Mixture Models in Statistics and Psychometrics – Detecting Subgroups and Differential Item Functioning

PhD Thesis

in Mathematics

Faculty of Mathematics, Computer Science and Physics, Universität Innsbruck

by

Hannah Frick

Advisors:

Achim Zeileis, Carolin Strobl, and Christel Geiss

Innsbruck, August 2014

Suggested reviewing committee:

1. Bettina Grün, Johannes Kepler Universität Linz
2. Carolin Strobl, Universität Zürich
3. David Magis, Université de Liège

Abstract

Mixture models are a flexible tool to uncover latent groups for which separate models hold. The Rasch model can be used to measure latent traits by modeling the probability of a subject solving an item through the subject's ability and the item's difficulty. A crucial assumption of the Rasch model is measurement invariance: each item measures the latent trait in the same way for all subjects. Measurement invariance is violated if, e.g., an item is of different difficulty for different (groups of) subjects. Mixtures of Rasch models can be used to check if one Rasch model with a single set of item difficulties holds for all subjects and thus measurement invariance is not violated. However, estimation of the item difficulties in a Rasch mixture model is not independent of the specification of the score distribution, which is based on the abilities. The latent groups detected with such a Rasch mixture model are not solely based on the item difficulties but also – or even only – the scores and thus subject abilities. If the aim is to detect violations of measurement invariance, only latent groups based on item difficulties are of interest because different ability groups do not infringe on measurement invariance.

This thesis aims at making three different yet connected contributions: The methodological, psychometric contribution is a new specification of the Rasch mixture model. It ensures that latent classes uncovered by a Rasch mixture model are based solely on the item difficulties and thus increases the model's suitability as a tool to detect violations of measurement invariance. The computational contribution is open-source software in form of the R package **psychomix** for estimation of various flavors of the Rasch mixture model with or without concomitant variables and several options for the score distribution including the newly suggested specification. The statistical contribution connects and compares mixture models to model-based recursive partitioning. This is another method to detect subgroups in the data for which a stable set of model parameters holds and has also been applied to Rasch models to detect violations of measurement invariance. Here, mixture models and model-based recursive partitioning are presented in a unifying framework and the relative (dis-)advantages are illustrated in a simulation study.

Acknowledgments

This work would not have been possible without the support of several people whom I would like to thank here.

A particularly big thank you goes to Achim Zeileis for offering me this opportunity in the first place, accompanying me all the way, and leaving a lasting impression with regard to academic work in its many aspects and beyond.

Carolin Strobl introduced me to psychometrics, and gave valuable advise throughout the project. In addition, I am thankful for her being a champion of a productive work-life balance and the *Wort zum Montag*.

Christel Geiss facilitated the exchange between the Mathematics and Statistics Departments.

Friedrich Leisch sparked my interest in statistical computing.

All members of our department contributed to a great workplace environment and made me feel welcome amidst the many mountains. Helene Roth taught me proper Austrian German (it's *Bankomat*, not *Geldautomat*) and shared the everyday office life with me. Niki Umlauf and Jakob Meßner were reliably hungry at noon and pointed the way to post-PhD life. Julia Kopf shared the journey, albeit from Munich, and occasionally her couch.

Thank you, family and friends, for all the non-academic support along the way.

Contents

Chapter 1

Introduction

1.1 Mixture models

A very general and very common assumption in statistics is that all observations of a sample stem from the same distribution. However, sometimes a sample consists of data from different groups, which do not necessarily all follow the same distribution and group membership is unknown. In such situations, mixture models are an adequate and popular tool (McLachlan and Peel, 2000, Chapter 1). One of the first major applications of mixture models was the analysis by Pearson (1894). His colleague Weldon (1893) collected the forehead to body length ratio of crabs sampled from the Bay of Naples and suspected from the asymmetry in the histogram that two subspecies were evolving. To account for both, the asymmetry and the two (suspected) subspecies, Pearson fitted a weighted mixture of two normal densities to the data. Each density represented one of the two subspecies and had its own mean and variance. The resulting mixture density fitted the histogram considerably better than a single normal density.

The basic form of a mixture model is a weighted sum over several components. In the example of the crabs data, those components are plain densities. However, statistical models can also be used as such components. If the component is, e.g., a normal distribution and its mean is specified to depend on a set of covariates through a linear predictor, the component is also a linear model (McLachlan and Peel, 2000, Chapter 3.6). In such cases, the components do not only represent different groups with different means (as in the crabs example), but since the mean depends on covariates, the different groups can also be described in terms of this relationship between response and regressors. As each group is described through its component, each group can be described through its corresponding statistical model. Or in other words, a different model holds for each of the groups in the data. Note that since mixture models are usually employed with a finite number of components, finite mixture models are commonly referred to as mixture models.

These groups in the data are considered latent as group membership is unknown and not determined by any covariates – covariates employed *within* component models do not

directly determine which observations belong to which group. Posterior probabilities of belonging to a component (or the latent class it represents) can be derived for every observation. If additional covariates are available which influence group membership rather than the relationship modeled within each component, they can be incorporated in the mixture model through a so-called concomitant variable model (Dayton and Macready, 1988). A common choice for such a concomitant variable model is a multinomial logit model. The resulting probabilities are used as weights for the components in the mixture model.

Mixture models constitute a flexible tool to model data for which one model with one set of parameters is not adequate. However, it can also be used to check if such a deviation from one model with a single set of parameters is necessary. This is usually done by comparing the fit of a single model to that of mixture models with several components via an information criterion such as the Akaike Information Criterion (AIC, Akaike, 1973) or Schwarz's Bayesian Information Criterion (BIC, Schwarz, 1978). Note that regularity conditions for the likelihood ratio test are not fulfilled (McLachlan and Peel, 2000, Chapter 6.4).

Here, the focus will be on determining whether or not latent groups are present in the data which require different model parameters. In a broader statistical context this topic is also referred to as parameter stability whereas in the field of psychometrics it is often approached in the context of measurement invariance.

1.2 Measuring latent traits with the Rasch model

To measure latent traits such as intelligence, knowledge (be it of a specific form or general knowledge), mathematical or reading skills as in the PISA study, etc., subjects are usually asked to respond to a set of items, e.g., through a questionnaire which is then referred to as a psychological test or an instrument. Item response theory provides several models for estimating such a latent trait. The Rasch model (Rasch, 1960) is the state-of-the-art model for binary items, which will be the focus here. For polytomous items, several extensions of the Rasch model exist, which include the rating scale model (RSM, Andrich, 1978) and the partial credit model (PCM, Masters, 1982).

In the Rasch model, the probability of a subject to solve an item is based on the subject's ability – the latent trait – and the item's difficulty. A Rasch model thus contains two types of parameters: person and item parameters for abilities and difficulties, respectively. Joint estimation of both types of parameters is inconsistent for a fixed number of items (Molenaar, 1995a). Two alternative approaches have been developed, which both aim at untangling the problem by transforming the likelihood so that first the item parameters and then the person parameters can be estimated. In the marginal maximum likelihood (MML) approach (e.g., Molenaar, 1995a), a distribution across abilities is assumed and used to drop the person parameters from the joint likelihood via integration. In the conditional maximum likelihood (CML) approach (e.g., Molenaar, 1995a), the person parameters are shed by conditioning on a sufficient statistic for the abilities, the

so-called scores. The score of a subject is the number of correctly solved items. The joint likelihood can then be split into two parts: one part conditional on the scores which does not depend on the person parameters and is thus used to estimate the item parameters, and the second part for the conditioning scores. Here, the CML approach is adopted as no assumption on the distribution of the abilities – or the scores – needs to be made in order to estimate the item difficulties. Given estimates for the item parameters, the person parameters are estimated in a second step.

If the Rasch model is used as a measurement tool, one is mainly interested in the ability parameters as they are the estimates for the subjects' latent traits. However, in order to make fair comparisons between subjects based on these estimates, the crucial assumption of measurement invariance needs to be fulfilled: each item needs to measure the latent trait in the same way for all subjects. This is violated if, e.g., an item is of different difficulty for different (groups of) subjects. Difficulty may (and should, for a good instrument) vary across items but no item may vary in its difficulty across subjects. If this is the case for an item, this is usually referred to as *differential item functioning (DIF)* and is a violation of the assumption of measurement invariance. From a statistical viewpoint this can also be seen as parameter instability since one item difficulty (parameter) does not hold for all subjects if the item is easier for some subjects and more difficult for others. In order to avoid unfair comparisons between subjects, all items of an instrument are checked for such violations like DIF. Many tests are available to establish whether or not DIF is present. Some tests such as the Mantel-Haenszel test (Holland and Thayer, 1988) are item-wise tests which test each item separately for DIF. Global tests such as the likelihood ratio (LR) test (Andersen, 1972; Gustafsson, 1980) assess a set of item difficulties (e.g., for all items of a questionnaire) at once for DIF in any of the items. An alternative approach for DIF detection are mixture models.

1.3 Mixtures of Rasch models

As indicated earlier, mixtures of Rasch models (Rost, 1990) can be seen as a tool to relax the assumption of measurement invariance by allowing for several sets of item difficulties instead of just one set which is required to hold for all observations. However, it can also be seen as a tool to investigate whether this is necessary in the first place. It follows the straightforward rationale that if a mixture of Rasch models (with two or more components) fits the data better than a single Rasch model, the data consist of several groups of subjects for which different Rasch models hold. Therefore, measurement invariance is violated and a single Rasch model is not applicable to the data.

In its basic form a Rasch mixture model is a weighted sum over several Rasch models. On the basis of the CML approach, the Rasch models are written as the product of the conditional likelihood of the item difficulties and the score probabilities. In the introduction of Rasch mixture models, Rost (1990) suggested a saturated parametrization of the score distribution with one probability parameter for each possible score. This can be challenging in model estimation and/or model selection. Thus, a more parsimonious parametrization has been suggested by Rost and von Davier (1995), which uses only two

parameters: one for the mean and one for the variance.

While Rasch mixture model do not constitute a significance test like the DIF tests mentioned above, they have the advantage of detecting latent DIF groups. For the DIF tests, the groups which are tested against need to be pre-specified and are usually based on covariates such as gender and (categorized) age. Rasch mixture models, however, do not a priori assume specific (DIF) groups but rather establish them in a data-driven way (Rost and von Davier, 1995). In particular, the detection of latent classes can also be achieved if only the item responses are available but no additional covariates.

1.4 Software for Rasch mixture models

Rasch models can be fitted through a large variety of software, whereas the selection is smaller for mixtures of Rasch models. Software for all major platforms is available, however, not all programs are available for all platforms. Among the commercial programs, WINMIRA 2001 (von Davier, 2000) is one of the most well-known options. Among the open-source options, the R system for statistical computing (R Core Team, 2014), available for all major platforms, is one of the most versatile tools for statistical analysis and beyond. A base distribution is provided by the R Core Team and add-on packages for a large variety of tasks are contributed by numerous useRs. Two notable examples for fitting Rasch models are the packages **eRm** (Mair and Hatzinger, 2007) and **ltm** (Rizopoulos, 2006). Rasch mixture models can be fitted with the packages **mixRasch** (Willse, 2011) and **mRm** (Preinerstorfer and Formann, 2011), using joint and conditional maximum likelihood estimation, respectively. Unfortunately, neither package allows for different score specifications or the inclusion of concomitant variables. To close this gap, the **psychomix** package (Frick, Strobl, Leisch, and Zeileis, 2012) leverages the framework for mixture models (including concomitant variable models) provided by the **flexmix** package (Leisch, 2004; Grün and Leisch, 2008) together with the **psychotools** package (Zeileis, Strobl, and Wickelmaier, 2011) for estimating psychometric models. The options for different specifications of the score distribution are implemented directly in the **psychomix** package. Chapter 2 is a slightly modified version of Frick *et al.* (2012) which contains an introduction to the package, implementation details and hands-on guidance for practical use.

1.5 Specification of the score distribution

For a single Rasch model, the CML approach allows for an estimation of the item parameters without a specification of the score distribution. In the context of a Rasch mixture model, however, the estimation of the item difficulties is *not* independent of the score distribution. As the components of a Rasch mixture model are full Rasch models, the components are made up by both parts of the factorized likelihood, the conditional likelihood of the item parameters and the score distribution. The estimation of the Rasch mixture model is therefore influenced by both difficulties *and* abilities. In the context of

DIF detection, this not desirable because only latent groups based on the item difficulties are of interest. Latent ability groups based on the person parameters are possible but not a violation of measurement invariance (Ackerman, 1992).

Chapter 3 is a slightly modified version of Frick, Strobl, and Zeileis (2014a). It is explained how the score distribution influences the estimation of the Rasch mixture model – even if the CML approach is employed – and a new specification of the score distribution is suggested which restores the independence of item parameter estimation from the specification of the score distribution in Rasch mixture models. The basic idea is to restrict the score distribution to be equal across all components of the mixture. This corresponds to a mixture of just the conditional likelihoods of the item parameters as the score distribution is *not* component-specific and thus does not influence the mixture. The resulting latent classes are then based on the item parameters only and thus corresponds to DIF groups only. The accompanying simulation study illustrates how DIF detection via Rasch mixture models with a saturated or mean-variance specification for the scores can be misleading if ability groups are present in the data. It also illustrates how this can be avoided by the restricted score distribution. The implementation of all discussed specifications for the score distribution, including the restricted specification, are included in the **psychomix** package.

1.6 Beyond mixtures: Detecting subgroups with tree models

The first aim of DIF detection, of course, is to decide if any DIF, and thus any DIF groups, is present in the data. In a second step, it is usually of interest which subjects are in which DIF group and how those can be described through covariates. This can be done in a post-hoc analysis once group membership is estimated/established. In the context of mixture model, this can also be part of the initial analysis when such covariates are included through a concomitant variable model. Subjects then have different probabilities of belonging to a certain DIF group based on their covariates characteristics. Another fairly recent approach to detect DIF, which also establishes DIF groups in a data-driven way, are Rasch trees (Strobl, Kopf, and Zeileis, 2014). In this approach, the sample is split into DIF groups based on covariates such that for each group one set of item difficulties holds. Since this method uses significance tests to determine whether or not to split the sample, it may seem more closely related to classical DIF tests than Rasch mixture models. Nonetheless, it shares a lot of similarities with Rasch mixture models. Both methods can be embedded in a larger statistical framework, namely mixture models and model-based recursive partitioning (Zeileis, Hothorn, and Hornik, 2008).

Chapter 4 is a slightly modified version of Frick, Strobl, and Zeileis (2014b) and gives a brief introduction to both frameworks. The basic ideas are explained and illustrated via the linear model. The differences and similarities between the two methods are pointed out and systematically investigated by means of a simulation study. The results suggest that neither method universally outperforms the other and thus both methods should

be part of a statistical toolbox for assessing parameter stability and model assumptions.

Chapter 2

Flexible Rasch Mixture Models with Package psychomix

This chapter is a (slightly) modified version of Frick et al. *(2012), published in the* Journal of Statistical Software.

Abstract:
Measurement invariance is an important assumption in the Rasch model and mixture models constitute a flexible way of checking for a violation of this assumption by detecting unobserved heterogeneity in item response data. Here, a general class of Rasch mixture models is established and implemented in R, using conditional maximum likelihood estimation of the item parameters (given the raw scores) along with flexible specification of two model building blocks: (1) Mixture weights for the unobserved classes can be treated as model parameters or based on covariates in a concomitant variable model. (2) The distribution of raw score probabilities can be parametrized in two possible ways, either using a saturated model or a specification through mean and variance. The function `raschmix()` in the R package **psychomix** provides these models, leveraging the general infrastructure for fitting mixture models in the **flexmix** package. Usage of the function and its associated methods is illustrated on artificial data as well as empirical data from a study of verbally aggressive behavior.

2.1 Introduction

In item response theory (IRT), latent traits are usually measured by employing probabilistic models for responses to sets of items. One of the most prominent examples for such an approach is the Rasch model (Rasch, 1960) which captures the difficulty (or equivalently easiness) of binary items and the respondent's trait level on a single common scale. Generally, a central assumption of most IRT models (including the Rasch model) is measurement invariance, i.e., that all items measure the latent trait in the same way for all subjects. If violated, measurements obtained from such a model provide no fair comparisons of the subjects. A typical violation of measurement invariance in the Rasch

model is differential item functioning (DIF), see Ackerman (1992).

Therefore, assessing the assumption of measurement invariance and checking for DIF is crucial when establishing a Rasch model for measurements of latent traits. Hence, various approaches have been suggested in the literature that try to assess heterogeneity in (groups of) subjects either based on observed covariates or unobserved latent classes. If covariates are available, classical tests like the Wald or likelihood ratio test can be employed to compare model fits between some reference group and one or more focal groups (Glas and Verhelst, 1995). Typically, these groups are defined by the researcher based on categorical covariates or arbitrary splits in either numerical covariates or the raw scores (Andersen, 1972). More recently, extensions of these classical tests have also been embedded into a mixed model representation (Van den Noortgate and De Boeck, 2005). Another recently suggested technique is to recursively define and assess groupings in a data-driven way based on all available covariates (both numerical and categorical) in so-called Rasch trees (Strobl, Kopf, and Zeileis, 2011a).

Heterogeneity occurring in latent classes only (i.e., not observed or captured by covariates), however, is typically addressed by mixtures of IRT models. Specifically, Rost (1990) combined a mixture model approach with the Rasch model. If any covariates are present, they can be used to predict the latent classes (as opposed to the item parameters themselves) in a second step (Cohen and Bolt, 2005). More recently, extensions to this mixture model approach have been suggested that encompass this prediction, see Tay, Newman, and Vermunt (2011) for a unifying framework.

DIF cannot be separated from multidimensionality when items measure several different latent traits which also differ in (groups of) subjects (cf., e.g., Ackerman, 1992, and the references therein). In this sense, mixtures of Rasch models can also be used to assess the assumption of unidimensionality, see Rijmen and De Boeck (2005) on the relationship of between-item multidimensional models and mixtures of Rasch models.

In this paper, we introduce the **psychomix** package for the R system for statistical computing (R Development Core Team, 2011) that provides software for fitting a general and flexible class of Rasch mixture models along with comprehensive methods for model selection, assessment, and visualization. The package leverages the general and object-oriented infrastructure for fitting mixture models from the **flexmix** package (Leisch, 2004; Grün and Leisch, 2008), combining it with the function `RaschModel.fit()` from the **psychotools** package (Zeileis *et al.*, 2011) for the estimation of Rasch models. All packages are freely available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/`.

The reason for using `RaschModel.fit()` as opposed to other previously existing (and much more powerful and flexible) R packages for Rasch modeling – such as **ltm** (Rizopoulos, 2006) or **eRm** (Mair and Hatzinger, 2007) – is reduced computational complexity: `RaschModel.fit()` is intended to provide a "no frills" implementation of simple Rasch models, useful when refitting a model multiple times in mixtures or recursive partitions (see also Strobl *et al.*, 2011a).

While **psychomix** was under development, another R implementation of the Rost (1990) model became available in package **mRm** (Preinerstorfer and Formann, 2011). As this builds on specialized C++ code, it runs considerably faster than the generic **flexmix** approach – however, it only covers this one particular type of model and offers fewer methods for specifying, inspecting, and assessing (fitted) models. In **psychomix**, both approaches are reconciled by optionally employing the **mRm** solution as an input to the **flexmix** routines.

In the following, we first briefly review both Rasch and mixture models and combine them in a general Rasch mixture framework (Section 2.2). Subsequently, the R implementation in **psychomix** is introduced (Section 2.3), illustrated by means of simulated data, and applied in practice to a study of verbally aggressive behavior (Section 2.4). Concluding remarks are provided in Section 2.5.

2.2 Rasch mixture models

In the following, we first provide a short introduction to the Rasch model, subsequently outline the basics of mixture models in general, and finally introduce a general class of Rasch mixture models along with the corresponding estimation techniques.

2.2.1 The Rasch model

Latent traits can be measured through a set of items to which binary responses are given. Success of solving an item or agreeing with it is coded as "1", while "0" codes the opposite response. The model suggested by Rasch (1960) uses the person's ability θ_i $(i = 1, \ldots, n)$ and the item's difficulty β_j $(j = 1, \ldots, m)$ to model the response y_{ij} of person i to item j:

$$P(Y_{ij} = y_{ij}|\theta_i, \beta_j) \;=\; \frac{\exp\{y_{ij}(\theta_i - \beta_j)\}}{1 + \exp\{\theta_i - \beta_j\}}. \tag{2.1}$$

Under the assumption of independence – both across persons and items within persons (see Molenaar, 1995b) – the likelihood for the whole sample $y = (y_{ij})_{n\times m}$ can be written as the product of the likelihood contributions from Equation 2.1 for all combinations of subjects and items. It is parametrized by the vector of all person parameters $\theta = (\theta_1, \ldots, \theta_n)^\top$ and the vector of all item parameters $\beta = (\beta_1, \ldots, \beta_m)^\top$ (see Equation 2.2). Since the probability of solving an item is modeled only through the logit of the difference between person ability and item difficulty, these parameters are on an interval scale with an arbitrary zero point. To obtain a fixed reference point, usually one item difficulty (e.g., the first β_1) or the sum of item difficulties is restricted to zero. See Fischer (1995) for details.

On the basis of the number of correctly solved items, the so-called "raw" scores $r_i = \sum_{j=1}^{m} y_{ij}$, the likelihood for the full sample can be factorized into a conditional likelihood of the item parameters $h(\cdot)$ and the score probabilities $g(\cdot)$ (Equation 2.3). Because the

scores r are sufficient statistics for the person parameters θ, the likelihood of the item parameters β conditional on the scores r does not depend on the person parameters θ (Equation 2.4).

$$\begin{aligned} L(\theta, \beta) &= f(y|\theta, \beta) && (2.2)\\ &= h(y|r, \theta, \beta)g(r|\theta, \beta) && (2.3)\\ &= h(y|r, \beta)g(r|\theta, \beta). && (2.4) \end{aligned}$$

The conditional likelihood of the item parameters takes the form

$$h(y|r, \beta) = \prod_{i=1}^{n} \frac{\exp\{-\sum_{j=1}^{m} y_{ij}\beta_j\}}{\gamma_{r_i}(\beta)} \quad (2.5)$$

where $\gamma_{r_i}(\cdot)$ is the elementary symmetric function of order r_i, capturing all possible response patterns leading to a certain score (see Molenaar, 1995a, for details).

