



## Latent Class Probabilistic Latent Feature Analysis of Three-Way Three-Mode Binary Data

Michel Meulders  
KU Leuven

Philippe De Bruecker  
KU Leuven

---

### Abstract

The analysis of binary three-way data (i.e., persons who indicate which attributes apply to each of a set of objects) may be of interest in several substantive domains as sensory profiling, marketing research or personality assessment. Latent class probabilistic latent feature models (LCPLFMs) may be used to explain binary object-attribute associations on the basis of a small number of binary latent variables (called latent features). As LCPLFMs aim to model object-attribute associations using a small number of latent features they may be more suited to analyze data with many objects/attributes than standard multilevel latent class models which do not include such a dimension reduction. In this paper we describe new functions of the **plfm** package for analyzing binary three-way data with LCPLFMs. The new functions provide a flexible modeling approach as they allow to (1) specify different assumptions for modeling statistical dependencies between object-attribute pairs, (2) use different assumptions for modeling parameter heterogeneity across persons, (3) conduct a confirmatory analysis by constraining specific parameters to pre-specified values, (4) inspect results with **print**, **summary** and **plot** methods. As an illustration, the models are applied to analyze data on the perception of midsize cars, and to study the situational determinants of anger-related behavior.

*Keywords:* latent feature, three-way data, disjunctive model, conjunctive model, perceptual mapping, individual differences, EM algorithm, R.

---

## 1. Introduction

The analysis of three-way data is of interest in several substantive domains. In personality psychology, for instance, one may study the behavior of persons in different situations (Kuppens, Van Mechelen, and Meulders 2004; Mischel, Shoda, and Mendoza-Denton 2002; Vansteelandt and Van Mechelen 1998) or one may study the activation of emotion components (e.g. appraisals, action tendencies) when persons react to situations (Ceulemans, Kuppens,

and Van Mechelen 2012). In consumer research one may study consumers' perception of products in terms of attributes in order to derive a perceptual mapping of the product space (DeSarbo, Grewal, and Scott 2008; Monteiro, Dibb, and Almeida 2012; Torres and Bijmolt 2009). In sensory science it is common to study the sensory characterization of food samples by panellists (Meilgaard, Civille, and Carr 1999; Varela and Ares 2014).

Data from three-way studies (e.g., persons who judge object-attribute relations) may be analyzed in many ways. A common approach is to apply classical multivariate techniques (e.g., principal components analysis, factor analysis, correspondence analysis) to two-way data obtained by aggregating the three-way data across one of the modes (Varela and Ares 2014). However, as the analysis of such aggregated data provides only a partial view, specific three-way methods have been developed as well. In particular, with ratio-scaled three-way data popular three-way methods as Tucker3 and Candecomp/Parafac (Carroll and Chang 1970; Harshman 1970) can be used to summarize the entities of each mode using a limited number of components. Functions for estimating these models are available in the R packages **ThreeWay** (Giordani, Kiers, and Del Ferraro 2014), **PTAk** (Leibovici 2010) and **multiway** (Helwig 2017). Furthermore, as these classical three-way methods actually make the rather strict assumption that component scores assigned to entities of one mode are homogeneous across entities of other modes, methods that allow to capture such heterogeneity in the component scores have recently been proposed. In particular, one may use a clusterwise extension of the Parafac model (Wilderjans and Ceulemans 2013), or (clusterwise extensions of) simultaneous components analysis (De Roover, Ceulemans, Timmerman, Vansteelandt, Stouten, and Onghena 2012b; De Roover, Timmerman, Van Mechelen, and Ceulemans 2013b; De Roover, Ceulemans, Timmerman, Nezlek, and Onghena 2013a; Helwig 2013). Simultaneous components analysis and its (clusterwise) extensions can be estimated using the R package **multiway** or a software program based on MATLAB code (De Roover, Ceulemans, and Timmerman 2012a).

In addition to component-based methods, classification-based methods have been developed for three-way data to classify the elements of one mode while accounting simultaneously for information of the other two modes. In particular, Vichi (1999) described a least-squares approach to one-mode clustering of continuous three-way data and Vichi, Rocci, and Kiers (2007) further extended this approach to combine clustering and dimension reduction. Basford and McLachlan (1985) used a finite mixture of multivariate normal distributions to conduct one-mode classification of continuous three-way data. This approach was further extended by Hunt and Basford (1999) to model both continuous and categorical variables and by Hunt and Basford (2001) to deal with missing observations. Viroli (2011) considered finite mixtures of matrix-normal distributions to model continuous three-way data. Finally Vermunt (2007) used a hierarchical mixture approach and adapted multilevel latent class models to analyze continuous or categorical three-way data. Finite mixtures of normal distributions can be estimated with the R packages **mclust** (Fraley and Raftery 2002), **mixtools** (Benaglia, Chauveau, Hunter, and Young 2009) and **flexmix** (Leisch 2004; Grün and Leisch 2007, 2008). Hierarchical mixture models and special cases thereof can be estimated with Latent GOLD (Vermunt and Magidson 2005).

Although the analysis of continuous three-way data has received a lot of attention, the analysis of binary three-way data may be of special interest in studies where persons have to make object-attribute judgements. More specifically, check-all-that-apply (CATA) questions have recently gained in popularity in sensory product characterization with consumers (Ares, Deliza, Barreiro, Giménez, and Gámbaro 2010; Ares and Jaeger 2013). Responding to CATA

questions (i.e., selecting the relevant attributes from a list for each product) has been reported to be easier for participants while providing stable and reproducible product characterizations (Jaeger, Chheang, Yin, Bava, Giménez, Vidal, and Ares 2013; Ares, Antúnez, Giménez, Roigard, Pineau, Hunter, and Jaeger 2014).

As with continuous three-way data, methods for binary three-way data are often extensions of existing two-way methods. The deterministic hierarchical classes (HICLAS) model (De Boeck and Rosenberg 1988), for instance, explains two-way binary data by describing objects and attributes in terms of a limited set of binary latent variables (further called latent features). Several deterministic three-way extensions of the HICLAS model have been described (Leenen, Van Mechelen, and De Boeck 1999; Ceulemans, Van Mechelen, and Leenen 2003; Ceulemans and Van Mechelen 2004; Wilderjans, Ceulemans, and Kuppens 2012). In addition, Maris, De Boeck, and Van Mechelen (1996) developed a stochastic likelihood-based extension of the HICLAS model to analyze two-way object-by-attribute frequency data (obtained by aggregating three-way binary object  $\times$  attribute  $\times$  person data across persons). This model is further called the probabilistic latent feature model (PLFM). Meulders (2013) described the R package `plfm` to estimate the PLFM and to evaluate its fit. Furthermore several (multilevel) latent class extensions were proposed to extend PLFMs to the case of binary three-way data (Meulders, De Boeck, Kuppens, and Van Mechelen 2002; Meulders, De Boeck, and Van Mechelen 2003; Meulders, Tuerlinckx, and Vanpaemel 2013).

The goal of this paper is to describe the `LCplfm` function that has been added to the R package (R Development Core Team 2008) `plfm` (Meulders 2013) and that implements the (multilevel) LCPLFMs described by Meulders *et al.* (2003) and Meulders *et al.* (2013). The proposed `LCplfm` function offers a flexible modeling approach. First, it allows to specify different assumptions for modeling statistical dependencies between object-attribute pairs as well as different assumptions for modeling parameter heterogeneity across persons. Second, as the `LCplfm` function allows to constrain specific parameters to pre-specified values, it can handle confirmatory LCPLFM analysis. Third, in contrast to standard multilevel latent class models for three-way data (Vermunt 2007), LCPLFMs are suited to provide a parsimonious description of the data as they aim to model objects-attribute associations in terms of a small number of latent features. Finally, user-friendly `print`, `plot` and `summary` methods are added to enhance data analysis with the `LCplfm` function.

In Section 2 we first describe the original PLFM. Next, in Section 3 we describe multilevel LCPLFMs as described in Meulders *et al.* (2003) and Meulders *et al.* (2013). In Section 4 we describe the `LCplfm` function for probabilistic latent feature analysis of three-way binary data. In Section 5 we provide an example on the perception of midsize cars, and in Section 6 we use LCPLFMs to study the situational determinants of anger-related behavior. In Section 7 we conduct a small simulation study to investigate to what extent data generated using a specific decision rule can be fitted using another decision rule. Finally, in Section 8 we discuss topics that may be interesting to pursue in future research.

## 2. Probabilistic latent feature models

Probabilistic latent feature models (PLFMs) (Maris *et al.* 1996) can be used to model object-by-attribute frequency data that are derived by aggregating binary object  $\times$  attribute  $\times$  person data across persons. The models have been used to analyze data from several sub-

stantive domains as marketing research (Candel and Maris 1997; Meulders 2013), personality psychology (Meulders 2013), psychiatric diagnosis (Maris *et al.* 1996), cross cultural psychology (Meulders, De Boeck, Van Mechelen, Gelman, and Maris 2001) and the facial perception of emotions (Meulders, De Boeck, Van Mechelen, and Gelman 2005).

Observed object-attribute associations are described using the observed variable  $D_{ijk}$  which equals 1 if object  $j$  ( $j = 1, \dots, J$ ) has attribute  $k$  ( $k = 1, \dots, K$ ) according to person  $i$  ( $i = 1, \dots, I$ ), and 0 otherwise. To explain observed object-attribute associations, PLFMs assume that persons first classify objects and attributes in terms of  $F$  binary latent features and that, in a next step, observed associations are derived by applying a decision rule to these classifications.

As an example consider consumers who judge for each of a set of car models (e.g., ‘Porsche Cayenne’, ‘Ford S-Max’, ‘Toyota Prius’) and for each of a set of attributes (e.g., ‘low CO2 emissions’, ‘high status’, ‘comfortable’, ‘city-focused’, ‘powerful’, ‘spacious’, ‘safe’) whether or not an attribute applies to certain car model. To explain binary car-attribute associations, PLFMs assume that consumers classify car models and attributes in terms of so-called latent features. In this example, the latent features could represent basic car properties that may be ascribed to a car and that can be linked to certain car attributes. For instance, the latent feature ‘environmentally friendly’ is likely to be attributed to the ‘Toyota Prius’, and this feature is likely to be linked with attributes as ‘low CO2 emissions’, ‘city-focused’, and ‘high fuel efficiency’. Furthermore, the latent feature ‘family car’ is likely to be linked with the attributes ‘spacious’, ‘safe’, ‘easy to enter’ and this feature is likely to be attributed to a car model as the ‘Ford S-Max’. Next, PLFMs assume that car-attribute associations are obtained by applying a decision rule to the pattern of features perceived in a certain car and the pattern of features linked to the attribute.

As another example, consider persons who judge for each of a set of situations and for each of a set of behaviors whether or not they would they would display a certain behavior (e.g., suppress anger, shout, quarrel) when they are being frustrated by someone in a certain situation (e.g., you are falsely accused of cheating on an exam, you bang your shins against a park bench, you find out that someone has told lies about you). To explain situation-behavior judgements PLFMs assume that persons classify situations and behaviors in terms of a number of latent features. In this case the latent features could for instance represent latent situational encodings (e.g., whether or not you are alone in the situation, whether the person that frustrates you has a higher status than you) that may elicit a certain behavior (shout, become irritated, bottle up anger). PLFMs assume further that situation-behavior judgements are derived by applying a decision rule to situation and behavior classifications.

The two-stage process of PLFMs can be formally described as follows:

1. The classification of object  $j$  in terms of latent features is modeled using latent Bernoulli variables. In particular,  $X_{ijkf} \sim \text{Bern}(\sigma_{jf})$  equals 1 if object  $j$  has latent feature  $f$  ( $f = 1, \dots, F$ ) according to person  $i$  when she judges object-attribute pair  $(j, k)$ , and 0 otherwise. Likewise, the classification of attribute  $k$  in terms of latent features is modeled using latent Bernoulli variables. More specifically,  $Y_{ijkf} \sim \text{Bern}(\rho_{kf})$  equals 1 if attribute  $k$  is linked to latent feature  $f$  ( $f = 1, \dots, F$ ) according to person  $i$  when she judges object-attribute pair  $(j, k)$ , and 0 otherwise. Note that upper-case notation is used here to denote random variables and lower-case notation will be used to denote specific realizations of the corresponding random variable.

2. It is assumed that the observed judgement of person  $i$  on object-attribute pair  $(j, k)$  results from applying a decision rule  $C(\cdot)$  to the latent object and attribute classifications. Formally the decision rule specifies the conditional probability that an object is associated to an attribute given the corresponding object and attribute classifications, that is,

$$C(\mathbf{X}_{ijk}, \mathbf{Y}_{ijk}) = P(D_{ijk} = 1 | \mathbf{X}_{ijk}, \mathbf{Y}_{ijk})$$

with  $\mathbf{X}_{ijk} = (X_{ijk1}, \dots, X_{ijkF})$  and  $\mathbf{Y}_{ijk} = (Y_{ijk1}, \dots, Y_{ijkF})$ . For deterministic decision rules, the observed association directly follows from the object and attribute classifications, that is  $P(D_{ijk} = 1 | \mathbf{X}_{ijk}, \mathbf{Y}_{ijk})$  can only take the values 0 or 1 and  $D_{ijk} = C(\mathbf{X}_{ijk}, \mathbf{Y}_{ijk})$ . On the other hand, for probabilistic decision rules  $P(D_{ijk} = 1 | \mathbf{X}_{ijk}, \mathbf{Y}_{ijk})$  takes values in the interval  $[0, 1]$ , that is the decision rule specifies the degree of the object-attribute association given the object and attribute classifications.

Maris *et al.* (1996) discuss two deterministic non-compensatory decision rules. First, a disjunctive communality rule (DC) implies that an object is associated to an attribute if the object has at least one latent feature that is linked to the attribute, that is,

$$D_{ijk} = 1 \iff \exists f : X_{ijkf} = Y_{ijkf} = 1$$

or

$$C^{DC}(\mathbf{X}_{ijk}, \mathbf{Y}_{ijk}) = 1 - \prod_f (1 - X_{ijkf} Y_{ijkf}).$$

Second, a conjunctive dominance (CD) rule implies that an object is associated to an attribute if the object has all the latent features that are linked to the attribute, that is,

$$D_{ijk} = 1 \iff \forall f : X_{ijkf} \geq Y_{ijkf}$$

or

$$C^{CD}(\mathbf{X}_{ijk}, \mathbf{Y}_{ijk}) = \prod_f [1 - (1 - X_{ijkf}) Y_{ijkf}].$$

In addition to the described deterministic non-compensatory decision rules, we also propose a probabilistic compensatory so-called additive (ADD) decision rule which assumes that the conditional probability of an object-attribute association equals the average number of features that the object and attribute have in common, that is:

$$C^{ADD}(\mathbf{X}_{ijk}, \mathbf{Y}_{ijk}) = \frac{1}{F} \sum_f X_{ijkf} Y_{ijkf}.$$

From the above model assumptions  $X_{ijkf} \sim \text{Bern}(\sigma_{jf})$  and  $Y_{ijkf} \sim \text{Bern}(\rho_{kf})$  one may derive the object-attribute association probability:

$$P(D_{ijk} = 1 | \boldsymbol{\sigma}, \boldsymbol{\rho}) = \sum_{\mathbf{X}_{ijk}} \sum_{\mathbf{Y}_{ijk}} P(D_{ijk} = 1 | \mathbf{x}_{ijk}, \mathbf{y}_{ijk}) p(\mathbf{x}_{ijk} | \boldsymbol{\sigma}) p(\mathbf{y}_{ijk} | \boldsymbol{\rho}).$$

For the disjunctive communality (DC) rule one may derive that

$$\pi_{jk}^{DC} = P(D_{ijk} = 1 | \boldsymbol{\sigma}, \boldsymbol{\rho}) = 1 - \prod_f (1 - \sigma_{jf} \rho_{kf}).$$

Likewise, for the conjunctive dominance (CD) rule one may derive that

$$\pi_{jk}^{CD} = P(D_{ijk} = 1 | \boldsymbol{\sigma}, \boldsymbol{\rho}) = \prod_f [1 - (1 - \sigma_{jf}) \rho_{kf}].$$

Finally, for the additive decision rule one may derive that

$$\pi_{jk}^{ADD} = P(D_{ijk} = 1 | \boldsymbol{\sigma}, \boldsymbol{\rho}) = \frac{1}{F} \sum_f \sigma_{jf} \rho_{kf}.$$

Depending on the application at hand, and depending on available substantive theory a particular decision rule may be more appropriate. A disjunctive rule, for instance, may be used to formalize the idea that, when a feature is linked to the attribute, the presence of the feature in the object is a sufficient condition to observe an object-attribute association. For instance, to explain binary patient-symptom associations on the basis of latent syndromes a disjunctive rule can be used to formalize the idea that a patient will have a symptom if he/she has at least one syndrome that implies the symptom (Maris *et al.* 1996).

On the other hand, a conjunctive rule may be used to formalize the idea that, in order to observe an object-attribute association, all the features linked to the attribute need to be present in the object. For instance, a conjunctive decision rule may be used to test the hypothesis of configural encoding in the processing of facial expressions (Meulders *et al.* 2005). More specifically this hypothesis implies that an emotion will only be perceived in a facial expression if all the facial features linked to the emotion are also activated when processing the facial expression.

