Alma Mater Studiorum
Università di Bologna

DOTTORATO DI RICERCA IN
SCIENZE STATISTICHE

CICLO XXX

Settore Concorsuale: 13/D1
Settore Scientifico Disciplinare: SECS-S/01

Solution Path Clustering for Fixed-Effects Models in a Latent Variable Context

Presentata da: Francesco Giovinazzi

Coordinatore Dottorato:
Prof.ssa Alessandra Luati

Supervisore:
Prof.ssa Silvia Cagnone
Co-supervisore:
Prof. Francesco Bartolucci

Esame finale anno 2018

Abstract

The main drawback of estimating latent variable models with fixed effects is the direct dependence between the number of free parameters and the number of observations. We propose to apply a well suited penalization technique in order to regularize the parameter estimates. In particular, we promote sparsity based on the pairwise differences of subject-specific parameters, inducing the latter to shrink on each other. This method allows to group statistical units into clusters that are homogeneous with respect to a latent attribute, without the need to specify any distributional assumption, and without adopting random effects. In practice, applying the proposed penalization, the number of free parameters is reduced and the adopted model becomes more parsimonious. The estimation of the fixed effects is based on an algorithm that builds a solution path, in the form of a hierarchical aggregation tree, whose outcome depends on a single tuning parameter. The method is intended to be general, and in principle it can be applied on the likelihood of any latent variable model with fixed effects. We describe in detail its application to the Rasch model, for which we provide a real data example and a simulation study. We then extend the method to the case of a latent variable model for continuous data, where the number of fixed effects to be estimated is higher.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the era of big data statisticians are faced with the challenging task to develop more and more flexible and efficient methods in order to handle the increasing complexity of data structures. «We are drowning in information and starving for knowledge»is the motto cited by Hastie *et al.* (2015) in their inspiring book on statistical learning. Thus, a modern data scientist has the impelling need to dive into this huge mass of information, reducing it to its bare essential. In particular, when building a statistical model, one should always follow the basic principle for which «less is more», enhancing parsimony and promoting simplicity over complexity. A problem arises since simplicity is a general, not univocal concept, that may be attained and interpreted in many different ways. We refer specifically to the following interpretations:

- We interpret simplicity as a result of dimension reduction every time we try to synthesize high-dimensional data extracting a number of informative and non-redundant features. For example, the derivation of these features can be based on a direct projection of the data onto a new space with fewer dimensions, as in Principal Component Analysis (PCA), or it can be the result of a theoretical set of assumptions, as when we deal with a latent variable model (LVM);

- We interpret simplicity as sparsity every time we are interested in reducing the number of non-zero parameters in a model, for example in order to identify a smaller subset of important variables in a regression problem. Sparsity can be attained applying a penalization in the estimation process, and many regularization techniques have been developed to produce the so-called shrinkage effect.

The idea beyond this work is to enhance the first form of simplicity using the properties of the second one. In other words, we propose to overcome some limitations of latent variable modeling, in those cases when the number of parameters directly grows with the dimension of the data, by the means of a well designed sparsification strategy.

In latent variable models we assume that many observed variables are realizations of fewer unobservable ones, synthesizing a certain proportion of the available information (Bartholomew *et al.*, 2011). Latent variables can be defined as variables which are not susceptible of direct measurement, but that in some way affect a set of observed responses. Apart from dimension reduction, there are many reasons for which we may want to include latent variables in a statistical model, typically:

- Represent individual characteristics that cannot be directly measured, such as intelligence, satisfaction or ability;
- Account for measurement errors, where the latent variables represent the true outcomes of which the responses represent a disturbed version;
- Represent the effect of unobservable covariates and then accounting for the so-called unobserved heterogeneity between subjects;
- Account for particular data structures, especially in the presence of repeated observations, longitudinal/panel data and multilevel data.

The idea at the basis of latent variable models is the principle of local independence, for which if a latent variable underlies a series of observed variables, then conditioning on that latent variable makes the observed variables statistically independent. Latent variable models are a wide and heterogeneous family, and are usually classified according to two criteria (Skrondal & Rabe-Hesketh, 2007): the nature of the response variables and the nature of the latent variables, which could be in both cases discrete or continuous. Here we list some of the most important examples of latent variable models:

- Factor analysis (Child, 2006), a classical tool in multivariate statistics, which summarizes several continuous measurements through a small number of continuous latent traits, called factors;
- Generalized linear mixed models (GLMMs) (McCulloch & Neuhaus, 2001), also referred to as random-effects models, that represent an extension of the class of generalized linear models (GLM) for continuous or categorical responses which account for unobserved heterogeneity, beyond the effect of observable covariates;
- Item Response Theory (IRT) models (Hambleton & Swaminathan, 1985; Bartolucci *et al.*, 2015), commonly used in questionnaire analysis and Psychometrics. In IRT we typically model categorical items measuring a common latent trait, assumed to be continuous, or less often discrete, representing an ability or a psychological attitude;

- Latent class analyses (Lazarsfeld *et al.*, 1968), that models a set of categorical response variables with a discrete latent variable, the levels of which correspond to latent classes in the population;
- Finite mixture models (Lindsay, 1995; McLachlan & Peel, 2004), in which subjects are assumed to come from different subpopulations corresponding to different levels of a discrete latent variable, each population or cluster being characterized by a different distribution of the response variables. The conditional distribution of the responses and/or the distribution of the latent variable can also be affected by observable covariates in finite mixture regression models (Skrondal & Rabe-Hesketh, 2004);
- Latent Markov models (LMM) (Zucchini & MacDonald, 2009; Bartolucci *et al.*, 2012) for longitudinal data, in which the response variables are assumed to depend on a Markov process with unobservable discrete states.