There are several approaches to estimating the Rasch model: *Joint maximum likelihood (ML)* estimation of β and θ is inconsistent, thus two other approaches have been established. Both are two-step approaches but differ in the way the person parameters θ are handled. For *marginal ML* estimation a distribution for θ is assumed and integrated out in $L(\theta, \beta)$, or equivalently in $g(r|\theta, \beta)$. In the *conditional ML* approach only the conditional likelihood of the item parameters $h(y|r, \beta)$ from Equation 2.5 is maximized for estimating the item parameters. Technically, this is equivalent to maximizing $L(\theta, \beta)$ with respect to β if one assumes that $g(r|\delta) = g(r|\theta, \beta)$ does not depend on θ or β, but potentially other parameters δ.

In R, the **ltm** package (Rizopoulos, 2006) uses the marginal ML approach while the **eRm** package (Mair and Hatzinger, 2007) employs the conditional ML approach, i.e., uses and reports only the conditional part of the likelihood in the estimation of β. The latter approach is also taken by the `RaschModel.fit()` function in the **psychotools** package (Zeileis *et al.*, 2011).

2.2.2 Mixture models

Mixture models are a generic approach for modeling data that is assumed to stem from different groups (or clusters) but group membership is unknown.

The likelihood $f(\cdot)$ of such a mixture model is a weighted sum (with prior weights π_k) of the likelihood from several components $f_k(\cdot)$ representing the different groups:

$$f(y_i) = \sum_{k=1}^{K} \pi_k f_k(y_i).$$

Generally, the components $f_k(\cdot)$ can be densities or (regression) models. Typically, all K components $f_k(\cdot)$ are assumed to be of the same type $f(y|\xi_k)$, distinguished through their component-specific parameter vector ξ_k.

If variables are present which do not influence the components $f_k(\cdot)$ themselves but rather the prior class membership probabilities π_k, they can be incorporated in the model as so-called *concomitant variables* (Dayton and Macready, 1988). In the psychometric literature, such covariates predicting latent information are also employed, e.g., by Tay *et al.* (2011) who advocate a unifying IRT framework that also optionally encompasses concomitant information (labeled *MM-IRT-C* for mixed-measurement IRT with covariates). To embed such concomitant variables x_i into the general mixture model notation, a model for the component membership probability $\pi(k|x_i, \alpha)$ with parameters α is employed:

$$f(y_i|x_i, \alpha, \xi_1, \ldots, \xi_K) \;=\; \sum_{k=1}^{K} \pi(k|x_i, \alpha) f(y_i|\xi_k), \tag{2.6}$$

where commonly a multinomial logit model is chosen to parametrize $\pi(k|x_i, \alpha)$ (see e.g., Grün and Leisch, 2008; Tay *et al.*, 2011). Note that the multinomial model collapses to separate π_k $(k = 1, \ldots, K)$ if there is only an intercept and no real concomitants in x_i.

2.2.3 Flavors of Rasch mixture models

When combining the general mixture model framework from Equation 2.6 with the Rasch model based on Equation 2.1, several options are conceivable for two of the building blocks. First, the component weights can be estimated via a separate parameter π_k for each component or via a concomitant variable model $\pi(k|x_i, \alpha)$ with parameters α. Second, the full likelihood function $f(y_i|\xi_k)$ of the components needs to be defined. If a conditional ML approach is adopted, it is clear that the conditional likelihood $h(y_i|r_i, \beta)$ from Equation 2.5 should be one part, but various choices for modeling the score probabilities are available. One option is to model each score probability with its own parameter $g(r_i) = \psi_{r_i}$, while another (more parsimonious) option would be to adopt a parametric distribution of the score probabilities with fewer parameters (Rost and von Davier, 1995).

Note that while for a single-component model, the estimates of the item parameters $\hat{\beta}$ are invariant to the choice of the score probabilities (as long as it is independent from β), this is no longer the case for a mixture model with $K \geq 2$. The estimation of the item parameters in the mixture is invariant given the weights $\pi(k|x_i, \alpha)$ but the weights and thus the estimates of β may depend on the score distribution in a mixture model. (This dependency is introduced through the posterior probabilities calculated in the E-step of the algorithm that is explained Section 2.2.4.) Also note that the conditional ML approach employed here uses a model $g(r|\delta)$ directly on the score probabilities rather than a distribution on the person parameters θ as does the marginal ML approach.

Rost's original parametrization

One of these possible mixtures – the so-called "mixed Rasch model" introduced by Rost (1990) – is already well-established in the psychometric literature. It models the score probabilities through separate parameters $g(r_i) = \psi_{r_i}$ (under the restriction that they

sum to unity) and does not employ concomitant variables. The likelihood of Rost's mixture model can thus be written as

$$f(y|\pi, \psi, \beta) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k h(y_i|r_i, \beta_k)\psi_{r_i,k}. \tag{2.7}$$

This particular parametrization is implemented in the R package **mRm** (Preinerstorfer and Formann, 2011).

Since subjects who solve either none or all items (i.e., $r_i = 0$ or m, respectively) do not contribute to the conditional likelihood of the item parameters they cannot be allocated to any of the components in this parametrization. Hence, Rost (1990) proposed to remove those "extreme scorers" from the analysis entirely and fix the corresponding score probabilities ψ_0 and ψ_m at 0. However, if one wishes to include these extreme scorers in the analysis, the corresponding score probabilities can be estimated through their relative frequency (across all components) and the remaining score probabilities within each component are rescaled to sum to unity together with those extreme score probabilities. Nevertheless, the extreme scorers still do not contribute to the estimation of the mixture itself.

Other score distributions

As noted by Rost and von Davier (1995), the disadvantage of this saturated model for the raw score probabilities is that many parameters need to be estimated ($K \times (m-2)$, not counting potential extreme scorers) that are typically not of interest. To check for DIF, the item parameters are of prime importance while the raw score distribution can be regarded as a nuisance term. This problem can be alleviated by embedding the model from Equation 2.7 into a more general framework that also encompasses more parsimonious parametrizations. More specifically, a conditional logit model can be established

$$g(r|\delta) \;=\; \frac{\exp\{z_r^\top \delta\}}{\sum_{j=1}^{m-1} \exp\{z_j^\top \delta\}}, \tag{2.8}$$

containing some auxiliary regressors z_i with coefficients δ.

The saturated $g(r_i) = \psi_{r_i}$ model is a special case when constructing the auxiliary regressor from indicator/dummy variables for the raw scores $2, \ldots, m-1$: $z_i = (I_2(r_i), \ldots, I_{m-1}(r_i))^\top$. Then $\delta = (\log(\psi_2) - \log(\psi_1), \ldots, \log(\psi_{m-1}) - \log(\psi_1))^\top$ is a simple logit transformation of ψ.

As an alternative Rost and von Davier (1995) suggests a specification with only two parameters that link to mean and variance of the score distribution, respectively. More specifically, the auxiliary regressor is $z_i = (r_i/m, 4r_i(m-r_i)/m^2)^\top$ so that δ pertains to the vector of location and dispersion parameters of the score distribution. Thus, for $m > 4$ items, this parametrization is more parsimonious than the saturated model.

General Rasch mixture model

Combining all elements of the likelihood this yields a more general specification of the Rasch mixture model

$$f(y|\alpha,\beta,\delta) = \prod_{i=1}^{n}\sum_{k=1}^{K} \pi(k|x_i,\alpha)\ h(y_i|r_i,\beta_k)\ g(r_i|\delta_k) \tag{2.9}$$

with (a) the concomitant model $\pi(k|x_i,\alpha)$ for modeling component membership, (b) the component-specific conditional likelihood of the item parameters given the scores $h(y_i|r_i,\beta_k)$, and (c) the component-specific score distribution $g(r_i|\delta_k)$.

2.2.4 Parameter estimation

Parameter estimation for mixture models is usually done via the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977). It treats group membership as unknown and optimizes the complete-data log-likelihood including the group membership on basis of the observed values only. It iterates between two steps until convergence: estimation of group membership (E-step) and estimation of the components (M-step).

In the E-step, the posterior probabilities of each observation for the K components is estimated through:

$$\hat{p}_{ik} \ = \ \frac{\hat{\pi}_k f(y_i|\hat{\xi}_k)}{\sum_{g=1}^{K}\hat{\pi}_g f(y_i|\hat{\xi}_g)} \tag{2.10}$$

using the parameter estimates from the previous iteration for the component weights π and the model parameters ξ which encompass β and δ. In the case of concomitant variables, the weights are $\hat{\pi}_{ik} = \pi(k|x_i,\hat{\alpha})$.

In the M-step, the parameters of the mixture are re-estimated with the posterior probabilities as weights. Thus, observations deemed unlikely to belong to a certain component have little influence on estimation within this component. For each component, the weighted ML estimation can be written as

$$\begin{aligned}\hat{\xi}_k \ &= \ \underset{\xi_k}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{p}_{ik} \log f(y_i|\xi_k) \quad (k=1,\ldots,K) \\ &= \ \left\{ \underset{\beta_k}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{p}_{ik} \log h(y_i|r_i,\beta_k);\ \ \underset{\delta_k}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{p}_{ik} \log g(r_i|\delta_k) \right\}\end{aligned} \tag{2.11}$$

which for the Rasch model amounts to separately maximizing the weighted conditional log-likelihood for the item parameters and the weighted score log-likelihood.

The concomitant model can be estimated separately from the posterior probabilities:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}} \sum_{i=1}^{n}\sum_{k=1}^{K} \hat{p}_{ik} \log(\pi(k|x_i,\alpha)), \tag{2.12}$$

where $\pi(k|x_i, \alpha)$ could be, e.g, a multinomial model.

Finally, note that the number of components K is not a standard model parameter (because the likelihood regularity conditions do not apply) and thus it is not estimated through the EM algorithm. Either it needs to be chosen by the practitioner or by model selection techniques such as information criteria, as illustrated in the following examples.

2.3 Implementation in R

2.3.1 User interface

The function `raschmix()` can be used to fit the different flavors of Rasch mixture models described in Section 2.2.3: with or without concomitant variables in $\pi(k|x_i, \alpha)$, and with different score distributions $g(r_i|\delta_k)$ (saturated vs. mean/variance parametrization). The function's synopsis is

```
raschmix(formula, data, k, subset, weights, scores = "saturated",
  nrep = 3, cluster = NULL, control = NULL,
  verbose = TRUE, drop = TRUE, unique = FALSE, which = NULL,
  gradtol = 1e-6, deriv = "sum", hessian = FALSE, ...)
```

where the lines of arguments pertain to (1) data/model specification processed within `raschmix()`, (2) control arguments for fitting a single mixture model, (3) control arguments for iterating across mixtures over a range of numbers of components K, all passed to `stepFlexmix()`, and (4) control arguments for fitting each model component within a mixture (i.e., the M-step) passed to `RaschModel.fit()`. Details are provided below, focusing on usage in practice first.

A formula interface with the usual `formula`, `data`, `subset`, and `weights` arguments is used: The left-hand side of the formula sets up the response matrix y and the right-hand side the concomitant variables x (if any). The response may be provided by a single matrix or a set of individual dummy vectors, both of which may be contained in an optional data frame. Example usages are `raschmix(resp ~ 1, ...)` if the matrix `resp` is an object in the environment of the formula, typically the calling environment, or `raschmix(item1 + item2 + item3 ~ 1, data = d, ...)` if the `item*` vectors are in the data frame `d`. In both cases, `~ 1` signals that there are no concomitant variables – if there were, they could be specified as `raschmix(resp ~ conc1 + conc2, ...)`. As an additional convenience, the formula may be omitted entirely if there are no concomitant variables, i.e., `raschmix(data = resp, ...)` or alternatively `raschmix(resp, ...)`.

The `scores` of the model can be set to either `"saturated"` (see Equation 2.7) or `"meanvar"` for the mean/variance specification of Rost and von Davier (1995). Finally, the number of components K of the mixture is specified through `k`, which may be a vector resulting in a mixture model being fitted for each element.

To control the EM algorithm for fitting the specified mixture models, `cluster` may optionally specify starting probabilities $\hat{p}_{ik}$ and `control` can set certain control arguments through a named list or an object of class "`FLXcontrol`". One of these control arguments named `minprior` sets the minimum prior probability for every component. If in an iteration of the EM algorithm, any component has a prior probability smaller then `minprior`, it is removed from the mixture in the next iteration. The default is `0`, i.e., avoiding such shrinkage of the model. If `cluster` is not provided, `nrep` different random initializations are employed, keeping only the best solution (to avoid local optima). Finally, `cluster` can be set to `"mrm"` in which case the fast C++ implementation from **mRm** (Preinerstorfer and Formann, 2011) can be leveraged to generate optimized starting values. Again, the best solution of `nrep` runs of `mrm()` is used. Note that as of version 1.0 of **mRm** only the model from Equation 2.7 is supported in `mrm()`, resulting in suboptimal – but potentially still useful – posterior probabilities $\hat{p}_{ik}$ for any other model flavor.

Internally, `stepFlexmix()` is called to fit all individual mixture models and takes control arguments `verbose`, `drop`, and `unique`. If `k` is a vector, the whole set of models is returned by default but one may choose to select only the best model according to an information criterion. For example, `raschmix(resp, k = 1:3, which = "AIC", ...)` or `raschmix(resp ~ 1, data = d, k = 1:4, which = "BIC", ...)`.

The arguments `gradtol`, `deriv` and `hessian` are used to control the estimation of the item parameters in each M-step (Equation 2.11) carried out via `RaschModel.fit()`.

Function `raschmix()` returns objects of class "`raschmix`" or "`stepRaschmix`", respectively, depending on whether a single or multiple mixture models are fitted. These classes extend "`flexmix`" and "`stepFlexmix`", respectively, for more technical details see the next section. For standard methods for extracting or displaying information, either for "`raschmix`" directly or by inheritance, see Table 2.1 for an overview.

2.3.2 Internal structure

As briefly mentioned above, `raschmix()` leverages the **flexmix** package (Leisch, 2004; Grün and Leisch, 2008) and particularly its `stepFlexmix()` function for the estimation of (sets of) mixture models.

The **flexmix** package is designed specifically to provide the infrastructure for flexible mixture modeling via the EM algorithm, where the type of a mixture model is determined through the model employed in the components. In the estimation process, this component model definition corresponds to the definition of the M-step (Equation 2.11). Consequently, the **flexmix** package provides the framework for fitting mixture models by leveraging the modular structure of the EM algorithm. Provided with the right M-step, **flexmix** takes care of the data handling and iterating estimation through both E-step and M-step.

The M-step needs to be provided in the form of a `flexmix` driver inheriting from class "`FLXM`" (see Grün and Leisch, 2008, for details). The **psychomix** package includes such

Function	Class	Description
`summary()`	"`raschmix`"	display information about the posterior probabilities and item parameters; returns an object of class "`summary.raschmix`" containing the relevant summary statistics (which has a `print()` method)
`parameters()`	"`raschmix`"	extract estimated parameters of the model for all or specified components, extract either parameters α of the concomitant model or item parameters β and/or score parameters δ
`worth()`	"`raschmix`"	extract the item parameters β under the restriction $\sum_{j=1}^{m} \beta_j = 0$
`scoreProbs()`	"`raschmix`"	extract the score probabilities $g(r\|\delta)$
`plot()`	"`raschmix`"	base graph of item parameter profiles in all or specified components
`xyplot()`	"`raschmix`"	lattice graph of item parameter profiles of all or specified components in a single or multiple panels
`histogram()`	"`raschmix`"	lattice rootogram or histogram of posterior probabilities
`print()`	"`stepFlexmix`"	simple printed display of number of components, log-likelihoods, and information criteria
`plot()`	"`stepFlexmix`"	plot information criteria against number of components
`getModel()`	"`stepFlexmix`"	select model according to either an information criterion or the number of components
`print()`	"`flexmix`"	simple printed display with cluster sizes and convergence
`clusters()`	"`flexmix`"	extract predicted class memberships
`posterior()`	"`flexmix`"	extract posterior class probabilities
`logLik()`	"`flexmix`"	extract fitted log-likelihood
`AIC()`; `BIC()`	"`flexmix`"	compute information criteria AIC, BIC

Table 2.1: Methods for objects of classes "`raschmix`", "`flexmix`", and "`stepFlexmix`".

a driver function: `FLXMCrasch()` relies on the function `RaschModel.fit()` from the **psychotools** package for estimation of the item parameters (i.e., maximization of the conditional likelihood from Equation 2.5) and adds different estimates of raw score probabilities depending on their parametrization. The driver can also be used directly with functions `flexmix()` and `stepFlexmix()`. The differences in model syntax and functionality for the classes of the resulting objects are illustrated in the Appendix A. As noted in the introduction, the reason for employing `RaschModel.fit()` rather than one of the more established Rasch model packages such as **eRm** or **ltm** is speed.

In the **flexmix** package, two fitting functions are provided. `flexmix()` is designed for fitting one model once and returns an object of class "`flexmix`". `stepFlexmix()` extends this so that either a single model or several models can be fitted. It also provides the functionality to fit each model repeatedly to avoid local optima.

When fitting models repeatedly, only the solution with the highest likelihood is returned. Thus, if `stepFlexmix()` is used to repeatedly fit a single model, it returns an object of class "`flexmix`". If `stepFlexmix()` is used to fit several different models (repeatedly or just once), it returns an object of class "`stepFlexmix`".

This principle extends to `raschmix()`: If it is used to fit a single model, the returned object is of class "`raschmix`". If used for fitting multiple models, `raschmix()` returns an object of class "`stepRaschmix`". Both classes extend their **flexmix** counterparts.

2.3.3 Illustrations

For illustrating the flexible usage of `raschmix()`, we employ an artificial data set drawn from one of the three data generating processes (DGPs) suggested by Rost (1990) for the introduction of Rasch mixture models. All three DGPs are provided in the function `simRaschmix()` setting the `design` to `"rost1"`, `"rost2"`, or `"rost3"`, respectively. The DGPs contain mixtures of $K = 1$, and 2, and 3 components, respectively, all with $m = 10$ items.

The DGP `"rost1"` is intended to illustrate the model's capacity to correctly determine when no DIF is present. Thus, it includes only one latent class with item parameters $\beta^{(1)} = (2.7, 2.1, 1.5, 0.9, 0.3, -0.3, -0.9, -1.5, -2.1, -2.7)^\top$. (Rost originally used opposite signs to reflect item easiness parameters but since difficulty parameters are estimated by `raschmix()` the signs have been reversed.) The DGP `"rost2"` draws observations from two latent classes of equal sizes with item parameters of opposite signs: $\beta^{(1)}$ and $\beta^{(2)} = -\beta^{(1)}$, respectively (see Figure 2.2 for an example). Finally, for the DGP `"rost3"` a third component is added with item parameters $\beta^{(3)} = (-0.5, 0.5, -0.5, 0.5, -0.5, 0.5, -0.5, 0.5, -0.5, 0.5)^\top$. The prior probabilities for the latent classes with item parameters $\beta^{(1)}$, $\beta^{(2)}$, and $\beta^{(3)}$ are 4/9, 2/9, and 3/9 respectively. In all three DGPs, the person parameters θ are drawn from a discrete uniform distribution on $\{2.7,\ 0.9, -0.9, -2.7\}$, except for the third class of DGP `"rost3"` which uses only one level of ability, drawn from the before-mentioned set of four ability levels. In all DGPs, response vectors for 1800 subjects are initially drawn but the extreme scorers who solved either none or all items are excluded.

Here, a dataset from the second DGP is generated along with two artificial covariates `x1` and `x2`. Covariate `x1` is an informative binary variable (i.e., correlated with the true group membership) while `x2` is an uninformative continuous variable.

```
R> set.seed(1)
R> r2 <- simRaschmix(design = "rost2")
R> d <- data.frame(
+    x1 = rbinom(nrow(r2), prob = c(0.4, 0.6)[attr(r2, "cluster")], size = 1),
+    x2 = rnorm(nrow(r2))
+  )
R> d$resp <- r2
```

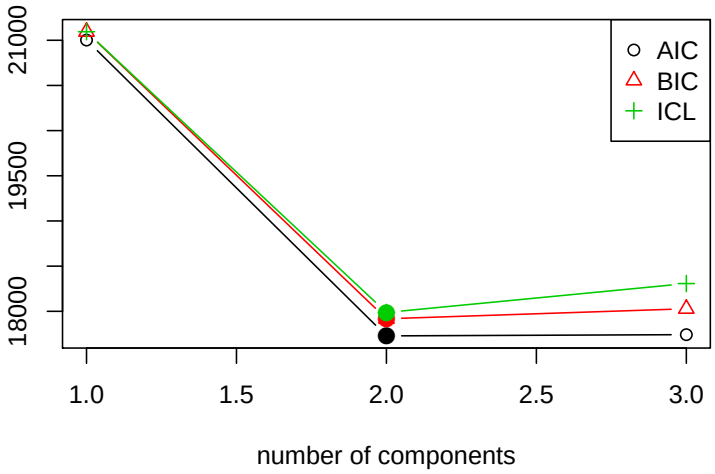

Figure 2.1: Information criteria for Rost's model with $K = 1, 2, 3$ components for the artificial scenario 2 data.