The fact that observed variables  $D_{ijk}$  are derived as a mapping of independent (Bernoulli distributed) latent variables implies that the likelihood of the PLFM can be expressed as a product of Bernoulli distributed variables:

$$p(\mathbf{d} | \boldsymbol{\sigma}, \boldsymbol{\rho}) = \prod_i \prod_j \prod_k (\pi_{jk})^{d_{ijk}} (1 - \pi_{jk})^{1-d_{ijk}}$$

Note that  $D_{ijk}$  is used to indicate the observed random variable and  $d_{ijk}$  represents a specific realization. The vector  $\mathbf{d}$  comprises the  $I \times J \times K$  observations  $d_{ijk}$  ( $i = 1, \dots, I; j = 1, \dots, J; k = 1, \dots, K$ ).

Furthermore, as the basic PLFM assumes homogeneity of object-attribute association probabilities across persons, it is easy to see that the PLFM is actually a model for the aggregated  $J \times K$  two-way two-mode frequency matrix that is obtained by summing the  $I \times J \times K$  three-way three-mode array of binary observations across persons. The likelihood of the basic PLFM reads

$$p(\mathbf{n} | \boldsymbol{\sigma}, \boldsymbol{\rho}) \propto \prod_j \prod_k \text{Binomial}(n_{jk}, I)$$

with  $n_{jk} = \sum_{i=1}^I d_{ijk}$ .

### 3. Latent class probabilistic latent feature models

Due to the assumptions involved, the basic PLFM is not well suited to model the observed three-way binary associations. First, the assumption that all observations  $D_{ijk}$  are statistically independent may be unrealistic because object-attribute associations with a common



object or attribute tend to be correlated across persons. As dependencies between object-attribute associations with a common element may actually be caused by qualitative individual differences in object or attribute perception (i.e., persons differ in the pattern of features they ascribe to objects or attributes), and as such individual differences are likely to occur, dependence is a realistic model assumption. Second, the assumption that object-attribute association probabilities are homogeneous across persons may be unrealistic in practice as it implies that there are no individual differences. To deal with these problems, different types of latent class extensions of the PLFM have been proposed.

### 3.1. Modeling dependencies

To deal with unrealistic independence assumptions, [Maris \(1999\)](#) and [Meulders \*et al.\* \(2003\)](#) developed PLFMs with adapted stochastic assumptions. More specifically, to model dependencies between object-attribute associations with a common element, one may assume that object-feature classifications are not realized again at each new judgement, but that they remain constant when a person judges which attributes apply to a certain object. Likewise, one may model dependencies between observations with a common attribute by assuming that attribute-feature classifications are not realized again at each new judgement, but that they remain constant when a person judges whether the attribute applies to each of the objects.

Models that assume constant object-feature classifications or constant attribute-feature classifications per person, model dependencies between object-attribute pairs with a common element by including qualitative individual differences in the model. In particular, a model with constant object-feature classifications assumes that persons differ in the pattern of features they ascribe to an object. In the same way, a model with constant attribute-feature classifications assumes that the pattern of features linked to each attribute is a source of individual differences.

Formally, models in which object-feature classifications or attribute-feature classifications remain constant across all the judgements made by a person can be described as constrained latent class models ([Formann 1992](#)). Consider for instance a model in which object-feature classifications remain constant across the judgements made by a person and in which attribute-feature classifications are renewed at each new judgement made by the person. Let  $\mathbf{x}_{ij} = (x_{ij1}, \dots, x_{ijF})$  indicate the pattern of features assigned by person  $i$  to object  $j$ . Assuming that judgements of different persons are statistically independent, and that judgements  $\mathbf{D}_{ij} = (D_{ij1}, \dots, D_{ijK})$  of person  $i$  about the attributes that apply to object  $j$  are conditionally independent given  $\mathbf{x}_{ij}$ , the likelihood of a model with constant object-feature classifications reads:

$$p(\mathbf{d}|\boldsymbol{\sigma}, \boldsymbol{\rho}) = \prod_i \prod_j \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij}|\boldsymbol{\sigma}_j) \prod_k (\pi_{jk|\mathbf{x}})^{d_{ijk}} (1 - \pi_{jk|\mathbf{x}})^{1-d_{ijk}} \quad (1)$$

with

$$p(\mathbf{x}_{ij}|\boldsymbol{\sigma}_j) = \prod_f (\sigma_{jf})^{x_{ijf}} (1 - \sigma_{jf})^{1-x_{ijf}} \quad (2)$$

and with  $\pi_{jk|\mathbf{x}}$  the conditional probability of an object-attribute association which depends on the decision rule at hand. For instance, assuming a disjunctive communality rule one may derive that

$$\pi_{jk|\mathbf{x}} = \sum_{\mathbf{y}_{ijk}} P(D_{ijk} = 1|\mathbf{x}_{ij}, \mathbf{y}_{ijk})p(\mathbf{y}_{ijk}|\boldsymbol{\rho}_k) = 1 - \prod_f (1 - x_{ijf}\rho_{kf}). \quad (3)$$

The model defined by (1), (2) and (3) can be regarded as a constrained latent class model with  $F$  binary latent variables ( $X$ ) at the object-person level and with  $F$  binary latent variables ( $Y$ ) at the object-attribute-person level. The model classifies persons in  $2^F$  classes for each object, and it models  $2^F \times K$  conditional probabilities with  $K \times F$  attribute parameters. Note that a model with constant attribute-feature classifications can be obtained by switching the role of objects and attributes in Equation 1.

As an alternative one may consider a model in which attribute-feature classifications remain constant across the judgements made by a person and in which object-feature classifications are renewed at each new judgement made by the person. Let  $\mathbf{y}_{ik} = (y_{ik1}, \dots, y_{ikF})$  indicate the pattern of features assigned by person  $i$  to attribute  $k$ . Assuming that judgements of different persons are statistically independent, and that judgements  $\mathbf{D}_{ik} = (D_{i1k}, \dots, D_{iJk})$  of person  $i$  are conditionally independent given  $\mathbf{y}_{ik}$ , the likelihood of the model reads:

$$p(\mathbf{d}|\boldsymbol{\sigma}, \boldsymbol{\rho}) = \prod_i \prod_k \sum_{\mathbf{y}_{ik}} p(\mathbf{y}_{ik}|\boldsymbol{\rho}_j) \prod_j (\pi_{jk|\mathbf{y}})^{d_{ijk}} (1 - \pi_{jk|\mathbf{y}})^{1-d_{ijk}} \quad (4)$$

with

$$p(\mathbf{y}_{ik}|\boldsymbol{\rho}_k) = \prod_f (\rho_{kf})^{y_{ikf}} (1 - \rho_{kf})^{1-y_{ikf}} \quad (5)$$

and with  $\pi_{jk|\mathbf{y}}$  the conditional probability of an object-attribute association which depends on the decision rule at hand. For instance, assuming a disjunctive communality rule one may derive that

$$\pi_{jk|\mathbf{y}} = \sum_{\mathbf{x}_{ijk}} P(D_{ijk} = 1|\mathbf{x}_{ijk}, \mathbf{y}_{ik}) p(\mathbf{x}_{ijk}|\boldsymbol{\sigma}_j) = 1 - \prod_f (1 - y_{ikf} \sigma_{jf}). \quad (6)$$

The model defined by (4), (5) and (6) can be regarded as a constrained latent class model with  $F$  binary latent variables ( $Y$ ) at the attribute-person level and with  $F$  binary latent variables ( $X$ ) at the object-attribute-person level. The model classifies persons in  $2^F$  classes for each attribute, and it models  $2^F \times J$  conditional probabilities with  $J \times F$  object parameters.

Finally, we note that a latent class extension of the PLFM in which both object-feature classifications and attribute-feature classifications remain constant within a person is not feasible if the involved decision rule is deterministic because it is not always possible to define latent object and attribute classifications  $(\mathbf{x}_{ij}, \mathbf{y}_{ik})$  so that  $\mathbf{d}_i = C(\mathbf{x}_{ij}, \mathbf{y}_{ik})$ .

### 3.2. Modeling heterogeneity

To relax the assumption of homogeneous object-attribute association probabilities among persons one may use a latent class extension of the basic PLFM that includes class-specific object and/or attribute parameters (Meulders *et al.* 2013). Let  $Z_{it}$  indicate a binary latent variable that equals 1 if person  $i$  belongs to class  $t$  ( $t = 1, \dots, T$ ), and 0 otherwise. It is further assumed that  $P(Z_{it} = 1) = \xi_t$  with  $\sum_t \xi_t = 1$ . Assuming that judgements of different persons are independent and that, within persons, judgements are conditionally independent given latent class membership, the likelihood of a model with both class-specific object and attribute parameters equals

$$p(\mathbf{d}|\boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\xi}) = \prod_i \sum_t \xi_t \prod_j \prod_k (\pi_{jk|t})^{d_{ijk}} (1 - \pi_{jk|t})^{1-d_{ijk}} \quad (7)$$



with  $\pi_{jk|t}$  being the conditional object-attribute association probability which depends on the decision rule of the model. For instance, using a disjunctive communality rule one may derive that

$$\pi_{jk|t} = 1 - \prod_f (1 - \sigma_{jft} \rho_{kft}) \quad (8)$$

The assumption that both object and attribute parameters are class-specific implies that the interpretation of the features may be different in each of the latent classes. As an alternative one may also constrain the object or attribute parameters in Equation 8 to be homogeneous across classes. In particular, using only class-specific object parameters the conditional probabilities equal

$$\pi_{jk|t} = 1 - \prod_f (1 - \sigma_{jft} \rho_{kft}).$$

Likewise, using only class-specific attribute parameters, the conditional probabilities equal

$$\pi_{jk|t} = 1 - \prod_f (1 - \sigma_{jft} \rho_{kft}).$$

A model with only class-specific object or only class-specific attribute parameters may be interesting from a substantive point of view. For instance, a model with class-specific object parameters and homogeneous attribute parameters implies that the features have the same interpretation for all persons, but that persons of different classes may have different probabilities to ascribe a particular feature to an object. When applying the model to analyze product-by-attribute judgements of consumers, for instance, it could be meaningful to assume that product perception is driven by the same set of features for all consumers (i.e., there is a common reality), but that the probability to ascribe a latent feature to a product depends on the latent class the person belongs to (Meulders *et al.* 2013).

Finally, note that the model defined by Equation 7 and Equation 8 can be regarded as a constrained latent class model with  $T$  binary latent variables ( $Z$ ) at the person level and with  $2 \times F$  binary latent variables ( $X, Y$ ) at the object-attribute-person level. The model classifies persons in  $T$  classes, and it models  $J \times K \times T$  conditional probabilities with  $J \times F \times T$  object parameters and  $K \times F \times T$  attribute parameters.

### 3.3. Modeling both dependency and heterogeneity

To model heterogeneity in the object-attribute association probability among persons as well as dependencies between object-attribute pairs with a common element, Meulders *et al.* (2013) developed multilevel LCPLFMs with constant object-feature or attribute-feature classifications that include class-specific object and/or attribute parameters. Consider for instance a model that assumes constant object-feature classifications per person as well as class-specific object and attribute parameters. This type of model actually includes two types of person classifications. First, the model classifies persons in one of  $T$  latent classes to model heterogeneity in the object and attribute parameters. This (outer) classification of persons is represented by a  $T$ -element vector  $\mathbf{Z}_i = (Z_{i1}, \dots, Z_{iT})$  in which  $Z_{it}$  equals 1 if person  $i$  belongs to class  $t$  and in which  $Z_{it}$  equals 0 otherwise. It is further assumed that  $P(Z_{it} = 1) = \xi_t$  with  $\sum_t \xi_t = 1$ . Second, a model with constant object-feature classifications classifies persons in  $2^F$  latent classes for each object. This (inner) classification is represented by  $\mathbf{x}_{ij}$  which indicates the pattern of latent features assigned by person  $i$  to object  $j$ . Assuming that judgements of

different persons are independent, and that judgements of the same person are conditionally independent given latent class membership, the likelihood of this model can be expressed as follows:

$$p(\mathbf{d}|\boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\xi}) = \prod_i \sum_t \xi_t \prod_j \sum_{\mathbf{x}_{ij}} \pi_{\mathbf{x}|t} \left[ \prod_k (\pi_{jk|\mathbf{x},t})^{d_{ijk}} (1 - \pi_{jk|\mathbf{x},t})^{1-d_{ijk}} \right] \quad (9)$$

with

$$\pi_{\mathbf{x}|t} = P(\mathbf{x}_{ij}|Z_{it} = 1, \boldsymbol{\sigma}_{jt}) = \prod_f (\sigma_{jft})^{x_{ijf}} (1 - \sigma_{jft})^{1-x_{ijf}}. \quad (10)$$

and with  $\pi_{jk|\mathbf{x},t}$  a conditional probability that depends on the decision rule involved. More specifically, for a disjunctive communality rule one may derive that:

$$\pi_{jk|\mathbf{x},t} = P(D_{ijk} = 1|\mathbf{x}_{ij}, Z_{it} = 1, \boldsymbol{\rho}_{kt}) = 1 - \prod_f (1 - x_{ijf} \rho_{kft}). \quad (11)$$

### 3.4. Taxonomy of models

Table 1 provides an overview of (multilevel) LCPLFMs that can be obtained by combining different dependency and heterogeneity assumptions. The basic PLFM (Maris *et al.* 1996; Meulders 2013) is indicated in Table 1 as m0. This model can be estimated with the `plfm` function of the `plfm` package. Models D and E have been described by Meulders *et al.* (2003) for three-way three-mode binary data, and by Maris (1999) for two-way two-mode binary data. Models A to C and m1 to m6 have been described by Meulders *et al.* (2013). In this paper we present the function `LCplfm` which has been added to the `plfm` package to estimate disjunctive, conjunctive and additive models m1 to m6 and models D and E. Models that only allow to model heterogeneity in the model parameters (i.e., models A to C) are not included in the `LCplfm` function because they are usually outperformed by models that allow to model both dependencies between object-attribute pairs with a common element and heterogeneity in the model parameters (i.e., models m1 to m6).

Why do models m1 to m6 typically outperform model A to C on real data? First, in real data, object-attribute pairs with a common element (i.e., a common object or attribute) tend to be correlated and, unlike models A to C, models m1 to m6 are specifically designed to model such dependencies. Therefore, when applying the LCPLFMs to real data, models m1 to m6 typically have a higher likelihood than models A to C. Second, as the distribution of object-feature classification patterns (in models m1 to m3) and the distribution of attribute-feature classification patterns (in models m4 to m6) is modeled using an independence model (see Equation 2 and Equation 5), modeling dependencies between object-attribute pairs with a common element does not increase the number of model parameters. That is, assuming  $T=t$  latent classes are used to model heterogeneity among object and/or attribute parameters, models  $\{A, m1, m4\}$  are equally complex. In the same way, models  $\{B, m2, m5\}$  and models  $\{C, m3, m6\}$  have the same number of model parameters. Therefore, if models m1 to m6 have a higher likelihood than models A to C they will also be selected by AIC or BIC because the models are equally complex.

As some of the models in Table 1 are equivalent, six different algorithms are sufficient to estimate 24 models (i.e., disjunctive, conjunctive and additive models of type m1 to m6, D

and E). First, three different algorithms (i.e., for disjunctive models m1 to m3) are needed to estimate disjunctive models m1 to m6 and models D and E. More specifically, by switching the role of objects and attributes, model m5 can be obtained from m1, m4 can be obtained from m2, and m6 can be obtained from m3. Furthermore, models D and E can be obtained respectively by fitting models m1 and m4 with one latent class ( $T = 1$ ). Second, as  $\pi_{jk}^{CD}(1 - \sigma, \rho) = 1 - \pi_{jk}^{DC}(\sigma, \rho)$  conjunctive models can be obtained by fitting the corresponding disjunctive model to transformed data  $D_{ijk}^* = 1 - D_{ijk}$  and by applying a transformation to the object parameters, namely  $\sigma^* = 1 - \sigma$ . Third, using the same reasoning as for disjunctive models, only three algorithms (i.e., for additive models m1 to m3) are needed to estimate additive models of type m1 to m6, D and E.

Object-feature classification	Attribute-feature classification	Class-specific parameters			
		None	Object	Attribute	Object and attribute
renewed	renewed	m0	A	B	C
constant	renewed	D	m1	m2	m3
renewed	constant	E	m4	m5	m6

Table 1: Taxonomy of latent class probabilistic latent feature models.

## 4. Latent class probabilistic latent feature analysis

### 4.1. Components of the package

The R package **plfm** comprises the following components for conducting latent class probabilistic latent feature analysis:

- The function `LCplfm` can be used to compute point estimates of disjunctive or conjunctive LCPLFMs with a specific number of features, a specific dependency assumption (i.e., constant object classification per person or constant attribute classification per person) and with a specific heterogeneity assumption (i.e., class-specific object parameters and/or class specific attribute parameters). In addition, the function computes standard errors of the parameters as well as criteria for model selection and descriptive model fit.
- The function `stepLCplfm` can be used to fit a series of disjunctive, conjunctive or additive LCPLFMs with different numbers of features and different numbers of latent classes.
- There are `print` and `plot` methods provided for each function. The `plot` method of the `LCplfm` function, can be used to visualize (class-specific) point estimates of object or attribute parameters and corresponding 95% confidence intervals (CI) for each feature. The `plot` method of the `stepLCplfm` function can be used to visualize the value of model selection criteria (i.e., AIC, BIC) or descriptive fit measures for models with different numbers of latent classes and with different numbers of features.