Beyond the nature of the involved variables, another important distinction between latent variable models concerns the estimation process. We can distinguish two alternative approaches, based on different sets of assumptions on the latent structure of the underlying model, and facing different computational challenges:

- The random-effects approach, where we consider the latent variables as random variables attaining a specific value for each sample unit;
- The fixed-effects approach, where we treat the values attained by the latent variables for each sample unit as fixed parameters.

The random-effects approach is widely used, because of the lower number of parameters to be estimated, and the high flexibility in modeling data structures. It however has several drawbacks, since it involves an a priori assumption on the distribution of the latent variable, that could be misspecified. Under the fixed-effects approach we avoid such assumption but, on the other hand, we end up with a number of parameters to be estimated that is proportional to the size of the data. In this case latent variables are interpreted as sets of individual-specific parameters, and if we are modeling time variant constructs as in LMM then they generate both individual- and time-specific parameters. Namely, if we specify a latent variable model for longitudinal data, an increase in the sample size or in the number of time points will correspond to an increase in the complexity of the model itself. We propose to act in the opposite direction, following the example by Tutz

& Oelker (2017), to promote simplicity by the means of a well suited penalization, inducing sparsity.

But what do we mean with sparsity? Generally speaking, we can say that sparsity is high in a model where only a small number of parameters plays an important role (Hastie *et al.*, 2015). We we call such a model sparse. For example, we can say that a linear regression model is sparse when only a limited subgroup of the predictors is considered to be meaningful and is associated with non-zero coefficients. This is the reason why a sparsity inducing process, that allows features selection, can be seen as a classical statistical learning tool (James *et al.*, 2013). This learning task in the framework of the least squares estimation of linear models can be performed with the use of alternative methods, both classical and modern:

- We can perform a classic subset selection. In a model with J predictors we can use inferential tools and stepwise methods to identify a subset of size J' of the predictors, with $J >> J'$, which we believe are significantly related to the response, and then fit the model on the reduced set of the J' selected covariates;

- We can perform dimension reduction before fitting the model, for example by the means of Principal Component Analysis or Factor Analysis. In other words we can project a J-dimensional problem into an J'-dimensional subspace, and then fit the model;

- We can use a regularization technique introducing a shrinkage operator, which consists in adding a penalty term when solving the least squares optimization problem. In this case the model is fitted with all the J predictors, but the estimated coefficients are shrunk towards zero. Depending on the type of penalty some coefficients may be shrunk exactly to zero, performing this way an automatic variable selection.

By retaining a subset of the predictors and discarding the rest, subset selection produces a model that is more interpretable and that has possibly lower prediction error than the full model. However, because it is a discrete process, since at each step variables are either retained or discarded, it often exhibits high variance. Shrinkage methods are more continuous, and do not suffer as much from high variability (Friedman *et al.*, 2001).

The two best-known regularization techniques for a regression problem are ridge regression (Hoerl & Kennard, 1970) and the lasso (Tibshirani, 1996):

- Ridge regression is particularly useful when there are many correlated variables in a linear regression model. It shrinks the regression coefficients toward zero by imposing a penalty on their size, which alleviates the effects of multicollinearity. The ridge coefficients minimize the objective function penalized by the sum of squares of the parameters. Since the shrunk parameters simultaneously reach zero, ridge regression does not perform an automatic feature selection;

- The lasso in a regression problem allows a continuous subset selection, and it consists in penalizing the objective function by the sum of the absolute value of the parameters. Forcing the sum of the absolute value of the regression coefficients to be less than a fixed value, it induces certain coefficients to be set exactly to zero, effectively choosing a simpler model that does not include those coefficients.

The lasso has known a huge popularity and it has been generalized to fit many different needs. For example, with a well suited penalty we can force some parameters to be shrunk to a given value, rather than zero, or to be similar among each other, penalizing the objective function on their differences. Some of the most interesting extensions of the lasso are:

- The elastic net, which combines and generalizes ridge and lasso regularization introducing a tuning parameter that controls the prevalence of one type of shrinkage on the other;

- The group lasso, which allows for a grouped shrinkage of the parameters, so that the coefficients in the same group are shrunk to zero in the same moment;

- The fused lasso, which penalizes the objective function by the absolute value of the differences of consecutive coefficients, so that the coefficients are forced to vary in a smooth fashion, for example in order to respect a certain spatial or temporal structure.

