). Finally, we provide equations for relevant statistics such as the Bayesian information criterion (BIC; S.6).

S.1. Data Notation

The observations are denoted by y_{ijt} with $i = 1, \dots, I$ referring to the subjects, with $j = 1, \dots, J$ indicating the variables, and with $t = 1, \dots, T$,[1] referring to the measurement occasions. The observation are collected in the $J \times 1$ vectors $\mathbf{y}_{it} = \left(y_{i1t}, y_{i2t}, \dots, y_{iJt}\right)'$ that are themselves stored in the $T \times J$ data matrices $\mathbf{Y}_i = (\mathbf{y}_{i1}', \mathbf{y}_{i2}', \dots, \mathbf{y}_{iT}')'$. The covariate scores are denoted by z_{itu} (with $u = 1, \dots, U$ referring to the covariates) and are collected in the $U \times 1$ vectors $\mathbf{z}_{it} = (z_{it1}, z_{it2}, \dots, z_{itU})'$, which are themselves collected in the $T \times J$ matrices $\mathbf{Z}_i = (\mathbf{z}_{i1}', \mathbf{z}_{i2}', \dots, \mathbf{z}_{iT}')$.

S.2. LMFA

The LMFA model consists of a transition model (i.e., the continuous-time (CT-)LMM) and state-specific measurement models (MMs) (i.e., the FA models). The conceptual ideas behind the models were described in the main body of the paper. In the following, we describe the technical details.

S.2.1. CT-LMM

The CT-LMM makes two assumptions. First, the first-order Markov assumption states that the probability to be in state k (with $k = 1, \dots, K$) at time-point t depends only on the state-membership l (with $l = 1, \dots, K$) at time-point $t - 1$. Second, the local independence assumption states that the responses $\mathbf{y}_{it}$ at time-point t depend only on the state-membership k at this time-point. The CT-LMM, for subject i, is defined as follows:

[1] Note that the number of time-points T may differ across subjects but we mostly omit the index i for the sake of simplicity.

$$p(\mathbf{Y}_i, \mathbf{S}_i|\mathbf{Z}_i) = p\big(\mathbf{y}_{i1}, \dots, \mathbf{y}_{iT}, \mathbf{s}_{i1}, \dots, \mathbf{s}_{iT}|\mathbf{z}_{i1}, \dots, \mathbf{z}_{iT}\big)$$
$$= \overbrace{p(\mathbf{s}_{i1}|\mathbf{z}_{i1})}^{\substack{initial\ state\\ probabilities}} \prod_{t=2}^{T} \overbrace{p_{\delta_{ti}}(\mathbf{s}_{it}|\mathbf{s}_{it-1}, \mathbf{z}_{it})}^{\substack{transition\\ probabilities}} \prod_{t=1}^{T} \overbrace{p(\mathbf{y}_{it}|\mathbf{s}_{it})}^{\substack{response\\ probabilities}}. \quad \text{(S1)}$$

The $K \times 1$ vectors $\mathbf{s}_{it} = (s_{it1}, \dots, s_{itK})'$ contain binary indicators that determine the state memberships at time-point t, with $s_{itk} = 1$ for one state k and $s_{itk} = 0$ for all other states. As can be seen in Equation S1, the model contains three types of probabilities:

1. $p(\mathbf{s}_{i1}|\mathbf{z}_{i1})$ is a $K \times 1$ vector with the initial state probabilities that define the probabilities to start in a certain state at the first time-point and thus sum to one.
2. $p_{\delta_{ti}}(\mathbf{s}_{it}|\mathbf{s}_{it-1}, \mathbf{z}_{it})$ is a $K \times K$ matrix $\mathbf{P}_{\delta_{ti}}$ with transition probabilities that define the probabilities to stay in a state or transition to another state at two consecutive measurement occasions with rowsums equal to 1. As indicated by the index δ_{ti}, the transition probabilities are a function of the interval δ_{ti} between two consecutive observations and the $K \times K$ transition intensity matrix $\mathbf{Q}$ with the off-diagonal elements $q_{lk} = \lim_{\delta \to 0} \frac{p(s_{itk}=1|s_{it-\delta,l}=1, \mathbf{z}_{it})}{\delta}$, which define transitions between the origin state l and the destination state k for a very small time unit. The diagonal elements are equal to the negative row sums (Cox & Miller, 1965). The transition probability matrix $\mathbf{P}_{\delta_{ti}}$ is obtained with $\mathbf{Exp}(\mathbf{Q} \times \delta_{ti})$, where $\mathbf{Exp}(\cdot)$ denotes the matrix exponential.
3. $p(\mathbf{y}_{it}|\mathbf{s}_{it})$ determine the state-specific response probabilities and, thus, the probabilities of having a certain response pattern depending on the state-membership at time-point t. In LMFA, the probabilities depend on the state-specific MMs.