## 4.2. Function for latent class probabilistic latent feature analysis

The function `LCplfm` can be used to compute parameter estimates for disjunctive, conjunctive or additive LCPLFMs with a particular number of latent features and with a specific number of latent classes. In addition, the function computes standard errors of the parameter estimates, as well as criteria for model selection, and measures of descriptive model fit. The `LCplfm` function takes the following arguments:

- `data` is a  $I \times J \times K$  array of binary observations. Observation  $(i, j, k)$  equals 1 if object  $j$  ( $j = 1 \dots, J$ ) is associated to attribute  $k$  ( $k = 1, \dots, K$ ) according to person  $i$  ( $i = 1, \dots, I$ ).
- `F` indicates number of latent features included in the model.
- `T` indicates the number of latent classes used to classify persons.
- `M` indicates number of runs that should be conducted using random starting points.
- `maprule` is used to specify a disjunctive (`maprule = "disj"`), conjunctive (`maprule = "conj"`) or additive (`maprule = "add"`) decision rule for the latent class probabilistic latent feature model.
- `emcrit1` is used to specify the convergence criterion for estimating `M` candidate models with the EM algorithm.
- `emcrit2` is used to specify the convergence criterion for estimating the best model (out of `M` models) with the EM algorithm.
- `model` is used to specify the type of dependency and heterogeneity assumption included in the model. More specifically, `model = 1`, `model = 2`, and `model = 3` represent models with a constant object classification per person and with, respectively, class-specific object parameters, class-specific attribute parameters, and class-specific object and attribute parameters (i.e., models `m1`, `m2` and `m3` in Table 1). Furthermore, `model = 4`, `model = 5`, and `model = 6` represent models with a constant attribute classification per person, and with, respectively, class-specific object parameters, class-specific attribute parameters and class-specific object and attribute parameters (i.e., models `m4`, `m5` and `m6` in Table 1).
- `start.objectparameters` is an array of object parameters to be used as a starting point in each of `M` runs. The size of the array equals  $J \times F \times T \times M$  when `model = 1, 4, 3, 6` and  $J \times F \times M$  when `model = 2, 5`. If `start.objectparameters = NULL`, randomly generated starting values are used.
- `start.attributeparameters` is an array of attribute parameters to be used as a starting point in each of `M` runs. The size of the array equals  $K \times F \times T \times M$  when `model = 2, 5, 3, 6` and  $K \times F \times M$  when `model = 1, 4`. If `start.attributeparameters = NULL`, randomly generated starting values are used.
- `start.sizeparameters` is a  $T \times M$ -vector of latent class size parameters to be used as starting point in each of `M` runs. If `start.sizeparameters = NULL`, randomly generated starting values are used.

- `delta` indicates the precision used to compute standard errors of the parameters with the method of finite differences.
- `printrun` is used to specify printing options. If `printrun = TRUE` the mapping rule (i.e., disjunctive, conjunctive or additive), the number of features (`F`), the number of latent classes (`T`) and the number of the run (out of `M` runs) are printed to the output screen, whereas `printrun = FALSE` suppresses the printing.
- `update.objectparameters` is a binary valued array that indicates for each object parameter whether it has to be estimated from the data or constrained to the starting value. A value of 1 means that the corresponding object parameter is estimated and a value of 0 means that the corresponding object parameter is constrained to the starting value provided by the user. The size of the array equals  $J \times F \times T$  when `model = 1, 4, 3, 6` and  $J \times F$  when `model = 2, 5`. If `update.objectparameters = NULL` all object parameters are estimated from the data.
- `update.attributeparameters` is a binary valued array that indicates for each attribute parameter whether it has to be estimated from the data or constrained to the starting value. A value of 1 means that the corresponding attribute parameter is estimated and a value of 0 means that the corresponding attribute parameter is constrained to the starting value provided by the user. The size of the array equals  $K \times F \times T$  when `model = 2, 5, 3, 6` and  $K \times F$  when `model = 1, 4`. If `update.attributeparameters = NULL` all attribute parameters are estimated from the data.
- `Nbootstrap` indicates the number of bootstrap iterations to be used for simulating the reference distribution of log odds-ratio dependency measures.

### *EM algorithm for computation of the posterior mode(s) of multilevel LCPLFMs*

As the complete-data likelihood of LCPLFMs has a simple structure, it is convenient to compute maximum-likelihood estimates of the model parameters with an EM algorithm (Maris *et al.* 1996; Meulders *et al.* 2003, 2013). As an example, Meulders *et al.* (2013) provide a detailed derivation of the EM algorithm for model `m1` in Table 1. The derivation of the EM algorithm for other models in Table 1 is similar.

Furthermore, as maximum likelihood estimates will not always exist in the interior of the parameter space, it is convenient to impose a convex prior distribution on the model parameters as this will guarantee the existence of posterior mode estimates within the interior of the parameter space. In particular, for the LCPLFM with class-specific object and attribute parameters we specify the following conjugate convex prior distribution:

$$\begin{aligned}
 p(\boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\xi}) &= \prod_j \prod_f \prod_t \text{Beta}(\sigma_{jft} | 1 + \frac{\alpha_\sigma}{JT}, 1 + \frac{\beta_\sigma}{JT}) \\
 &\quad \times \prod_k \prod_f \prod_t \text{Beta}(\rho_{kft} | 1 + \frac{\alpha_\rho}{KT}, 1 + \frac{\beta_\rho}{KT}) \\
 &\quad \times \text{Dir}(\boldsymbol{\xi} | 1 + \frac{\gamma_1}{T}, \dots, 1 + \frac{\gamma_T}{T})
 \end{aligned} \tag{12}$$

with  $\alpha_\sigma = \beta_\sigma = \alpha_\rho = \beta_\rho = 1$  and  $\gamma_t = 2$  ( $t = 1, \dots, T$ ). Note that these constants and their weights in Equation 12 are chosen to obtain the default prior used by the standard software Latent Gold (Vermunt and Magidson 2005, 2008).

#### *Computation of asymptotic standard errors*

Let  $\boldsymbol{\theta} = (\boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\xi})$  and let  $\hat{\boldsymbol{\theta}}$  denote the posterior mode of the model. The standard error of the parameter  $\theta_i$  can be computed as follows:

$$SE(\theta_i) = \left[ -\frac{\partial^2 \log p(\boldsymbol{\theta}|\mathbf{d})}{\partial \theta_i^2} \Big|_{\hat{\boldsymbol{\theta}}} \right]^{-\frac{1}{2}}$$

To compute the second derivative of the log posterior with respect to the parameter  $\theta_i$  we use a central difference approximation to numerically differentiate the gradient  $g(\boldsymbol{\theta}) = \partial \log p(\boldsymbol{\theta}|\mathbf{d}) / \partial \boldsymbol{\theta}$ . More specifically,

$$\frac{\partial^2 \log p(\boldsymbol{\theta}|\mathbf{d})}{\partial \theta_i^2} \approx \frac{g(\boldsymbol{\theta}^{(-i)}, \theta_i + \delta) - g(\boldsymbol{\theta}^{(-i)}, \theta_i - \delta)}{2\delta}$$

with  $\boldsymbol{\theta}^{(-i)}$  a vector in which the  $i$ -th parameter has been omitted. Section A of the appendix describes how the gradient  $g(\boldsymbol{\theta})$  can be analytically derived for the disjunctive model. For additive models, the derivation is similar.

#### *Estimation strategy*

For  $F > 1$  the posterior distribution of LCPLFMs is always multimodal. Different local maxima may exist, and in addition, for each local maximum the posterior distribution of a model with  $F$  latent features contains  $F!$  identical posterior modes because the labels of the latent features can be switched.

As the posterior distribution can be highly multimodal, a two-stage estimation strategy is used to efficiently locate the model with the highest posterior density. In a first stage, the EM algorithm is used to estimate  $M$  candidate models using random starting points and using the less strict convergence criterion `emcrit1` with default value  $10^{-3}$ . In a second stage, the EM-algorithm is used to estimate the best model (i.e., with the highest posterior density among the  $M$  candidate models) using the more strict convergence criterion `emcrit2` with default value  $10^{-8}$ . As many local maxima may exist, it is recommended to estimate a sufficiently high number of candidate models, that is specify  $M \geq 50$ .

#### *Model selection criteria*

When applying LCPLFMs to real data one needs to select an appropriate model to describe the data. As there are many possible models one could start by fitting the basic PLFM to get an idea about the approximate number of features needed to explain the object  $\times$  attribute associations. In a second step, LCPLFMs with different dependency and heterogeneity assumptions could be fitted and model selection criteria could be used to determine the best among the candidate models. In a third step, descriptive fit measures can be inspected to further evaluate the validity of the selected model. Finally, as LCPLFMs are often used as an exploratory tool, practical or substantive arguments can also play a role to select a final model, for instance, meaningful feature interpretation, sufficient size of extracted latent classes, accuracy of the parameter estimates, and so on.



To choose the best model among a set of candidate LCPLFMs one may use model selection criteria as AIC (Akaike 1974) or BIC (Schwarz 1978). Both criteria are computed as the sum of a badness-of-fit term (minus twice the loglikelihood of the fitted model) and a penalty term which measures the complexity of the model. For AIC and BIC the penalty term equals  $2k$  and  $k \log(I)$ , respectively with  $k$  being the number of free model parameters and  $I$  being the number of persons who have judged the object-attribute pairs. AIC and BIC both have desirable properties. AIC is an efficient model selection criterion. This means that when the true model is infinite dimensional, the prediction error in the model selected with AIC is asymptotically the same as the prediction error of the best candidate model. On the other hand BIC is asymptotically consistent, which means that it will select the true model from a set of candidate models if the true model is finite dimensional and among the candidate models (Liu and Yang 2011, p. 2075). A simulation study with LCPLFMs has shown that BIC has a good performance to select the true model from a set of candidate models (Meulders *et al.* 2003). However, in real data applications it is unlikely that the true model is among the candidate models and hence AIC may be more appropriate than BIC.

### *Measures of model fit*

When evaluating the fit of LCPLFMs we are especially interested in two aspects, (1) the fit of the model to the marginal object-by-attribute table, and (2) to what extent observed individual differences are captured by the model.

First, to describe the fit of the model to the marginal object-by-attribute table, the `LCplfm` function computes the correlation and squared correlation (i.e., variance accounted for (VAF)) between observed and expected frequencies in the marginal object  $\times$  attribute table.

Furthermore, the `plfm` function includes an absolute Pearson- $\chi^2$  goodness-of-fit test that can be used to test the null hypothesis that model `m0` fits the marginal object  $\times$  attribute frequency table. However, with increasing sample size this test may become very powerful to detect differences between the true and the fitted model, and as the fitted model is usually incorrect in some sense the test is likely to be significant. Therefore, it may also be interesting to study the descriptive fit of the model. For the basic PLFM (model `m0`), the correlation between observed and expected frequencies in the object  $\times$  attribute table is a useful measure of descriptive fit, because it indicates to what extent the latent features included in the model can model the object  $\times$  attribute interaction.

In contrast to PLFMs, LCPLFMs do not directly fit the object  $\times$  attribute table of frequencies. For instance, an LCPLFM of type `m1` models frequencies of patterns  $\mathbf{d}_{ij}$ . The relevant goodness-of-fit test for this type of model is a Pearson- $\chi^2$  test that compares observed and expected frequencies of the patterns  $\mathbf{d}_{ij}$ . However, as LCPLFMs are usually applied to data sets with many objects and attributes, the number of patterns  $\mathbf{d}_{ij}$  is usually large (e.g., if  $K = 10$  there are  $2^{10} = 1024$  patterns for object  $j$ ). As a result, the Pearson-chi square statistic will not have a  $\chi^2$  distribution because the table of observed frequencies is sparse and expected frequencies are too small.

Although the `LCplfm` function does not directly fit the frequencies in the object  $\times$  attribute table it still reports the correlation between observed and expected frequencies for this marginal table, because this correlation gives an indication of how well the object  $\times$  attribute interaction can be explained by the latent features included in the model.

Second, LCPLFMs model observed individual differences by classifying persons with respect

to the pattern of features they ascribe to an object or attribute, or by assuming class-specific object or attribute parameters. If the involved classifications represent the observed individual differences well, observed object-attribute associations will be conditionally independent (i.e., conditional on latent class membership), and dependencies between pairs of observed associations will match dependencies predicted by the model.

To check whether observed dependencies between pairs of object-attribute associations  $D_{ijk}$  and  $D_{ij'k'}$  are fitted by the model, one may use a parametric bootstrap procedure to simulate the distribution of the dependencies under the model, and check whether observed dependencies are in the corresponding  $(1 - \alpha)\%$  CI.

The dependency between  $D_{ijk}$  and  $D_{ij'k'}$  can be measured using the log-odds ratio computed on the  $D_{ijk} \times D_{ij'k'}$  cross-table (de la Torre 2008):

$$\text{OR}_{jkj'k'} = \log \left\{ \frac{[0.5 + \sum_i D_{ijk} D_{ij'k'}] [0.5 + \sum_i (1 - D_{ijk})(1 - D_{ij'k'})]}{[0.5 + \sum_i D_{ijk}(1 - D_{ij'k'})] [0.5 + \sum_i (1 - D_{ijk}) D_{ij'k'}]} \right\}$$

Note that we add 0.5 to each cell of the cross-table to make sure that log-odds ratio can be computed if a cell frequency equals 0.

As LCPLFMs especially focus on modeling dependencies between object-attribute associations with a common object or attribute the `LCplfm` function reports the observed OR statistic and the corresponding 95% and 99% simulated CIs for object-attribute pairs with a common object, that is,  $\text{OR}_{jkj'k'}$  ( $j = 1, \dots, J; k = 1, \dots, K; k' = 1, \dots, K; k < k'$ ) and  $\text{OR}_{jkj'k}$  ( $k = 1, \dots, K; j = 1, \dots, J; j' = 1, \dots, J; j < j'$ ). OR dependencies between object-attribute pairs without a common element are omitted from the reporting because the total number of dependencies can become very large (i.e., if  $J = K = 30$  then  $JK(JK - 1)/2 = 404550$ ).

### 4.3. Function for estimating a series of LCPLFMs

The function `stepLCplfm` can be used to fit a series of disjunctive or conjunctive latent class probabilistic latent feature models that assume `minF` to `maxF` latent features and `minT` to `maxT` latent classes. As `stepLCplfm` repeatedly calls the `LCplfm` function, it takes the same arguments except the number of features and the number of latent classes which are specified with the following arguments:

- `minF` is the minimum number of latent features.
- `maxF` is the maximum number of latent features.
- `minT` is the minimum number of latent classes.
- `maxT` is the maximum number of latent classes.

## 5. Example: Perception of midsize cars

The list `car2` contains data of 147 persons who indicated for each of 12 midsize cars whether or not they have each of 23 attributes. In an initial analysis we use the basic PLFM (i.e., model `m0` in Table 1) to model car-attribute associations on the basis of binary latent features.