The Rost (1990) version of the Rasch mixture model – i.e., with a saturated score model and without concomitant variables – is fitted for one to three components. As no concomitants are employed in this model flavor, the matrix `r2` can be passed to `raschmix()` without formula:

```
R> set.seed(2)
R> m1 <- raschmix(r2, k = 1:3)
R> m1


Call:
raschmix(formula = r2, k = 1:3)

  iter converged k k0     logLik      AIC      BIC      ICL
1    2      TRUE 1  1 -10484.227 21002.45 21094.39 21094.39
2    9      TRUE 2  2  -8829.038 17728.08 17917.35 17987.30
3   67      TRUE 3  3  -8817.848 17741.70 18028.32 18307.13
```

To inspect the results, the returned object can either be printed, as illustrated above, or plotted yielding a visualization of information criteria (see Figure 2.1). Both printed display and visualization show a big difference in information criteria across number of components K, with the minimum always being assumed for $K = 2$, thus correctly recovering the two latent classes constructed in the underlying DGP.

The values of the information criteria can also be accessed directly via the functions of the corresponding names. To select a certain model from a "`stepRaschmix`" object, the `getModel()` function from the **flexmix** package can be employed. The specification of

`which` model is to be selected can either be an information criterion, or the number of components as a string, or the index of the model in the original vector `k`. In this particular case, `which = "BIC"`, `which = "2"`, and `which = 2` would all return the model with $K = 2$ components.

```
R> BIC(m1)
```

```
       1        2        3
21094.39 17917.35 18028.32
```

```
R> m1b <- getModel(m1, which = "BIC")
R> summary(m1b)
```

```
Call:
raschmix(formula = r2, k = 2)

       prior size post>0 ratio
Comp.1   0.5  819   1285 0.637
Comp.2   0.5  830   1301 0.638

Item Parameters:
           Comp.1     Comp.2
Item01 -2.5461200  2.6278031
Item02 -2.1053835  2.1250746
Item03 -1.6716294  1.3812228
Item04 -1.0293901  0.8535203
Item05 -0.2233486  0.3129260
Item06  0.2737782 -0.2937386
Item07  0.9561390 -0.8701059
Item08  1.5512468 -1.4464421
Item09  2.0670820 -2.0540182
Item10  2.7276257 -2.6362421

'log Lik.' -8829.038 (df=35)
AIC: 17728.08   BIC: 17917.35
```

To inspect the main properties of the model, `summary()` can be called. The information about the components of the mixture includes a priori component weights π_k and sizes as well as the estimated item parameters $\hat{\beta}$ per component. Additionally, the fitted log-likelihood and the information criteria AIC and BIC are reported. As one of the item parameters in the Rasch model is not identified, a restriction needs to be applied to the item parameters. In the output of the `summary()` function, the item parameters of each component are scaled to sum to zero.

Two other functions, `worth()` and `parameters()`, can be used to access the item parameters. The sum restriction employed in the `summary()` output is also applied by `worth()`. Additionally, `worth()` provides the possibilities to select several or just one specific `component` and to transform item `difficulty` parameters to item easiness parameters. The function `parameters()` also offers these two options but restricts the first item parameter to be zero (rather than to restrict the sum of item parameters), as this restriction is used in the internal computations. Thus, for the illustrative dataset with 10 items, `parameters()` returns 9 item parameters, leaving out the first item parameter restricted to zero while `worth()` returns 10 item parameters summing to zero. The latter corresponds to the parametrization employed by Rost (1990) and `simRaschmix()`. For convenience reasons, the true parameters are attached to the simulated dataset as an attribute named `"difficulty"`. These are printed below and visualized in Figure 2.2 (left), showing that all item parameters are recovered rather well. Note that the ordering of the components in mixture models is generally arbitrary.

```
R> parameters(m1b, "item")
```

```
                Comp.1     Comp.2
item.Item01         NA         NA
item.Item02  0.4407365 -0.5027285
item.Item03  0.8744906 -1.2465803
item.Item04  1.5167298 -1.7742828
item.Item05  2.3227714 -2.3148771
item.Item06  2.8198982 -2.9215417
item.Item07  3.5022589 -3.4979090
item.Item08  4.0973667 -4.0742453
item.Item09  4.6132020 -4.6818213
item.Item10  5.2737457 -5.2640452
```

```
R> worth(m1b)
```

```
            Comp.1     Comp.2
Item01 -2.5461200  2.6278031
Item02 -2.1053835  2.1250746
Item03 -1.6716294  1.3812228
Item04 -1.0293901  0.8535203
Item05 -0.2233486  0.3129260
Item06  0.2737782 -0.2937386
Item07  0.9561390 -0.8701059
Item08  1.5512468 -1.4464421
Item09  2.0670820 -2.0540182
Item10  2.7276257 -2.6362421
```

```
R> attr(r2, "difficulty")
```

```
      [,1] [,2]
 [1,]  2.7 -2.7
 [2,]  2.1 -2.1
 [3,]  1.5 -1.5
 [4,]  0.9 -0.9
 [5,]  0.3 -0.3
 [6,] -0.3  0.3
 [7,] -0.9  0.9
 [8,] -1.5  1.5
 [9,] -2.1  2.1
[10,] -2.7  2.7
```

In addition to the item parameters, the `parameters()` function can also return the parameters of the `"score"` model and the `"concomitant"` model (if any). The type of parameters can be set via the `which` argument. Per default `parameters()` returns both item and score parameters.

A comparison between estimated and true class membership can be conducted using the `clusters()` function and the corresponding attribute of the data, respectively. As already noticeable from the item parameters, the first component of the mixture matches the second true group of the data and vice versa. This label-switching property of mixture models in general can also be seen in the cross-table of class memberships. We thus have 32 misclassifications among the 1649 observations.

```
R> table(model = clusters(m1b), true = attr(r2, "cluster"))
```

```
     true
model   1   2
    1  14 805
    2 812  18
```

For comparison, a Rasch mixture model with mean/variance parametrization for the score probabilities, as introduced in Section 2.2.3, is fitted with one to three components and the best BIC model is selected.

```
R> set.seed(3)
R> m2 <- raschmix(data = r2, k = 1:3, scores = "meanvar")
```

```
R> m2
```

```
Call:
raschmix(data = r2, k = 1:3, scores = "meanvar")
```

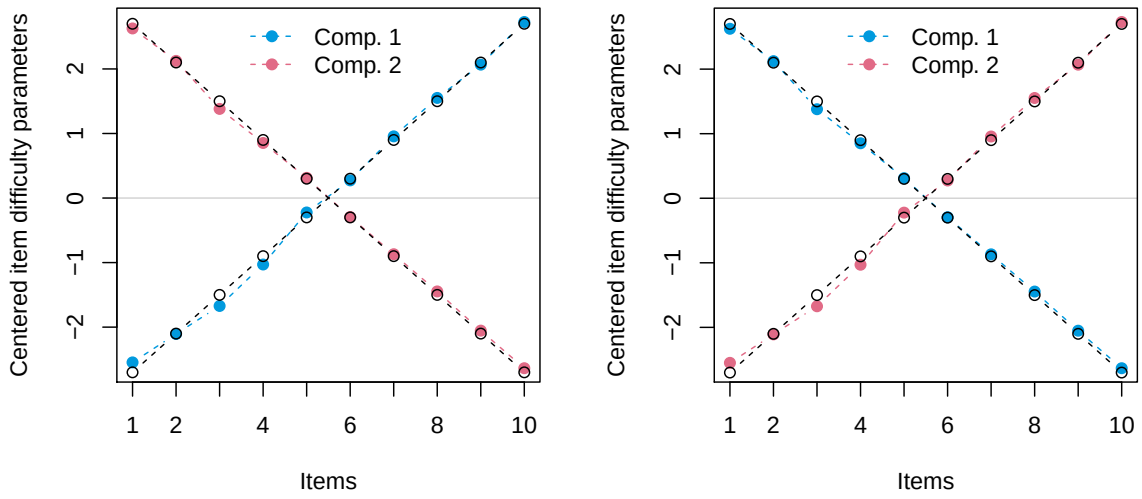

Figure 2.2: True (black) and estimated (blue/red) item parameters for the two model specifications, `"saturated"` (left) and `"meanvar"` (right), for the artificial scenario 2 data.

```
  iter converged k k0     logLik      AIC      BIC      ICL
1    2      TRUE 1  1 -10486.816 20995.63 21055.12 21055.12
2    8      TRUE 2  2  -8834.887 17715.77 17840.16 17910.21
3  121      TRUE 3  3  -8827.652 17725.30 17914.58 18352.78
```

```
R> m2b <- getModel(m2, which = "BIC")
```

As in the saturated version of the Rasch mixture model, all three information criteria prefer the two-component model. Thus, this version of a Rasch mixture model is also capable of recognizing the two latent classes in the data while using a more parsimonious parametrization with 23 instead of 35 parameters.

```
R> logLik(m2b)

'log Lik.' -8834.887 (df=23)

R> logLik(m1b)

'log Lik.' -8829.038 (df=35)
```

The estimated parameters of the distribution of the score probabilities can be accessed through `parameters()` while the full set of score probabilities is returned by `scoreProbs()`. The estimated score probabilities of the illustrative model are approximately equal across components and roughly uniform.

```
R> parameters(m2b, which = "score")
```

```
                     Comp.1     Comp.2
score.location    0.1662452  0.1001153
score.dispersion -0.1679713 -0.2514496
```

```
R> scoreProbs(m2b)
```

```
          Comp.1    Comp.2
 [1,] 0.0000000 0.0000000
 [2,] 0.1104995 0.1170100
 [3,] 0.1071901 0.1101524
 [4,] 0.1053864 0.1058038
 [5,] 0.1050149 0.1036919
 [6,] 0.1060603 0.1036871
 [7,] 0.1085652 0.1057891
 [8,] 0.1126327 0.1101268
 [9,] 0.1184334 0.1169719
[10,] 0.1262176 0.1267671
[11,] 0.0000000 0.0000000
```

The resulting item parameters for this particular data set are virtually identical to those from the saturated version, as can be seen in Figure 2.2.

To demonstrate the use of a concomitant variable model for the weights of the mixture, the two artificial variables `x1` and `x2` are employed. They are added on the right-hand side of the formula, yielding a multinomial logit model for the weights (only if `k = 2` or more components are specified).

```
R> set.seed(4)
R> cm2 <- raschmix(resp ~ x1 + x2, data = d, k = 1:3, scores = "meanvar")
```

The BIC is used to compare the models with and without concomitant variables and different number of components. The two true groups are recognized correctly with and without concomitant variables, while the model with concomitants manages to employ the additional information and reaches a somewhat improved model fit.

```
R> rbind(m2 = BIC(m2), cm2 = BIC(cm2))
```

```
            1        2        3
m2  21055.12 17840.16 17914.58
cm2 21055.12 17776.30 17867.25
```

While the likelihood ratio (LR) test cannot be employed to choose the number of components in a mixture model, it can be used to assess the concomitant variable model for a mixture model with a fixed number of components. Testing the 3-component model with concomitant variables against the 3-component model without concomitant variables yields a test statistic of 78.67 ($p < 0.001$).

As mentioned above, the parameters of the concomitant model can be accessed via the `parameters()` function, setting `which = "concomitant"`. The influence of the informative covariate `x1` is reflected in the large absolute coefficient while the estimated coefficient for the noninformative covariate `x2` is close to zero.

```
R> cm2b <- getModel(cm2, which = "BIC")
R> parameters(cm2b, which = "concomitant")
```

```
            1           2
(Intercept) 0  0.45759154
x1          0 -0.91231698
x2          0  0.02909326
```

The corresponding estimated item parameters `parameters(cm2b, "item")` are not very different from the previous models (and are hence not shown here). This illustrative application shows that the inclusion of concomitant variables can provide additional information, e.g., that `x1` but not `x2` is associated with the class membership. Note also that this is picked up although a rather weak association was simulated here.

```
R> table(x1 = d$x1, clusters = clusters(cm2b))
```

```
   clusters
x1    1   2
  0 318 501
  1 505 325
```

2.4 Empirical application: Verbal aggression

The verbal aggression dataset (De Boeck and Wilson, 2004) contains item response data from 316 first-year psychology students along with gender and trait anger (assessed by the Dutch adaptation of the state-trait anger scale) as covariates (Smits, De Boeck, and Vansteelandt, 2004). The 243 women and 73 men responded to 24 items constructed the following way: Following the description of a frustrating situation, subjects are asked to agree or disagree with a possible reaction. The situations are described by the following four sentences:

- S1: A bus fails to stop for me.
- S2: I miss a train because a clerk gave me faulty information.
- S3: The grocery store closes just as I am about to enter.
- S4: The operator disconnects me when I had used up my last 10 cents for a call.

Each reaction begins with either "I want to" or "I do" and is followed by one of the three verbally aggressive reactions "curse", "scold", or "shout", e.g., "I want to curse", "I do curse", "I want to scold", or "I do scold".

For our illustration, we use only the first two sentences which describe situations in which the others are to blame. Extreme-scoring subjects agreeing with either none or all responses are removed.

```
R> data("VerbalAggression", package = "psychotools")
R> VerbalAggression$resp2 <- VerbalAggression$resp2[, 1:12]
R> va12 <- subset(VerbalAggression,
+    rowSums(resp2) > 0 & rowSums(resp2) < 12)
R> colnames(va12$resp2)
```

```
 [1] "S1WantCurse" "S1DoCurse"   "S1WantScold" "S1DoScold"
 [5] "S1WantShout" "S1DoShout"   "S2WantCurse" "S2DoCurse"
 [9] "S2WantScold" "S2DoScold"   "S2WantShout" "S2DoShout"
```

We fit Rasch mixture models with the mean/variance score model, one to four components, and with and without the two concomitant variables, respectively (the single component model being only fitted without covariates).

```
R> set.seed(1)
R> va12_mix1 <- raschmix(resp2 ~ 1, data = va12, k = 1:4, scores = "meanvar")
R> set.seed(2)
R> va12_mix2 <- raschmix(resp2 ~ gender + anger, data = va12, k = 1:4,
+    scores = "meanvar")
```

The corresponding BIC for all considered models can be computed by

```
R> rbind(BIC(va12_mix1), BIC(va12_mix2))
```

```
           1        2        3        4
[1,] 3874.632 3857.549 3854.353 3889.432
[2,] 3874.632 3859.120 3854.823 3881.705
```

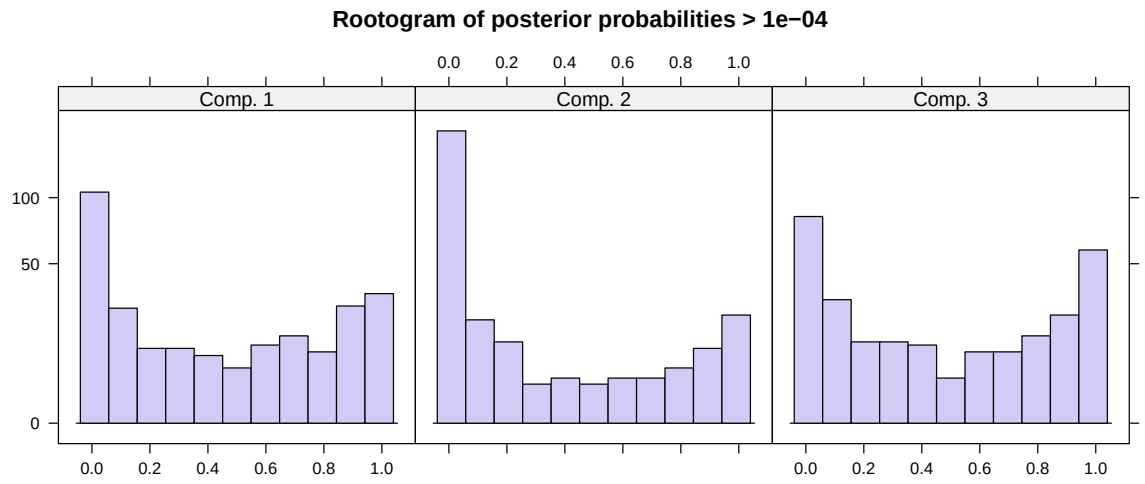

Figure 2.3: Rootogram of posterior probabilities in the 3-component Rasch mixture model on verbal aggression data.

```
R> va12_mix3 <- getModel(va12_mix2, which = "3")
```

showing that three components are preferred regardless of whether or not concomitant variables are used. The difference in BIC between the models with and without concomitant variables is very small and the LR test yields a test statistic of 21.97 ($p < 0.001$), thus the 3-component model with concomitant variables is chosen.

The posterior probabilities for the three components can be visualized via `histogram(va12_mix3)` – by default using a square-root scale, yielding a so-called rootogram – as shown in Figure 2.3. In the ideal case, posterior probabilities of the observations for each component are either high or low, yielding a U-shape in all panels. In this case here, the components are separated acceptably well.

The item profiles in the three components can be visualized via `plot(va12_mix3)` or `xyplot(va12_mix3)` with the output of the latter being shown in Figure 2.4. The first six items are responses to the first sentence (bus), the remaining six refer to the second sentence (train). The six reactions are grouped in "want"/"do" pairs: first for "curse", then "scold", and finally "shout".

The second component displays a zigzag pattern which indicates that subjects in this component always find it easier or less extreme to "want to" react a certain way rather than to actually "do" react that way. In the other two components this want/do relationship is reversed, except for the shouting response (to either situation) and the scolding response to the train situation (S2) in the first component.

In the third component, there are no big differences in the estimated item parameters. Neither any situation (S1 or S2) nor any type of verbal response (curse, scold, or shout) is perceived as particularly extreme by subjects in this component. In components 1 and 2, the situation is also not very relevant but subjects differentiate between the three verbal responses. This is best visible in component 1 where item difficulty is clearly

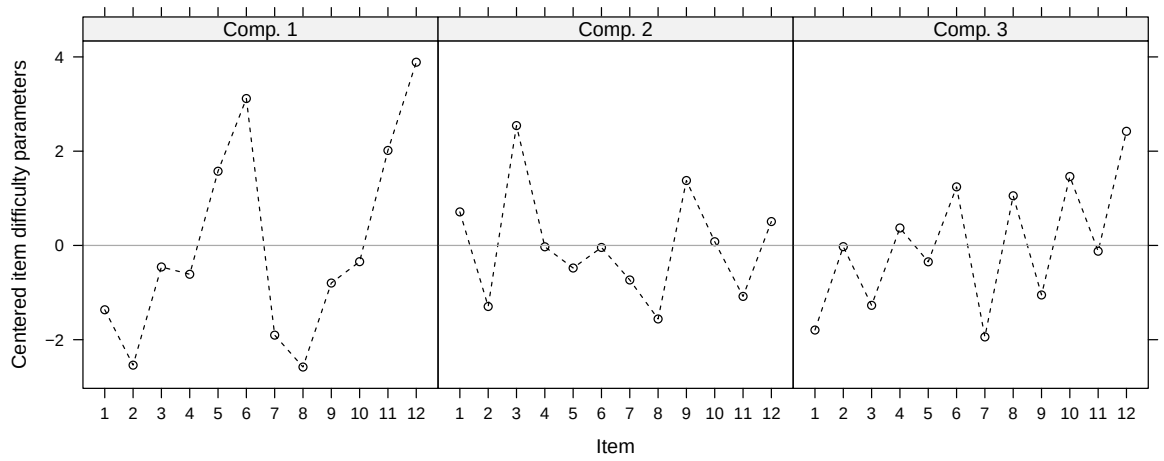

Figure 2.4: Item profiles for the 3-component Rasch mixture model on verbal aggression data. Items 1–6 pertain to situation S1 (bus), items 7–12 to situation S2 (train), each in the following order: want to curse, do curse, want to scold, do scold, want to shout, do shout.

increasing from response "curse" to response "shout". Thus, shouting is perceived as the most extreme verbal response while cursing is considered a comparably moderate response. In component 2 this pattern is also visible albeit not as prominently as in component 1.

The link between the covariates and the latent classes is described through the concomitant variable model:

```
R> parameters(getModel(va12_mix2, which = "3"), which = "concomitant")
```

```
            1          2           3
(Intercept) 0 -2.9113698  0.76066767
gendermale  0 -0.2473712 -1.66472846
anger       0  0.1152369 -0.01154008
```

The absolute sizes of the coefficients reflect that there may be some association with gender but less with the anger score. However, as there is a slight increase in BIC compared to the model without concomitants, the association with the covariates appears to be relatively weak. In comparison to other approaches exploring the association of class membership with covariates *ex post* (e.g., as in Cohen and Bolt, 2005), the main advantage of the concomitant variables model lies in the *simultaneous* estimation of the mixture and the influence of covariates.

2.5 Summary

Mixtures of Rasch models are a flexible means of checking measurement invariance and testing for differential item functioning. Here, we establish a rather general unifying conceptual framework for Rasch mixture models along with the corresponding computational tools in the R package **psychomix**. In particular, this includes the original model specification of Rost (1990) as well as more parsimonious parametrizations (Rost and von Davier, 1995), along with the possibility to incorporate concomitant variables predicting the latent classes (as in Tay *et al.*, 2011).

The R implementation is based on the infrastructure provided by the **flexmix** package, allowing for convenient model specification and selection. The rich set of methods for **flexmix** objects is complemented by additional functions specifically designed for Rasch models, e.g., extracting different types of parameters in different transformations and visualizing the estimated component-specific item parameters in various ways. Optionally, speed gains can be obtained from utilizing the C++ implementation in the **mRm** package for selecting optimal starting values. Thus, **psychomix** provides a comprehensive and convenient toolbox for the application of Rasch mixture models in psychometric research practice.

Acknowledgments

We are thankful to the participants of the "Psychoco 2011" workshop at Universität Tübingen for helpful feedback and discussions.

Chapter 3

Rasch Mixture Models for DIF Detection: A Comparison of Old and New Score Specifications

This chapter is a (slightly) modified version of Frick et al. (2014a), published in Educational and Psychological Measurement.