As can be seen from Equation (S1), the initial state and transition intensities may depend on the covariates $\mathbf{z}_{it}$. Note that the covariates and the observations at time-point t are assumed to be conditionally independent given the state-membership at that time-point (i.e., the covariates affect only the state-membership and not the indicators directly). The covariates are included by means of regression (Bartolucci, Farcomeni, & Pennoni, 2014; Vermunt, Langeheine, & Böckenholt, 1999). Note that a logit model is used for the initial state probabilities and a log-linear model is used for the transition intensities. Specifically, the initial state probabilities are modelled as:

$$\log \frac{p(s_{i1k} = 1|\mathbf{z}_{i1})}{p(s_{i11} = 1|\mathbf{z}_{i1})} = \beta_{0k} + \boldsymbol{\beta}_k' \mathbf{z}_{it=1}; \text{ for } k = 2, \dots, K, \quad \text{(S2)}$$

with β_{0k} indicating the initial state intercepts and $\boldsymbol{\beta}'_k = \left(\beta_{k,z_{i11}}, \ldots, \beta_{k,z_{i1U}}\right)'$ indicating the slopes (i.e., the covariate effects). For the transition intensities, the model is

$$\log q_{lk} = \gamma_{0lk} + \boldsymbol{\gamma}'_{lk}\mathbf{z}_{it}; \text{ for } k \neq \text{l}, \tag{S3}$$

with γ_{0lk} as transition intercepts and $\boldsymbol{\gamma}'_{lk} = \left(\gamma_{lk,z_{it1}}, \ldots, \gamma_{lk,z_{itU}}\right)$ as slopes.

S.2.2. State-specific FA models

The state-specific FA models determine what the MMs look like. The state-specific FA model for subject i being in state k at time-point t (i.e., $s_{itk} = 1$) is:

$$[\mathbf{y}_{it}|s_{itk} = 1] = \boldsymbol{\nu}_k + \boldsymbol{\Lambda}_k \mathbf{f}_{it} + \mathbf{e}_{it}, \tag{S4}$$

where $\boldsymbol{\nu}_{\text{k}}$ is a state-specific $J \times 1$ intercept vector; $\boldsymbol{\Lambda}_k$ is a state-specific $J \times F_k$ loading matrix, where F_k indicates the number of factors for state *k*; $\mathbf{f}_{it} \sim MVN(\mathbf{0}, \boldsymbol{\Phi}_k)$ is a subject-specific and time-point-specific $F_k \times 1$ vector of factor scores with $\boldsymbol{\Phi}_k$ being the state-specific factor covariance matrix. Furthermore, $\mathbf{e}_{it} \sim MVN(\mathbf{0}, \boldsymbol{\Psi}_k)$ is a subject- and time-point-specific $J \times 1$ vector of residuals, where $\boldsymbol{\Psi}_k$ contains the unique variances ψ_{kj} on the diagonal and zeros on the off-diagonal. This implies that the response probabilities, $p(\mathbf{y}_{it}|\mathbf{s}_{it})$ in Equation S1 are determined by state-specific multivariate normal distributions with covariance matrices $\boldsymbol{\Sigma}_k = \boldsymbol{\Lambda}_k\boldsymbol{\Phi}_k\boldsymbol{\Lambda}'_k + \boldsymbol{\Psi}_k$ and mean vectors $\boldsymbol{\nu}_k$.

S.3. Step 1 in *lmfa*

The first step of the 3S procedure consists of obtaining the maximum likelihood (ML) estimates for the state-specific MMs (and thus the FA models) by means of mixture factor analysis (McLachlan & Peel, 2000; McNicholas, 2016). In *lmfa*, this is done by means of the `step1()` function. Details about the arguments and the output are provided in the Appendix of the paper. In the following, we first show the loglikelihood (log *L*) function that has to be optimized. Then, we explain how the model is estimated in the *lmfa* package, followed by the description of the inherent convergence criteria, the implemented algorithm, and, finally, the multistart procedure.