Fused-type penalties have also been used to perform unsupervised classification of statistical units, inducing sparsity on the pairwise differences of group centroids. Many different specifications of the pairwise fused lasso penalty have been proposed under different names in relatively recent literature. We will use the term convex clustering to refer in general to this whole family of clustering methods exploiting shrinkage techniques to perform grouping.

Our proposal is to apply a peculiar pairwise fused lasso penalty in the framework of the fixed-effects estimation of latent variable models in order to induce a natural clustering on a subset of the parameters space. The idea of clustering fixed effects has been widely explored in the work of Tutz, we refer in particular to Tutz & Oelker (2017), where the unobserved heterogeneity is treated using the grouping property of pairwise fused lasso regularization, and Berger & Tutz (2018), where the fixed effects are grouped using a recursive partitioning method.

In the present work we face this issue by adapting the Solution Path Clustering (SPC) algorithm, which was originally proposed in Marchetti *et al.* (2014) as a general unsupervised clustering method, to the estimation of grouped fixed effects. We believe the SPC to be particularly suited for our purpose for the following reasons. It applies a very flexible non-convex penalty on the pairwise differences of the parameters, promoting sparser models than the ℓ_1 penalty with the same or superior accuracy. It does not require previous knowledge on the number of groups, and it produces a complete solution path, regulated by a single tuning parameter. The tuning process is data-driven, and it can be adapted to datasets of different complexity. Furthermore, the SPC has the important property of naturally isolating singletons or very small clusters of outliers, that could bring severe bias if included in the estimation of group parameters.

We are the first to propose the SPC as an estimation tool in the fixed-effects latent variable models framework. We think that this method can lead to very good results in terms of accuracy of the estimates, particularly in models with a very large number of free parameters. The aim of this thesis is to develop a general estimation procedure based on the SPC, and to evaluate its performance, also compared to other methods, in latent variable models characterized by a growing amount of complexity. We are interested in exploring the case of both categorical and continuous data, evaluating the accuracy of the SPC estimates under different scenarios, with growing sample size and number of item variables. We choose to focus first on the most popular IRT model for binary data, the Rasch model (Fischer & Molenaar, 2012), and then on a linear latent variable model for continuous items, where the number of fixed effects to be estimated is higher.

The work is structured as follows:

- In Chapter 2 we discuss the fixed-effects estimation of latent variable models. We present the theoretical framework under the fixed-effects approach of the classic Rasch model for binary data;

- In Chapter 3 we illustrate the lasso technique applied to linear regression, and we present the main generalizations of the lasso penalty. We then discuss convex clustering, with particular attention to the algorithm by Marchetti *et al.* (2014);

- In Chapter 4 we present the proposed estimation method for the penalized fixed-effects Rasch model. We introduce the classification likelihood as an alternative approach to group fixed effects in this context. We perform a simulation study to compare the performance of the proposed method with the latent class approach and the classification likelihood. We show the results of a real data example on INVALSI data.

- In Chapter 5 we present the proposed estimation method for the penalized fixed-effects latent variable model for continuous data. We perform a simulation study and develop a real data example on INVALSI scores.

The concluding remarks are dedicated to the future development of the work, in particular to the extension of the proposed estimation method to the penalized fixed-effects variable-intercept model for longitudinal data.

Chapter 2

Fixed-Effects Latent Variable Models

2.1 Latent Variables as Fixed or Random Effects

Latent variables are theoretical constructs which cannot be directly observed, but whose values can be inferred on the basis of several other manifest variables (Bartholomew *et al.*, 2011). The use of latent variables is extremely popular in many fields of human knowledge, from Economics to Biology, from Medicine to Sociology, but they are historically relevant in particular to the field of Psychology (Borsboom *et al.*, 2003). The conceptual framework of latent variable analysis originates indeed in the context of intelligence testing with the work of Spearman (1904). For instance, it is not possible to directly measure the mathematical ability of a student, but we can easily measure his or her performance at a number of test items in mathematics. The observed answers to the items of this questionnaire are then assumed to be proxies of the latent ability (Crocker & Algina, 1986; Raykov & Marcoulides, 2011). Hence, the latent variable is indirectly measured through a statistical model. In this chapter we focus on Item Response Theory (IRT) models (Hambleton & Swaminathan, 2013; van der Linden & Hambleton, 2013), where the probability to provide a certain answer to a questionnaire item is defined as a function of a set of parameters characterizing the items, and of a person's level on the latent trait.

The conceptual status of this unobservable entity is tightly connected to the mathematical formulation of the statistical model. So, the distinction between the fixed-effects approach and the random-effects approach in latent variable models contributes not only to determine their parametrization, since it primarily concerns the nature itself of the latent constructs. In the random-effects approach, the individual levels of the latent trait are considered to be realizations of a random variable, characterized by a certain distribution in the population from which the sample has been drawn. This distribution has to be postulated, and it can be either continuous, usually normal, or discrete, allowing for the identification of latent classes in the

population. On the other hand, under the fixed-effects approach, the individual levels of the latent trait are included in the model as unknown fixed subject-specific parameters.