More specifically, after loading the data, we use the function `stepplfm` to estimate disjunctive PLFMs with 1 up to 7 latent features. For each model  $M$  ( $=50$ ) runs with random starting points are conducted. A `set.seed()` statement is used to guarantee the reproducibility of the results.

```
R> data("car2")
R> set.seed(723756)
R> car2.m0.lst <- stepplfm(freq1 = car2$freq1, freqtot = car2$freqtot, M =
+   50, minF = 1, maxF = 7, emcrit1 = 10e-5, emcrit2 = 10e-8)
```

A print of the results shows that a model with 5 latent features has lowest BIC (i.e., 36566) and that a model with six features has lowest AIC (i.e., 35952). Furthermore, the results indicate that models with 5 and 6 features fit the frequencies in the marginal car-by-attribute table rather well (i.e., the correlation between observed and predicted frequencies equals .968 and 0.983, respectively), although both models fail to fit the car-by-attribute frequencies in an absolute sense ( $\chi^2 = 433, df = 101, p < .01$  and  $\chi^2 = 278, df = 66, p < .01$ , respectively).

```
R> print(car2.m0.lst)
```

yields

INFORMATION CRITERIA:

	LogLik	LogPost	Deviance	AIC	BIC
F=1	-20295	-20348	40591	40661	40765
F=2	-18752	-18904	37504	37644	37854
F=3	-18134	-18386	36267	36477	36791
F=4	-17975	-18322	35951	36231	36649
F=5	-17846	-18293	35692	36042	36566
F=6	-17766	-18319	35532	35952	36580
F=7	-17731	-18381	35462	35952	36685

PEARSON CHI SQUARE TEST OBJECT X ATTRIBUTE TABLE:

	Chisquare	df	p-value
F=1	5675.831	241	0
F=2	2383.139	206	0
F=3	1006.579	171	0
F=4	671.833	136	0
F=5	433.054	101	0
F=6	277.697	66	0
F=7	207.503	31	0

DESCRIPTIVE FIT OBJECT X ATTRIBUTE TABLE:

	Correlation	VAF
F=1	0.510	0.260
F=2	0.816	0.666

```

F=3      0.924 0.853
F=4      0.951 0.904
F=5      0.968 0.938
F=6      0.983 0.965
F=7      0.988 0.976

```

In a subsequent analysis, we use LCPLFMs to model person differences in car perception, and to model dependencies between car-attribute pairs with a common element. Although the six-feature model has a lower AIC value, we present the results for the five-feature model because interpretation and descriptive fit of both models are similar. More specifically we use the function `stepLCplfm` to estimate disjunctive five-feature latent class PLFMs with only class-specific car parameters ( $T = 1, \dots, 3$ ) and with either a constant car-feature classification per person (i.e., model m1) or a constant attribute-feature classification per person (i.e., model m4). The assumption that attribute parameters are constrained to be equal across classes is convenient as it implies that the latent features have the same interpretation for each of the latent classes.

```

R> set.seed(91673)
R> car2.m1.lst <- stepLCplfm(data = car2$data, M = 50, minF = 5, maxF = 5,
+   minT = 1, maxT = 3, printrun = TRUE, model = 1, Nbootstrap = 5000)
R> set.seed(27492)
R> car2.m4.lst <- stepLCplfm(data = car2$data, M = 50, minF = 5, maxF = 5,
+   minT = 1, maxT = 3, printrun = TRUE, model = 4, Nbootstrap = 5000)

```

To further inspect the fit of the models we may apply the `summary` method to the result of each analysis:

```
R> summary(car2.m1.lst)
```

yields

	LogLik	LogPost	Deviance	AIC	BIC	Correlation	VAF
model=1 F=5 T=1	-17935	-17976	35871	36221	36744	0.95	0.9
model=1 F=5 T=2	-17794	-17836	35588	36060	36765	0.95	0.9
model=1 F=5 T=3	-17740	-17783	35480	36074	36962	0.95	0.9
	P.0Rattpair.in95CI		P.0Robjpair.in95CI				
model=1 F=5 T=1			0.83		0.78		
model=1 F=5 T=2			0.83		0.80		
model=1 F=5 T=3			0.84		0.82		
	P.0Rattpair.in99CI		P.0Robjpair.in99CI				
model=1 F=5 T=1			0.93		0.91		
model=1 F=5 T=2			0.94		0.91		
model=1 F=5 T=3			0.94		0.92		

```
R> summary(car2.m4.lst)
```

yields

			LogLik	LogPost	Deviance	AIC	BIC	Correlation
model=4	F=5	T=1	-17455	-17485	34909	35259	35782	0.96
model=4	F=5	T=2	-17258	-17290	34515	34987	35693	0.95
model=4	F=5	T=3	-17172	-17207	34343	34937	35826	0.95
			VAF		P.0Rattpair.in95CI		P.0Robjpair.in95CI	
model=4	F=5	T=1	0.92		0.83			0.89
model=4	F=5	T=2	0.91		0.84			0.89
model=4	F=5	T=3	0.91		0.85			0.90
			P.0Rattpair.in99CI		P.0Robjpair.in99CI			
model=4	F=5	T=1			0.93			0.97
model=4	F=5	T=2			0.95			0.97
model=4	F=5	T=3			0.95			0.97

As can be seen from a comparison of AIC and BIC values, LCPLFMs with a constant attribute-feature classification per person (i.e., `model = 4`) outperform LCPLFMs with a constant car-feature classification per person (i.e., `model = 1`) or basic PLFMs in which both car-feature classifications and attribute-feature classifications are renewed at each judgement made by a person. In addition, model m4 can better fit dependencies between object-attribute pairs with a common attribute than model m1. More specifically, for 1-, 2- and 3- class models of type m1, 78%, 80%, 82% of the observed OR dependencies between pairs with a common attribute are in the simulated 95% CI, compared to 89%, 89% and 90% for model m4. On the other hand, models m1 and m4 perform equally well in fitting dependencies between object-attribute pairs with a common object. In particular, for 1-, 2- and 3- class models of type m1, 83%, 83%, 84% of the observed OR dependencies between pairs with a common object are in the simulated 95% CI, compared to 83%, 84% and 85% for model m4.

Furthermore, among models with constant attribute-feature classification, there is evidence for heterogeneity of car parameters across respondents. In particular AIC is lowest for the three-class model (i.e., 34937), whereas BIC is lowest for the two-class model (i.e., 35693). In what follows, the parameters of the five-feature two-class model will be interpreted more in detail. We present the results of the two-class solution rather than the three-class solution because one of the classes in the three-class solution is very small (4% of the respondents), resulting in inaccurate estimates for the corresponding car parameters.

When using the `stepLCplfm` function to compute a series of disjunctive (`maprule = "disj"`), conjunctive (`maprule = "conj"`) or additive (`maprule = "add"`) LCPLFMs with `minF` up to `maxF` features and `minT` up to `maxT` latent classes, the results of subsequent `LCplfm` analyses are stored in a matrix of lists with `maxF-minF+1` rows and `maxT-minT+1` columns. Each list is an object of class "LCplfm". Using `names()`, a list of all attached entries can be obtained. For instance, for the five-feature two-class model with class-specific parameters and a constant attribute-feature classification per person:

```
R> names(car2.m4.lst[[1, 2]])
```

yields

```
[1] "call"           "logpost.runs"   "best"
[4] "objpar"        "attpar"         "sizepar"
```

```

[7] "SE.objpar"      "SE.attpar"      "SE.sizepar"
[10] "gradient.objpar" "gradient.attpar" "gradient.sizepar"
[13] "fitmeasures"    "postprob"       "margprob.JK"
[16] "condprob.JKT"

```

The listed entries have the following contents:

- `call` contains the parameters used to call the function.
- `logpost.runs` is a list containing the logarithm of the posterior density for each of the  $M$  computed models.
- `best` is an index which indicates the model with the highest posterior density among each of the  $M$  computed models.
- `objpar` is a  $J \times F$  matrix or a  $J \times F \times T$  array of estimated object parameters for the best model.
- `attpar` is a  $K \times F$  matrix or a  $K \times F \times T$  array of estimated attribute parameters for the best model.
- `sizepar` is a  $T$ -vector of estimated class size parameters for the best model.
- `SE.objpar` is a  $J \times F$  matrix or a  $J \times F \times T$  array of estimated standard errors for the object parameters of the best model.
- `SE.attpar` is a  $K \times F$  matrix or a  $K \times F \times T$  array of estimated standard errors for the attribute parameters of the best model.
- `SE.sizepar` is a  $T$ -vector of estimated standard errors for the class size parameters of the best model.
- `gradient.objpar` is a  $J \times F$  matrix or a  $J \times F \times T$  array containing the gradients of object parameters for the best model.
- `gradient.attpar` is a  $K \times F$  matrix or a  $K \times F \times T$  array containing the gradients of attribute parameters for the best model.
- `gradient.sizepar` is a  $T$ -vector containing the gradients of class size parameters for the best model.
- `fitmeasures` is a list of fit measures for the best model including loglikelihood, log posterior density, deviance, information criteria (AIC and BIC), and two measures of descriptive fit on the object-by-attribute table of frequencies (i.e., correlation between observed and expected frequencies and variance accounted for by the model).
- `postprob` is a  $I \times T$  matrix of posterior probabilities for the best model.
- `margprob.JK` is a  $J \times K$  matrix of expected marginal object-attribute association probabilities for the best model.



- `condprob.JKT` is a  $J \times K \times T$  array of expected conditional object-attribute association probabilities (i.e., expected probability of object-attribute association given latent class membership).
- `report.OR.attpair` is a matrix that contains per object for all attribute pairs the observed OR dependency (`OR.obs`), the expected OR dependency (`OR.mean`) and the upper and lower bounds of the corresponding simulated 95 and 99 percent confidence interval (`OR.p025`, `OR.p975`, `OR.p005`, `OR.p995`).
- `report.OR.objpair` is a matrix that contains per attribute for all object pairs the observed OR dependency (`OR.obs`), the expected OR dependency (`OR.mean`) and the upper and lower bounds of the corresponding simulated 95 and 99 percent confidence interval (`OR.p025`, `OR.p975`, `OR.p005`, `OR.p995`).

In order to interpret the latent features, one may use the `plot` method to visualize point estimates and a 95% CI of car- and attribute parameters for a specific feature. For instance, to visualize the car- and attribute parameters for feature 1 one may use the following code:

```
R> plot(car2.m4.lst[[1, 2]], element = "attribute", feature = 1,
+       positionlabel = -0.6, main = "Attribute parameters feature 1")
R> plot(car2.m4.lst[[1, 2]], element = "object", feature = 1, main =
+       "Car parameters feature 1")
```

The visualization of the parameters for other features is similar. The top panels of Figure 1 to Figure 5 show point estimates and 95% CIs for the attribute parameters of feature 1 to feature 5, respectively. The bottom panels of Figure 5 contain point estimates and 95% CIs for the car parameters of the two latent classes. Note that, classes 1 and 2 contain respectively 18% and 82% of the persons in the sample.

As can be seen from an inspection of estimated attribute parameters in Figure 1 to Figure 5, the latent features have a meaningful interpretation. In particular, feature 2 is ascribed to environmentally friendly cars with high fuel efficiency that have low CO<sub>2</sub> emissions (e.g., ‘Toyota Prius’). Feature 1 and feature 3 are similar in that they are ascribed to comfortable and spacious cars that are suited for family use. However, these features also have a different interpretation as feature 1 is ascribed to cars that are also being perceived as safe and reliable (e.g., ‘Volkswagen Passat’) whereas feature 3 is ascribed to cars that are also easy to exit or enter. Feature 4 and feature 5 are similar in that they are ascribed to luxurious sporty cars with a nice design. In contrast, these features also have a different interpretation as feature 5 is ascribed to cars that are also being perceived as powerful and as having a high status and a high trade-in value (e.g., ‘Mercedes C-Class’, ‘BMW Series 3’, and ‘Audi A4’) whereas feature 4 (which is ascribed to ‘Peugeot 508’) is not linked to these attributes.

Further inspection of the car parameters indicates that, depending on the latent class they belong to, persons may have a different perception of specific car models. For instance, unlike persons of class 1, persons of class 2 have a much higher probability to ascribe feature 3 to a ‘Citroen C5’ (i.e., .57 versus .02). In contrast, persons of class 1 are more likely to ascribe feature 1 to a ‘Citroen C5’ (i.e., .34 versus .10). Hence persons of class 2 perceive the ‘Citroen C5’ as a comfortable and spacious family car that is also easy to exit or enter, but that is not particularly reliable or safe. On the other hand, persons of class 1 perceive the ‘Citroen C5’ as a comfortable, spacious family car that is also reliable and has good safety, but that

is not particularly easy to exit or enter. As another example, we see that persons of class 2 are more likely to ascribe feature 5 (luxurious, sporty, nice design, powerful, high status, high trade-in value) to ‘Mercedes C-class’, ‘Audi A4’ and ‘BMW Series 3’ than persons of class 1. On the other hand, persons of class 1 are more likely to ascribe feature 4 (luxurious, sporty, nice design, comfortable but lacking high status, high trade-in value) to ‘BMW Series 3’ than persons of class 2.

## 6. Example: Situational determinants of anger-related behavior

As a second example we analyze data on the situational determinants of anger-related behavior (Kuppens *et al.* 2004). The data, which are stored in the list `anger2`, consist of the binary judgements of 115 first-year psychology students who indicated whether or not they would display each of 14 anger-related behaviors when being angry at someone in each of 9 situations. The 14 behaviors consist of 7 pairs that reflect a particular behavior type that can be elicited in situations in which one is angry at someone (1) anger-out (you flew off the handle, you started a fight), (2) avoidance (you avoided a confrontation, you went out of the other’s way), (3) social sharing (you unburdened your heart to others, you told others what had happened), (4) assertive behavior (you said what was bothering you in a direct and sober way, you calmly explained what was bothering you), (5) indirect behavior (you showed something was bothering you without saying anything, you started to sulk), (6) anger-in (you suppressed your anger, you bottled up your anger), and (7) reconciliation (you reconciled, you talked things out). The 9 situations are constructed by crossing the levels of two factors with three levels: (1) the extent to which one likes the instigator of anger (like, unfamiliar, dislike), and (2) the status of the instigator of anger (lower status, equal status, higher status).

An important goal of personality research is to identify personality types with stable situation-behavior relations (Mischel *et al.* 2002). As will be illustrated in this section, LCPLFMs are especially suited to meet this challenge. First, they aim for a parsimonious description of situation-behavior associations using latent features that represent basic behavior types. Second, they allow to model heterogeneity in the situation-behavior association probability by assuming latent person classes with class-specific situation-feature and/or behavior-feature parameters. Note that in the present example behavior-feature parameters indicate to what extent a behavior reflects a certain behavior type, and situation-feature parameters indicate to what extent a certain behavior type is appropriate in a situation.

To study the occurrence of the 7 behavior types hypothesized in the design of the study across situations we will adopt a confirmatory approach. In particular we use a model with 7 latent features each of which represents a behavior type hypothesized in the study. Furthermore, we assume that a behavior-feature parameter equals 0 if the behavior does not reflect the behavior type hypothesized in the study. Table 2 shows the matrix of behavior-feature parameters to be estimated assuming that these parameters are equal across persons. On the other hand, as we want to study to what extent situations affect the occurrence of specific behavior types and how this differs among persons, we consider a model with class-specific situation parameters. To conduct the analysis we first load the data

```
R> data("anger2")
```

Next, we create a matrix that indicates which behavior parameters should be updated in the

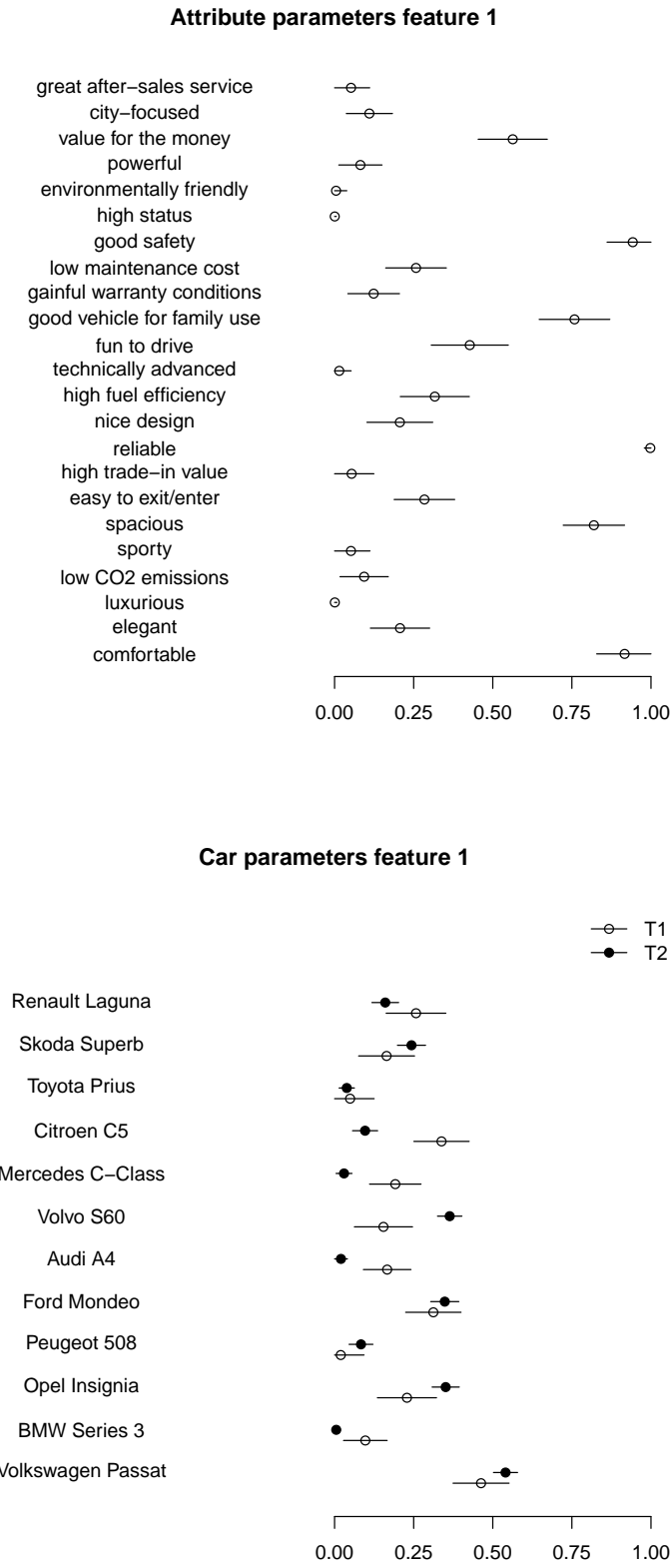


Figure 1: Point estimates and 95% confidence interval for attribute parameters (top panel) and car parameters (bottom panel) of feature 1.