S.3.1. Likelihood function

In order to obtain the ML estimates, the following log *L* function has to be maximized:

$$\log L_{STEP1} = \sum_{i=1}^{I}\sum_{t=1}^{T}\log\left(\sum_{k=1}^{K} p(s_{itk}=1)\, p(\mathbf{y}_{it}|s_{itk}=1)\right), \tag{S5}$$

where $p(s_{itk}=1)$ are the state proportions, $p(\mathbf{y}_{it}|s_{itk}=1) = MVN(\mathbf{y}_{it}|\boldsymbol{\nu}_k, \boldsymbol{\Sigma}_k)$ are the response probabilities for a specific state, and $\boldsymbol{\Sigma}_k$ was defined before (S.2.2). In `step1()`, the proportions and response probabilities are estimated by means of an expectation-maximization (EM) algorithm described next.

S.3.2. Estimation

The observed-data log L in Equation (S5) is complicated by the latent-state memberships and the latent factor scores. The EM algorithm solves this problem by iterating though the following steps. First, in the expectation- (E-)step, the model parameters are assumed to be given and the posterior state-membership probabilities are calculated accordingly (i.e., under current estimates of the model parameters). Subsequently, in the maximization- (M-)step, the posterior state-membership probabilities are treated as observed and the model parameters are updated (i.e., optimized) one by one. In fact, in the M-step, another EM algorithm with a limited number of iterations is used to update the factor parameters for each state. The algorithm iterates through the E- and M-steps until convergence. In the following, we describe the specific steps of the EM algorithm of *lmfa* (S.3.4), including convergence criteria (S.3.3), and the multistart procedure (S.3.5).

S.3.3. Convergence

The EM algorithm (S.3.4) stops when reaching a convergence criterion. In the `step1()` function, the convergence is evaluated with respect to both the $\log L_{STEP1}$:

$$\Delta_{logL} = \log L^{v}_{STEP1} - \log L^{v-1}_{STEP1}, \tag{S6}$$

where v refers to the iteration number, and with respect to the sum of the absolute changes in the parameter estimates:

$$\Delta_{\hat{\theta}_{EM}} = \sum_{r=1}^{R}\left|\frac{\hat{\theta}^{v}_{r} - \hat{\theta}^{v-1}_{r}}{\hat{\theta}^{v-1}_{r}}\right|, \tag{S7}$$

with $r = 1, \ldots, R$ referring to the separate parameters. More specifically, the algorithm stops when one of the two criteria (or the maximum number of specified iterations) is reached. Within the EM algorithm, the M-step is either repeated as long as the parameters still change a lot, which is evaluated with

$$\Delta_{\hat{\theta}_{M-step}} = \sum_{r=1}^{R} \left| \frac{\hat{\theta}_r^b - \hat{\theta}_r^{b-1}}{\hat{\theta}_r^{b-1}} \right|, \tag{S8}$$

where b refers to the M-step iteration number, or until a certain number of M-step iterations is reached.

S.3.4. EM algorithm

In the following description of the EM algorithm, we use specific values to specify, for instance, the convergence tolerance and the maximum number of iterations. These are just the default values, which may be changed by the user.

START

Set the iteration number: $v = 0$ (or equal to the final iteration number of the chosen start set as described in S.3.5). **While** Δ_{logL} and $\Delta_{\hat{\theta}_{EM}} > 1e-06$ and $v < 1000$:

1. Update the iteration number: $v = v + 1$.
2. Update posterior state-membership probabilities $p(s_{itk} = 1|\mathbf{y}_{it})$ as follows:

$$p(s_{itk} = 1|\mathbf{y}_{it}) = \frac{p(s_{itk} = 1)p(\mathbf{y}_{it}|s_{itk} = 1)}{p(\mathbf{y}_{it})} = \frac{p(s_{itk} = 1)p(\mathbf{y}_{itk}|s_{itk} = 1)}{\sum_{k=1}^{K} p(s_{itk} = 1)p(\mathbf{y}_{itk}|s_{itk} = 1)}. \tag{S9}$$