Abstract:
Rasch mixture models can be a useful tool when checking the assumption of measurement invariance for a single Rasch model. They provide advantages compared to manifest DIF tests when the DIF groups are only weakly correlated with the manifest covariates available. Unlike in single Rasch models, estimation of Rasch mixture models is sensitive to the specification of the ability distribution even when the conditional maximum likelihood approach is used. It is demonstrated in a simulation study how differences in ability can influence the latent classes of a Rasch mixture model. If the aim is only DIF detection, it is not of interest to uncover such ability differences as one is only interested in a latent group structure regarding the item difficulties. To avoid any confounding effect of ability differences (or impact), a new score distribution for the Rasch mixture model is introduced here. It ensures the estimation of the Rasch mixture model to be independent of the ability distribution and thus restricts the mixture to be sensitive to latent structure in the item difficulties only. Its usefulness is demonstrated in a simulation study and its application is illustrated in a study of verbal aggression.

3.1 Introduction

Based on the Rasch model (Rasch, 1960), Rost (1990) introduced what he called the "mixed Rasch model", a combination of a latent class approach and a latent trait approach

to model qualitative and quantitative ability differences. As suggested by Rost (1990), it can also be used to examine the fit of the Rasch model and check for violations of measurement invariance such as differential item functioning (DIF). Since the model assumes latent classes for which separate Rasch models hold, it can be employed to validate a psychological test or questionnaire: if a model with two or more latent classes fits better than a model with one latent class, measurement invariance is violated and a single Rasch model is not suitable because several latent classes are present in the data which require separate Rasch models with separate sets of item difficulties. These classes are latent in the sense that they are not determined by covariates.

As the model assesses a questionnaire – or instrument as it will be referred to in the following – as a whole, it works similar to a global test like the likelihood ratio (LR) test (Andersen, 1972; Gustafsson, 1980), not an item-wise test like the Mantel-Haenszel test (Holland and Thayer, 1988). Hence, it is the set of item parameters for all items, which is tested for differences between groups rather than each item parameter being tested separately.

The mixed Rasch model – here called Rasch mixture model to avoid confusion with mixed (effects) models and instead highlight its relation to mixture models – has since been extended by Rost and von Davier (1995) to different score distributions and by Rost (1991) and von Davier and Rost (1995) to polytomous responses. The so-called "mixed ordinal Rasch model" is a mixture of partial credit models (PCM, Masters, 1982) and includes a mixture of rating scale models (RSM, Andrich, 1978) as a special case.

The original dichotomous model as well as its polytomous version have been applied in a variety of fields. Zickar, Gibby, and Robie (2004) use a mixture PCM to detect faking in personality questionnaires, while Hong and Min (2007) identify three types/classes of depressed behavior by applying a mixture RSM to a self-rating depression scale. Another vast field of application are tests in educational measurement. Baghaei and Carstensen (2013) identify different reader types from a reading comprehension test using a Rasch mixture model. Maij-de Meij, Kelderman, and van der Flier (2010) also apply a Rasch mixture model to identify latent groups in a vocabulary test. Cohen and Bolt (2005) use a Rasch mixture model to detect DIF in a mathematics placement test.

Rasch mixture models constitute a legitimate alternative to DIF tests for manifest variables such as the LR test or the recently proposed Rasch trees (Strobl *et al.*, 2014). These methods are usually used to test DIF based on observed covariates, whereas Maij-de Meij *et al.* (2010) show that mixture models are more suitable to detect DIF if the "true source of bias" is a latent grouping variable. The simulation study by Preinerstorfer and Formann (2011) suggests that parameter recovery works reasonably well for Rasch mixture models. While they did not study in detail the influence of DIF effect size or the effect of different ability distributions, they deem such differences relevant for practical concern but leave it to further research to establish just how strongly they influence estimation accuracy.

As the Rasch model is based on two aspects, subject ability and item difficulty, Rasch mixture models are sensitive not only to differences in the item difficulties – as in DIF – but also to differences in abilities. Such differences in abilities are usually called

impact and do not infringe on measurement invariance (Ackerman, 1992). In practice, when developing a psychological test, one often follows two main steps. First, the item parameters are estimated, e.g., by means of the conditional maximum likelihood (CML) approach, checked for model violations and problematic items are possibly excluded or modified. Second, the final set of items is used to estimate person abilities. The main advantage of the CML approach is that, for a single Rasch model, the estimation and check of item difficulties are (conditionally) independent of the abilities and their distribution. Other global assessment methods like the LR test and the Rasch trees are also based on the CML approach to achieve such independence. However, in a Rasch mixture model, the estimation of the item difficulties is not independent of the ability distribution, even when employing the CML approach. DeMars and Lau (2011) find that a difference in mean ability between DIF groups affects the estimation of the DIF effect sizes. Similarly, other DIF detection methods are also affected by impact, e.g., inflated type I error rates occur in the Mantel-Haenszel and logistic regression procedures if impact is present (Li, Brooks, and Johanson, 2012; DeMars, 2010).

When using a Rasch mixture model for DIF detection, an influence of impact alone on the mixture is undesirable as the goal is to uncover DIF groups based on item difficulties, not impact groups based on abilities. To avoid such confounding effects of impact, we propose a new version of the Rasch mixture model specifically designed to detect DIF, which allows for the transfer of the crucial property of CML from a single Rasch model to the mixture: estimation and testing of item difficulties is independent of the abilities and their distribution.

A simulation study is conducted to illustrate how previously suggested versions and this new version of the Rasch mixture model react to impact, either alone or in combination with DIF, and how this affects the suitability of the Rasch mixture model as a DIF detection method.

In the following, we briefly discuss the Rasch model and Rasch mixture models to explain why the latter are sensitive to the specification of the score distribution despite employing a conditional maximum likelihood approach for estimation. This Section 3.2 is concluded with our suggested new score distribution. We illustrate and discuss the behavior of Rasch mixture models with different options for the score distribution in a Monte Carlo study in Section 3.3. The suggested approach for DIF detection via Rasch mixture models is illustrated through an empirical application to a study on verbally aggressive behavior in Section 3.4. Concluding remarks are provided in Section 3.5.

3.2 Theory

3.2.1 The Rasch model

The Rasch model, introduced by Georg Rasch (1960), models the probability for a binary response $y_{ij} \in \{0, 1\}$ by subject i to item j as dependent on the subject's ability θ_i and the item's difficulty β_j. Assuming independence between items given the subject, the

probability for observing a vector $y_i = (y_{i1}, \ldots, y_{im})^\top$ with responses to all m items by subject i can be written as

$$P(Y_i = y_i|\theta_i, \beta) \;=\; \prod_{j=1}^{m} \frac{\exp\{y_{ij}(\theta_i - \beta_j)\}}{1 + \exp\{\theta_i - \beta_j\}}, \tag{3.1}$$

depending on the subject's ability θ_i and the vector of all item difficulties $\beta = (\beta_1, \ldots, \beta_m)^\top$. Capital letters denote random variables and lower case letters denote their realizations.

Since joint maximum likelihood (JML) estimation of all abilities and difficulties is not consistent for a fixed number of items m (Molenaar, 1995a), conditional maximum likelihood (CML) estimation is employed here. This exploits that the number of correctly scored items, the so-called raw score $R_i = \sum_{j=1}^{m} Y_{ij}$, is a sufficient statistic for the ability θ_i (Molenaar, 1995a). Therefore, the answer probability from Equation 3.1 can be split into two factors where the first factor is conditionally independent of θ_i:

$$\begin{aligned} P(Y_i = y_i|\theta_i, \beta) &= P(Y_i = y_i|r_i, \theta_i, \beta)\; P(R_i = r_i|\theta_i, \beta) \\ &= \underbrace{P(Y_i = y_i|r_i, \beta)}_{h(y_i|r_i,\beta)}\; \underbrace{P(R_i = r_i|\theta_i, \beta)}_{g(r_i|\theta_i,\beta)}. \end{aligned}$$

Due to this separation, consistent estimates of the item parameters β can be obtained by maximizing only the conditional part of the likelihood $h(\cdot)$:

$$h(y_i|r_i, \beta) \;=\; \frac{\exp\{-\sum_{j=1}^{m} y_{ij}\beta_j\}}{\gamma_{r_i}(\beta)}, \tag{3.2}$$

with $\gamma_j(\cdot)$ denoting the elementary symmetric function of order j. The resulting CML estimates $\hat{\beta}$ are consistent, asymptotically normal, and asymptotically efficient (Molenaar, 1995a).

If not only the conditional likelihood but the full likelihood is of interest – as in Rasch mixture models – then the score distribution $g(\cdot)$ needs to be specified as well. The approach used by Rost (1990) and Rost and von Davier (1995) is to employ some distribution for the raw scores r_i based on a set of auxiliary parameters δ. Then the probability density function for y_i can be written as:

$$f(y_i|\beta, \delta) \;=\; h(y_i|r_i, \beta)\; g(r_i|\delta). \tag{3.3}$$

Based on this density, the following subsections first introduce mixture Rasch models in general and then discuss several choices for $g(\cdot)$. CML estimation is used throughout for estimating the Rasch model, i.e., the conditional likelihood $h(\cdot)$ is always specified by Equation 3.2.

3.2.2 Rasch mixture models

Mixture models are essentially a weighted sum over several components, i.e., here over several Rasch models. Using the Rasch model density function from Equation 3.3, the

likelihood $L(\cdot)$ of a Rasch mixture model with K components for data from n respondents is given by

$$\begin{aligned} L(\pi^{(1)}, \ldots, \pi^{(K)}, \beta^{(1)}, \ldots, \beta^{(K)}, \delta^{(1)}, \ldots, \delta^{(K)}) &= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi^{(k)} f(y_i | \beta^{(k)}, \delta^{(k)}) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi^{(k)} h(y_i | r_i, \beta^{(k)}) \; g(r_i | \delta^{(k)}) \end{aligned} \tag{3.4}$$

where the (k)-superscript denotes the component-specific parameters: the component weight $\pi^{(k)}$, the component-specific item parameters $\beta^{(k)}$, and the component-specific score parameters $\delta^{(k)}$ for $k = 1, \ldots, K$.

This kind of likelihood can be maximized via the expectation-maximization (EM) algorithm (Dempster *et al.*, 1977) which alternates between maximizing the component-specific likelihoods for obtaining parameter estimates and computing expectations for each observations belonging to each cluster.

More formally, given (initial) estimates for the model parameters $\hat{\pi}^{(k)}$, $\hat{\beta}^{(k)}$, $\hat{\delta}^{(k)}$ for all components $k = 1, \ldots, K$, posterior probabilities of each observation i belonging to a component, or latent class, k are calculated in the E-step. This is simply i's relative contribution to component k compared to the sum of all its contributions:

$$\hat{p}_{ik} = \frac{\hat{\pi}^{(k)} \; f(y_i | \hat{\beta}^{(k)}, \hat{\delta}^{(k)})}{\sum_{\ell=1}^{K} \hat{\pi}^{(\ell)} \; f(y_i | \hat{\beta}^{(\ell)}, \hat{\delta}^{(\ell)})} = \frac{\hat{\pi}^{(k)} \; h(y_i | r_i, \hat{\beta}^{(k)}) \; g(r_i | \hat{\delta}^{(k)})}{\sum_{\ell=1}^{K} \hat{\pi}^{(\ell)} \; h(y_i | r_i, \hat{\beta}^{(\ell)}) \; g(r_i | \hat{\delta}^{(\ell)})} \,. \tag{3.5}$$

In the M-step of the algorithm, these posterior probabilities are used as the weights in a weighted ML estimation of the model parameters. This way, an observation deemed unlikely to belong to a certain latent class does not contribute strongly to its estimation. Estimation can be done separately for each latent class. Using CML estimation for the Rasch Model, the estimation of item and score parameters can again be done separately. For all components $k = 1, \ldots, K$:

$$\begin{aligned} (\hat{\beta}^{(k)}, \hat{\delta}^{(k)}) &= \underset{\beta^{(k)}, \delta^{(k)}}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{p}_{ik} \log f(y_i | \beta^{(k)}, \delta^{(k)}) \\ &= \left\{ \underset{\beta^{(k)}}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{p}_{ik} \log h(y_i | r_i, \beta^{(k)}); \;\; \underset{\delta^{(k)}}{\operatorname{argmax}} \sum_{i=1}^{n} \hat{p}_{ik} \log g(r_i | \delta^{(k)}) \right\} \end{aligned} \tag{3.6}$$

Estimates of the class probabilities can be obtained from the posterior probabilities by averaging:

$$\hat{\pi}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_{ik}. \tag{3.7}$$

The E-step (Equation 3.5) and M-step (Equations 3.6 and 3.7) are iterated until convergence, always updating either the weights based on current estimates for the model parameters or vice versa.

Note that the above implicitly assumes that the number of latent classes K is given or known. However, this is typically not the case in practice and K needs to be chosen based

on the data. As K is not a model parameter – regularity conditions for the likelihood ratio test are not fulfilled (McLachlan and Peel, 2000, Chapter 6.4) – it is often chosen via some information criterion that balances goodness of fit (via the likelihood) with a penalty for the number of model parameters. Since the various information criteria differ in their penalty term, the decision which model is considered "best" may depend on the information criterion chosen. In the following, the BIC (Bayesian information criterion, Schwarz, 1978) is used, which Li, Cohen, Kim, and Cho (2009) found to be a suitable model selection method for dichotomous mixture item response theory models. Note that this is not a formal significance test because one does not control a type I error rate.

3.2.3 Score distribution

In a single Rasch model, the estimation of the item parameters is invariant to the score distribution because of the separation in Equation 3.3. In the mixture context, this invariance property holds only *given the weights* in Equation 3.6. However, these posterior weights depend on the full Rasch likelihood, including the score distribution (Equation 3.5). Therefore, the estimation of the item parameters in a Rasch mixture model is *not* independent of the score distribution for $K > 1$, even if the CML approach is employed. Hence, it is important to consider the specification of the score distribution when estimating Rasch mixture models and to assess the consequences of potential misspecifications.

Saturated and mean-variance specification

In his introduction of the Rasch mixture model, Rost (1990) suggests a discrete probability distribution on the scores with a separate parameter for each possible score. This requires $m-2$ parameters per latent class as the probabilities need to sum to 1 (and the extreme scores, $r = 0$ and $r = m$, do not contribute to the likelihood).

Realizing that this saturated specification requires a potentially rather large number of parameters, Rost and von Davier (1995) suggest a parametric distribution with one parameter each for mean and variance.

Details on both specifications can be found in Rost (1990) and Rost and von Davier (1995), respectively. Here, the notation of Frick *et al.* (2012) is adopted, which expresses both specifications in a unified way through a conditional logit model for the score $r = 1, \ldots, m-1$:

$$g(r|\delta^{(k)}) \;=\; \frac{\exp\{z_r^\top \delta^{(k)}\}}{\sum_{j=1}^{m-1} \exp\{z_j^\top \delta^{(k)}\}},$$

with different choices for z_r leading to the saturated and mean-variance specification, respectively. For the former, the regressor vector is $(m-2)$-dimensional with

$$z_r = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$$

and the 1 at position $r-1$. Consequently, if $r = 1$, z_r is a vector of zeros. For the mean-variance specification, the regressor vector is 2-dimensional and given by

$$z_r = \left(\frac{r}{m}, \frac{4r(m-r)}{m^2}\right)^\top.$$

Restricted specification

In the following we suggest a new specification of the score distribution in the Rasch mixture model, which aims at obtaining independence of the item parameter estimates from the specification of the score distribution and therefore enabling the Rasch mixture model to distinguish between DIF and impact. Other global DIF detection methods like the LR test and Rasch trees are able to make this distinction (Ankenmann, Witt, and Dunbar, 1999; Strobl *et al.*, 2014) because they are based only on the conditional part of the likelihood (Equation 3.2). Analogously, we suggest a mixture of only this conditional part rather than the full likelihood (Equation 3.3) of the Rasch model so that the mixture model will only be influenced by differences in the item parameters.

Mixing only the conditional likelihood $h(\cdot)$ means that the sum over the K latent classes in the likelihood of the Rasch mixture model in Equation 3.4 only applies to $h(\cdot)$ but not to the score distribution $g(\cdot)$. The mixture is then only based on latent structure in the item difficulties, not on latent structure in both difficulties and scores. Moreover, such a Rasch mixture model based only on the conditional likelihood without any score distribution is equivalent to a Rasch mixture model where the score distribution is independent of the latent class $k = 1, \ldots, K$:

$$g(r|\delta^{(k)}) = g(r|\delta) \qquad (k = 1, \ldots, K),$$

because then the factor $g(r|\delta)$ is a constant that can be moved out of the sum over the components k in Equation 3.4. Consequently, compared to the case without any score distribution, the log-likelihood just changes by an additional constant without component-specific parameters. In either case, the estimation of the component-specific parameters item parameters as well as the selection of the number of components K is independent of the specification of the score distribution.

This equivalence and independence from the score distribution can also be seen easily from the definition of the posterior weights (Equation 3.5): If restricted, $g(\cdot)$ can be moved out of the sum and then cancels out, preserving only the dependence on $h(\cdot)$. Thus, the $\hat{p}_{ik}$ depend only on $\hat{\pi}^{(k)}$ and $\hat{\beta}^{(k)}$ but not $\hat{\delta}^{(k)}$. Therefore, the component weights and component-specific item parameters can be estimated without any specification of the score distribution.

Subsequently, we adopt the restricted perspective rather than omitting $g(\cdot)$ completely, when we want to obtain a mixture model where the mixture is independent of the score distribution. From a statistical point of view this facilitates comparisons of the restricted Rasch mixture model with the corresponding unrestricted counterpart.

Overview

The different specifications of the score distribution vary in their properties and implications for the whole Rasch mixture model.

- The saturated model is very flexible. It can model any shape and is thus never misspecified. However, it needs a potentially large number of parameters which can be challenging in model estimation and selection.
- The mean-variance specification of the score model is more parsimonious as it only requires two parameters per latent class. While this is convenient for model fit and selection, it also comes at a cost: since it can only model unimodal or U-shaped distributions (see Rost and von Davier, 1995), it is partially misspecified if the score distribution is actually multimodal.
- A restricted score model is even more parsimonious. Therefore, the same advantages in model fit and selection apply. Furthermore, it is invariant to the latent structure in the score distribution. If a Rasch mixture model is used for DIF detection, this is favorable as only differences in the item difficulties influence the mixture. However, it is partially misspecified if the latent structure in the scores and item difficulties coincides.

3.3 Monte Carlo study

The simple question *"DIF or no DIF?"* leads to the question whether the Rasch mixture model is suitable as a tool to detect such violations of measurement invariance.

As the score distribution influences the estimation of the Rasch mixture model in general, it is of particular interest how it influences the estimation of the number of latent classes, the measure used to determine Rasch scalability.

3.3.1 Motivational example

As a motivation for the simulation design, consider the following example: The instrument is a knowledge test which is administered to students from two different types of schools and who have been prepared by one of two different courses for the knowledge test. Either of the two groupings might be the source of DIF (or impact). If the groupings are available as covariates to the item responses of the students, then a test for DIF between either school types or course types can be easily carried out using the LR test. However, if the groupings are not available (or even observed) as covariates, then a DIF test is still possible by means of the Rasch mixture model. The performance of such a DIF assessment is investigated in our simulation study for different effects of school and course type, respectively.

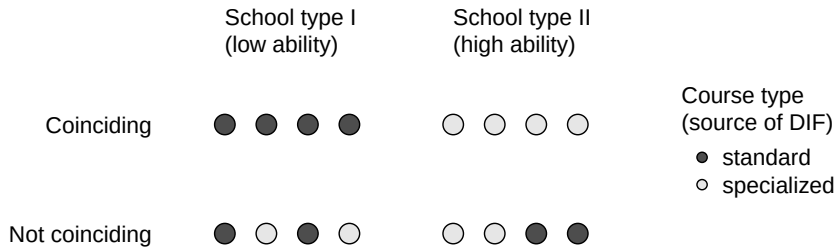

Figure 3.1: Grouping structure in the motivational example.

In the following we assume that the school type is linked to ability difference (i.e., impact but not DIF) while the course type is the source of DIF (but not impact). This can be motivated in the following way (see also Figure 3.1): When the students from the two school types differ in their mean ability, this is impact between these two groups. The courses might be a standard course and a new specialized course. While the standard course covers all topics of the test equally, the specialized course gives more emphasis to a relatively advanced topic and due to time constraints less emphasis to a relatively basic topic. This may lead to DIF between the students in the standard and the specialized course. See the left panel of Figure 3.2 for illustrative item profiles of the standard course (in dark gray) and the specialized course (in light gray).

Finally, the ability groups by school and the DIF groups by course can either coincide or not. If all students in the first school type are being taught the standard course while all students in the second school type are being taught the specialized course, the DIF groups *coincide* with the ability groups. The DIF and ability groups do *not coincide* but only overlap partly if both course types are taught in both school types: each DIF group (based on the type of course taught) consists of a mix of students from both schools and therefore from both ability groups. An illustration of coinciding and not coinciding ability and DIF groups is provided in the upper and lower row of Figure 3.1, respectively. Ability groups, based on school type, are shown in the columns, while DIF groups, based on course type, are illustrated with dark and light gray for the standard course and specialized course, respectively. This difference of coinciding or not coinciding DIF and ability groups might have an influence on the Rasch mixture model's ability to detect the DIF because in the former case the score distributions differ between the two DIF groups while in the latter case they do not.

Subsequently, a Monte Carlo study is carried out to investigate how the Rasch mixture model performs in situations where such groupings are present in the underlying data-generating process but are not available as observed covariates. Moreover, we vary whether or not all students come from the same school type (i.e., from the same ability distribution), whether or not all students receive the standard course (i.e., whether there is DIF), and whether both school types use the same or different courses (i.e., whether the groupings coincide or not). For all computations, the R system for statistical computing (R Core Team, 2013) is used along with the add-on packages **psychomix** (Frick *et al.*, 2012) and **clv** (Nieweglowski, 2013).