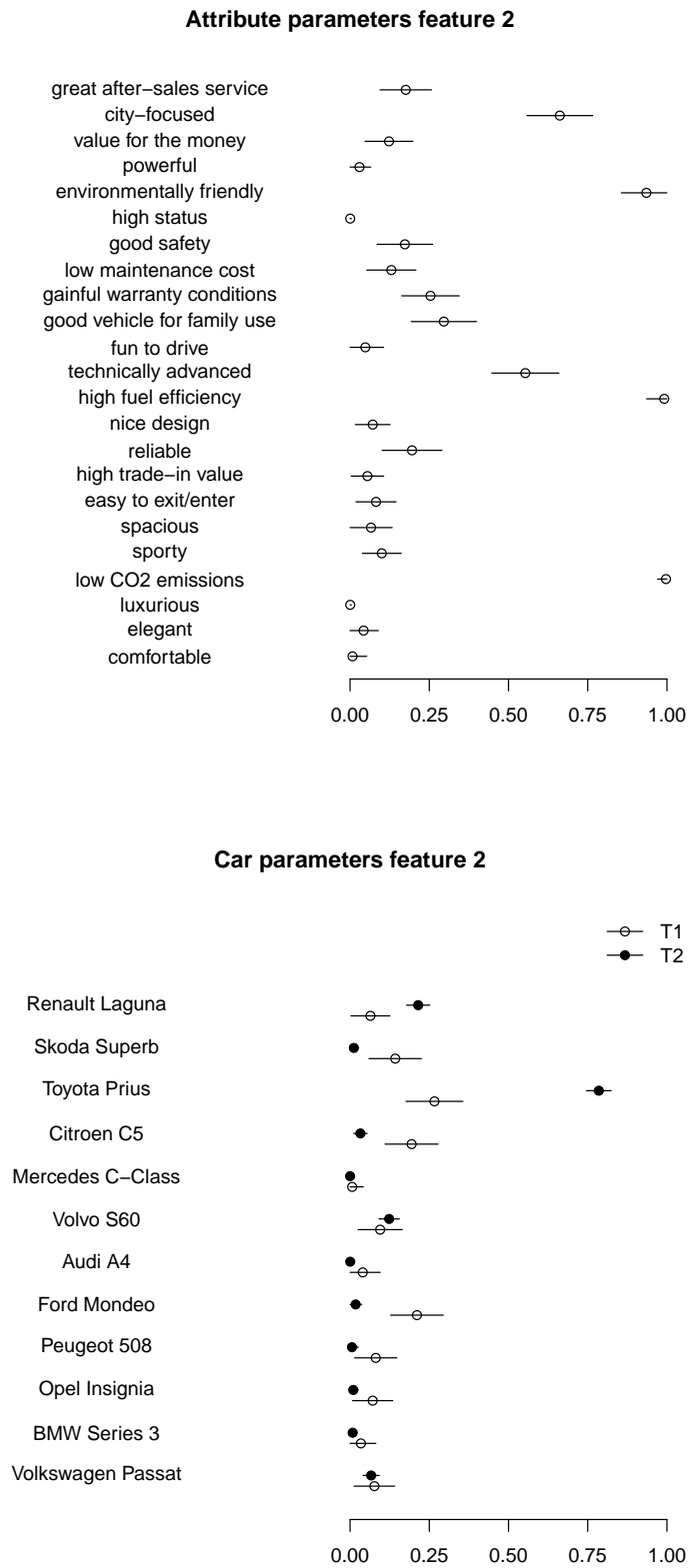


Figure 2: Point estimates and 95% confidence interval for attribute parameters (top panel) and car parameters (bottom panel) of feature 2.

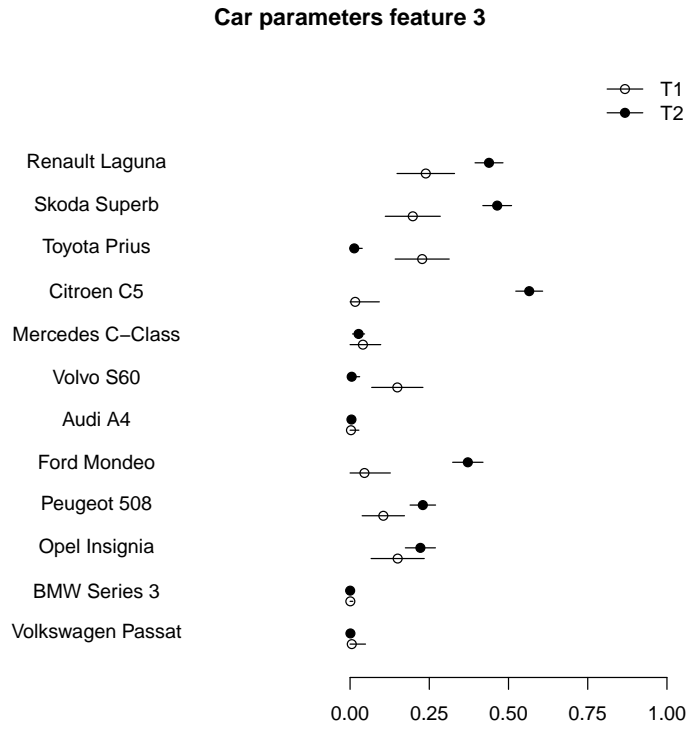
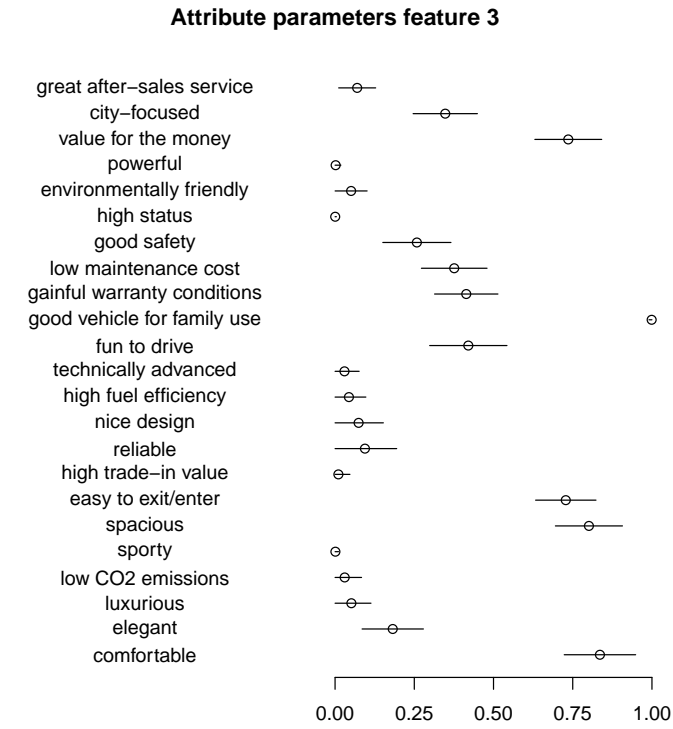


Figure 3: Point estimates and 95% confidence interval for attribute parameters (top panel) and car parameters (bottom panel) of feature 3.

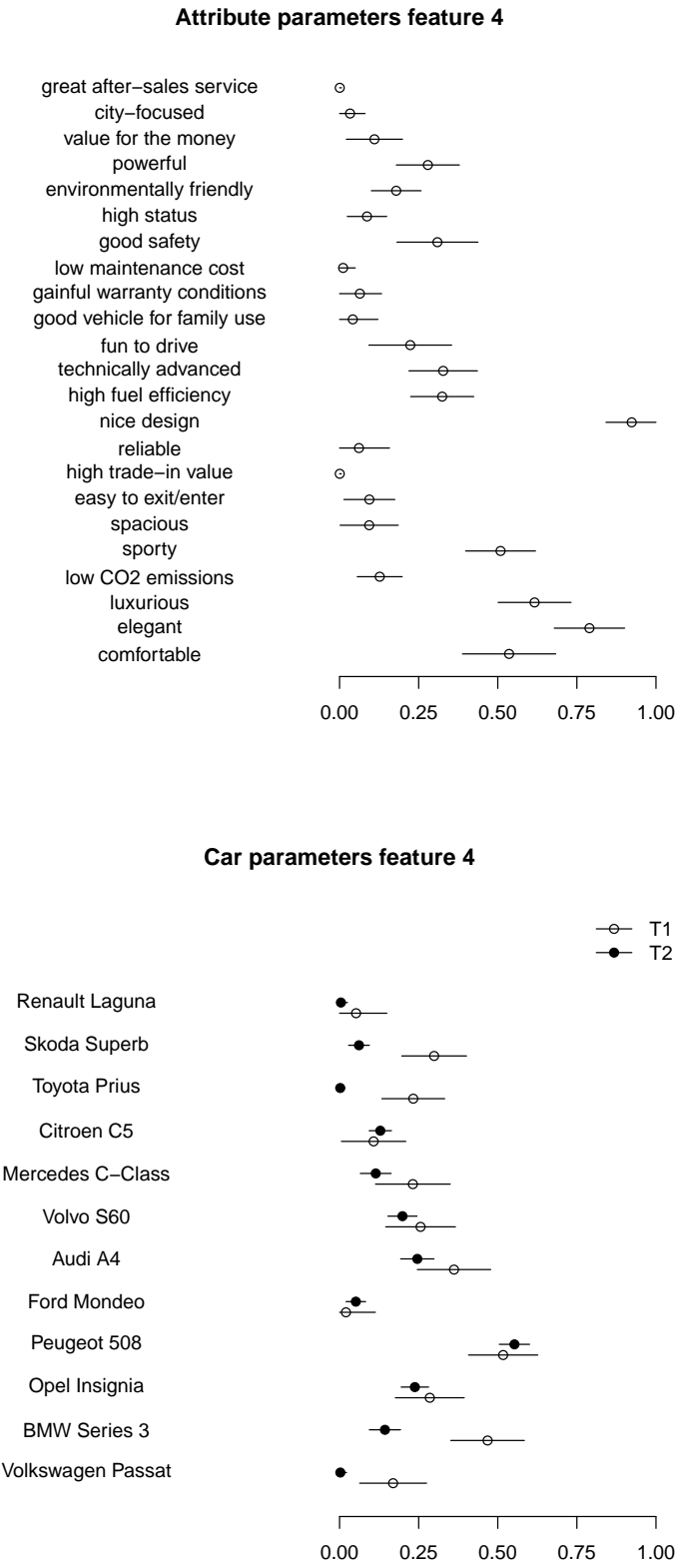


Figure 4: Point estimates and 95% confidence interval for attribute parameters (top panel) and car parameters (bottom panel) of feature 4.



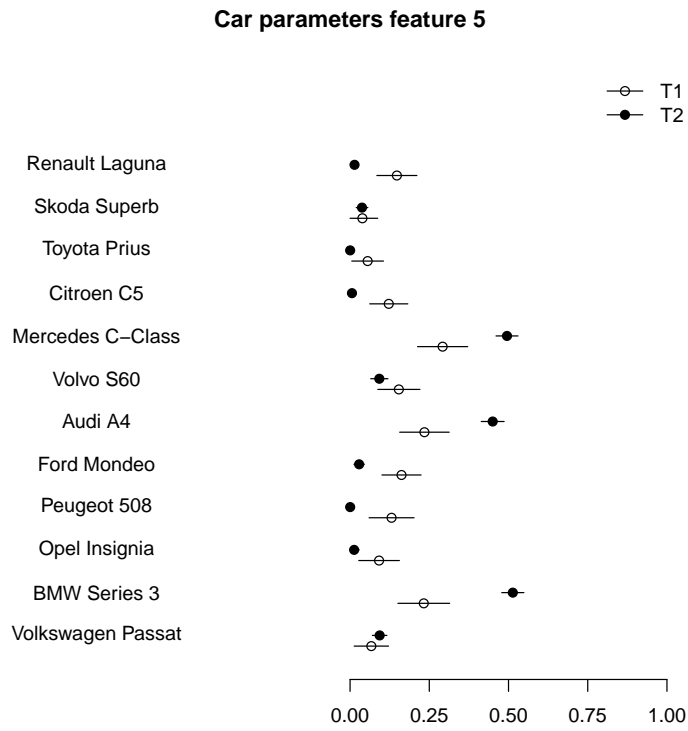
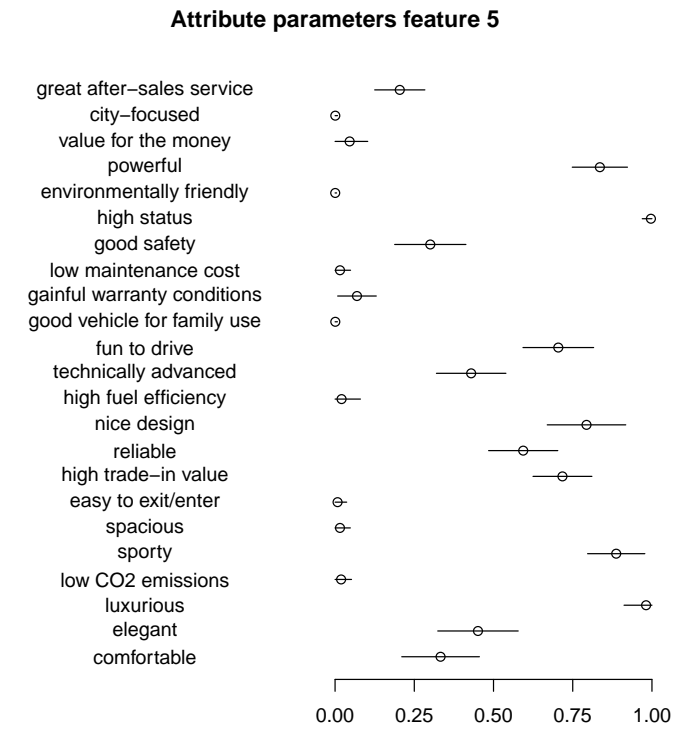


Figure 5: Point estimates and 95% confidence interval for attribute parameters (top panel) and car parameters (bottom panel) of feature 5.

Behavior	F1	F2	F3	F4	F5	F6	F7
Fly off the handle	$\rho_{11}$	0	0	0	0	0	0
Start a fight	$\rho_{21}$	0	0	0	0	0	0
Avoid a confrontation	0	$\rho_{32}$	0	0	0	0	0
Go out of the other's way	0	$\rho_{42}$	0	0	0	0	0
Unburden your heart to others	0	0	$\rho_{53}$	0	0	0	0
Tell others what has happened	0	0	$\rho_{63}$	0	0	0	0
Say what is bothering you	0	0	0	$\rho_{74}$	0	0	0
Calmly explain what is bothering you	0	0	0	$\rho_{84}$	0	0	0
Show something bothers you without speaking	0	0	0	0	$\rho_{95}$	0	0
Start to sulk	0	0	0	0	$\rho_{10,5}$	0	0
Suppress your anger	0	0	0	0	0	$\rho_{11,6}$	0
Bottle up your anger	0	0	0	0	0	$\rho_{12,6}$	0
Reconcile	0	0	0	0	0	0	$\rho_{14,7}$
Talk things out	0	0	0	0	0	0	$\rho_{14,7}$

Table 2: Matrix of behavior parameters assuming latent features reflect the behavior types hypothesized in the study.

course of the estimation process

```
R> update.att <- matrix(c(
  1, 0, 0, 0, 0, 0, 0,
  1, 0, 0, 0, 0, 0, 0,
  0, 1, 0, 0, 0, 0, 0,
  0, 1, 0, 0, 0, 0, 0,
  0, 0, 1, 0, 0, 0, 0,
  0, 0, 1, 0, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0,
  0, 0, 0, 1, 0, 0, 0,
  0, 0, 0, 0, 1, 0, 0,
  0, 0, 0, 0, 1, 0, 0,
  0, 0, 0, 0, 0, 1, 0,
  0, 0, 0, 0, 0, 0, 1,
  0, 0, 0, 0, 0, 0, 1), ncol = 7, byrow = T)
```

Finally, we use the function `LCplfm` to estimate disjunctive seven-feature LCPLFMs with class specific situation parameters ( $T = 1, \dots, 3$ ) and with a constant situation-feature classification per person (i.e., model m1). For each model  $M$  (= 50) runs using random starting points are conducted. To constrain part of the behavior parameters to equal 0 as indicated in Table 2, we specify starting values close to zero (i.e.,  $10^{-6}$ ) for parameters that are not updated in the course of the estimation process.

In particular, the following code can be used to subsequently estimate models with one up to three latent classes

```
R> set.seed(571125)
R> start.att1 <- array(runif(14 * 7 * 50), c(14, 7, 50))
```

```

R> start.att1[update.att %>% rep(1, 50) == 0] <- 1e-6
R> m1.constr.T1 <- LCplfm(data = anger2$data, M = 50, F = 7, T = 1, printrun
+   = TRUE, model = 1, update.attributeParameters = update.att,
+   start.attributeParameters = start.att1, Nbootstrap = 5000)

R> set.seed(837182)
R> start.att2 <- array(runif(14 * 7 * 50), c(14, 7, 50))
R> start.att2[update.att %>% rep(1, 50) == 0] <- 1e-6
R> m1.constr.T2 <- LCplfm(data = anger2$data, M = 50, F = 7, T = 2, printrun
+   = TRUE, model = 1, update.attributeParameters = update.att,
+   start.attributeParameters = start.att2, Nbootstrap = 5000)

R> set.seed(155983)
R> start.att3 <- array(runif(14 * 7 * 50), c(14, 7, 50))
R> start.att3[update.att %>% rep(1, 50) == 0 ] <- 1e-6
R> m1.constr.T3 <- LCplfm(data = anger2$data, M = 50, F = 7, T = 3, printrun
+   = TRUE, model = 1, update.attributeParameters = update.att,
+   start.attributeParameters = start.att3, Nbootstrap = 5000)

```

Inspection of the results shows that models with 1, 2 and 3 latent classes have AIC values equal to 16265, 16051 and 15941 and BIC values equal to 16476, 16438 and 16504, respectively. Hence the three class model has lowest AIC and the two-class model has lowest BIC. Furthermore, for models with 1, 2 and 3 classes 85%, 89% and 91% of the OR dependencies between object-attribute pairs with a common object are in the simulated 99% CI and 53%, 64% and 73% of the OR dependencies between pairs with a common attribute are in the simulated 99% CI. Finally, models with 1, 2 and 3 latent classes all explain 93% of the variance in the observed situation-behavior frequency table. Although the three-class model has lowest AIC and yields a somewhat better fit to observed OR dependencies, we will further discuss the two-class model (with lowest BIC) as the latent classes in this model are easier to interpret. Note that the first and second latent class contain respectively 34% and 66% of the students in the sample.