3. Update the state-specific expected sample size N_k, state proportions $p(s_{itk} = 1)$, and $J \times 1$ state-specific intercept vectors $\boldsymbol{\nu}_k$ with

$$N_k = \sum_{i=1}^{I} \sum_{t=1}^{T} p(s_{itk} = 1|\mathbf{y}_{it}), \tag{S10}$$

$$p(s_{itk} = 1) = \frac{N_k}{I \times T}, \text{and} \tag{S11}$$

$$\boldsymbol{\nu}_k = \frac{\sum_{i=1}^{I} \sum_{t=1}^{T} [p(s_{itk} = 1|\mathbf{y}_{it})\mathbf{y}_{it}]}{N_k}. \tag{S12}$$

4. Compute the $J \times J$ state-specific expected observed covariance matrices $\mathbf{C}_k$ with

$$\mathbf{C}_k = \frac{\sum_{i=1}^{I} \sum_{t=1}^{T} [p(s_{itk} = 1|\mathbf{y}_{it})(\mathbf{y}_{it} - \boldsymbol{\nu}_k)(\mathbf{y}_{it} - \boldsymbol{\nu}_k)']}{N_k}. \tag{S13}$$

5. Set the M-step iteration number: $b = 0$.
6. While $\Delta_{\widehat{\theta}_{M-step}} > 1e-03$ and $b < 10$:
    a. Update the M-step iteration number: $b = b + 1$.
    b. Compute the $F_k \times J$ regression-weight vectors $\boldsymbol{\beta}_k$ (from regressions of the items on the latent factors) and the $F_k \times F_k$ matrices $\boldsymbol{\Theta}_k$ containing the expectations of the factor covariances based on the current parameters and the observed data with

$$\boldsymbol{\beta}_k = \boldsymbol{\Lambda}_k'(\boldsymbol{\Lambda}_k\boldsymbol{\Lambda}_k' + \mathbf{D}_k)^{-1} \text{ and} \tag{S14}$$

$$\boldsymbol{\Theta}_k = \mathbf{I}_{F_k} - \boldsymbol{\beta}_k\boldsymbol{\Lambda}_k + \boldsymbol{\beta}_k\mathbf{C}_k\boldsymbol{\beta}_k'. \tag{S15}$$

    c. Update loadings $\boldsymbol{\Lambda}_k$ and unique variances $\boldsymbol{\Psi}_k$ with

$$\boldsymbol{\Lambda}_k = \mathbf{C}_k\boldsymbol{\beta}_k'\boldsymbol{\Theta}_k^{-1} \text{ and} \tag{S16}$$

$$\boldsymbol{\Psi}_k = diag(\mathbf{C}_k - \boldsymbol{\Lambda}_k\boldsymbol{\beta}_k\mathbf{C}_k). \tag{S17}$$

    Updating the loadings and unique variances basically comes down to calculating the regression parameters and residual variance in a series of simple linear regressions for items with the factors as predictors. Note that small unique variances can lead to numerical problems. Therefore, if the variances fall below a certain threshold value, they are fixed to this value. The item-specific threshold values are constant across states and determined by multiplying the observed variance of an item with 1e-06.[2]

    d. Compute the change in parameters $\Delta_{\widehat{\theta}_{M-step}}$ (Equation (S8)).
7. Compute the log L_{STEP1} value (Equation (S5)).
8. Compute the change in parameters $\Delta_{\widehat{\theta}_{EM}}$ and log L_{STEP1} Δ_{logL} (Equations (S7) and (S6)).

END

S.3.5. Multistart procedure

In order to increase the chances of finding the global maximum, the following multistart procedure with multiple start sets is used, where the number of starts is equal to the specified number of model-based clusterings plus 10 times the specified number of start sets (i.e., by default, $5 + 10 \times 25 = 255$).

[2] Note that this value cannot be changed by the user.