A wide literature is dedicated to a comparison between the two approaches, mainly in the framework of generalized linear mixed models (Skrondal & Rabe-Hesketh, 2004), and in particular on multilevel models for hierarchical, grouped or longitudinal data (Gardiner *et al.*, 2009; Clarke *et al.*, 2010; Townsend *et al.*, 2013). Regardless of a specific model, the choice between fixed and random effects depends on a number of theoretical and practical considerations listed in the following:

- Ontology, we have to clarify the nature of the latent construct. Under the fixed-effects approach the latent variable is considered as an intrinsic attribute of the individual, while under the random-effects approach it is an attribute of the population, characterized by a known distribution. The flexibility in defining this distribution allows us to handle a high level of complexity (multidimensionality, hierarchies, group structures, time dependencies) that under the fixed effect approach would be problematic.

- Parsimony, the number of parameters to be estimated increases dramatically under the fixed-effects approach. In particular, apart from the structural parameters of the chosen latent variable model, we also have to estimate at least one parameter measuring the latent trait for each individual in the sample. On the other hand, the random-effects approach requires us to only estimate the parameters of the postulated distribution, together with the structural parameters.

- Assumptions, the subjectivity in defining the latent variable distribution under the random-effects approach may lead to misspecification and related problems. Many works inquiry about the impact of misspecification of the random effect distribution on the estimators efficiency (Heagerty & Kurland, 2001; Agresti *et al.*, 2004; Litière *et al.*, 2008; McCulloch & Neuhaus, 2011). A wide literature faces this issue avoiding any distributional assumption through non-parametric estimation methods, as for example the recent work of Kelava *et al.* (2017), or going back to Aitkin (1995) and Laird (1978).

Following from the above example about a questionnaire in mathematics, if we are interested in estimating the latent ability of the respondents, we should consider initially whether or not it is reasonable to formulate a distributional assumption about it, and then choose which one fits best our data. Is the

ability normally distributed? Or is its distribution discrete? Can we isolate groups of respondents with homogeneous ability level?

In order to answer to these questions, in the present chapter we focus on a particular class of IRT model: the binary Rasch model (Rasch, 1960, 1961) or 1PL model, according to the nomenclature in Birnbaum (1968). We illustrate its basic formulation, and we discuss further details about its interpretation and estimation respectively under the fixed-effects and the random-effects approach. The Rasch Model is the most widespread among the IRT models for binary data, and we have chosen to use it as a baseline to emphasize the limitations of the fixed-effects approach in latent variable models for cross-sectional data.

2.2 The Rasch Model

2.2.1 Notation

We observe the responses of n subjects to J binary items, and we denote with y_{ij} the response of subject i to item j, where $i = 1, \dots, n$ and $j = 1, \dots, J$. We indicate with $y_{ij} = 1$ the correct answer of responded i to item j (where y_{ij} is the realization of the random variable Y_{ij}). We also denote by $\mathbf{y}_i = (y_{i1}, \dots, y_{iJ})'$ the response pattern of subject i, and with $\mathbf{Y}$ the data matrix of size $n \times J$, with columns $\mathbf{y}_j = (y_{1j}, \dots, y_{nj})'$:

$$\mathbf{Y} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1J} \\ y_{21} & y_{22} & \cdots & y_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nJ} \end{pmatrix}. \tag{2.1}$$

We can now define some quantities of interests such as the total score $y_{i\cdot}$ of respondent i (row score) and the total score $y_{\cdot j}$ for item j (column score)

$$y_{i\cdot} = \sum_{j=1}^{J} y_{ij} \quad \text{and} \quad y_{\cdot j} = \sum_{i=1}^{n} y_{ij} \tag{2.2}$$

which correspond respectively to the number of items endorsed by subject i and the number of subjects who endorsed item j. These descriptive statistics represent a first raw approximation of the individual's ability and easiness of the item (Baker, 2001).

2.2.2 The model

The Rasch model is the most popular IRT model for binary data. It was introduced by Rasch (1960), and it is based on an item characteristic curve

(ICC) of logistic type. The ICC defines the conditional probability $p_j(\theta_i)$ of a correct response of individual i to item j as a function of the individual latent trait level, indicated with θ_i. We can write:

$$p_j(\theta_i) = p(Y_{ij} = 1|\theta_i) = \frac{e^{\theta_i - \beta_j}}{1 + e^{\theta_i - \beta_j}} = \text{logit}^{-1}(\theta_i - \beta_j), \qquad (2.3)$$

where β_j is a parameter which describes the difficulty of item j. Both θ_i and β_j are defined in $\mathbb{R}$ and they are measured on the same scale, allowing for a direct comparison. The item difficulty β_j can be interpreted as the level of ability for which subject i has a probability equal to 0.5 of giving the correct answer to item j, given his ability level θ_i. In particular we have $p_j(\theta_i) = 0.5$ when $\theta_i = \beta_j$, $p_j(\theta_i) > 0.5$ when $\theta_i > \beta_j$, and $p_j(\theta_i) < 0.5$ when $\theta_i < \beta_j$. In this way an item characterized by a high difficulty will require a higher value of the latent ability in order to be endorsed. We can consider the item difficulty as a location parameter, since it identifies the point on the latent continuum at which the latent trait of a subject is located.