To evaluate whether the behaviors included in the study actually measure the hypothesised behavior types of the study design, we may inspect the estimated behavior parameters and the corresponding standard errors.

```
R> round(m1.constr.T2$attpar, 2)
```

yields

	F1	F2	F3	F4	F5	F6	F7
fly off the handle	0.72	0.00	0.00	0.00	0.00	0.00	0.00
fight	0.55	0.00	0.00	0.00	0.00	0.00	0.00
avoid	0.00	0.67	0.00	0.00	0.00	0.00	0.00
leave	0.00	0.81	0.00	0.00	0.00	0.00	0.00
unburden heart	0.00	0.00	0.81	0.00	0.00	0.00	0.00
tell story	0.00	0.00	0.91	0.00	0.00	0.00	0.00
explain in direct/sober way	0.00	0.00	0.00	0.76	0.00	0.00	0.00

calmly explain	0.00	0.00	0.00	0.70	0.00	0.00	0.00
show that something bothers you	0.00	0.00	0.00	0.00	0.62	0.00	0.00
sulk	0.00	0.00	0.00	0.00	0.29	0.00	0.00
suppress anger	0.00	0.00	0.00	0.00	0.00	0.86	0.00
bottle up anger	0.00	0.00	0.00	0.00	0.00	0.59	0.00
reconcile	0.00	0.00	0.00	0.00	0.00	0.00	0.91
talk things out	0.00	0.00	0.00	0.00	0.00	0.00	0.81

and

```
R> round(m1.constr.T2$SE.attpar, 2)
```

yields

	F1	F2	F3	F4	F5	F6	F7
fly off the handle	0.02	0.00	0.00	0.00	0.00	0.00	0.00
fight	0.02	0.00	0.00	0.00	0.00	0.00	0.00
avoid	0.00	0.02	0.00	0.00	0.00	0.00	0.00
leave	0.00	0.02	0.00	0.00	0.00	0.00	0.00
unburden heart	0.00	0.00	0.02	0.00	0.00	0.00	0.00
tell story	0.00	0.00	0.01	0.00	0.00	0.00	0.00
explain in direct/sober way	0.00	0.00	0.00	0.02	0.00	0.00	0.00
calmly explain	0.00	0.00	0.00	0.02	0.00	0.00	0.00
show that something bothers you	0.00	0.00	0.00	0.00	0.02	0.00	0.00
sulk	0.00	0.00	0.00	0.00	0.02	0.00	0.00
suppress anger	0.00	0.00	0.00	0.00	0.00	0.02	0.00
bottle up anger	0.00	0.00	0.00	0.00	0.00	0.02	0.00
reconcile	0.00	0.00	0.00	0.00	0.00	0.00	0.01
talk things out	0.00	0.00	0.00	0.00	0.00	0.00	0.02

Hence, we can conclude that behavior parameters are accurately estimated (i.e., standard errors are small) and that the link between the behaviors and the assumed behavior type is generally rather strong, except for the behavior ‘starting to sulk’, which is only weakly linked (.29) to the intended behavior type (i.e., indirect behavior).

To study to what extent situations affect the occurrence of specific behavior types and to evaluate to what extent these situation-behavior relations differ across persons, we visualize the situation-feature parameters using the plot method of the LCplfm function:

```
R> par(mfrow = c(2, 2), pty = "s")
R> plot(m1.constr.T2, element = "object", feature = 1, main = "anger-out",
+   positionlabel = -0.5, xlegend = "topright", y.intersplegend = 0.7,
+   cexsymb = 0.8)
R> plot(m1.constr.T2, element = "object", feature = 4, main =
+   "assertive behavior", positionlabel = -0.5, xlegend = "topright",
+   y.intersplegend = 0.7, cexsymb = 0.8)
R> plot(m1.constr.T2, element = "object", feature = 7, main =
+   "reconciliation", positionlabel = -0.5, xlegend = "topright",
```

```

+   y.intersplegend = 0.7, cexsymb = 0.8)

R> par(mfrow = c(2, 2), pty = "s")
R> plot(m1.constr.T2, element = "object", feature = 2, main = "avoidance",
+   positionlabel = -0.5, xlegend = 0.4, ylegend = 45.5, y.intersplegend =
+   0.7, cexsymb = 0.8)
R> plot(m1.constr.T2, element = "object", feature = 5, main =
+   "indirect behavior", positionlabel = -0.5, xlegend = 0.4, ylegend =
+   45.5, y.intersplegend = 0.7, cexsymb = 0.8)
R> plot(m1.constr.T2, element = "object", feature = 6, main = "anger-in",
+   positionlabel = -0.5, xlegend = 0.4, ylegend = 45.5, y.intersplegend =
+   0.7, cexsymb = 0.8)
R> plot(m1.constr.T2, element = "object", feature = 3, main =
+   "social sharing", positionlabel = -0.5, xlegend = 0.4, ylegend = 45.5,
+   y.intersplegend = 0.7, cexsymb = 0.8)

```

Figure 6 shows the situation parameters for so-called approach behaviors (i.e., anger-out, assertion and reconciliation) and Figure 7 displays the situation parameters for so-called avoidance behaviors (i.e., avoidance, indirect behavior, anger-in) and social sharing.

As can be seen in Figure 6 and Figure 7 the probability that a certain behavior type is considered appropriate in a situation by a person depends on the latent class the person belongs to and on situation characteristics. First, it turns out that the two latent classes can be interpreted as an ‘approach’ and an ‘avoidance’ class. That is, when being angry at someone, persons in class 1 are generally more likely to consider approach behaviors appropriate, and persons in class 2 are more likely to consider avoidance behaviors and social sharing appropriate. Second, situation characteristics have a systematic effect on the extent to which behavior types are considered appropriate in the situation. Anger-in behaviors, for instance, are more considered appropriate if the status of the person one is angry with increases. In the same way, anger-out behaviors are more considered appropriate if the status of the person one is angry with decreases. Assertive behaviors are more likely considered adequate if the person one is angry with is liked or if he/she has a lower status. Reconciliation is especially considered appropriate if one likes the person one is angry with and it is considered inappropriate if the person one is angry with has a higher/equal status and is not liked/unfamiliar.

## 7. Example: Simulated data

When applying LCPLFMs one may wonder to what extent a model using a specific decision rule can fit data generated by another decision rule, and relatedly, which decision rule is more flexible to fit data generated by other decision rules. Furthermore, one may wonder about the quality of the parameter recovery when the fitted model involves a correct or incorrect decision rule.

To research these questions, we conduct a small simulation study. Data generated using a disjunctive, conjunctive or additive decision rule are fitted using models with a disjunctive, conjunctive or additive rule. Apart from the decision rule, true and fitted models are assumed to be equivalent. That is, true and fitted models always have the same number of features  $F$  ( $F = 1, \dots, 4$ ) and always are of type m1 (i.e., assuming constant object-feature classification

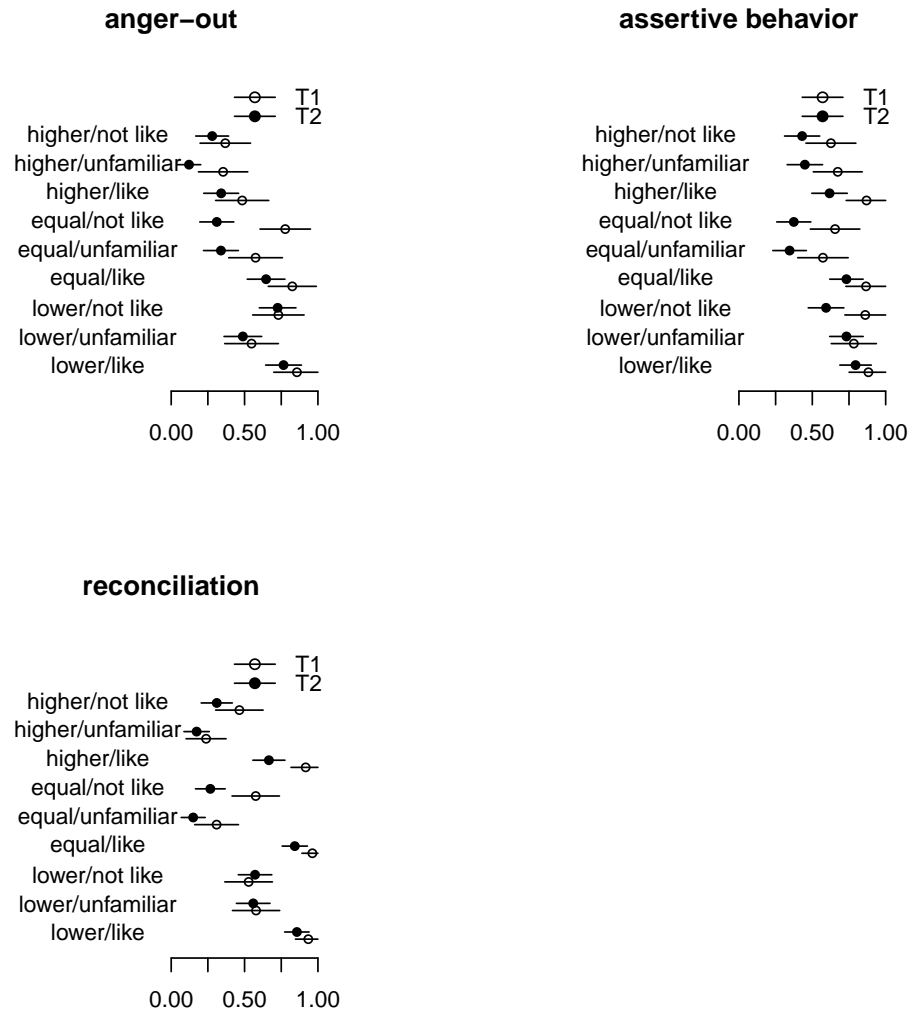


Figure 6: Point estimates and 95% confidence interval of situation parameters for approach behaviors anger-out, assertion and reconciliation.

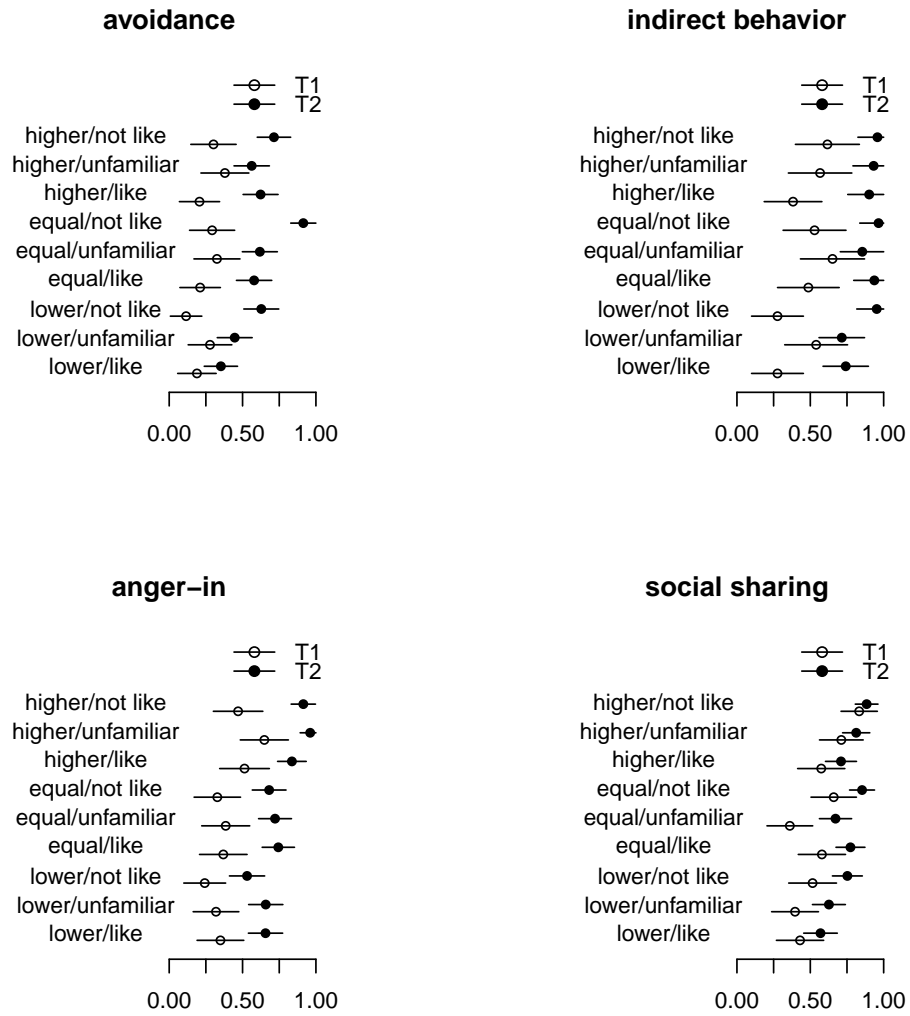


Figure 7: Point estimates and 95% confidence interval of situation parameters for avoidance behaviors (avoidance, indirect behavior, anger-in) and social sharing.

and class-specific object parameters) with  $T=2$ . For each of 12 true models (i.e., disjunctive, conjunctive and additive models with 1 up to 4 features) 100 data sets are generated using randomly sampled true parameter values and assuming 10 ( $= J$ ) objects, 15 ( $= K$ ) attributes and 200 ( $= I$ ) persons. Each model is fitted using  $M=40$  runs of the EM-algorithm with random starting points.

The following code illustrates the simulations that are conducted for the true disjunctive model with 2 features:

```
R> J <- 10
R> K <- 15
R> I <- 200
R> F <- 2
R> T <- 2
R> M <- 40
R> Niter <- 100
R> outgenDC.F2 <- array(rep(0, Niter * 11 * 3), c(Niter, 3, 11))
R> dimnames(outgenDC.F2)[[2]] <- c("DC", "CD", "ADD")
R> set.seed(972645)

R> for (iter in 1 : Niter){
+   objpar <- array(runif(J * F * T),c(J, F, T))
+   attpar <- matrix(runif(K * F), c(K, F))
+   sizepar <- rep(1 / T, T)

+   gendatDC <- gendatLCplfm(N = I, objpar = objpar, attpar =
+     attpar, sizepar = sizepar, maprule = "disj", model = 1)

+   mDC <- LCplfm(data = gendatDC$data, F = F, T = T, model = 1, maprule =
+     "disj", M = M)

+   mCD <- LCplfm(data = gendatDC$data, F = F, T = T, model = 1, maprule =
+     "conj", M = M)

+   mADD <- LCplfm(data = gendatDC$data, F = F, T = T, model = 1, maprule =
+     "add", M = M)

+   outgenDC.F2[iter, 1, ] <- mDC$fitmeasures
+   outgenDC.F2[iter, 2, ] <- mCD$fitmeasures
+   outgenDC.F2[iter, 3, ] <- mADD$fitmeasures}
```

Figure 8 displays a boxplot of the likelihood values for disjunctive, conjunctive and additive models fitted to data generated from a disjunctive, conjunctive or additive model. Note that the models presented in each boxplot always have the same number of parameters and so likelihood values are a valid basis for model comparison.