Specifically, the procedure starts with five model-based clusterings by means of the *mclust* package (Scrucca, Fop, Murphy, & Raftery, 2016) to assign observations to states. Then, parameters are initialized for each of the five start sets as follows. The state-specific sample size N_k, state proportions $p(s_{itk} = 1)$, and state-specific intercepts $\boldsymbol{\nu}_k$, as well as the state-specific weighted sample covariance matrices $\mathbf{C}_k$ are computed as in Equations (S12) and (S13). Subsequently, probabilistic principal component analysis (Tipping & Bishop, 1999) is used in order to obtain the state-specific loading matrices $\boldsymbol{\Lambda}_k$ and unique variances $\boldsymbol{\Psi}_k$. To this end, the following eigendecomposition is conducted:

$$\widetilde{\mathbf{C}_k} = \mathbf{M}_k \mathbf{v}_k {\mathbf{M}_k}^{-1}, \tag{S18}$$

where the state-specific $J \times J$ matrix $\mathbf{M}_k$ is the matrix of eigenvectors and the state-specific $J \times 1$ vector $\mathbf{v}_k$ the eigenvalues. Moreover, the state-specific $J \times F_k$ matrix $\mathbf{M}_{k,F_k}$contains the first F_k eigenvectors, the state-specific $J \times (J - F_k)$ matrix $\mathbf{M}_{k,-F_k}$contains the disregarded eigenvectors, and the state-specific $F_k \times F_k$ diagonal matrix $\mathbf{V}_{k,F_k}$ contains the first F_k eigenvalues of $\mathbf{v}_k$ on its diagonal. The loadings and unique variances are then obtained with

$$\boldsymbol{\Lambda}_k = \mathbf{M}_{k,F_k} \sqrt{\mathbf{V}_{k,F_k} - \frac{\sum \mathbf{M}_{k,-F_k}}{J - F_k} \mathbf{I}_{F_k}} \text{ and with} \tag{S19}$$

$$\boldsymbol{\Psi}_k = \frac{\sum \mathbf{M}_{k,-F_k}}{J - F_k} \mathbf{I}_J, \tag{S20}$$

where $\sum \mathbf{M}_{k,-F_k}$ denotes the sum of the disregarded eigenvalues and $\mathbf{I}_{F_k}$ and $\mathbf{I}_J$ denote $F_k \times F_k$ and $J \times J$ identity matrices, respectively. Next, the loadings $\boldsymbol{\Lambda}_k$ and unique variances $\boldsymbol{\Psi}_k$ are updated once as in Equations (S16) and (S17). Subsequently, the value of $\log L_{STEP1}$ is obtained (Equation (S5)). Then, the partitions are ranked according to their $\log L_{STEP1}$ values.

From the *mclust* start set with the largest $\log L_{STEP1}$ value, 250 random start sets are generated by iteratively assigning 30 percent of the assignments, that is, for start set 1, 30 percent of the *mclust* assignments are reassigned, for start set 2, 30 percent of the assignments from start set 1 are reassigned, and so on. Then, for each of the start sets, parameters are again initialized and the $\log L_{STEP1}$ values are computed as described above. Next, the partitions of the random start sets and the best *mclust* set are ranked according to their $\log L_{STEP1}$ values and the best 25 start sets (i.e., the number of specified start sets) are selected as start partitions. For each start set, 15 iterations are performed by the EM algorithm (S.3.4). Subsequently, the testing strategy selects the start set with the highest $\log L_{STEP1}$ and saves the parameter estimates, $\hat{\theta}_r^{best}$, which serve as the initial values in the EM algorithm. Note that, instead of setting the number of iterations $v = 0$ (as at the beginning of the first iterations through the start sets), the algorithm

continues with the number of iterations that have already been performed; that is $v = 15$.

S.4. Step 2 in *lmfa*

In step 2, the subject- and time-point-specific observations are classified into the states $\mathbf{w}_{it} = (w_{it1}, \dots, w_{itK})$ based on the largest estimated posterior probability to belong to a state (i.e., based on a so-called "modal" state assignment). Thus, the observations are assigned to the MM that is most likely underlying the item responses. This can be expressed as $p(w_{itm} = 1|\mathbf{y}_{it}) = 1$ for state k with the largest $p(s_{itm} = 1|\mathbf{y}_{it})$. It is important to understand that the assignment of almost any observation includes some amount of uncertainty or "classification error" and can be calculated by conditioning the assigned state membership on the true state membership; that is $p(\mathbf{w}_{it} \mid \mathbf{s}_{itk})$. For details, see S.6.5. As stated before, the amount of classification error is related to the degree of state separation, which is quantified by the entropy-based R-squared measure $R^2_{entropy}$. The larger the state separation, the smaller the classification error. In *lmfa*, the state assignments, classification errors, and the $R^2_{entropy}$ can be obtained with the `step2()` function. For details about the arguments and the output, see the Appendix in the main body of the paper.