Equation (2.3) incorporates three fundamental assumptions (Crocker & Algina, 1986) that must be respected by all IRT models for binary data that are:

- UNIDIMENSIONALITY, which states that the responses to the J items by every individual i depend solely on a singular latent trait level $\theta_i \in \mathbb{R}$, and no other variables are involved in the response process.

- MONOTONICITY, according to which the ICC is a monotonic non-decresing function of θ_i. This is true for the logistic link function, that has an increasing S-shape, which approaches 0 for $\theta_i \to -\infty$ and 1 for $\theta_i \to +\infty$. This assumption guarantees that the probability of endorsing an item increases with an increase of the ability level of the respondent. Conversely the probability of success decreases as the difficulty parameters β_j increases. Figure 2.1 represents the ICC as a function of θ_i for five items having different difficulty levels $\beta_1 < \beta_2 < \beta_3 < \beta_4 < \beta_5$.

- LOCAL INDEPENDENCE, which states that the responses to the J items for each subject i are conditionally independent given the latent ability level θ_i. In other words, if the true levels of ability were known, the response of individual i to an item j would not give any additional information in predicting the response of the same individual to any other item. In the same way an individual with an higher ability level will respond better to any item with respect to an individual on a lower

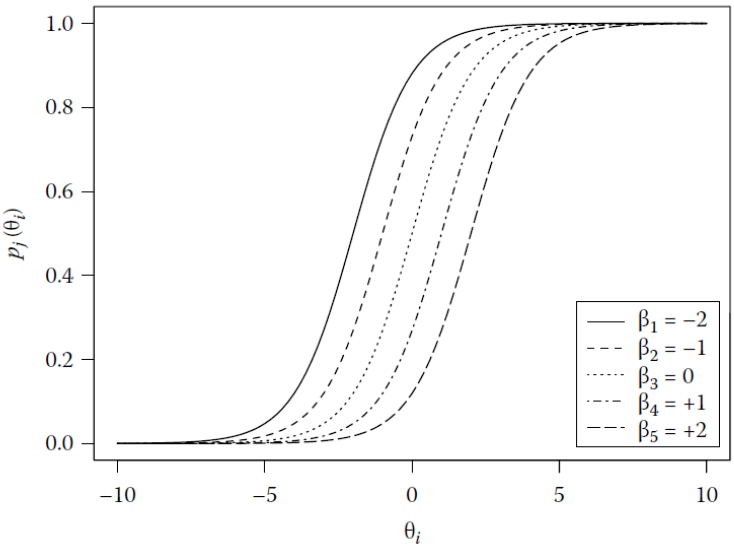

Figure 2.1: Item characteristic curves of a Rasch model for 5 items with increasing difficulty levels.

position on the latent trait. Thanks to this assumption we can write the joint distribution of a response pattern $\mathbf{y}_i$ given θ_i as follows:

$$p\left(\mathbf{y}_i|\theta_i\right) = \prod_{j=1}^{J} p_j\left(\theta_i\right)^{y_{ij}} \left[1 - p_j\left(\theta_i\right)\right]^{1-y_{ij}}. \tag{2.4}$$

Plugging in Equation (2.3) in (2.4) we obtain the explicit expression of the conditional probability for the Rasch model

$$p\left(\mathbf{y}_i|\theta_i\right) = \prod_{j=1}^{J} \frac{e^{y_{ij}(\theta_i - \beta_j)}}{1 + e^{\theta_i - \beta_j}} = \frac{e^{y_{i\cdot}\theta_i - \sum_{j=1}^{J} y_{ij}\beta_j}}{\prod_{j=1}^{p}\left(1 + e^{\theta_i - \beta_j}\right)}. \tag{2.5}$$

Recalling the implicit assumption that the response vectors corresponding to different subjects in the sample are independent to each other, we can write the conditional probability of observing the response matrix $\mathbf{Y}$ given the ability vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)'$, as:

$$p\left(\mathbf{Y}|\boldsymbol{\theta}\right) = \prod_{i=1}^{n} p\left(\mathbf{y}_i|\theta_i\right) = \prod_{i=1}^{n} \frac{e^{y_{i\cdot}\theta_i - \sum_{j=1}^{p} y_{ij}\beta_j}}{\prod_{j=1}^{J}\left(1 + e^{\theta_i - \beta_j}\right)} = \frac{e^{\sum_{i=1}^{n} y_{i\cdot}\theta_i - \sum_{j=1}^{J} y_{\cdot j}\beta_j}}{\prod_{i=1}^{n}\prod_{j=1}^{J}\left(1 + e^{\theta_i - \beta_j}\right)} \tag{2.6}$$

.