First, the results show that one-feature disjunctive and additive models fit data generated by the one-feature additive (or disjunctive) model equally well. This could be expected because one-feature additive and disjunctive models are formally equivalent (i.e.,  $C^{DC}(\mathbf{x}_{ij}, \mathbf{y}_{ijk}) =$



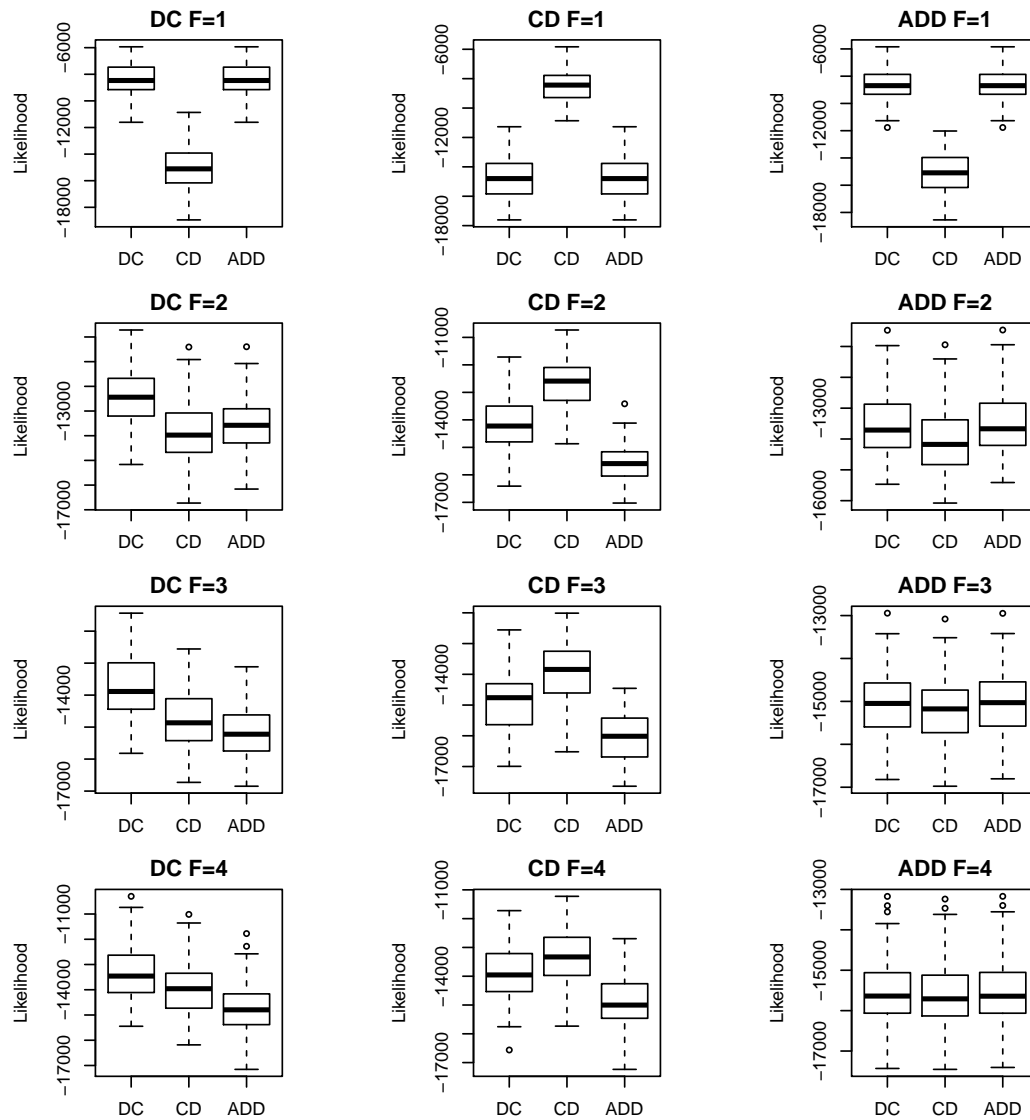


Figure 8: Boxplot of likelihood values for disjunctive (DC), conjunctive (CD) and additive (ADD) models fitted to data generated from a disjunctive (DC, left column), conjunctive (CD, center column) or additive model (ADD, right column). True and fitted models have the same number of features  $F$  ( $F = 1, \dots, 4$ ) and are always of type m1 with  $T = 2$ .

$C^{ADD}(\mathbf{x}_{ij}, \mathbf{y}_{ijk}) = x_{ij1}y_{ijk1}$ ). On the other hand, data generated by the one-feature conjunctive model cannot be fitted well by a disjunctive or additive model, and data generated by the one feature disjunctive and additive model cannot be fitted well by the one-feature conjunctive model.

Second the results indicate that, when the true model is disjunctive or conjunctive, models with a correct decision rule will outperform models with an incorrect decision rule. Furthermore, additive models fit data generated from disjunctive (conjunctive) models less well than conjunctive (disjunctive) models.

Third, the results indicate that when the true model is additive, the additive model only has a slightly better fit than the disjunctive or conjunctive model. In fact, especially the disjunctive model can fit data generated by the additive model well. Moreover, the difference in fit between additive, disjunctive and conjunctive models decreases if the number of features increases.

In sum, we may conclude that when  $F > 1$  and with an increasing number of features, additive models become less flexible to fit data generated from disjunctive or conjunctive models. In addition, with an increasing number of features in the true model, conjunctive and especially disjunctive models can fit data generated by the additive model rather well. Finally, we note that other fit measures as the proportion of variance accounted for in the  $J \times K$  table and the proportion of OR dependencies between object-attribute pairs with a common object or attribute that are in the simulated 95% CI lead to the same conclusions.

Finally, further inspection of (a sample of) the simulations shows that parameter recovery is usually very good (i.e., correlations between true and estimated object/attribute parameters of .99) when the decision rule of the fitted model is also used for generating the data. Note however, that for models with  $F > 2$  it may be required to (manually) switch the feature/class labels of the estimated model so that they match the true feature/class labels.

On the other hand, when the decision rule of the estimated model is not the same as the decision rule of the true model, parameter recovery of object/attribute parameters is usually poor. In particular, fitting a disjunctive (conjunctive) model to data generated by a conjunctive (disjunctive) model often leads to a poor recovery of object and attribute parameters. This is probably because the probability of an observed association is a very different function of the object and attribute parameters in disjunctive and conjunctive models. On the other hand, when fitting a disjunctive (additive) model to data generated by an additive (disjunctive) model, correlations between true and estimated parameters may be rather high (i.e.,  $>0.85$ ), but estimated object/attribute parameters may have considerable bias. Finally, note that recovery of true latent class sizes is generally quite good, also if the incorrect decision rule is used.

To summarize, parameter recovery of object/attribute parameters is expected to be satisfactory if the true model involving the correct decision rule is used for fitting the data and is expected to be poor if a model with an incorrect decision rule is fitted. Note that the poor recovery of models with an incorrect decision rule should not be considered problematic because the simulation study shows that models with a correct decision rule will generally fit better than models with an incorrect decision rule. In addition, a meaningful substantive interpretation of the features should also help to select the appropriate decision rule.

## 8. Discussion

For LCPLFMs several types of model extensions could be interesting to pursue in future research. A first extension is related to modeling correlations among the latent object or attribute variables. Currently, the models included in the `LCplfm` function all use an independence model for the marginal distribution of the latent feature patterns (see Equation 2 and Equation 5). Using an independence model has the important advantage that models are parsimonious and that objects and attributes are described using a limited number of parameters. However, if needed correlations among latent object or attribute variables can be modeled in several ways. As an example we consider modeling correlations among the latent object variables in models of type `m1`. A first approach to model correlations among the latent variables  $X_{ij1}, \dots, X_{ijF}$  is to specify a saturated model for the  $2^F$  patterns  $\mathbf{x}_{ij}$ , that is  $p(\mathbf{x}_{ij}|\boldsymbol{\sigma}_j) = \sigma_{jq}$  with  $q = 1, \dots, 2^F$  and  $\sum_q \sigma_{jq} = 1$  (Maris 1999). A disadvantage of this approach is that the number of parameters increases rapidly as a function of the number of latent features, leading to overparameterized models. A second approach is to model the multinomial probability  $p(\mathbf{x}_{ij}|\boldsymbol{\sigma}_j)$  with a multinomial logit model including main-effects, second-order interactions, third-order interactions and so on (Vermunt and Magidson 2005). For instance, the following multinomial logit model includes main-effects and second-order interactions among the latent variables  $X_{ij1}, \dots, X_{ijF}$ :

$$p(\mathbf{x}_{ij}|\boldsymbol{\sigma}_j) = \frac{\exp\left(\sum_{f \in h} x_{ijf} \sigma_{jff} + \sum_{f \in h, f' \in h, f < f'} x_{ijf} x_{ijf'} \sigma_{jff'}\right)}{\sum_{\mathbf{x}_{ij}} \exp\left(\sum_{f \in h} x_{ijf} \sigma_{jff} + \sum_{f \in h, f' \in h, f < f'} x_{ijf} x_{ijf'} \sigma_{jff'}\right)}$$

with  $h = \{1, \dots, F\}$ . LCPLFMs using this kind of parameterization to include interactions among latent variables can be estimated using the syntax module of Latent Gold (Vermunt and Magidson 2008). For an example of estimating LCPLFMs with Latent Gold see (Meulders *et al.* 2013). A possible drawback of models including interactions among latent variables is that selecting the model with the appropriate correlation structure may be difficult as there are many alternative models. A third approach to model correlation among the latent variables is to use a mixture of independence models with class-specific parameters, that is, assuming a mixture with  $Q$  components  $q = 1, \dots, Q$ :

$$p(\mathbf{x}_{ij}|\boldsymbol{\sigma}_j) = \sum_{q=1}^Q \gamma_q \prod_f (\sigma_{jfq})^{x_{ijf}} (1 - \sigma_{jfq})^{1-x_{ijf}}.$$

LCPLFMs that use a mixture to model the marginal distribution of the latent feature patterns can be estimated using the syntax module of Latent Gold (Vermunt and Magidson 2008).

A second extension that may be of interest is adapting LCPLFMs for other data structures. The models that are now included in the `LCplfm` function have been developed to analyze fully crossed binary three-way three-mode data. However, it could for instance be interesting to adapt the models to the case of nested binary two-way three-mode data. An example of such data are respondents  $i$  ( $i = 1, \dots, I_j$ ) of country  $j$  ( $j = 1, \dots, J$ ) who indicate for  $k$  ( $k = 1, \dots, K$ ) aspects of national pride (e.g., the performance of the economy, the quality of the social security system, quality of the educational system, an effective health-care system, a powerful army, past achievements in sports, science etc.) whether or not they take pride in their country with respect to this aspect. Let the binary variable  $D_{ijk}$  equal 1 if respondent  $i$

of country  $j$  is proud of his/her country with respect to aspect  $k$ , and 0 otherwise. Using an LCPLFM to model the data we assume that pride in the  $K$  aspects can be explained from  $F$  more basic sources of national pride (e.g., being proud about past achievements, being proud about nowadays achievements, etc.). Suppose the binary latent variable  $X_{ijf} \sim \text{Bern}(\sigma_{jf})$  equals 1 if respondent  $i$  of country  $j$  takes pride in the basic source  $f$ , and 0 otherwise. Furthermore, the latent variable  $Y_{ijkf} \sim \text{Bern}(\rho_{kf})$  equals 1 if aspect  $k$  measures the basic source of pride  $f$  when respondent  $i$  of country  $j$  judges statement  $k$ , and 0 otherwise. Using a disjunctive LCPLFM we assume that a respondent is proud of an aspect if he/she takes pride in at least one basic source of pride measured by the aspect. An LCPLFM that models dependencies among the responses of each respondent has the following likelihood:

$$p(\mathbf{d}|\boldsymbol{\sigma}, \boldsymbol{\rho}) = \prod_{j=1}^J \prod_{i=1}^{I_j} \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij}|\boldsymbol{\sigma}) \prod_{k=1}^K p(\mathbf{d}_{ijk}|\mathbf{x}_{ij}, \boldsymbol{\rho}_k).$$

The previous model assumes measurement invariance because aspect-feature parameters  $\rho_{kf}$  are the same for respondents of different countries. The measurement invariance assumption can now be relaxed by assuming that respondents belong to latent classes with class-specific aspect-feature parameters. Let  $Z_{ijt}$  be equal to 1 if respondent  $i$  of country  $j$  belongs to class  $t$  ( $t = 1, \dots, T$ ), and 0 otherwise. Furthermore it is assumed that  $P(Z_{ijt} = 1|\boldsymbol{\xi}) = \xi_t$  with  $\sum_t \xi_t = 1$  and that  $Y_{ijkf}|Z_{ijt} = 1 \sim \text{Bern}(\rho_{kft})$ . An LCplfm that models dependencies among the responses of each respondent, and that relaxes the assumption of measurement invariance by assuming class-specific aspect-feature parameters has the following likelihood:

$$p(\mathbf{d}|\boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\xi}) = \prod_{j=1}^J \prod_{i=1}^{I_j} \sum_t \xi_t \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij}|\boldsymbol{\sigma}) \prod_{k=1}^K p(\mathbf{d}_{ijk}|\mathbf{x}_{ij}, Z_{ijt} = 1, \boldsymbol{\rho}_{kt}).$$

Third, the present paper uses a latent class extension of PLFMs for modeling qualitative individual differences in object or attribute classification (i.e., the pattern of features a person assigns to an object or attribute) and for modeling person heterogeneity in object and/or attribute parameters. As PLFMs are actually constrained latent class models, using a latent class approach for modeling qualitative individual differences and person heterogeneity in the model parameters was straightforward. An advantage of using a latent class approach is that individual differences can be described in terms of person types. In domains as personality psychology or marketing this may be a natural way to proceed. However, a disadvantage of the latent class approach is that many local maxima may exist leading to many different solutions with a comparable fit. Furthermore, the assumption that reality is categorical may be unrealistic. As alternative, it could be interesting to investigate how a random effects approach can be used for modeling person heterogeneity in object or attribute parameters of (LC)PLFMs. Furthermore, it would be interesting to compare existing LCPLFMs or future random effects extensions with (random effects) structural equations models that have been developed for binary three-way three-mode data (González, De Boeck, and Tuerlinckx 2008).

Finally, to speed up the computation of LCPLFMs it would be interesting to use parallel computing in the `LCplfm` function when using the EM algorithm to estimate  $M$  candidate models from random starting points (Hofert and Mächler 2016; Schmidberger, Morgan, Eddebuettel, Yu, Tierney, and Mansmann 1978).

## References

- Akaike H (1974). “A New Look at the Statistical Model Identification.” *IEEE Transactions on Automatic Control*, **19**, 716–723.
- Ares G, Antúnez L, Giménez A, Roigard CM, Pineau B, Hunter DC, Jaeger SR (2014). “Further Investigations into the Reproducibility of Check-All-That-Apply (CATA) Questions for Sensory Product Characterization Elicited by Consumers.” *Food Quality and Preference*, **36**, 111–121.
- Ares G, Deliza R, Barreiro C, Giménez A, Gámbaro A (2010). “Comparison of Two Sensory Profiling Techniques Based on Consumer Perception.” *Food Quality and Preference*, **21**, 417–426.
- Ares G, Jaeger SR (2013). “Check-All-That-Apply Questions: Influence of Attribute Order on Sensory Product Characterization.” *Food Quality and Preference*, **28**, 141–153.
- Basford KE, McLachlan G (1985). “The Mixture Method of Clustering Applied to Three-Way Data.” *Journal of Classification*, **2**, 109–125.
- Benaglia T, Chauveau D, Hunter DR, Young D (2009). “**mixtools**: An R Package for Analyzing Finite Mixture Models.” *Journal of Statistical Software*, **32**(6), 1–29. doi:10.18637/jss.v032.i06. URL <http://www.jstatsoft.org/v32/i06/>.
- Candel MJJM, Maris E (1997). “Perceptual Analysis of Two-Way Two-Mode Frequency Data: Probability Matrix Decomposition and Two Alternatives.” *International Journal of Research in Marketing*, **14**, 321–339.
- Carroll JD, Chang JJ (1970). “Analysis of Individual Differences in Multidimensional Scaling via an n-Way Generalization of Eckart-Young Decomposition.” *Psychometrika*, **35**, 283–319.
- Ceulemans E, Kuppens P, Van Mechelen I (2012). “Capturing the Structure of Distinct Types of Individual Differences in the Situation-Specific Experience of Emotions: The Case of Anger.” *European Journal of Personality*, **26**, 484–495.
- Ceulemans E, Van Mechelen I (2004). “Tucker2 Hierarchical Classes Analysis.” *Psychometrika*, **69**(3), 375–399.
- Ceulemans E, Van Mechelen I, Leenen I (2003). “Tucker3 Hierarchical Classes Analysis.” *Psychometrika*, **68**(3), 413–433.
- De Boeck P, Rosenberg S (1988). “Hierarchical Classes: Model and Data Analysis.” *Psychometrika*, **53**, 361–381.
- de la Torre J (2008). “Model Evaluation and Multiple Strategies in Cognitive Diagnosis: An Analysis of Fraction Subtraction Data.” *Psychometrika*, **73**(4), 595–624.
- De Roover K, Ceulemans E, Timmerman ME (2012a). “How to Perform Multiblock Component Analysis in Practice.” *Behavior Research Methods*, **44**, 41–56.