S.5. Step 3 in *lmfa*

The third step of the 3S procedure consists of estimating the transition model by means of a single indicator CT-LMM (with covariates), which automatically corrects for the classification uncertainty from step 2. More specifically, the single indicator model is:

$$p(\mathbf{W}_i|\mathbf{Z}_i) = \sum_{\mathbf{s}_{i1}} \cdots \sum_{\mathbf{s}_{iT}} p(\mathbf{s}_{i1}|\mathbf{z}_{i1}) \prod_{t=2}^{T} p(\mathbf{s}_{it}|\mathbf{s}_{it-1}, \mathbf{z}_{it}) \prod_{t=1}^{T} p(\mathbf{w}_{it}|\mathbf{s}_{it}), \tag{S21}$$

where the response probabilities $p(\mathbf{w}_{it}|\mathbf{s}_{it})$ are fixed to the classification errors of step 2 and $\mathbf{W}_i = (\mathbf{w}_{i1}, \mathbf{w}_{i2}, \dots, \mathbf{w}_{iT})$ are manifest single indicators (containing error) of the latent (error free) latent states $\mathbf{S}_i$ (Di Mari, Oberski, & Vermunt, 2016; Vogelsmeier, Vermunt, Bülow, & De Roover, 2019). Note that $\mathbf{W}_i$ and $\mathbf{S}_i$ may differ, which is increasingly more likely for larger classification errors. In *lmfa*, the CT-LMM can be estimated with the `step3()` function. Details about the arguments and the output are given in the Appendix of the paper. In the following, we first show the log L function that has to be maximized to obtain the ML estimates. Afterwards, we describe how the estimation is performed in the *lmfa* package, including the inherent convergence criteria and a multistart procedure.

S.5.1. Likelihood function

The following log L function has to be maximized to obtain the ML parameter estimates.

$$\log L_{STEP3} = \sum_{i=1}^{I} \log\big(p(\mathbf{W}_i|\mathbf{Z}_i)\big). \tag{S22}$$

Note that all parameters have been defined before.

S.5.2. Estimation

The model is estimated by means of an optimization routine from the *msm* package (Jackson, 2011). The corresponding function `msm()` can be used to estimate various types of CT-LMMs. In order to obtain the estimates, `msm()` itself uses the `optim()` function, which performs "general-purpose optimization" (R Core Team, 2020). In brief, instead of maximizing the log L, `optim()` minimizes a loss function that is equal to -2 times the log L. For details about the estimation procedure, we refer to the function documentations, that can be called with the commands `?msm` and `?optim`. The `step3()` function in *lmfa* is tailored to the type of model that needs to be estimated in the third step of LMFA (i.e., a single indicator CT-LMM with response probabilities fixed to the classification errors that results from the modal state assignment in step 2). Thus, `step3()` can be seen as a wrapper that facilitates the usage of `msm()` by providing proper parameter specifications and constraints and showing the desired parameter estimates, including significance tests for covariate effects by means of Wald tests.

S.5.3. Convergence

The optimization of the loss function (S.5.2) stops when the convergence criterion (say, 1e-10, the default) is reached. More specifically, the optimization stops when the loss function (i.e., $-2\log L$) can no further be reduced by a factor equal to the specified tolerance times the sum of the absolute value of the loss function and the tolerance. Thus, when defining the reduction in the loss function as

$$\Delta_{-2logL} = -2\log L_{STEP3}^{v-1} - (-2\log L_{STEP3}^{v}), \tag{S23}$$

the estimation stopes when $\Delta_{-2logL} < 1\mathrm{e}-10$ ($|-2\log L_{STEP3}^{v}| + 1\mathrm{e}-10$).

S.5.4. Multistart procedure

The results of the CT-LMM are very sensitive to the start values of the log transition intensities. More specifically, as previously stated, the intensities are directly related to the size of the time unit in the dataset. If the unit of the intensities from which the algorithm starts are too far from the actual unit, the model estimation will likely end up in a local maximum. Therefore, the following multistart procedure has been implemented: First, (by default) 25 random diagonal transition probability matrices **P** are sampled with staying probabilities on the diagonal that lie between 0.5 and 1. Per row, the off-diagonal probabilities are set equal (considering the constraint that rows must sum to 1). Subsequently, the transition intensity matrices **Q** are obtained by taking the matrix logarithm of the transition probability matrices **P**. Then, the **Q** matrix is rescaled by dividing it by the average length of the time intervals. Next, for each start set, 10 initial iterations of the CT-LMM analysis are performed. For each set, the $\log L$ values are obtained and ranked and the solution with the best $\log L$ value is used for the final analysis. Note that the estimation is not sensitive to the start values of the initial state logits and the covariate effects, which are, therefore, simply initialized to zero (i.e., covariate effects are absent and probabilities to start in a state are equally likely).