Scenario		Latent class I		Latent class II	
		Mean abilities	Difficulties	Mean abilities	Difficulties
No impact ($\Theta = 0$)					
1	no DIF ($\Delta = 0$)	$\{0\}$	β^I	—	—
2	DIF ($\Delta > 0$)	$\{0\}$	β^I	$\{0\}$	β^{II}
Impact ($\Theta > 0$)					
3	no DIF ($\Delta = 0$)	$\{-\Theta/2, +\Theta/2\}$	β^I	—	—
4	DIF ($\Delta > 0$), not coinciding	$\{-\Theta/2, +\Theta/2\}$	β^I	$\{-\Theta/2, +\Theta/2\}$	β^{II}
5	DIF ($\Delta > 0$), coinciding	$\{-\Theta/2\}$	β^I	$\{+\Theta/2\}$	β^{II}

Table 3.1: Simulation design. The latent-class-specific item parameters β^I and β^{II} differ by Δ for two elements and thus coincide for $\Delta = 0$, leaving only a single latent class.

3.3.2 Simulation design

The simulation design combines ideas from the motivational example with aspects from the simulation study conducted by Rost (1990). Similar to the original simulation study, the item parameters represent an instrument with increasingly difficult items. Here, 20 items are employed with corresponding item parameters β^I which follow a sequence from -1.9 to 1.9 with increments of 0.2 and hence sum to zero.

$$\begin{aligned} \beta^I &= (-1.9, -1.7, \ldots, 1.7, 1.9)^\top \\ \beta^{II} &= (-1.9, -1.7, \ldots, -1.1 + \Delta, \ldots, 1.1 - \Delta, \ldots, 1.7, 1.9)^\top \end{aligned}$$

To introduce DIF, a second set of item parameters β^{II} is considered where items 5 and 16 are changed by $\pm\Delta$. This approach is similar in spirit to that of Rost (1990) – who reverses the full sequence of item parameters to generate DIF – but allows for gradually changing from small to large DIF effect sizes. Subject abilities are drawn with equal weights from two normal distributions with means $-\Theta/2$ and $+\Theta/2$ and standard deviation 0.3, thus creating a sample with two groups of subjects: one group with a lower mean ability and one with a higher mean ability.

In the simulations below, the DIF effect size Δ ranges from 0 to 4 in steps of 0.2

$$\Delta \in \{0, 0.2, \ldots, 4\}$$

while the impact Θ covers the same range in steps of 0.4:

$$\Theta \in \{0, 0.4, \ldots, 4\}.$$

Impact and DIF, or lack thereof, can be combined in several ways. Table 3.1 provides an overview and Figures 3.2, 3.3, and 3.4 show illustrations. In the following, the different combinations of impact and DIF are explained in more detail and connected to the motivational example:

- If the simulation parameter Δ for the DIF effect size is set to zero, both sets of item parameters, β^I and β^{II}, are identical and no DIF is present. Since CML

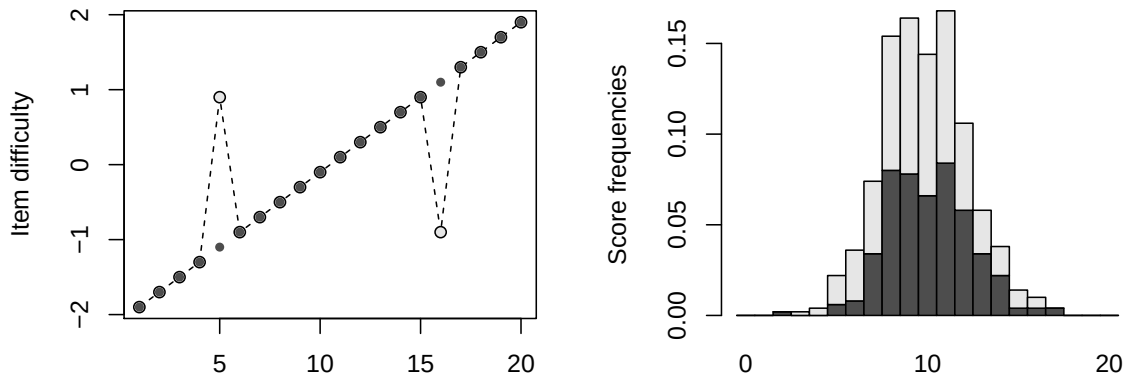

Figure 3.2: Scenario 2. Left: Item difficulties with DIF ($\Delta = 2$). Right: Stacked histogram of unimodal score distribution with homogeneous abilities ($\Theta = 0$).

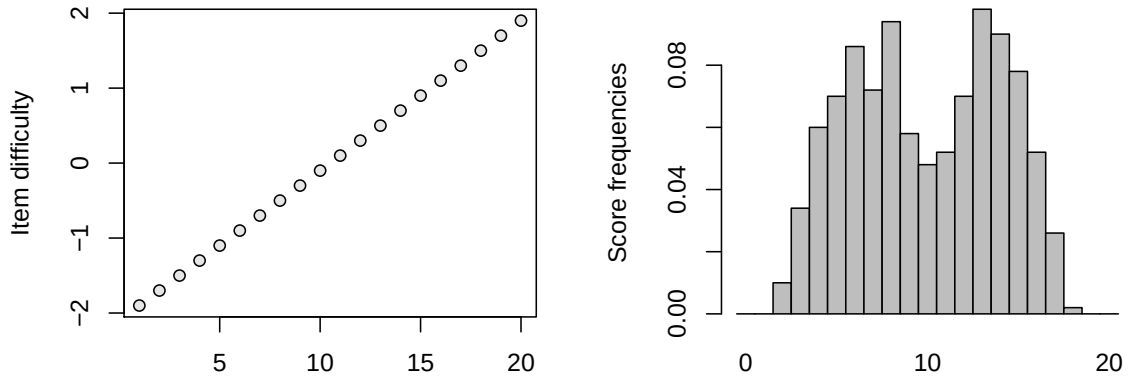

Figure 3.3: Scenario 3. Left: Item difficulties without DIF ($\Delta = 0$). Right: Histogram of bimodal score distribution with impact ($\Theta = 2$).

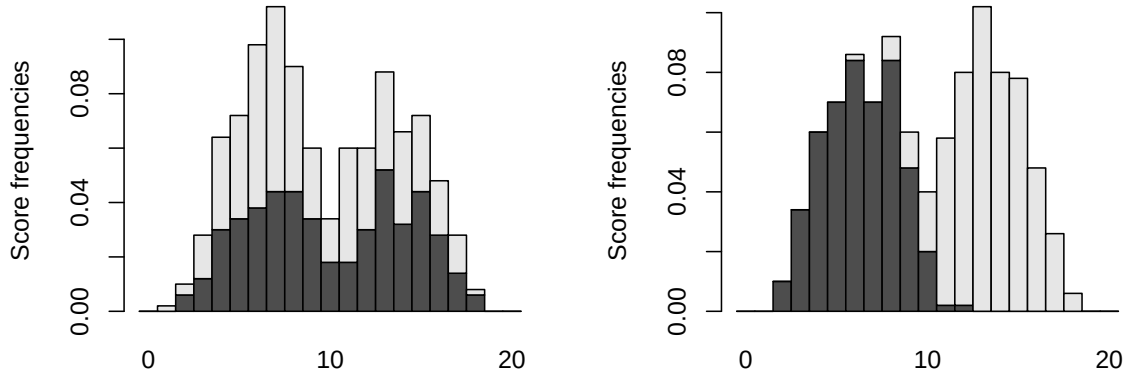

Figure 3.4: Stacked histograms of score distributions for Scenarios 4 (left) and 5 (right) with DIF ($\Delta = 2$). Left: impact and DIF, not coinciding ($\Theta = 2$). Right: impact and DIF, coinciding ($\Theta = 2$). For item difficulties see Figure 3.2 (left).

is employed, model selection and parameter estimation is typically expected to be independent of whether or not an impact is present (Scenario 1 and 3 in Table 3.1).

In the example: Only the standard course is taught and hence no DIF exists.

- If $\Delta > 0$, the item parameter set β^{II} is different from β^{I}. Hence, there is DIF and two latent classes exist (Scenarios 2, 4, and 5). Both classes are chosen to be of equal size in this case. For an illustration see the left panel of Figure 3.2.

  In the example: Both courses are taught, thus leading to DIF. The standard course corresponds to the straight line as the item profile while the specialized course corresponds to the spiked item profile with relatively difficult item 16 being easier and the relatively easy item 5 being more difficult for students in this specialized course than for students in the standard course.

- If the simulation parameter Θ for the impact is set to zero (Scenarios 1 and 2), then the resulting score distribution is unimodal. For an illustration of such a unimodal score distribution see the right panel of Figure 3.2. This histogram illustrates specifically Scenario 2 where no impact is present but DIF exists. The histogram is shaded in light and dark gray for the two DIF groups and thus to be read like a "stacked histogram".

  In the example: All students are from the same school and hence there is no impact. However, both types of courses may be taught in this one school, thus leading to DIF as in Scenario 2.

- If $\Theta > 0$, subject abilities are sampled with equal weights from two normal distributions with means $\{-\Theta/2, +\Theta/2\}$, thus generating impact. When no DIF is included (Scenario 3), the resulting score distribution moves from being unimodal to being bimodal with increasing Θ. The two modi of high and low scores represent the two groups of subjects with high and low mean abilities, respectively. However, only a medium gray is used to shade the illustrating histogram in Figure 3.3 as *no DIF groups* are present.

  In the example: Only the standard course is taught in both school types. Hence no DIF is present but impact between the school types.

- If there is DIF (i.e., $\Delta > 0$) in addition to impact (i.e., $\Theta > 0$), subjects can be grouped both according to mean ability (high vs. low) and difficulty (straight vs. spiked profile in β^{I} and β^{II}, respectively).

  These groups can *coincide*: For subjects with low mean ability $-\Theta/2$, item difficulties β^{I} hold, while for subjects with high mean ability $+\Theta/2$, item difficulties β^{II} hold. This is simulated in Scenario 5 and labeled *Impact and DIF, coinciding*. The resulting score distribution is illustrated in the right panel of Figure 3.4. Subjects for whom item difficulties β^{I} hold are shaded in dark gray and as they also have lower mean abilities, their scores are all relatively low. Conversely, subjects for whom item difficulties β^{II} hold are shaded in light gray and as they also have higher mean abilities, their scores are all relatively high.

  Additionally, the DIF groups and ability groups can also *not coincide*: Subjects in either DIF group may stem from both ability groups, not just one. This is

simulated in Scenario 4 and labeled *Impact and DIF, not coinciding*. The resulting score distribution is illustrated in the left panel of Figure 3.4. Again, subjects for whom item difficulties β^I and β^{II} hold are shaded in dark and light gray, respectively. As subjects stem from both ability groups (high vs. low abilities), both score distributions are bimodal.

In the example: Students from both school types and from both course types are considered, thus leading to both impact and DIF. Either both courses are taught at both schools (Scenario 4, not coinciding) or the standard course is only taught in the first school and the specialized course is only taught at the second school (Scenario 5, coinciding).

Note that Scenario 1 is a special case of Scenario 2 where Δ is reduced to zero as well as a special case of Scenario 3 where Θ is reduced to zero. Therefore, in the following, Scenario 1 is not inspected separately but included in both the setting of *No impact with DIF* (Scenario 2) and the setting of *Impact without DIF* (Scenario 3) as a reference point. Similarly, Scenarios 4 and 5 both can be reduced to Scenario 3 if Δ is set to zero. It is therefore also included in both the setting of *Impact and DIF, not coinciding* (Scenario 4) and the setting of *Impact and DIF, coinciding* (Scenario 5) as a reference point.

For each considered combination of Δ and Θ, 500 datasets of 500 observations each are generated. Larger numbers of datasets or observations lead to very similar results. Observations with raw scores of 0 or m are removed from the dataset as they do not contribute to the estimation of the Rasch mixture model (Rost, 1990). For each dataset, Rasch mixture models for each of the saturated, mean-variance, and restricted score specifications are fitted for $K = 1, 2, 3$.

3.3.3 False alarm rate and hit rate

The main objective here is to determine how suitable a Rasch mixture model, with various choices for the score model, is to recognize DIF or the lack thereof.

For each dataset and type of score model, models with $K = 1, 2, 3$ latent classes are fitted and the $\hat{K}$ associated with the minimum BIC is selected. Choosing one latent class ($\hat{K} = 1$) then corresponds to assuming measurement invariance while choosing more than one latent class ($\hat{K} > 1$) corresponds to assuming violations of measurement invariance. While Rasch mixture models do not constitute a formal significance test, the empirical proportion among the 500 datasets with $\hat{K} > 1$ corresponds in essence to the power of DIF detection if $\Delta > 0$ (and thus two true latent classes exist) and to the associated type I error of a corresponding test if $\Delta = 0$ (and thus only one true latent class exists). If the rate corresponds to power, it will be referred to as *hit rate* whereas if it corresponds to a type I error it will be referred to as *false alarm rate*.

In the following subsections, the key results of the simulation study will be visualized. The exact rates for all conditions are included as a dataset in the R package **psychomix**,

for details see the section on computational details.

Scenario 2: No impact with DIF

This scenario is investigated as a case of DIF that should be fairly simple to detect. There is no impact as abilities are homogeneous across all subjects so the only latent structure to detect is the group membership based on the two item profiles. This latent structure is made increasingly easy to detect by increasing the difference between the item difficulties for both latent groups. In the graphical representation of the item parameters (left panel of Figure 3.2) this corresponds to enlarging the spikes in the item profile.

Figure 3.5 shows how the rate of choosing a model with more than one latent class ($\hat{K} > 1$) increases along with the DIF effect size Δ. At $\Delta = 0$, this is a false alarm rate. It is around 7% for the saturated model and very close to zero for the mean-variance and the saturated score model ($< 1\%$). With increasing $\Delta > 0$, the rate is a hit rate. For low values of Δ the two more parsimonious versions of the Rasch mixture model (with mean-variance and restricted score distribution) are not able to pick up the DIF but at around $\Delta = 3$ the hit rate for the two models increases and almost approaches 1 at $\Delta = 4$. Not surprisingly, the restricted score specification performs somewhat better because in fact the raw score distributions do not differ between the two latent classes. The baseline hit rate of the saturated model for low values of Δ is the same as the false alarm rate for $\Delta = 0$. It only increases beyond the same threshold ($\Delta = 3$) as the hit rate of the other two models. However, its rate is much lower compared to the other two score model (only around 30%). The reason is that it requires 18 additional score parameters for an additional latent class which is "too costly" in terms of BIC. Hence, $\hat{K} = 1$ is chosen for most Rasch mixture models using a saturated score distribution.

The number of iterations in the EM algorithm which are necessary for the estimation to converge is much lower for the mean-variance and the restricted model than for the saturated model. Since the estimation of the saturated model is more extensive due to the higher number of parameters required by this model, it does not converge in about 10% of the cases before reaching the maximum number of iterations which was set to 400. The mean-variance and saturated model usually converge within the first 200 iterations.

Brief summary: The mean-variance and restricted model have higher hit rates than the saturated model in the absence of impact.

Scenario 3: Impact without DIF

Preferably, a Rasch mixture model should not only detect latent classes if the assumption of measurement invariance is violated but it should also indicate a lack of latent structure if indeed the assumption holds. In this scenario, the subjects all stem from the same class, meaning each item is of the same difficulty for every subject. However, subject abilities are simulated with impact resulting in a bimodal score distribution as illustrated in Figure 3.3.

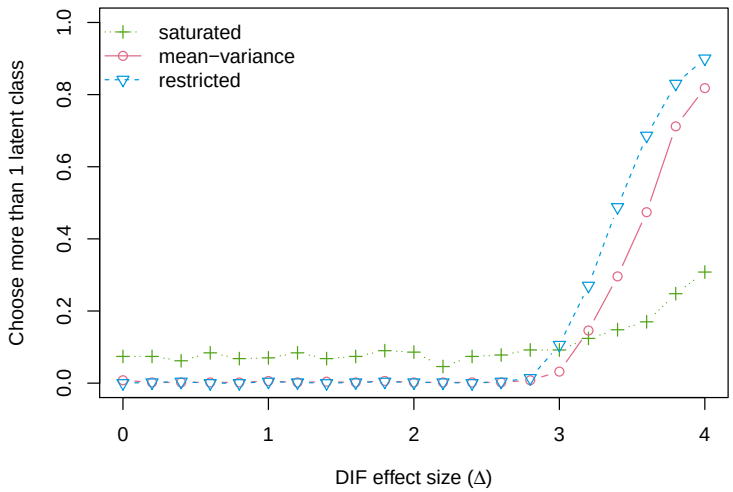

Figure 3.5: Rate of choosing a model with $\hat{K} > 1$ latent classes for data from Scenario 2 (DIF without impact, i.e., $\Theta = 0$).

Here, the rate of choosing more than one latent class can be interpreted as a false alarm rate (Figure 3.6). The restricted score model is invariant against any latent structure in the score distribution and thus almost always ($\leq 0.2\%$) suggests $\hat{K} = 1$ latent class based on the DIF-free item difficulties. The rate does not approach any specific significance level as the Rasch mixture model, regardless of the employed score distribution, is not a formal significance test. The saturated model also picks $\hat{K} = 1$ in most of the simulation. This might be due to its general reluctance to choose more than one latent class as illustrated in Figure 3.5 or the circumstance that it can assume any shape (including bimodal patterns). However, the mean-variance score distribution can only model unimodal or U-shaped distributions as mentioned above. Hence, with increasing impact and thus increasingly well-separated modes in the score distribution, the Rasch mixture model with this score specification suggests $\hat{K} > 1$ latent classes in up to 53% of the cases. Note, however, that these latent classes do not represent the DIF groups (as there are none) but rather groups of subjects with high vs. low abilities. While this may be acceptable (albeit unnecessarily complex) from a statistical mixture modeling perspective, it is misleading from a psychometric point of view if the aim is DIF detection. Only one Rasch model needs to be estimated for this type of data, consistent item parameter estimates can be obtained via CML and all observations can be scaled in the same way.

Brief summary: If measurement invariance holds but ability differences are present, the mean-variance model exhibits a high false alarm rate while the saturated and restricted model are not affected.

Scenario 4: Impact and DIF, not coinciding

In this scenario, there is DIF (and thus two true latent classes) if $\Delta > 0$. Again, Scenario 3 with $\Delta = 0$ (and thus without DIF) is included as a reference point. However,

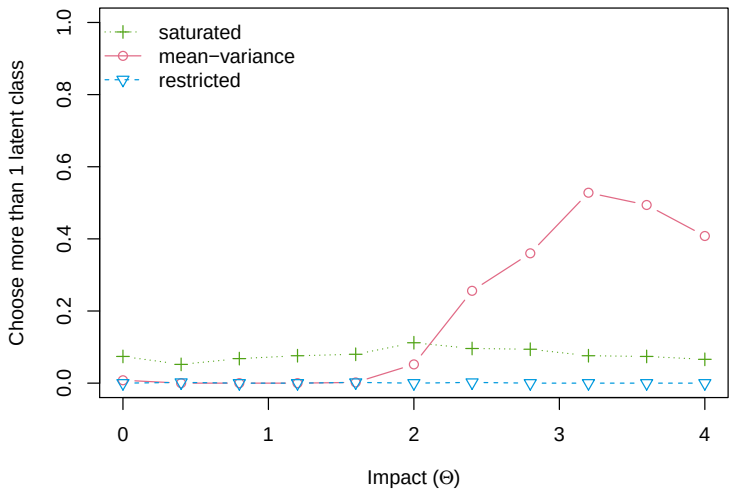

Figure 3.6: Rate of choosing a model with $\hat{K} > 1$ latent classes for data from Scenario 3 (impact without DIF, i.e., $\Delta = 0$).

unlike in Scenario 2, the abilities within the latent classes are not homogeneous but two ability groups exist, which do not coincide with the two DIF groups. Nonetheless, the score distribution is the same across both latent classes (illustrated in the left panel of Figure 3.4).

Figure 3.7 again shows the rate of choosing $\hat{K} > 1$ for increasing DIF effect size Δ for two levels of impact ($\Theta = 2.4$ and 3.6), exemplary for medium and high impact. If impact is small (e.g., $\Theta = 0.4$), the rates are very similar to the case of completely homogeneous abilities without impact (Figure 3.5 with $\Theta = 0$) and thus not visualized here. While the rates for the restricted and the saturated score model do not change substantially for an increased impact ($\Theta = 2.4$ and 3.6), the mean-variance model is influenced by this change in ability differences. While the hit rate is increased to around 20% over the whole range of Δ, the false alarm rate at $\Delta = 0$ is increased to the same extent. Moreover, the hit rate only increases noticeably beyond the initial false alarm rate at around $\Delta = 3$, i.e., the same DIF effect size at which the restricted and mean-variance specifications have an increasing hit rate given homogeneous abilities without impact. Thus, given rather high impact ($\Theta = 3.6$) the hit rate is not driven by the DIF detection but rather the model's tendency to assign subjects with high vs. low abilities into different groups (as already seen in Figure 3.6).