2.2.3 Fixed Effects or Random Effects

As mentioned in Section (2.1), under the random-effects approach the level θ_i of the latent ability in a Rasch model is considered as a realization of a random variable Θ_i with density function $f(\theta_i)$. Consequently, starting by Equation (2.4), we can obtain the marginal distribution of the response pattern $\mathbf{y}_i$ by integrating out the latent trait

$$p(\mathbf{y}_i) = \int_{\mathbb{R}} p(\mathbf{Y}|\boldsymbol{\theta})\, f(\theta_i)\, d\theta_i \tag{2.7}$$

where the density function $f(\theta_i)$ is common to every subject. The quantity $p(\mathbf{y}_i)$ is also known as manifest distribution (Bartolucci *et al.*, 2015). The random-effects approach has to be adopted when the group of respondents is considered a sample drawn from a population, the ability of which we know is characterized by a certain continuous or discrete distribution.

On the other hand, under the fixed-effects approach the ability levels $\theta_1, \ldots, \theta_n$ are considered as subject-specific parameters to be estimated along with the difficulty parameters. We can interpret the row scores $(y_{1\cdot}, \ldots, y_{n\cdot})'$ as a set of minimal sufficient statistics for $\boldsymbol{\theta}$, since the distribution with probability expressed by Equation (2.6) belongs to the exponential family, whose canonical parameters are a linear function of θ and β. The sufficiency of $y_{i\cdot}$ implies that individuals sharing the same number of endorsed items will also obtain the same estimate of the ability, independently from possible differences in their specific response patterns. In a similar way, the column scores $(y_{\cdot 1}, \ldots, y_{\cdot J})'$ represent a minimal sufficient statistic for the parameter vector $\boldsymbol{\beta}$. Adopting the fixed-effects approach we are not making any assumption concerning the population, we refer to a fixed group of subjects, each carrying an intrinsic value of the parameter, and we assume that the variability between repeated responses of the same subject to the same item is due only to accidental factors.

The choice between fixed-effects and random-effects implies different estimation strategies, each one characterized by its own advantages and disadvantages (Bartolucci *et al.*, 2015). We distinguish:

- JOINT MAXIMUM LIKELIHOOD (JML) under the fixed-effects approach, it consists in maximizing the likelihood of the model with respect to the abilities and the difficulties jointly. This is the method we will use to estimate the latent ability, and it will be extensively described in the next paragraph.

- CONDITIONAL MAXIMUM LIKELIHOOD (CML) under the fixed-effects approach, it consists in expressing the likelihood of the model as a

function of only one set of parameters, either abilities θ_i or difficulties β_j. The conditional likelihood, given the sufficient statistics for one set of parameters, is maximized with respect to the other using a Newton-Raphson algorithm.

- MARGINAL MAXIMUM LIKELIHOOD (MML) under the random-effects approach, it consists in maximizing the likelihood of the model after the ability parameters have been integrated out on the basis of a common known distribution. The ability is usually assumed to be Gaussian with arbitrary parameters μ and σ^2. The maximization is performed using the expectation maximization (EM) algorithm (Dempster *et al.*, 1977).

2.2.4 Estimation with the Joint Maximum Likelihood

The JML method involves the maximization of the Rasch model likelihood with respect to both ability and difficulty parameters simultaneously. For this reason we aggregate all the fixed-effects parameters in a single vector $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\beta})'$, and directly use Equation (2.6) to define the model likelihood function as:

$$L(\boldsymbol{\psi}) = \frac{e^{\sum_{i=1}^{n} y_{i\cdot}\theta_i - \sum_{j=1}^{J} y_{\cdot j}\beta_j}}{\prod_{i=1}^{n}\prod_{j=1}^{J}\left(1 + e^{\theta_i - \beta_j}\right)}, \tag{2.8}$$

with corresponding log-likelihood:

$$\ell\left(\boldsymbol{\psi}\right) = \log L\left(\boldsymbol{\psi}\right) = \sum_{i=1}^{n} y_{i\cdot}\theta_i - \sum_{j=1}^{J} y_{\cdot j}\beta_j - \sum_{i=1}^{n}\sum_{j=1}^{J} \log\left(1 + e^{\theta_i - \beta_j}\right). \tag{2.9}$$

The above quantity can be expressed in vector notation

$$\ell\left(\boldsymbol{\psi}\right) = \mathbf{y}_r'\boldsymbol{\theta} - \mathbf{y}_c'\boldsymbol{\beta} - \mathbb{1}_n' \log\left(1 + e^{\boldsymbol{\theta}\mathbb{1}_J' - \mathbb{1}_n\boldsymbol{\beta}'}\right)\mathbb{1}_J, \tag{2.10}$$

where $\mathbf{y}_r = (y_{1\cdot}, \ldots, y_{n\cdot})'$ and $\mathbf{y}_c = (y_{\cdot 1}, \ldots, y_{\cdot J})'$ are respectively the vectors of the row and column scores (containing the sufficient statistics for the model parameters), the symbol $\mathbb{1}_h$ indicates a vector of ones of generic length h, and $\log(1 + e^{\boldsymbol{\theta}\mathbb{1}_J' - \mathbb{1}_n\boldsymbol{\beta}'})$ is a $n \times J$ matrix of elements $\log(1 + e^{\theta_i - \beta_j})$, with $i = 1, \ldots, n$ and $j = 1, \ldots, J$.