S.6. Statistics

S.6.1. Number of free parameters

The number of free parameters, fp, is obtained as follows:

$$fp = \overbrace{K-1}^{state\ proportions} + \overbrace{K \times J}^{intercepts} + \overbrace{K \times J}^{unique\ variances} + \overbrace{K \times J \times F_k}^{loadings}. \quad \text{(S24)}$$

Note that the number of activated constraints for small variances (see S.3.4) would be subtracted from fp.

S.6.2. BIC

The BIC considers complexity and parsimony by penalizing models with more parameters (fp) and larger samplesize (N) as follows:

$$\text{BIC} = -2 \times \log L_{STEP1} + fp \times log(N), \quad \text{(S25)}$$

with $\log L_{STEP1}$ as in Equation (S5), fp as in Equation (S24), and $N = \sum_{k=1}^{K} N_k$ with N_k as in Equation (S10).

S.6.3. CHull scree-test value

For all models (but the least and most complex model) on the upper boundary of the convex hull (CHull), the following scree-test value, st, is obtained:[3]

$$st_n = \frac{\left(\frac{\log L_{STEP1,n} - \log L_{STEP1,n-1}}{fp_n - fp_{n-1}}\right)}{\left(\frac{\log L_{STEP1,n+1} - \log L_{STEP1,n}}{fp_{n+1} - fp_n}\right)}, \tag{S26}$$

where the index n denotes the nth hull model. Note that the numerator and the denominator pertain to the slopes of two consecutive parts of the upper boundary of the CHull. Large values of st_n indicate that model n fits clearly better than model $n-1$, while model $n+1$ only leads to a small increase in model fit (Bulteel et al., 2013).

S.6.4. Explained variance

The amount of explained variance, EV, is calculated by taking the sum of squares of the standardized loadings per state, SS_k, dividing them by the number of items, J, weighting them by the state proportions, $p(s_{itk} = 1)$, and , finally, adding them up across states K. Thus,

$$SS_k = \sum_{j=1}^{J} \sum_{f=1}^{F_k} \left(\lambda_{kjf}\right)^2, \tag{S27}$$

$$EV = \sum_{k=1}^{K} SS_k/J \times\ p(s_{itk} = 1). \tag{S28}$$

S.6.5. Classification error

The classification error is obtained by conditioning the assigned state $w_{itm} = 1$ on the true, latent state $s_{itk} = 1$ for all states $m, k = 1, \ldots, K$. That is:

$$p(w_{itm} = 1 \mid s_{itk} = 1) = \frac{\frac{1}{I \times T} \sum_{i=1}^{I} \sum_{t=1}^{T} p(w_{itm} = 1|\mathbf{y}_{it}) p(s_{itk} = 1|\mathbf{y}_{it})}{p(s_{itk} = 1)}. \tag{S29}$$

[3] For the procedure to obtain the models on the upper boundary, see Bulteel, Wilderjans, Tuerlinckx, and Ceulemans (2013) and Vervloet, Wilderjans, Durieux, and Ceulemans (2017).

For a derivation, see Vogelsmeier et al. (2019). Note that $p(s_{itk} = 1)$ are the state proportions (Equation (S11) and $p(w_{itm} = 1|\mathbf{y}_{it})$ are the modal state assignments based on the posterior state-membership probabilities $p(s_{itk} = 1|\mathbf{y}_{it})$ (Equation (S9).