- De Roover K, Ceulemans E, Timmerman ME, Nezlek JB, Onghena P (2013a). “Modeling Differences in the Dimensionality of Multiblock Data by Means of Clusterwise Simultaneous Component Analysis.” *Psychometrika*, **129**, 98–107.
- De Roover K, Ceulemans E, Timmerman ME, Vansteelandt K, Stouten J, Onghena P (2012b). “Clusterwise Simultaneous Component Analysis for Analyzing Structural Differences in Multivariate Multiblock Data.” *Psychological Methods*, **17**(1), 100–119.
- De Roover K, Timmerman ME, Van Mechelen I, Ceulemans E (2013b). “On the Added Value of Multiset Methods for Three-Way Data Analysis.” *Chemometrics and Intelligent Laboratory Systems*, **129**, 98–107.
- DeSarbo WS, Grewal R, Scott CJ (2008). “A Clusterwise Bilinear Multidimensional Scaling Methodology for Simultaneous Segmentation and Positioning Analyses.” *Journal of Marketing Research*, **45**, 280–292.
- Formann AK (1992). “Linear Logistic Latent Class Analysis for Polytomous Data.” *Journal of the American Statistical Association*, **87**, 476–486.
- Fraley C, Raftery AE (2002). “Model-Based Clustering, Discriminant Analysis and Density Estimation.” *Journal of the American Statistical Association*, **97**, 611–631.
- Giordani P, Kiers HAL, Del Ferraro MA (2014). “Three-Way Component Analysis Using the R Package **ThreeWay**.” *Journal of Statistical Software*, **57**(7), 1–23. doi:10.18637/jss.v057.i07. URL <http://www.jstatsoft.org/v57/i07/>.
- González J, De Boeck P, Tuerlinckx F (2008). “A Double-Structure Structural Equation Model for Three-Mode Data.” *Psychological Methods*, **13**(4), 337–353.
- Grün B, Leisch F (2007). “Fitting Finite Mixtures of Generalized Linear Regressions in R.” *Computational Statistics & Data Analysis*, **51**(11), 5247–5252. doi:10.1016/j.csda.2006.08.014.
- Grün B, Leisch F (2008). “**FlexMix** Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters.” *Journal of Statistical Software*, **28**(4), 1–35. doi:10.18637/jss.v028.i04. URL <http://www.jstatsoft.org/v28/i04/>.
- Harshman RA (1970). “Foundations of the Parafac Procedure: Models and Conditions for an Exploratory Multimodal Factor Analysis.” *UCLA Working Papers in Phonetics*, **16**, 1–84.
- Helwig NE (2013). “The Special Sign Indeterminacy of the Direct-Fitting Parafac2 Model: Some Implications, Cautions, and Recommendations for Simultaneous Component Analysis.” *Psychometrika*, **78**(4), 725–739.
- Helwig NE (2017). *multiway: Component Models for Multi-Way Data*. R package version 1.0-3, URL <https://CRAN.R-project.org/package=multiway>.
- Hofert M, Mächler M (2016). “Parallel and Other Simulations in R Made Easy: An End-to-End Study.” *Journal of Statistical Software*, **69**(4), 1–44.
- Hunt LA, Basford KE (1999). “Fitting a Mixture Model to Three-Mode Three-Way Data with Categorical and Continuous Variables.” *Journal of Classification*, **16**, 283–296.

- Hunt LA, Basford KE (2001). “Fitting a Mixture Model to Three-Mode Three-Way Data with Missing Information.” *Journal of Classification*, **18**, 209–226.
- Jaeger SR, Chheang SL, Yin Y, Bava CM, Giménez A, Vidal L, Ares G (2013). “Check-All-That-Apply (CATA) Responses Elicited by Consumers: Within-Assessor Reproducibility and Stability of Sensory Product Characterizations.” *Food Quality and Preference*, **30**, 56–67.
- Kuppens P, Van Mechelen I, Meulders M (2004). “Every Cloud Has a Silver Lining: Interpersonal and Individual Differences Determinants of Anger-Related Behaviors.” *Personality and Social Psychology Bulletin*, **30**, 1550–1564.
- Leenen I, Van Mechelen I, De Boeck P (1999). “INDCLAS: A Three-Way Hierarchical Classes Model.” *Psychometrika*, **64**(1), 9–24.
- Leibovici DG (2010). “Spatio-Temporal Multiway Decompositions Using Principal Tensor Analysis on k-Modes: The R Package **PTAk**.” *Journal of Statistical Software*, **34**(10), 1–34. doi:10.18637/jss.v034.i10. URL <http://www.jstatsoft.org/v34/i10/>.
- Leisch F (2004). “**FlexMix**: A General Framework for Finite Mixture Models and Latent Class Regression in R.” *Journal of Statistical Software*, **11**(8), 1–18. doi:10.18637/jss.v011.i08. URL <http://www.jstatsoft.org/v11/i08/>.
- Liu W, Yang Y (2011). “Parametric or non-parametric? A parametricness index for model selection.” *The Annals of Statistics*, **39**(4), 2074–2102.
- Maris E (1999). “Estimating Multiple Classification Latent Class Models.” *Psychometrika*, **64**, 187–212.
- Maris E, De Boeck P, Van Mechelen I (1996). “Probability Matrix Decomposition Models.” *Psychometrika*, **61**, 7–29.
- Meilgaard M, Civille GV, Carr BT (1999). *Sensory Evaluation Techniques*. 5th edition. CRC Press, Boca Raton, FL.
- Meulders M (2013). “An R package for Probabilistic Latent Feature Analysis of Two-Way Two-Mode Frequencies.” *Journal of Statistical Software*, **54**(14), 1–29. doi:10.18637/jss.v054.i14. URL <https://www.jstatsoft.org/article/view/v054i14>.
- Meulders M, De Boeck P, Kuppens P, Van Mechelen I (2002). “Constrained Latent Class Analysis of Three-Way Three-Mode Data.” *Journal of Classification*, **19**, 277–302.
- Meulders M, De Boeck P, Van Mechelen I (2003). “A Taxonomy of Latent Structure Assumptions for Probability Matrix Decomposition Models.” *Psychometrika*, **68**, 61–77.
- Meulders M, De Boeck P, Van Mechelen I, Gelman A (2005). “Probabilistic Feature Analysis of Facial Perception of Emotions.” *Applied Statistics*, **54**, 781–793.
- Meulders M, De Boeck P, Van Mechelen I, Gelman A, Maris E (2001). “Bayesian Inference with Probability Matrix Decomposition Models.” *Journal of Educational and Behavioral Statistics*, **26**, 153–179.



- Meulders M, Tuerlinckx F, Vanpaemel W (2013). “Constrained Multilevel Latent Class Models for the Analysis of Three-Way Three-Mode Binary Data.” *Journal of Classification*, **30**(3), 306–337.
- Mischel W, Shoda Y, Mendoza-Denton R (2002). “Situation-Behavior Profiles as a Locus of Consistency in Personality.” *Current Directions in Psychological Science*, **11**, 50–54.
- Monteiro CMF, Dibb S, Almeida LT (2012). “Revealing Doctors’ Prescribing Choice Dimensions with Multivariate Tools: A Perceptual Mapping Approach.” *European Journal of Operational Research*, **201**, 909–920.
- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Schmidberger M, Morgan M, Eddelbuettel D, Yu H, Tierney L, Mansmann U (1978). “State of the Art in Parallel Computing with R.” *Journal of Statistical Software*, **31**(1), 1–27.
- Schwarz G (1978). “Estimating the Dimensions of a Model.” *The Annals of Statistics*, **6**, 461–464.
- Torres A, Bijmolt THA (2009). “Assessing Brand Image through Communalities and Asymmetries in Brand-to-Attribute and Attribute-to-Brand Associations.” *European Journal of Operational Research*, **195**, 628–640.
- Vansteelandt K, Van Mechelen I (1998). “Individual Differences in Situation-Behavior Profiles: A Triple Typology Model.” *Journal of Personality and Social Psychology*, **75**(3), 751–765.
- Varela P, Ares G (2014). *Novel Techniques in Sensory Characterization and Consumer Profiling*. CRC Press, Boca Raton, FL.
- Vermunt JK (2007). “A Hierarchical Mixture Model for Clustering Three-Way Data Sets.” *Computational Statistics & Data Analysis*, **51**, 5368–5376.
- Vermunt JK, Magidson J (2005). *Latent GOLD 4.0 User’s Guide*. Belmont, Massachusetts: Statistical Innovations Inc.
- Vermunt JK, Magidson J (2008). *LG-Syntax User’s Guide: Manual for Latent GOLD 4.5 Syntax Module*. Belmont, Massachusetts: Statistical Innovations Inc.
- Vichi M (1999). “One-Mode Classification of a Three-Way Data Set.” *Journal of Classification*, **16**, 27–44.
- Vichi M, Rocci R, Kiers HAL (2007). “Simultaneous Component and Clustering Models for Three-Way Data: Within and Between Approaches.” *Journal of Classification*, **24**, 71–98.
- Viroli C (2011). “Finite Mixtures of Matrix Normal Distributions for Classifying Three-Way Data.” *Statistics and Computing*, **21**, 511–522.
- Wilderjans TF, Ceulemans E (2013). “Clusterwise Parafac to Identify Heterogeneity in Three-Way Data.” *Chemometrics and Intelligent Laboratory Systems*, **129**, 87–97.



Wilderjans TF, Ceulemans E, Kuppens P (2012). “Clusterwise HICLAS: A Generic Modeling Strategy to Trace Similarities and Differences in Multiblock Binary Data.” *Behavior Research Methods*, **44**(2), 532–545.

## A. Derivation gradient

To describe the gradient of the model parameters in models m1, m2 and m3, we introduce the following notation:

$$\begin{aligned}\delta_i &= p(\mathbf{d}_i | \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{\xi}) \\ \delta_{i|t} &= p(\mathbf{d}_i | Z_{it} = 1, \boldsymbol{\sigma}, \boldsymbol{\rho}) \\ \delta_{ij|t} &= p(\mathbf{d}_{ij} | Z_{it} = 1, \boldsymbol{\sigma}_j, \boldsymbol{\rho}) \\ \delta_{ij|\mathbf{x},t} &= p(\mathbf{d}_{ij} | \mathbf{x}_{ij}, Z_{it} = 1, \boldsymbol{\rho}) \\ \delta_{ijk|\mathbf{x},t} &= p(d_{ijk} | \mathbf{x}_{ij}, Z_{it}=1, \boldsymbol{\rho}) \\ \delta_{ijk0|\mathbf{x},t} &= P(D_{ijk} = 0 | \mathbf{x}_{ij}, Z_{it} = 1, \boldsymbol{\rho})\end{aligned}$$

### A.1. Derivation gradient model m1

Using a prior as in Equation 12, the logarithm of the posterior density (LP) for model m1 equals :

$$\begin{aligned}& \sum_j \sum_f \sum_t \frac{1}{JT} [\log(\sigma_{jft}) + \log(1 - \sigma_{jft})] \\ & + \sum_k \sum_f \frac{1}{K} [\log(\rho_{kf}) + \log(1 - \rho_{kf})] \\ & + \sum_t \frac{2}{T} \log(\xi_t) \\ & + \sum_i \log \sum_t \xi_t \prod_j \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij} | \boldsymbol{\sigma}_j, Z_{it} = 1) \prod_k p(d_{ijk} | \mathbf{x}_{ij}, \boldsymbol{\rho}_k)\end{aligned}$$

Assuming a disjunctive model, we can now derive that

$$\begin{aligned}\frac{\partial LP}{\partial \sigma_{j'f't'}} &= \sum_i \frac{1}{\delta_i} \xi_{t'} \frac{\delta_{i|t'}}{\delta_{ij'|t'}} \sum_{\mathbf{x}_{ij'}} \delta_{ij'|t', \mathbf{x}} \frac{p(\mathbf{x}_{ij'} | \boldsymbol{\sigma}_{j'}, Z_{it'} = 1)}{p(x_{ij'f'} | \sigma_{j'f'}, Z_{it'} = 1)} (2x_{ij'f'} - 1) \\ & + \frac{1}{JT\sigma_{j'f't'}} - \frac{1}{JT(1 - \sigma_{j'f't'})}\end{aligned}$$

$$\begin{aligned} \frac{\partial LP}{\partial \rho_{k'f'}} &= \sum_i \frac{1}{\delta_i} \sum_t \xi_t \sum_j \frac{\delta_{i|t}}{\delta_{ij|t}} \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij} | \sigma_j, Z_{it} = 1) \frac{\delta_{ij|\mathbf{x},t}}{\delta_{ijk'|\mathbf{x},t}} \frac{\delta_{ijk'0|\mathbf{x},t}}{(1 - x_{ijf'} \rho_{k'f'})} (2d_{ijk'} - 1) x_{ijf'} \\ &+ \frac{1}{K \rho_{k'f'}} - \frac{1}{K(1 - \rho_{k'f'})} \end{aligned}$$

$$\frac{\partial LP}{\partial \gamma_{t'}} = \sum_i \frac{\delta_{i|t'}}{\delta_i} + \left(\frac{2}{T}\right) \left(\frac{1}{\gamma_{t'}}\right) - (I + 2).$$

## A.2. Derivation gradient model m2

Using a prior as in Equation 12, the logarithm of the posterior density for model m2 equals :

$$\begin{aligned} &\sum_j \sum_f \frac{1}{J} [\log(\sigma_{jf}) + \log(1 - \sigma_{jf})] \\ &+ \sum_k \sum_f \sum_t \frac{1}{KT} [\log(\rho_{kft}) + \log(1 - \rho_{kft})] \\ &+ \sum_t \frac{2}{T} \log(\xi_t) \\ &+ \sum_i \log \sum_t \xi_t \prod_j \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij} | \sigma_j) \prod_k p(d_{ijk} | \mathbf{x}_{ij}, Z_{it} = 1, \rho_k) \end{aligned}$$

Assuming a disjunctive model, we can now derive that

$$\begin{aligned} \frac{\partial LP}{\partial \sigma_{j'f'}} &= \sum_i \frac{1}{\delta_i} \sum_t \xi_t \frac{\delta_{i|t}}{\delta_{ij'|t}} \sum_{\mathbf{x}_{ij'}} \delta_{ij'|\mathbf{x},t} \frac{p(\mathbf{x}_{ij'} | \sigma_{j'})}{p(x_{ij'f'} | \sigma_{j'f'})} (2x_{ij'f'} - 1) \\ &+ \frac{1}{J \sigma_{j'f'}} - \frac{1}{J(1 - \sigma_{j'f'})} \end{aligned}$$

$$\begin{aligned} \frac{\partial LP}{\partial \rho_{k'f',t'}} &= \sum_i \frac{\xi_{t'}}{\delta_i} \sum_j \frac{\delta_{i|t'}}{\delta_{ij|t'}} \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij} | \sigma_j) \frac{\delta_{ij|\mathbf{x},t'}}{\delta_{ijk'|\mathbf{x},t'}} \frac{\delta_{ijk'0|\mathbf{x},t'}}{(1 - x_{ijf'} \rho_{k'f',t'})} (2d_{ijk'} - 1) x_{ijf'} \\ &+ \frac{1}{KT \rho_{k'f',t'}} - \frac{1}{KT(1 - \rho_{k'f',t'})} \end{aligned}$$

$$\frac{\partial LP}{\partial \gamma_{t'}} = \sum_i \frac{\delta_{i|t'}}{\delta_i} + \left(\frac{2}{T}\right) \left(\frac{1}{\gamma_{t'}}\right) - (I + 2).$$

### A.3. Derivation gradient model m3

Using a prior as in Equation 12, the logarithm of the posterior density for model m3 equals :

$$\begin{aligned}
& \sum_j \sum_f \sum_t \frac{1}{JT} [\log(\sigma_{jft}) + \log(1 - \sigma_{jft})] \\
& + \sum_k \sum_f \sum_t \frac{1}{KT} [\log(\rho_{kft}) + \log(1 - \rho_{kft})] \\
& + \sum_t \frac{2}{T} \log(\xi_t) \\
& + \sum_i \log \sum_t \xi_t \prod_j \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij} | \sigma_j, Z_{it} = 1) \prod_k p(d_{ijk} | \mathbf{x}_{ij}, Z_{it} = 1, \rho_k)
\end{aligned}$$

Assuming a disjunctive model, we can now derive that

$$\begin{aligned}
\frac{\partial LP}{\partial \sigma_{j'f't'}} &= \sum_i \frac{1}{\delta_i} \xi_{t'} \frac{\delta_{i|t'}}{\delta_{ij'|t'}} \sum_{\mathbf{x}_{ij'}} \delta_{ij'|t', \mathbf{x}} \frac{p(\mathbf{x}_{ij'} | Z_{it'} = 1, \sigma_{j't'})}{p(x_{ij'f'} | Z_{it'} = 1, \sigma_{j'f't'})} (2x_{ij'f'} - 1) \\
& + \frac{1}{JT \sigma_{j'f't'}} - \frac{1}{JT(1 - \sigma_{j'f't'})}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial LP}{\partial \rho_{k'f't'}} &= \sum_i \frac{\xi_{t'}}{\delta_i} \sum_j \frac{\delta_{i|t'}}{\delta_{ij|t'}} \sum_{\mathbf{x}_{ij}} p(\mathbf{x}_{ij} | Z_{it'} = 1, \sigma_{j't'}) \frac{\delta_{ij|\mathbf{x}, t'}}{\delta_{ijk'|\mathbf{x}, t'}} \frac{\delta_{ijk'0|\mathbf{x}, t'}}{(1 - x_{ijf'} \rho_{k'f't'})} (2d_{ijk'} - 1) x_{ijf'} \\
& + \frac{1}{KT \rho_{k'f't'}} - \frac{1}{KT(1 - \rho_{k'f't'})}
\end{aligned}$$

$$\frac{\partial LP}{\partial \gamma_{t'}} = \sum_i \frac{\delta_{i|t'}}{\delta_i} + \left(\frac{2}{T}\right) \left(\frac{1}{\gamma_{t'}}\right) - (I + 2).$$

**Affiliation:**

Michel Meulders

Faculty of Economics and Business

KU Leuven

1000 Brussels, Belgium

E-mail: [michel.meulders@kuleuven.be](mailto:michel.meulders@kuleuven.be)

URL: <http://www.kuleuven.be/wieiswie/en/person/u0010822>

*and*

Department of Psychology

KU Leuven

Tiensestraat 102

B-3000 Leuven, Belgium

URL: <http://ppw.kuleuven.be/okp/home/>