As Rasch mixture models with $K = 1, 2, 3$ classes are considered, selecting $\hat{K} > 1$ classes can either mean selecting the correct number of $K = 2$ or overselecting $\hat{K} = 3$ classes. For the saturated and restricted specifications overselection is rare (occurring with rates of less than 9% or less than 1%, respectively). However, similar to Scenario 3 overselection is not rare for the mean-variance specification. Figure 3.8 depicts the rates of selecting $\hat{K} = 2$ and $\hat{K} = 3$ classes, respectively, for increasing Δ at $\Theta = 3.6$. If the chances of finding the correct number of classes increase with the DIF effect size Δ, the rate for overselection ($\hat{K} = 3$) should drop with increasing Δ. For this Scenario 4, denoted with hollow symbols, this rate stays largely the same (around 25%) and even slightly increases

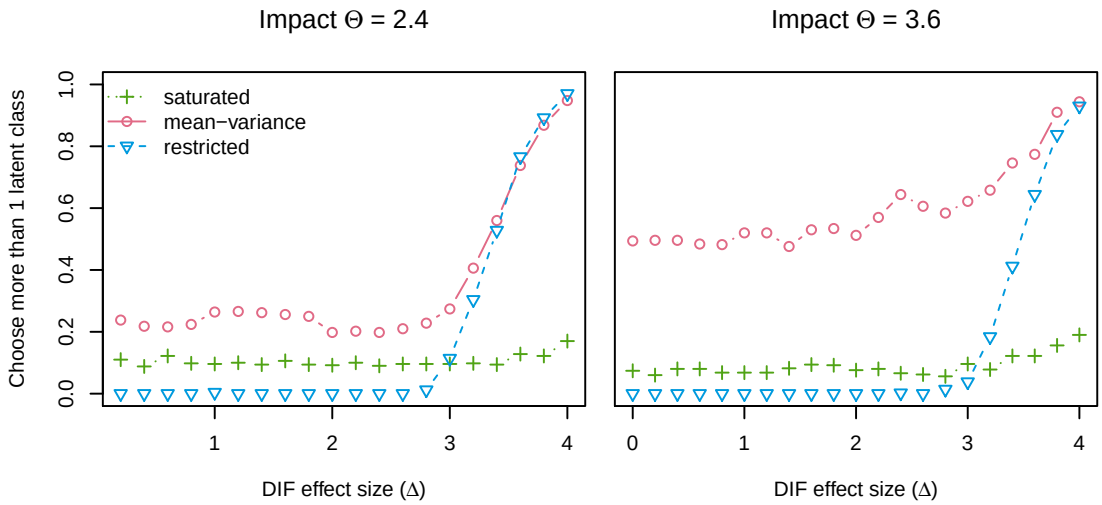

Figure 3.7: Rate of choosing a model with $\hat{K} > 1$ latent classes for data from Scenario 4 (impact and DIF, not coinciding).

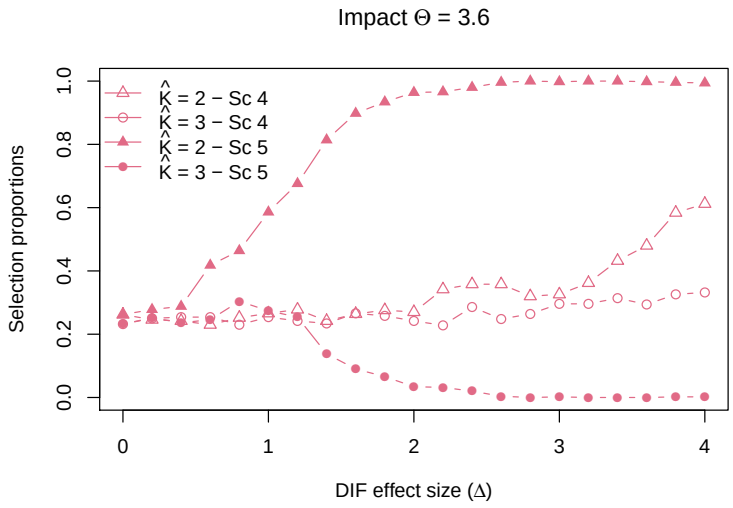

Figure 3.8: Rates of choosing the correct number of classes ($\hat{K} = 2$) or overselecting the number of classes ($\hat{K} = 3$) for the Rasch mixture model with mean-variance score specification in Scenarios 4 (hollow, impact within DIF groups) and 5 (solid, impact between DIF groups).

beyond this level, starting from around $\Delta = 3$. This illustrates again the pronounced tendency of the mean-variance model for overselection in cases of high impact.

Brief summary: If impact is simulated within DIF groups, the mean-variance model has higher hit rates than the saturated and restricted models. However, the latent classes estimated by the mean-variance model are mostly based on ability differences if the DIF effect size is low. If the DIF effect size is high, the mean-variance model tends to overestimate the number of classes.

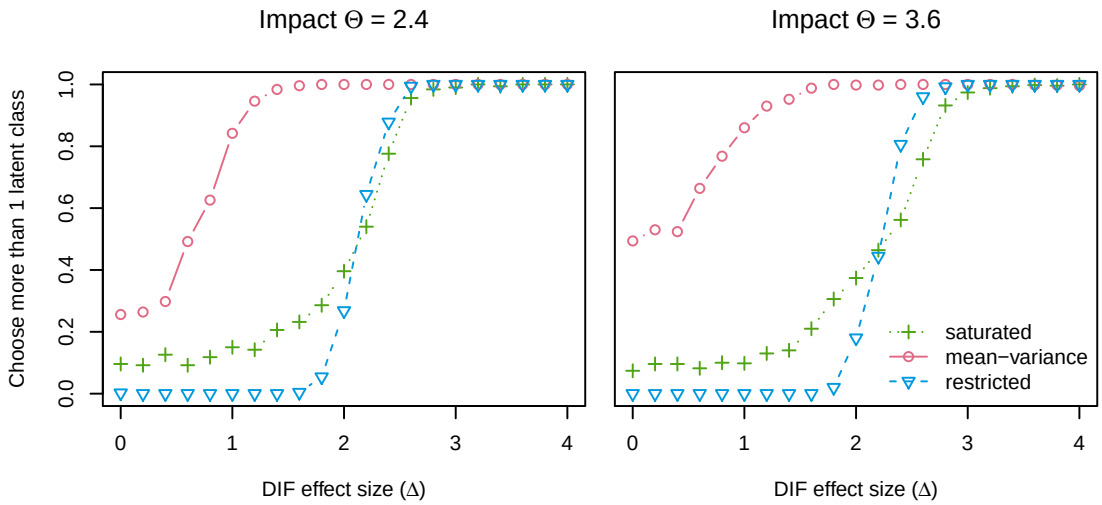

Figure 3.9: Rate of choosing a model with $\hat{K} > 1$ latent classes for data from Scenario 5 (impact and DIF, coinciding).

Scenario 5: Impact and DIF, coinciding

In Scenario 5, there is also DIF (i.e., $\Delta > 0$) and impact. However, in contrast to Scenario 4 the ability and DIF groups coincide (see the right panel of Figure 3.4). Furthermore, Scenario 3 is included also here as the reference point without DIF ($\Delta = 0$).

Again, small ability differences do not strongly influence the rate of choosing more than one latent class (rates for low levels of impact, such as $\Theta = 0.4$, are similar to those for $\Theta = 0$ as depicted in Figure 3.5). Recall, both mean-variance and restricted specification have comparable hit rates for DIF detection starting from around $\Delta = 3$ while the saturated specification has lower hit rates.

As impact increases (Figure 3.9), the hit rates of all models increases as well because the ability differences contain information about the DIF groups: separating subjects with low and high abilities also separates the two DIF groups (not separating subjects within each DIF group as in the previous setting). However, for the mean-variance model these increased hit rates are again coupled with a highly increased false alarm rate at $\Delta = 0$ of 26% and 50% for $\Theta = 2.4$ and 3.6, respectively. The restricted score model, on the other hand, is invariant to latent structure in the score distribution and thus performs similarly as in previous DIF scenarios, suggesting more than one latent class past a certain threshold of DIF intensity, albeit this threshold being a bit lower than when ability groups and DIF groups do not coincide (around $\Delta = 2$). The saturated model detects more than one latent class at a similar rate to the restricted score model for medium or high impact but its estimation converges more slowly and requires more iterations of the EM algorithm than the other two score models.

Finally, the potential issue of overselection can be considered again. Figure 3.8 (solid symbols) shows that this problem disappears for the mean-variance specification if both

DIF effect size Δ and impact are large *and* coincide. For the restricted model overselection is again very rare throughout (occurring in less than 1% of all cases) while the saturated model overselects in up to 29% of the datasets.

Brief summary: If abilities differ between DIF groups, the mean-variance model detects the violation of measurement invariance for smaller DIF effect sizes than the saturated and restricted model. While the mean-variance model does not overselect the number of components in this scenario, the high hit rates are connected to a high false alarm rate when no DIF is present but impact is high. This does not affect the other two score models.

3.3.4 Quality of estimation

Although here the Rasch mixture model is primarily used analogously to a global DIF test, model assessment goes beyond the question whether or not the correct number of latent classes is found. Once the number of latent classes is established/estimated, it is of interest how well the estimated model fits the data. Which groups are found? How well are the parameters estimated? In the context of Rasch mixture models with different score distributions, both of these aspects depend heavily on the posterior probabilities $\hat{p}_{ik}$ (Equation 3.5) as the estimation of the item parameters depends on the score distribution only through these. If the $\hat{p}_{ik}$ were the same for all three score specifications, the estimated item difficulties were the same as well. Hence, the focus here is on how close the estimated posterior probabilities are to the true latent classes in the data. If the similarity between these is high, CML estimation of the item parameters within the classes will also yield better results for all score models.

This is a standard task in the field of cluster analysis and we adopt the widely used Rand index (Rand, 1971) here: Each observation is assigned to the latent class for which its posterior probability is highest yielding an estimated classification of the data which is compared to the true classification. For this comparison, pairs of observations are considered. Each pair can either be in the same class in both the true and the estimated classification, in different classes for both classifications or it can be in the same class for one but not the other classification. The Rand index is the proportion of pairs for which both classifications agree. Thus, it can assume values between 0 and 1, indicating total dissimilarity and similarity, respectively.

In the following, the Rand index for models with the true number of $K = 2$ latent classes in Scenarios 4 and 5 (with DIF) is considered. Thus, the question of DIF detection (or model selection) is not investigated again but only the quality of latent class recovery (assuming the number of classes K to be known or correctly selected). The top row of Figure 3.10 depicts the average Rand index for data from Scenario 4 (impact and DIF, not coinciding). Here, all three score specifications find similarly well matching classifications, while the Rand index generally decreases with increasing impact (left to right panel). In particular, while the mean-variance score model has problems finding the *correct number* of latent classes in this scenario, it only performs slightly worse than the other two specifications in determining the *correct classes* if the number were known.

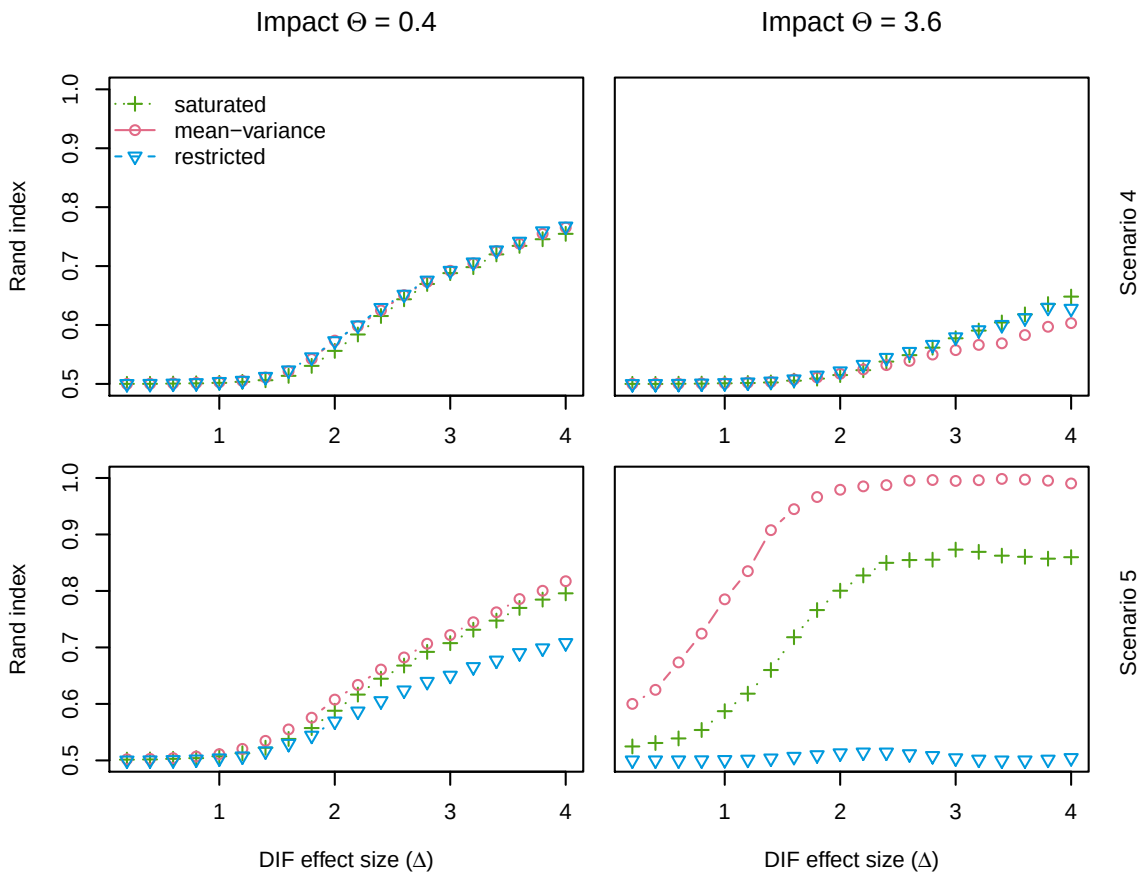

Figure 3.10: Average Rand index for models with $K = 2$ latent classes. Top row: Scenario 4 (impact and DIF, not coinciding). Bottom row: Scenario 5 (impact and DIF, coinciding).

Similarly, if it is provided with the correct number of classes, the saturated model also identifies the correct classes equally well compared to the other models – despite its difficulties with convergence for higher DIF effect sizes.

However, in Scenario 5 where the score distribution contains information about the DIF groups, the three score specifications perform very differently as the bottom row of Figure 3.10 shows. Given the correct number of classes, the mean-variance model is most suitable to uncover the true latent classes, yielding Rand indices close to 1 if both DIF effect size and impact are large. The saturated specification follows a similar pattern albeit with poorer results, reaching values of up to 0.87. However, the classifications obtained from the restricted score specification do not match the true groups well in this scenario, remaining below 0.52 if impact is high. The reason is that the restricted score model is partially misspecified as the score distributions differ substantially across DIF groups.

3.3.5 Summary and implications for practical use

Given various combinations of DIF and ability impact, the score models are differently suitable for the two tasks discussed here – DIF detection and estimation of item parameters in subgroups. Starting with a summary of the results for DIF detection:

- The saturated score model has much lower hit rates than the other two specifications, i.e., violation of measurement invariance remains too often undetected. Only if high impact and high DIF effect sizes coincide does the saturated model perform similarly well as the restricted model.
- The mean-variance model has much higher hit rates. However, if impact is present in the abilities, this specification has highly inflated false alarm rates. Hence, if the mean-variance model selects more than one latent class it is unclear whether this is due to DIF or just varying subject abilities. Thus, measurement invariance might still hold even if more than one latent class is detected.
- The restricted score model also has high hit rates, comparable to the mean-variance model if abilities are rather homogeneous. But unlike the mean-variance specification, its false alarm rate is not distorted by impact. Its performance is not influenced by the ability distribution and detecting more than one latent class reliably indicates DIF, i.e., a violation of measurement invariance.

Hence, if the Rasch mixture model is employed for assessing measurement invariance or detecting DIF, then the restricted score specification appears to be most robust. Thus, the selection of the number of latent classes should only be based on this specification.

DeMars (2010) illustrate how significance tests based on the observed (raw) scores in reference and focal groups suffer from inflated type I error rates with an increased sample size if impact is present. This does not apply to the false alarm rate of Rasch mixture models because not a significance test but rather model selection via BIC is carried out. The rate of the BIC selecting the correct model increases with larger sample size if the true model is a Rasch mixture model. Since consistent estimates are employed, a larger sample size also speeds up convergence, which is particularly desirable for the saturated model if the number of latent classes and thus the number of parameters is high.

Given the correct number of classes, the different score models are all similarly suitable to detect the true classification if ability impact does not contain any additional information about the DIF groups. However, if ability impact is highly correlated with DIF groups in the data and the ability groups thus coincide with the DIF groups, this information can be exploited by the unrestricted specifications while it distracts the restricted model.

Thus, while the selection of the number of latent classes should be based only on the restricted score specification, the unrestricted mean-variance and saturated specifications might still prove useful for estimating the Rasch mixture model (after $\hat{K}$ has been selected).

We therefore recommend a two-step approach for DIF detection via a Rasch mixture model. First, the number of latent classes is determined via the restricted score model. Second, if furthermore the estimation of the item difficulties is of interest, the full selection of score models can the be utilized. While the likelihood ratio test is not suitable to test for the number of latent classes, it can be used to establish the best fitting score model, given the number of latent classes. If this approach is applied to the full range of score models (saturated and mean-variance, both unrestricted and restricted), the nesting structure of the models needs to be kept in mind.

3.4 Empirical application: Verbal aggression

We use a dataset on verbal aggression (De Boeck and Wilson, 2004) to illustrate this two-step approach of first assessing measurement invariance via a Rasch mixture model with a restricted score distribution and then employing all possible score models to find the best fitting estimation of the item difficulties.

Participants in this study are presented with one of two potentially frustrating situations (S1 and S2):

- S1: A bus fails to stop for me.
- S2: I miss a train because a clerk gave me faulty information.

and a verbally aggressive response (cursing, scolding, shouting). Combining each situation and response with either "I want to" or "I do" leads to the following 12 items:

S1WantCurse	S1DoCurse	S1WantScold	S1DoScold	S1WantShout	S1DoShout
S2WantCurse	S2DoCurse	S2WantScold	S2DoScold	S2WantShout	S2DoShout

First, we assess measurement invariance with regard to the whole instrument: we fit a Rasch mixture model with a restricted score distribution for $K = 1, 2, 3, 4$ and employ the BIC for model selection. Note that the restricted versions of the mean-variance and saturated model only differ in their log-likelihood by a constant factor and therefore lead to the same model selection. Results are presented in Table 3.2.

The BIC for a Rasch mixture model with more than one latent class is smaller than the BIC for a single Rasch model, thus indicating that measurement invariance is violated. The best fitting model has $\hat{K} = 3$ latent classes. Given this selection of K, we want to gain further insight in the data and thus want to establish the best fitting Rasch mixture model with $K = 3$ latent classes. Four different models are conceivable: either using a restricted or unrestricted score model, and either using a saturated or mean-variance specification. The results for all four options are presented in Table 3.3. Note that the models with restricted saturated score distribution and restricted mean-variance score

Model	k	#Df	$\log L$	BIC
restricted (mean-variance)	1	13	−1900.9	3874.6
restricted (mean-variance)	2	25	−1853.8	3847.8
restricted (mean-variance)	**3**	**37**	**−1816.9**	**3841.4**
restricted (mean-variance)	4	49	−1792.0	3858.8

Table 3.2: DIF detection by selecting the number of latent classes $\hat{K}$ using the restricted Rasch mixture model.

Model	k	#Df	$\log L$	BIC
saturated	3	65	−1795.2	3955.1
restricted (saturated)	3	45	−1814.1	3880.6
mean-variance	3	41	−1812.2	3854.4
restricted (mean-variance)	**3**	**37**	**−1816.9**	**3841.4**

Table 3.3: Selection of the score distribution given the number of latent classes $\hat{K} = 3$.

distribution lead to identical item parameter estimates. However, it is still of interest to fit them separately because each of the restricted specifications is nested within the corresponding unrestricted specification. Furthermore, the mean-variance distribution is nested within the saturated distribution.

As $K = 3$ is identical for all of these four models, standard likelihood ratio tests can be used for comparing all nested models with each other. Testing the most parsimonious score model, the restricted mean-variance model, against its unrestricted version and the restricted saturated model at a 5% level shows that a more flexible score model does not yield a significantly better fit. The p-value are 0.051 and 0.686, respectively. Hence, the restricted mean-variance distribution is adopted here which also has the lowest BIC.

To visualize how the three classes found in the data differ, the corresponding item profiles are shown in Figure 3.11.

- The latent class in the right panel (with 111 observations) shows a very regular zig-zag-pattern where for any type of verbally aggressive response actually "doing" the response is considered more extreme than just "wanting" to respond a certain way as represented by the higher item parameters for the second item, the "do-item", than the first item, the "want-item", of each pair. The three types of response (cursing, scolding, shouting) are considered increasingly aggressive, regardless of the situation (first six items vs. last six items).
- The latent class in the left panel (with 53 observations) distinguishes more strongly between the types of response. However, the relationship between wanting and doing is reversed for all responses except shouting. It is more difficult to agree to the item "I want to curse/scold" than to the corresponding item "I do curse/scold". This could be interpreted as generally more aggressive behavior where one is quick to react a certain way rather than just wanting to react that way. However, shouting is considered a very aggressive response, both in wanting and doing.

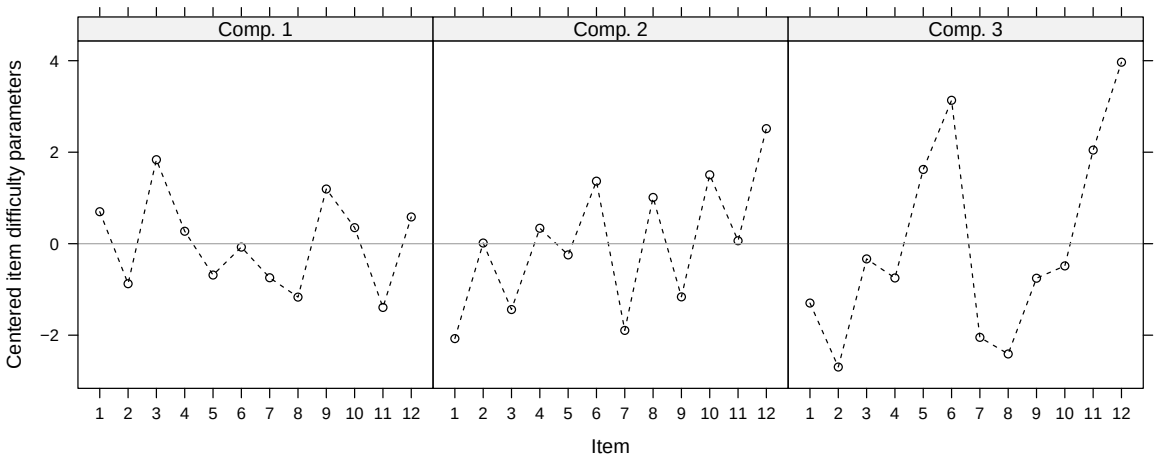

Figure 3.11: Item profiles for the Rasch mixture model with $\hat{K} = 3$ latent classes using a restricted mean-variance score distribution for the verbal aggression data.