S.6.6. R-squared entropy

The state separation in terms of the R-square measure $R^2_{entropy}$ can be calculated as follows:

$$R^2_{entropy} = \frac{Entropy(\mathbf{S}) - Entropy(\mathbf{S}|\mathbf{Y})}{Entropy(\mathbf{S})} = 1 - \frac{Entropy(\mathbf{S}|\mathbf{Y})}{Entropy(\mathbf{S})} \tag{S30}$$

with

$$Entropy(\mathbf{S}) = \sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{k=1}^{K} -p(s_{itk} = 1)\log p(s_{itk} = 1) \tag{S31}$$

and

$$Entropy(\mathbf{S}|\mathbf{Y}) = \sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{k=1}^{K} -p(s_{itk} = 1|\mathbf{y}_{it})\log p(s_{itk} = 1|\mathbf{y}_{it}). \tag{S32}$$

Thus, Equation (S30) shows that the $R^2_{entropy}$ measure determines the relative improvement in predicting the state memberships given the observations (numerator) versus predicting the state memberships without the observations (denominator).

S.6.7. Wald-test statistic

The Wald statistic W^2 is computed as follows:

$$W^2 = \theta_u' \widehat{\Sigma}(\theta_u)^{-1}\theta_u \tag{S33}$$

with θ_u and $\widehat{\Sigma}(\theta_u)^{-1}$ indicating the slopes and the estimated variance covariance matrix for covariate u, respectively. Note that the degrees of freedom (df) are equal to the number of constrained parameters (e.g., df = 6 when testing the significance of a covariate with six slopes).

S.6.8. Factor scores

The state-specific factor score estimates, $\hat{\mathbf{F}}_{ik}$, are obtained by means of the regression method (Thomson, 1934; Thurstone, 1935):

$$\hat{\mathbf{F}}_{ik} = \mathbf{Y}_i\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Lambda}_k\boldsymbol{\Phi}_k, \tag{S34}$$

with $\boldsymbol{\Sigma}_k = \boldsymbol{\Lambda}_k\boldsymbol{\Phi}_k\boldsymbol{\Lambda}_k' + \boldsymbol{\Psi}_k$.

References

Bartolucci, F., Farcomeni, A., & Pennoni, F. (2014). Comments on: Latent Markov models: a review of a general framework for the analysis of longitudinal data with covariates. *Test, 23*, 473–477. doi:10.1007/s11749-014-0387-1

Bulteel, K., Wilderjans, T. F., Tuerlinckx, F., & Ceulemans, E. (2013). CHull as an alternative to AIC and BIC in the context of mixtures of factor analyzers. *Behavior Research Methods, 45*, 782–791. doi:10.3758/s13428-012-0293-y

Cox, D. R., & Miller, H. D. (1965). *The Theory of Stochastic Process*. London: Chapman & Hall.

Di Mari, R., Oberski, D. L., & Vermunt, J. K. (2016). Bias-adjusted three-step latent Markov modeling with covariates. *Structural Equation Modeling: A Multidisciplinary Journal, 23*, 649–660. doi:10.1080/10705511.2016.1191015

Jackson, C. H. (2011). Multi-state models for panel data: The msm package for R. *Journal of Statistical Software, 38*, 1–29.

McLachlan, G. J., & Peel, D. (2000). Mixtures of factor analyzers. In P. Langley (Ed.), *Proceedings of the Seventeenth International Conference on Machine Learning.* (pp. 599–606). San Francisco: Morgan Kaufmann.

McNicholas, P. D. (2016). *Mixture Model-Based Classification*. Boca Raton: Chapman and Hall/CRC Press.

R Core Team. (2020). A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria.

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R journal, 8*, 289–317.

Thomson, G. H. (1934). The meaning of i in the estimate of g. *British Journal of Psychology, 25*, 92–99. doi:10.1111/j.2044-8295.1934.tb00728.x

Thurstone, L. L. (1935). *The vectors of mind*. Chicago, IL: University of Chicago Press.

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, 61*, 611–622. doi:10.1111/1467-9868.00196

Vermunt, J. K., Langeheine, R., & Böckenholt, U. (1999). Discrete-time discrete-state latent markov models with time constant and time-varying covariates. *Journal of Educational and Behavioral Statistics, 24*, 179–207. doi:10.2307/1165200

Vervloet, M., Wilderjans, T. F., Durieux, J., & Ceulemans, E. (2017). *multichull: A Generic Convex-Hull-Based Model Selection Method*.

Vogelsmeier, L. V. D. E., Vermunt, J. K., Bülow, A., & De Roover, K. (2019). *Evaluating covariate effects on ESM measurement model changes with latent Markov factor analysis: A three-step approach*. Preprint. Retrieved from https://doi.org/10.31234/osf.io/6ufrc