- The remaining latent class (with 109 observations considerably smaller), depicted in the middle panel, does not distinguish that clearly between response types, situations or wanting vs. doing.

Therefore, not just a single item or a small number of items have DIF but the underlying want/do relationship of the items is different across the three classes. This instrument thus works differently as a whole across classes.

In summary, the respondents in this study are not scalable to one single Rasch-scale but instead need several scales to represent them accurately. A Rasch mixture model with a restricted score distribution is used to estimate the number of latent classes. Given that number of classes, any type of score model is conceivable. Here, the various versions are all fairly similar and the restricted mean-variance specification is chosen based on likelihood ratio tests. Keep in mind that the resulting fits can be substantially different from each other as shown in the simulation study, in particular for the case of impact between DIF classes. The latent classes estimated here differ mainly in their perception of the type and the "want/do"-relationship of a verbally aggressive response.

3.5 Conclusion

Unlike in a single Rasch model, item parameter estimation is not independent of the score distribution in Rasch mixture models. The saturated and mean-variance specification of the score model are both well-established. A further option is the new restricted score specification introduced here. In the context of DIF detection, only the restricted score specification should be used as it prevents confounding effects of impact on DIF detection while exhibiting hit rates positively related to DIF effect size. Given the number of latent classes, it may be useful to fit the other score models as well, as they might improve

estimation of group membership and therefore estimation of the item parameters. The best fitting model can be selected via the likelihood ratio test or an information criterion such as the BIC. This approach enhances the suitability of the Rasch mixture model as a tool for DIF detection as additional information contained in the score distribution is only employed if it contributes to the estimation of latent classes based on measurement invariance.

Computational details

An implementation of all versions of the Rasch mixture model mentioned here is freely available under the General Public License in the R package **psychomix** from the Comprehensive R Archive Network. Accompanying the package at http://CRAN.R-project.org/package=psychomix is a dataset containing all the simulation results and a vignette with a replication of the verbal aggression example.

Acknowledgments

This work was supported by the Austrian Ministry of Science BMWF as part of the UniInfrastrukturprogramm of the Focal Point Scientific Computing at Universität Innsbruck.

Chapter 4

To Split or to Mix? Tree vs. Mixture Models for Detecting Subgroups

This chapter is a (slightly) modified version of Frick et al. (2014b), published in COMPSTAT 2014 - Proceedings in Computational Statistics.

Abstract:
A basic assumption of many statistical models is that the same set of model parameters holds for the entire sample. However, different parameters may hold in subgroups (or clusters) which may or may not be explained by additional covariates. Finite mixture models are a common technique for detecting such clusters and additional covariates (if available) can be included as concomitant variables. Another approach that relies on covariates for detecting the clusters are model-based trees. These recursively partition the data by splits along the covariates and fit one model for each of the resulting subgroups. Both approaches are presented in a unifying framework and their relative (dis)advantages for (a) detecting the presence of clusters and (b) recovering the grouping structure are assessed in a simulation study, varying both the parameter differences between the clusters and their association with the covariates.

4.1 Introduction

A basic assumption of many statistical models is that its set of parameters applies to all observations. However, subgroups may exist for which different sets of parameters hold, e.g., the relationship between some response and regressors might be different for younger and older individuals. If the breakpoint which separates "younger" and "older" were known, parameter stability can simply be assessed by checking for parameter differences between these two specific subgroups. However, if the breakpoint is unknown or if there is a smooth transition between "young" and "old", the subgroups can be still be detected in a data-driven way. Either a finite mixture model (McLachlan and Peel, 2000)

can be employed, possibly using age as a concomitant variable (Dayton and Macready, 1988) to model smooth transitions between clusters. Alternatively, model-based recursive partitioning (Zeileis *et al.*, 2008) can capture the difference by one or more splits in the partitioning variable age, yielding a tree structure (similar to classification and regression trees, Breiman, Friedman, Olshen, and Stone, 1984) where each leaf is associated with a parametric model.

Given their shared goal of establishing subgroups to capture parameter instabilities across subgroups, how do these mixture models and model-based trees compare? A unifying framework for both methods is presented and their relative (dis)advantages for (a) detecting the presence of clusters with different parameters and (b) recovering the underlying grouping structure are assessed in a simulation study.

4.2 Theory

Although both mixture models and model-based trees can be applied to general parametric models estimated by means of the maximum likelihood (ML) principle, we focus on linear regression here because it is the most simple and commonly applied model:

$$y_i = x_i^\top \beta + \varepsilon_i \tag{4.1}$$

with response y_i, regressor vector x_i, and errors ε_i for observations $i = 1, \ldots, n$. The unknown vector of regression coefficients β can be estimated by least squares, which yields the same estimates as ML estimation under the assumption of independent normal errors with variance σ^2. In the latter case the log-likelihood is given by $\sum_{i=1}^n \log \phi(y_i; x_i^\top \beta, \sigma^2)$ where $\phi(\cdot)$ denotes the probability density function of the normal distribution.

Within this framework, both trees and mixture models can assess whether the same coefficient vector β holds for all n observations and, if parameter stability is violated, simultaneously find clusters/subgroups and estimate the associated cluster-specific coefficients. Further covariates z_i can be used as concomitant or partitioning variables, respectively, to establish these clusters.

4.2.1 Finite mixture models

Finite mixture models assume that the data stem from K different subgroups with unknown subgroup membership and subgroup-specific parameters $\beta_{(k)}$ and $\sigma_{(k)}$ $(k = 1, \ldots, K)$. The full mixture model is a weighted sum over these separate models (or components):

$$f(y_i; x_i, z_i, \beta_{(1)}, \sigma_{(1)}, \ldots, \beta_{(K)}, \sigma_{(K)}) = \sum_{k=1}^{K} \pi_k(z_i) \cdot \phi(y_i; x_i^\top \beta_{(k)}, \sigma^2_{(k)}). \tag{4.2}$$

The component weights may depend on the additional covariates z_i through a concomitant variable model (Dayton and Macready, 1988), typically a multinomial logit model

$$\pi_k(z_i) = \frac{\exp(z_i^\top \alpha_{(k)})}{\sum_{g=1}^{K} \exp(z_i^\top \alpha_{(g)})} \tag{4.3}$$

with component-specific coefficients $\alpha_{(k)}$. For identifiability, one group (typically the first) is used as a reference group and the coefficients of this group are set to zero: $\alpha_{(1)} = 0$. This also includes the special case without concomitant variables, where $z_i = 1$ is just an intercept yielding component-specific weights $\pi_k(z_i) = \pi_{(k)}$.

Given the number of subgroups K, all parameters in the mixture model are typically estimated simultaneously by ML using the expectation-maximization (EM) algorithm. To choose the number of subgroups K, the mixture model is typically fitted for $K = 1, 2, \ldots$ and then the best-fitting model is selected by some information criterion. Here, we employ the Bayesian Information Criterion (BIC).

4.2.2 Model-based recursive partitioning

Model-based recursive partitioning (Zeileis *et al.*, 2008) can also detect subgroups for which different model parameters hold. These subgroups are separated by sample splits in the covariates z_i used for partitioning. The algorithm performs the following steps:

1. Estimate the model parameters in the current subgroup.
2. Test parameter stability along each partitioning variable z_{ij}.
3. If any instability is found, split the sample along the variable z_{ij*} with the highest instability. Choose the breakpoint with the highest improvement in model fit.
4. Repeat 2–4 on the resulting subsamples until no further instability is found.

Here, we only briefly outline how these parameter instability tests work and refer to Zeileis and Hornik (2007) for the theoretical details. The basic idea is that the scores, i.e., the derivative of $\log \phi(\cdot)$ with respect to the parameters, evaluated at the estimated coefficients behave similar to least squares residuals: They sum to zero and if the model fits well they should fluctuate randomly around zero. However, if the parameters change along one of the partitioning variables z_{ij}, there should be systematic departures from zero. Such departures *along* a covariate can be captured by a cumulative sum of the scores (ordered by the covariate) and aggregated to a test statistic, e.g., summing the absolute or squared cumulative deviations. Summing the scores along a categorical partitioning variable then leads to a statistic that has an asymptotic χ^2 distribution while aggregation along numeric partitioning variables can be done in a way that yields a maximally-selected score (or Lagrange multiplier) test. In either case, the p-value p_j can be obtained for each ordering along z_{ij} without having to re-estimate the model. The p-values are then Bonferroni-adjusted to account for testing along multiple orderings

and partitioning continues until there is no further significant instability (here at the 5% level).

Since each split can be expressed through an indicator function $I(\cdot)$ (for going left or right), each branch of the tree can be represented as a product of such indicator functions. Therefore, the model-based tree induced by recursive partitioning is in fact also a model of type (4.2), albeit with rather different weights:

$$\pi_k(z_i) = \prod_{j=1}^{J_k} I(s_{(j|k)} \cdot z_{i(j|k)} > b_{(j|k)}) \tag{4.4}$$

where $z_{(j|k)}$ denotes the j-th partitioning variable for terminal node k, $b_{(j|k)}$ is the associated breakpoint, $s_{(j|k)} \in \{-1, 1\}$ the sign (signaling splitting to the left or right), and J_k the number of splits leading up to node k.

4.2.3 Differences and similarities

While both methods are based on the same linear regression model and aim at detecting subgroups with stable parameters, certain differences arise:

- Because K is fixed for each mixture model, it is based on model selection via an information criterion whereas the selection of K through a tree is based on significance tests.
- Covariates z are optional for mixture models and latent subgroups can be estimated. For a tree, those covariates are required. Furthermore, if no covariates associated with the subgroups are available, the groups cannot be detected.
- The concomitant model (4.3) assumes a smooth, monotonic transition between subgroups. The sample splits of a tree (4.4) represent abrupt shifts, multiple splits in a covariate are able to represent a non-monotonic transition. While variable selection is inherent to trees, it requires an additional step for mixtures models.
- Trees yield a hard clustering and mixtures a probabilistic clustering of the observations.

To investigate how the aforementioned differences between the two methods affect their ability to detect parameter instability, a simulation study is conducted, which is described in the next section.

4.3 Simulation study

To determine how well mixture models and model-based trees detect parameter instability, two basic questions are asked. First, is any instability found at all? Second, if

so, are the correct subgroups recovered? These two aspects are potentially influenced by several factors: How does the relationship between the response y and the regressors x differ between the subgroups and how strongly does it differ? If there are any additional covariates z available, how and how strongly are those covariates connected to the subgroups? In general, we expect the following:

- Given the covariates z are associated strongly enough with the subgroups, trees are able to detect smaller differences in $\beta_{(k)}$ than mixtures because they employ a significance test for each parameter rather than an information criterion for full sets of parameters. In contrast, mixtures are more suitable to detect subgroups if they are only loosely associated with the covariates z, as long as the differences in $\beta_{(k)}$ are strong enough.
- If the association between covariates and subgroups is smooth and monotonic, mixtures are more suitable to detect the subgroups whereas trees are more suitable if the association is characterized by abrupt shifts and possibly non-monotonic.
- If several covariates determine the subgroups simultaneously, the mixture is more suitable, whereas trees are more suitable if z includes several noise variables unconnected to the subgroups.

Motivated by these considerations, the simulation design is explained in the next section.

4.3.1 Simulation design

A single regressor x and four additional covariates $z_1, \ldots, z_4$ are drawn from a uniform distribution on $[-1, 1]$. The response y is computed with errors drawn from a standard normal distribution. Two subgroups of equal size are simulated. How they differ is governed by the form of β – either in their *intercept*, *slope*, or *both* – and the magnitude of their differences is governed by the simulation parameter κ (Table 4.1). How the covariates are connected to the subgroups is governed by the form of $\pi_k(z)$ – either via a logistic or a step function – and the strength of this association is governed by simulation parameter ν. Here, three scenarios for $\pi_k(z)$ are considered: a smooth logistic transition along z_1, a smooth logistic transition along z_1 and z_2 simultaneously, and a sharp transition along z_1 with two breakpoints (labeled *axis1*, *diagonal*, and *double step*, respectively). The corresponding parameter vector of the logistic function and the breakpoints can be found in Table 4.1. The simulation parameters cover the following ranges: $\kappa \in \{0, 0.05, \ldots, 1\}$ and $\nu \in \{-1, -0.5, \ldots, 2\}$. Note that $\beta_{(1)}$ and $\beta_{(2)}$ are identical if $\kappa = 0$ and thus only one subgroup is simulated. Each coefficient scenario is combined with every covariate scenario and the sample size $n \in \{200, 500, 1000\}$ is varied. The covariates z_3 and z_4 are always noise variables and thus either included or excluded in z. For each of these conditions, 500 datasets are drawn and three methods applied: a model-based tree, a plain mixture, and a mixture with concomitant variables. Both mixtures are fitted with $K = \{1, \ldots, 4\}$ and $\hat{K}$ selected via BIC. For all computations, the R system for statistical computing (R Core Team, 2013) is used along with the add-on packages **partykit** (Hothorn and Zeileis, 2014) and **flexmix** (Grün and Leisch, 2008).

	Label	Details
Coefficients	intercept	$\beta_{(1)} = (\kappa, 0)^\top \quad \beta_{(2)} = (-\kappa, 0)^\top$
	slope	$\beta_{(1)} = (0, \kappa)^\top \quad \beta_{(2)} = (0, -\kappa)^\top$
	both	$\beta_{(1)} = (\kappa, -\kappa)^\top \quad \beta_{(2)} = (-\kappa, \kappa)^\top$
Covariates	axis1	logistic with $\alpha_{(2)} = (\exp(\nu), 0, 0, 0)^\top$
	diagonal	logistic with $\alpha_{(2)} = (\exp(\nu), -\exp(\nu), 0, 0)^\top$
	double step	tree with $\pi_2(z) = I(z_1 > -0.5)I(z_1 < 0.5)$

Table 4.1: Simulation scenarios for regression coefficients and covariates.

4.3.2 Outcome assessment

To address the first question of whether or not any instability is found, the *hit rate* is computed: This is the rate of selecting more than one subgroup – this corresponds to splitting at least once for a tree and to selecting $\hat{K} > 1$ for a mixture. To address the second question if the right subgroups are found, the estimated clustering is compared to the true clustering. Many external cluster indices such as the Rand index favor a "perfect match", i.e., splitting one true subgroup into several estimated subgroups (as might be unavoidable in a tree) is penalized by the index. Cramér's coefficient is invariant against such departures from a perfect match (Mirkin, 2001) and thus employed here.

4.3.3 Simulation results

Exemplary results are shown for the scenario with differences in *both* coefficients with $n = 200$ observations, without the noise variables z_3 and z_4, and the *double step* scenario as well as the logistic scenarios *axis1* and *diagonal* with three levels $\nu = \{-1, 0, 1\}$ of separation between subgroups. For the *double step* scenario, the hit rate for detecting instability is depicted in the left panel of Figure 4.1. The tree clearly outperforms both mixtures. For the logistic scenarios *axis1* and *diagonal*, the hit rates are depicted in Figure 4.2. If the covariates are only weakly associated with the subgroups ($\nu = -1$, left column), the tree is unable to detect the subgroups, regardless of how strongly they differ in their regression coefficients. Both mixtures are able to detect instability beyond a threshold of $\kappa = 0.7$ and reach hit rates of almost 1. For a medium association ($\nu = 0$), the tree is able to detect smaller differences in the regression coefficients than the mixtures but for larger differences both mixtures equally outperform the tree. If the association is strong ($\nu = 1$), the tree outperforms both mixtures. The mixture with concomitants in turn outperforms the plain mixture which is per definition invariant to (changes in) the association between covariates and subgroups. Interestingly, the tree performs rather similarly for the scenarios *axis1* and *diagonal*, indicating that approximation through sequential splits works rather well. For $\kappa = 0$ only one true subgroup exists and mixtures nearly always select $\hat{K} = 1$ while trees incorrectly select $\hat{K} > 1$ subgroups only in less than 5% of the cases (which is the significance level employed in the parameter stability tests).

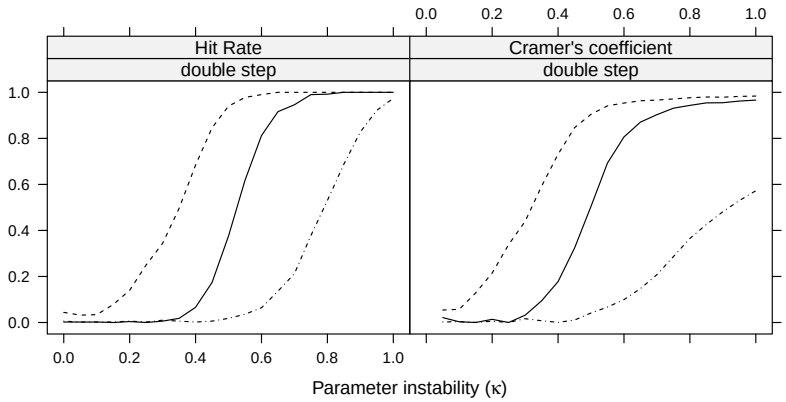

Figure 4.1: Hit Rate (left panel) and Cramér's coefficient (right panel) for the *double step* covariate scenario. Line type: dashed for tree, solid for mixture, dot-dashed for plain mixture.

The corresponding recovery of the subgroups as measured by Cramér's coefficient is depicted in the right panel of Figure 4.1 for the *double step* scenario. Similar to the detection of instability, the tree outperforms both mixtures. For the logistic scenarios, the Cramér's coefficients are shown in Figure 4.3. For low and medium levels of association between covariates and subgroups, both mixtures outperform the tree for stronger instabilities. For a medium level of association ($\nu = 0$) and small instabilities, the tree's advantage in detecting instabilities translates into an advantage of also uncovering the correct subgroups. However for a stronger association ($\nu = 1$), the mixture with concomitants recovers the true subgroups better than the other two methods once the hit rates are similar across methods. Despite its good hit rates, the tree never exceeds a Cramér's coefficient of about 0.6. This is the case regardless of how strong the regression coefficients differ, indicating that the tree's ability to uncover the correct subgroups is limited by the (relative) weakness of association between covariates and subgroups. For an even stronger association ($\nu > 1$, not depicted here), the tree recovers the subgroups as well as the concomitant mixture in the *axis1* scenario but fails to do so in the *diagonal* scenario.

For larger numbers of observations ($n = 500$ or 1000) and the other two coefficients scenarios (*intercept* and *slope*), results are similar to those shown here, just being generally more pronounced. Including two additional noise variables z_3 and z_4 affected both the tree and the concomitant mixture, with hit rates dropping slightly stronger for the mixture.

4.4 Discussion

Both methods are suitable to detect parameter instability (or lack thereof) and recover the subgroups (if any). Which method is more suitable depends largely on the association between the subgroups and covariates as well as how strongly the subgroups differ in

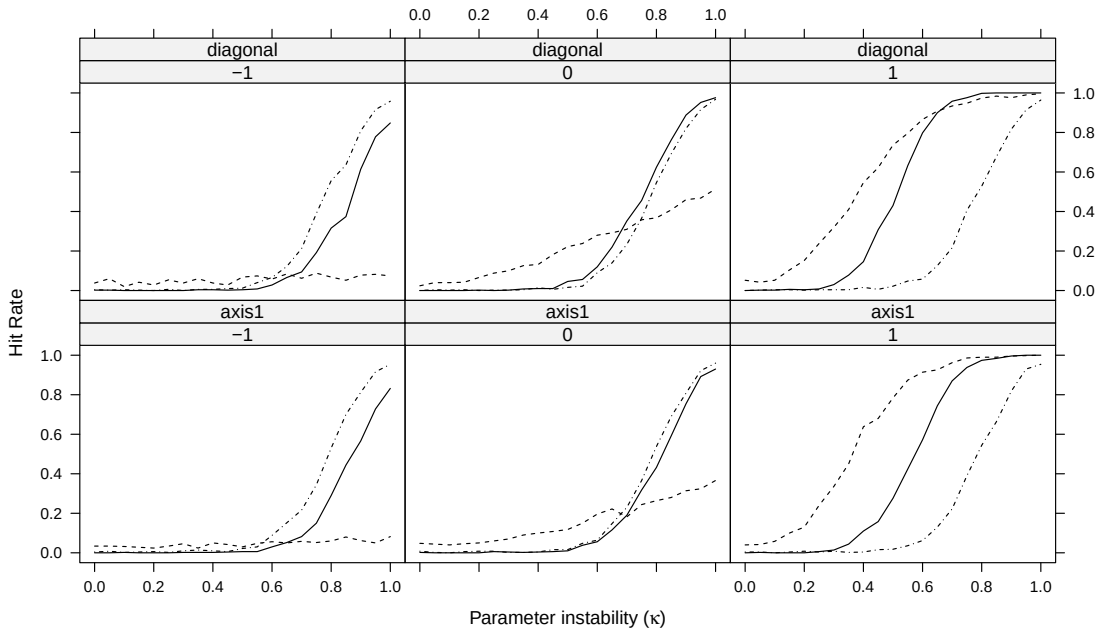

Figure 4.2: Hit Rate for the logistic covariate scenarios for three levels of $\nu \in \{-1, 0, 1\}$. Line type: dashed for tree, solid for mixture, dot-dashed for plain mixture.

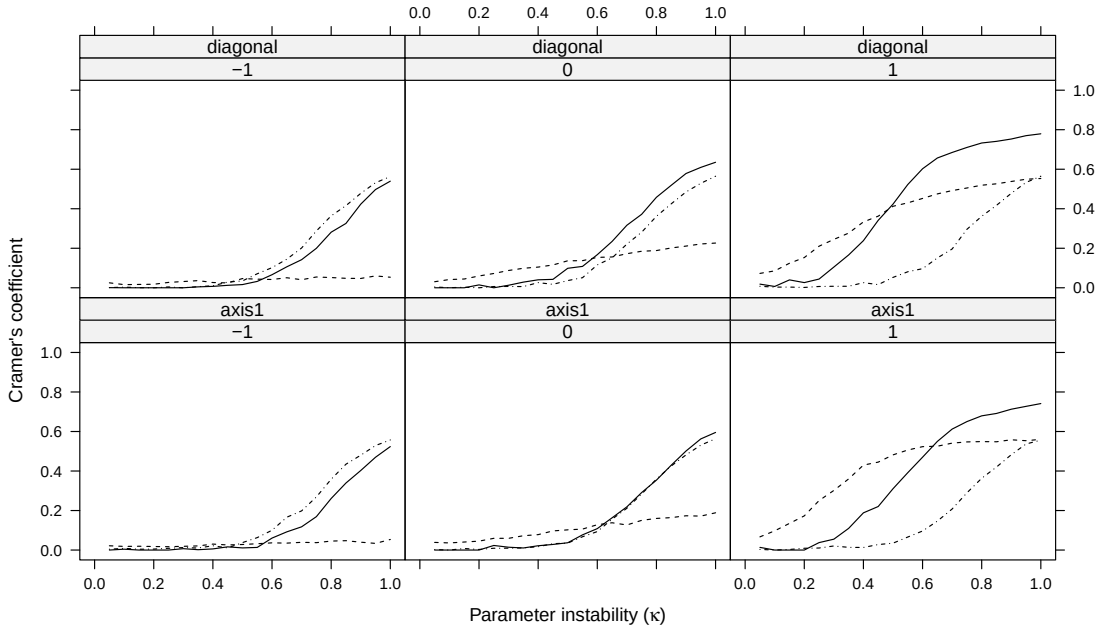

Figure 4.3: Cramér's coefficient for the logistic covariate scenarios for three levels of $\nu \in \{-1, 0, 1\}$. Line type: dashed for tree, solid for mixture, dot-dashed for plain mixture.

their respective parameter vectors. If the association between subgroups and covariates is strong, the tree is able to detect smaller differences in the parameters than the mixtures. The approximation of a smooth transition between classes through sample splits works rather well. In addition, the tree can represent a non-monotonic association which the mixtures cannot. If the association between subgroups and covariates is weak but the difference in the parameters reasonably strong, mixture models are more suitable than the tree. Mixture models are also capable of detecting latent subgroups without any association to covariates. It would be interesting to investigate whether these relationships also apply to situations with more subgroups and mixtures with higher numbers of components. Further questions for further research include the assessment of subgroup recovery on a test set rather than in-sample and variable selection for the concomitant models of mixtures, which could also be accomplished with an information criterion.

In summary, both methods have their relative advantages and thus are more suitable to detect parameter instability and uncover subgroups in different situations. As the exact structure is unknown in practice, we suggest using both methods to gain better insight into the data.

Chapter 5

Summary and Outlook

Mixture models are a flexible statistical tool, which can be applied to data stemming from different groups when group membership is unknown. This tool is here applied in a psychometric context to Rasch models. The R package **psychomix** provides a flexible open-source implementation of various extensions of the original model specification by Rost (1990), e.g., allowing for different score distributions and the specification of concomitant variable models (Dayton and Macready, 1988). This package extends the previously existing opportunities for fitting mixtures of Rasch models in R.

The Rasch model is used to measure latent traits by modeling a subject's probability of solving an item as dependent on the subject's ability and the item's difficulty. Accurate measurements and fair comparisons between subjects are only possible if each item is equally difficult for every subject, i.e., no groups of subjects exist for which different sets of item difficulties hold. Rasch mixture models can be used to assess this crucial assumption of measurement invariance. However, previously suggested specifications of the model are not invariant to the distribution of the scores, which are the number of correctly scored items. Thus, the latent groups found by the Rasch mixture model cannot only be based on the item difficulties but also – or even solely – on the subjects' abilities. While the first options are a violation of measurement invariance, the latter is not and thus not of interest in this context. The newly suggested specification of the Rasch mixture model restricts the score distribution in such a way that the latent groups found by the model are *only* based on the item difficulties. It is thus better suited to detect violations of measurement invariance, as illustrated in the simulation study comparing old and new score specifications.

Another method to assess measurement invariance by establishing groups based on stable item difficulties in a data-driven way are Rasch trees (Strobl *et al.*, 2014). These are based on model-based recursive partitioning which shares certain properties with mixture models. Here, these two general statistical methods are unified in a common framework and relative (dis-)advantages for the detection of parameter instability illustrated in a simulation study. Neither method outperforms the other in all scenarios. Therefore, it is suggested to employ both methods in an empirical analysis to overall gain a better insight into the data.

Since this thesis features aspects of statistics, psychometrics, and the corresponding statistical and psychometric computing, the outlook for future work also branches out in the various directions. Apart from mixtures of Rasch models, the **psychomix** package also includes mixtures of Bradley-Terry models (Bradley and Terry, 1952) and is intended to be extended to mixtures of further psychometric models such as the rating scale model (Andrich, 1978) and the partial credit model (Masters, 1982).

The comparison of mixture models and model-based recursive partitioning was done for the linear model as the group-specific model. An extension to a psychometric model as the group-specific model is an option to further strengthen the exchange between statistics and psychometrics by assessing how different methods for checking psychometric model assumptions compare. The Rasch model and the Bradley-Terry model could serve as a starting point. The corresponding implementations can be found in the R packages **psychotree** (Strobl, Wickelmaier, and Zeileis, 2011b) and **psychomix**.

The framework of the comparison introduced in Chapter 4 should in future work be extended to situations with more than two subgroups in the data. Also of interest are more complex examples of parameter instability, e.g., including multiple regressors for the linear model. The assessment of group recovery can be based on a test dataset rather than on the dataset which was also used to fit the model. Variable selection for the concomitant models of mixtures could be accomplished via an information criterion or a likelihood ratio test and potentially included in the simulation design.

A further extension of mixture models allows for a mix of constant and varying parameters in the component-specific model. The varying parameters differ for groups of observations whereas the constant parameters are fixed for the entire sample. In an application to Rasch models, this type of mixture model could be used to estimate constant parameters for DIF-free items and varying parameters for items with DIF. This type of model is similar to mixed-effects models (Pinheiro and Bates, 2000), however, here the distribution of the varying parameters is unknown and needs to be estimated. Grün and Leisch (2008) thus see this type of model more closely related to the varying-coefficients modeling framework (Hastie and Tibshirani, 1993). From the statistical viewpoint, a closer look towards both model types is warranted since they also deal with heterogeneity in model parameters as do mixture models and model-based recursive partitioning.

Bibliography

Ackerman TA (1992). "A Didactic Explanation of Item Bias, Item Impact, and Item Validity from a Multidimensional Perspective." *Journal of Educational Measurement*, **29**(1), 67–91.

Akaike H (1973). "Information Theory and an Extension of the Maximum Likelihood Principle." In *Proceedings of the Second International Symposium on Information Theory*, pp. 267–281.

Andersen EB (1972). "A Goodness of Fit Test for the Rasch Model." *Psychometrika*, **38**(1), 123–140.

Andrich D (1978). "A Rating Formulation for Ordered Response Categories." *Psychometrika*, **43**(4), 561–573.

Ankenmann RD, Witt EA, Dunbar SB (1999). "An Investigation of the Power of the Likelihood Ratio Goodness-of-Fit Statistic in Detecting Differential Item Functioning." *Journal of Educational Measurement*, **36**(4), 277–300.

Baghaei P, Carstensen CH (2013). "Fitting the Mixed Rasch Model to a Reading Comprehension Test: Identifying Reader Types." *Practical Assessment, Research & Evaluation*, **18**(5), 1–13.

Bradley RA, Terry ME (1952). "Rank Analysis of Incomplete Block Designs. I. The Method of Paired Comparisons." *Biometrika*, **39**(3/4), 324–345.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). *Classification and Regression Trees*. Wadsworth, California.

Cohen AS, Bolt DM (2005). "A Mixture Model Analysis of Differential Item Functioning." *Journal of Educational Measurement*, **42**(2), 133–148.

Dayton CM, Macready G (1988). "Concomitant-Variable Latent-Class Models." *Journal of the American Statistical Association*, **83**(401), 173–178.

De Boeck P, Wilson M (eds.) (2004). *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*. Springer-Verlag, New York.

DeMars CE (2010). "Type I Error Inflation for Detecting DIF in the Presence of Impact." *Educational and Psychological Measurement*, **70**(6), 961–972.

DeMars CE, Lau A (2011). "Differential Item Functioning Detection With Latent Classes: How Accurately Can We Detect Who Is Responding Differentially?" *Educational and Psychological Measurement*, **71**(4), 597–616.

Dempster A, Laird N, Rubin D (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B*, **39**(1), 1–38.

Fischer GH (1995). "Derivations of the Rasch Model." In Fischer and Molenaar (1995), chapter 2, pp. 15–38.

Fischer GH, Molenaar IW (eds.) (1995). *Rasch Models: Foundations, Recent Developments, and Applications*. Springer-Verlag, New York.

Frick H, Strobl C, Leisch F, Zeileis A (2012). "Flexible Rasch Mixture Models with Package **psychomix**." *Journal of Statistical Software*, **48**(7), 1–25. URL http://www.jstatsoft.org/v48/i07/.

Frick H, Strobl C, Zeileis A (2014a). "Rasch Mixture Models for DIF Detection: A Comparison of Old and New Score Specifications." *Educational and Psychological Measurement*. doi:10.1177/0013164414536183. Forthcoming.

Frick H, Strobl C, Zeileis A (2014b). "To Split or to Mix? Tree vs. Mixture Models for Detecting Subgroups." In M Gilli, G González-Rodríguez, A Nieto-Reyes (eds.), *COMPSTAT 2014 – Proceedings in Computational Statistics*, pp. 379–386. The International Statistical Institute/International Association for Statistical Computing. ISBN 978-2-8399-1347-8.

Glas CAW, Verhelst ND (1995). "Testing the Rasch Model." In Fischer and Molenaar (1995), chapter 5, pp. 69–95.

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. URL http://www.jstatsoft.org/v28/i04/.

Gustafsson JE (1980). "Testing and Obtaining Fit of Data in the Rasch Model." *British Journal of Mathematical and Statistical Psychology*, **33**(2), 205–233.

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

Holland PW, Thayer DT (1988). "Differential Item Performance and the Mantel-Haenszel procedure." In Wainer and Braun (1988), chapter 9, pp. 129–145.

Hong S, Min SY (2007). "Mixed Rasch Modeling of the Self-Rating Depression Scale: Incorporating Latent Class and Rasch Rating Scale Models." *Educational and Psychological Measurement*, **67**(2), 280–299.

Hothorn T, Zeileis A (2014). "partykit: A Modular Toolkit for Recursive Partytioning in R." *Working Paper 2014-10*, Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck. URL http://EconPapers.RePEc.org/RePEc:inn:wpaper:2014-10.

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. URL http://www.jstatsoft.org/v11/i08/.

Li F, Cohen AS, Kim SH, Cho SJ (2009). "Model Selection Methods for Mixture Dichotomous IRT Models." *Applied Psychological Measurement*, **33**(5), 353–373.

Li Y, Brooks GP, Johanson GA (2012). "Item Discrimination and Type I Error in the Detection of Differential Item Functioning." *Educational and Psychological Measurement*, **72**(5), 847–861.

Maij-de Meij AM, Kelderman H, van der Flier H (2010). "Improvement in Detection of Differential Item Functioning Using a Mixture Item Response Theory Model." *Multivariate Behavioral Research*, **45**(6), 975–999.

Mair P, Hatzinger R (2007). "Extended Rasch Modeling: The **eRm** Package for the Application of IRT Models in R." *Journal of Statistical Software*, **20**(9), 1–20. URL http://www.jstatsoft.org/v20/i09/.

Masters GN (1982). "A Rasch Model for Partial Credit Scoring." *Psychometrika*, **47**(2), 149–174.

McLachlan G, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons, New York.

Mirkin B (2001). "Eleven Ways to Look at Chi-squared Coefficients for Contingency Tables." *The American Statistician*, **55**(2), 111–120.

Molenaar IW (1995a). "Estimation of Item Parameters." In Fischer and Molenaar (1995), chapter 3, pp. 39–51.

Molenaar IW (1995b). "Some Background for Item Response Theory and the Rasch Model." In Fischer and Molenaar (1995), chapter 1, pp. 3–14.

Nieweglowski L (2013). *clv: Cluster Validation Techniques*. R package version 0.3-2.1, URL http://CRAN.R-project.org/package=clv.

Pearson K (1894). "Contributions to the Theory of Mathematical Evolution." *Philosophical Transactions of the Royal Society of London A*, **185**, 71–110.

Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in S and S-PLUS*. Springer-Verlag, New York.

Preinerstorfer D, Formann AK (2011). "Parameter Recovery and Model Selection in Mixed Rasch Models." *British Journal of Mathematical and Statistical Psychology*, **65**(2), 251–262.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

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

Rand WM (1971). "Objective Criteria for the Evaluation of Clustering Methods." *Journal of the American Statistical Association*, **66**(336), 846–850.

Rasch G (1960). *Probabilistic Models for Some Intelligence and Attainment Tests*. The University of Chicago Press.

R Development Core Team (2011). *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/.

Rijmen F, De Boeck P (2005). "A Relationship Between a Between-Item Multidimensional IRT Model and the Mixture Rasch Model." *Psychometrika*, **70**(3), 481–496.

Rizopoulos D (2006). "**ltm**: An R Package for Latent Variable Modeling and Item Response Theory Analyses." *Journal of Statistical Software*, **17**(5), 1–25. URL http://www.jstatsoft.org/v17/i05/.

Rost J (1990). "Rasch Models in Latent Classes: An Integration of Two Approaches to Item Analysis." *Applied Psychological Measurement*, **14**(3), 271–282.

Rost J (1991). "A Logistic Mixture Distribution Model for Polychotomous Item Responses." *British Journal of Mathematical and Statistical Psychology*, **44**(1), 75–92.

Rost J, von Davier M (1995). "Mixture Distribution Rasch Models." In Fischer and Molenaar (1995), chapter 14, pp. 257–268.

Schwarz G (1978). "Estimating the Dimension of a Model." *Annals of Statistics*, **6**(2), 461–464.

Smits DJM, De Boeck P, Vansteelandt K (2004). "The Inhibition of Verbally Aggressive Behaviour." *European Journal of Personality*, **18**(7), 537–555.

Strobl C, Kopf J, Zeileis A (2011a). "A New Method for Detecting Differential Item Functioning in the Rasch Model." *Working Paper 2011-01*, Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck. URL http://EconPapers.RePEc.org/RePEc:inn:wpaper:2011-01.

Strobl C, Kopf J, Zeileis A (2014). "A New Method for Detecting Differential Item Functioning in the Rasch Model." *Psychometrika*. doi:10.1007/s11336-013-9388-3. Forthcoming.

Strobl C, Wickelmaier F, Zeileis A (2011b). "Accounting for Individual Differences in Bradley-Terry Models by Means of Recursive Partitioning." *Journal of Educational and Behavioral Statistics*, **36**(2), 135–153. doi:10.3102/1076998609359791.

Tay L, Newman DA, Vermunt JK (2011). "Using Mixed-Measurement Item Response Theory with Covariates (MM-IRT-C) to Ascertain Observed and Unobserved Measurement Equivalence." *Organizational Research Methods*, **14**(1), 147–176.

Van den Noortgate W, De Boeck P (2005). "Assessing and Explaining Differential Item Functioning Using Logistic Mixed Models." *Journal of Educational and Behavioral Statistics*, **30**(4), 443–464.

von Davier M (2000). *Winmira 2001 – A Microsoft Windows program for analyses with the Rasch model, with the latent class analysis and with the mixed Rasch model [Computer Software]*. Available for download from http://winmira.von-davier.de.

von Davier M, Rost J (1995). "Polytomous Mixed Rasch Models." In Fischer and Molenaar (1995), chapter 20, pp. 371–379.

Wainer H, Braun HI (eds.) (1988). *Test Validity*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Weldon WFR (1893). "On Certain Correlated Variations in Carcinus Moenas." In *Proceedings of the Royal Society of London*, volume 54, pp. 318–329.

Willse JT (2011). "Mixture Rasch Models with Joint Maximum Likelihood Estimation." *Educational and Psychological Measurement*, **71**(1), 5–19.

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

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

Zeileis A, Strobl C, Wickelmaier F (2011). ***psychotools**: Infrastructure for Psychometric Modeling*. R package version 0.1-1, URL http://CRAN.R-project.org/package=psychotools.

Zickar MJ, Gibby RE, Robie C (2004). "Uncovering Faking Samples in Applicant, Incumbent, and Experimental Data Sets: An Application of Mixed-Model Item Response Theory." *Organizational Research Methods*, **7**(2), 168–190.

Appendix A

Using the FLXMCrasch() driver directly with stepFlexmix()

The "FLXM" driver FLXMCrasch(), underlying the raschmix() function, can also be used directly with flexmix() or stepFlexmix(), respectively, via the model argument. To do so, essentially the same arguments as in raschmix() (see Section ??) can be used, however, they need to be arranged somewhat differently. The formula just specifies the item responses, while the concomitant variables need to be passed to the concomitant argument in a suitable "FLXP" driver (see Grün and Leisch, 2008). Also some arguments, such as the scores distribution have to be specified in the FLXMCrasch() driver for the model argument. As an example, consider replication of the cm2 model fit on the artificial data from Section ??:

```
R> set.seed(4)
R> fcm2 <- stepFlexmix(resp ~ 1, data = d, k = 1:3,
+    model = FLXMCrasch(scores = "meanvar"),
+    concomitant = FLXPmultinom(~ x1 + x2))
```

Thus, there is some more flexibility but somewhat less convenience when using flexmix() or stepFlexmix() directly as opposed through the raschmix() interface. This is also reflected in the objects returned which are of class "flexmix" or "stepFlexmix", not class "raschmix" or "stepRaschmix", respectively. Thus, only the generic functions for those objects apply and not the additional ones specific to Rasch mixture models. In various cases, the methods are inherited or reused from **flexmix** and thus behave identically, e.g., for BIC() or getModel().

```
R> rbind(cm2 = BIC(cm2), fcm2 = BIC(fcm2))
```

```
            1       2       3
cm2  21055.12 17776.3 17867.25
fcm2 21055.12 17776.3 17867.25
```

```
R> fcm2b <- getModel(fcm2, which = "BIC")
```

For other methods, such as `parameters()`, the methods in **psychomix** offer more convenience. For example, the concomitant model coefficients can be accessed in the same way

```
R> cbind(parameters(cm2b, which = "concomitant"),
+    parameters(fcm2b, which = "concomitant"))
```

```
            1           2 1           2
(Intercept) 0  0.45759154 0  0.45759154
x1          0 -0.91231698 0 -0.91231698
x2          0  0.02909326 0  0.02909326
```

while the item and score parameters cannot be accessed separately (as it is possible for "`raschmix`" objects). They can only be accessed jointly as the `"model"` parameters. The method for "`raschmix`" objects also excludes non-identified or aliased parameters, e.g., the parameter for the anchor item.

```
R> parameters(fcm2b, which = "model")
```

```
                Comp.1     Comp.2
item.Item01  0.0000000  0.0000000
item.Item02  0.4458307 -0.5087512
item.Item03  0.8794596 -1.2541765
item.Item04  1.5175890 -1.7793843
item.Item05  2.3308845 -2.3289326
item.Item06  2.8255465 -2.9351253
item.Item07  3.5001225 -3.5059792
item.Item08  4.0926599 -4.0813289
item.Item09  4.6158654 -4.6979718
item.Item10  5.2655073 -5.2701331
score1       0.1008315  0.1638214
score2      -0.2485943 -0.1711337
```

To sum up, some (convenience) functionality which is specific for Rasch mixture models is only available for objects of class "`raschmix`", e.g., the score probabilities via `scoreProbs()` or the item profiles via the default `plot()` method among others. On the other hand, functionality which is applicable to mixture models in general is inherited or preserved as part of the additional methods.

Own Contributions

The authors of the three publications contained in Chapters 2, 3, and 4 have contributed in the following way:

- For Frick *et al.* (2012), all authors developed the general idea. I wrote the software and the manuscript. Achim Zeileis and Friedrich Leisch contributed to the code and turning it into an R package. Achim Zeileis and Carolin Strobl contributed to structuring and writing the manuscript.
- For Frick *et al.* (2014a), Carolin Strobl, Achim Zeileis, and myself developed the general idea. I wrote the software, conducted the simulation study, and wrote the manuscript. Achim Zeileis and Carolin Strobl contributed to the design of the simulation study as well as to structuring and writing the manuscript.
- For Frick *et al.* (2014b), Carolin Strobl, Achim Zeileis, and myself developed the general idea. I conducted the simulation study, and wrote the manuscript. Achim Zeileis and Carolin Strobl contributed to the design of the simulation study as well as to structuring and writing the manuscript.

Leopold-Franzens-Universität Innsbruck

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Dissertation eingereicht.

Datum	Unterschrift