$$\log(1+e^{\boldsymbol{\theta}\mathbb{1}_J' - \mathbb{1}_n\boldsymbol{\beta}'}) = \begin{bmatrix} \log(1 + e^{\theta_1 - \beta_1}) & \log(1 + e^{\theta_1 - \beta_2}) & \cdots & \log(1 + e^{\theta_1 - \beta_J}) \\ \log(1 + e^{\theta_2 - \beta_1}) & \log(1 + e^{\theta_2 - \beta_2}) & \cdots & \log(1 + e^{\theta_2 - \beta_J}) \\ \vdots & \vdots & \ddots & \vdots \\ \log(1 + e^{\theta_n - \beta_1}) & \log(1 + e^{\theta_n - \beta_2}) & \cdots & \log(1 + e^{\theta_n - \beta_J}) \end{bmatrix}. \tag{2.11}$$

A problem arises since both $L(\boldsymbol{\psi})$ and $\ell(\boldsymbol{\psi})$ are invariant with respect to translations of the parameters θ_i and β_j. The reason is that if in Equation (2.3) we add a constant to every ability parameter and every difficulty parameter, such that $\theta_i^* = \theta_i + c$ and $\beta_j^* = \beta_j + c$, then $p_j^*(\theta_i^*) = p_j(\theta_i)$. This makes the model non-identifiable, and in order to reach an estimate of $\boldsymbol{\psi}$ we have to put suitable identifiability constraints on some parameters. One may choose between:

- Setting to zero the first item difficulty $\beta_1 = 0$. In this case the first item is taken as a reference item, in the sense that we interpret the other difficulty parameters β_j where $j = 2, \ldots, J$ with respect to it.

- Setting to zero the average difficulty level $\sum_{j=1}^{J} \beta_j = 0$ or the average ability level $\sum_{i=1}^{n} \theta_j = 0$, so that the parameters estimates are interpreted as deviations from the mean.

The two constraints are equivalent, leading to a maximum likelihood value equal to the unconstrained maximum likelihood. Besides, the estimates obtained under one identification rule can be easily transformed into the estimates obtain under the other one. In this work we choose to adopt the first one, resulting the new vector of free parameters as $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\beta}^*)'$, where $\boldsymbol{\beta}^* = (\beta_2, \ldots, \beta_J)'$.

The log-likelihood (2.9) can be maximized using a Newton-Raphson iterative algorithm that at each step updates the parameter estimates until convergence (Bartolucci *et al.*, 2015). In more detail, let $h = 1, \ldots, H$ be the iteration index, and $\boldsymbol{\psi}^{(h)}$ be the $\boldsymbol{\psi}$ estimate obtained at step h, abilities and difficulties are updated as follows

$$\theta_i^{(h+1)} = \theta_i^{(h)} - \frac{\partial \ell(\boldsymbol{\psi}^{(h)})}{\partial \theta_i} \left[\frac{\partial^2 \ell(\boldsymbol{\psi}^{(h)})}{\partial \theta_i^2} \right]^{-1} \quad \text{and} \quad \beta_j^{(h+1)} = \beta_j^{(h)} - \frac{\partial \ell(\boldsymbol{\psi}^{(h)})}{\partial \beta_j} \left[\frac{\partial^2 \ell(\boldsymbol{\psi}^{(h)})}{\partial \beta_j^2} \right]^{-1} \tag{2.12}$$

where $i = 1, \ldots, n$ and $j = 2, \ldots, J$, the first and second derivatives with respect to θ_i are:

$$\frac{\partial \ell(\boldsymbol{\psi}^{(h)})}{\partial \theta_i} = y_{i\cdot} - \sum_{j=1}^{J} p_j(\theta_i) \quad \text{and} \quad \frac{\partial^2 \ell\left(\boldsymbol{\psi}^{(h)}\right)}{\partial \theta_i^2} = -\sum_{j=1}^{J} p_j(\theta_i)\left[1 - p_j(\theta_i)\right], \tag{2.13}$$

and the first and second derivatives with respect to β_j are:

$$\frac{\partial \ell(\boldsymbol{\psi}^{(h)})}{\partial \beta_j} = -\left[y_{\cdot j} - \sum_{i=1}^{n} p_j\left(\theta_i\right) \right] \quad \text{and} \quad \frac{\partial^2 \ell\left(\boldsymbol{\psi}^{(h)}\right)}{\partial \theta_i^2} = -\sum_{i=1}^{n} p_j\left(\theta_i\right)\left[1 - p_j\left(\theta_i\right)\right], \tag{2.14}$$

where $p_j(\theta_i)$ is defined as in Equation (2.3). Starting from expression (2.10), we can write the gradient $\nabla_{\boldsymbol{\psi}}$ and the Hessian $H_{\boldsymbol{\psi}}$ in vector notation:

$$\nabla_{\boldsymbol{\psi}} = \begin{bmatrix} \mathbf{y}_r - \mathbf{P}\mathbb{1}_J \\ -\mathbf{y}_c + \mathbf{P}'\mathbb{1}_n \end{bmatrix}, \tag{2.15}$$

$$H_{\boldsymbol{\psi}} = \begin{bmatrix} \operatorname{diag}\left\{\left[\mathbf{P}\left(1-\mathbf{P}\right)\right]\mathbb{1}_J\right\} & \left[\mathbf{P}\left(1-\mathbf{P}\right)\right] \\ \left[\mathbf{P}\left(1-\mathbf{P}\right)\right]' & \operatorname{diag}\left\{\left[\mathbf{P}\left(1-\mathbf{P}\right)\right]'\mathbb{1}_n\right\} \end{bmatrix}, \tag{2.16}$$

where $\mathbf{P}$ is a $n \times J$ matrix of elements

$$\mathbf{P} = \begin{pmatrix} p_1(\theta_1) & p_2(\theta_1) & \cdots & p_J(\theta_1) \\ p_1(\theta_2) & p_2(\theta_2) & \cdots & p_J(\theta_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\theta_n) & \cdots & \cdots & p_J(\theta_n) \end{pmatrix}. \tag{2.17}$$

and $[\mathbf{P}\left(1-\mathbf{P}\right)]$ is a matrix of the element-wise products $p_j\left(\theta_i\right)\left[1-p_j\left(\theta_i\right)\right]$.

The algorithm can be initialized with arbitrary values $\boldsymbol{\psi}^{(0)}$. The choice for the initialization is not extremely relevant being $\ell(\boldsymbol{\psi})$ a strictly concave function in the parameter space. Bartolucci *et al.* (2015) suggest to use a data driven initialization

$$\theta_i^{(0)} = \log \frac{y_{i\cdot}}{J - y_{i\cdot}} \quad \text{and} \quad \beta_j^{(0)} = \log \frac{n - y_{\cdot j}}{y_{\cdot j}} - \log \frac{n - y_{\cdot 1}}{y_{\cdot 1}} \tag{2.18}$$

for $i = 1, \ldots, n$ and $j = 2, \ldots, J$.

If properly initialized the algorithm converges reasonably fast to the JML estimate $\hat{\boldsymbol{\psi}} = (\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}})'$. A convergence rule based on both the maximum likelihood difference at consecutive steps and the distance between consecutive solutions can be adopted.

$$\ell(\boldsymbol{\psi}^{(h)}) - \ell(\boldsymbol{\psi}^{(h+1)}) < \varepsilon_1 \quad \text{and} \quad \max|\boldsymbol{\psi}^{(h)} - \boldsymbol{\psi}^{(h+1)}| < \varepsilon_2 \tag{2.19}$$

with ε_1 and ε_2 small constants.

However, the JML estimate is not guaranteed to exist, Fischer (1981) provides a set of conditions on the matrix $\mathbf{Y}$ that ensures the existence of

the JML estimate. A necessary condition is that in that $0 < y_{i\cdot} < J$ for $i = 1, \ldots, n$ and $0 < y_{\cdot j} < n$ for $j = 1, \ldots, J$, namely that there are no subjects that responded correctly or incorrectly to all items and that there are no items to which all subjects responded correctly or incorrectly. If there are rows and columns with all elements equal to zero or one they have to be eliminated from the dataset.

2.2.5 Latent Class Rasch Model

One unique property of the random-effects approach is the possibility to specify a discrete distribution for the latent construct. In this way, just as in latent class models (Lazarsfeld *et al.*, 1968; Goodman, 1974), we assume that the population under study is composed by a number of classes or subpopulations that are homogeneous in terms of the unobservable construct. A discreteness assumption is particularly convenient when we want to cluster individuals on the basis of their latent ability, or in those cases when we have many items and many different values of the sufficient statistics $y_{i\cdot}$. The latent class Rasch (LC-Rasch) model has been proposed by Rost (1990), other examples of this formulation and its extensions can be found in Lindsay *et al.* (1991), Formann (1995) and Bartolucci (2007).

In general, we assume that the random variable Θ_i, where $i = 1, \ldots, n$, has a discrete distribution with support points $\xi_1, \ldots, \xi_K$. Each support point measures the latent ability of the k-th latent class, with associated support points π_k, $k = 1, \ldots, K$, representing the probability that a subject belongs to class k, given by

$$\pi_k = p(\Theta_i = \xi_k), \tag{2.20}$$

with $\sum_{k=1}^{K} \pi_k$ and $\pi_k \geq 0$, $k = 1, \ldots, K$. Assuming that the difficulty parameters are constant across classes, we can write the ICC as:

$$p_j(\xi_k) = p(Y_{ij} = 1|\xi_k) = \frac{e^{\xi_k - \beta_j}}{1 + e^{\xi_k - \beta_j}}. \tag{2.21}$$

As in the case of continuous latent variable, the estimation of the parameters is based on the MML method and solved by the EM algorithm. Alternatively, in Lindsay *et al.* (1991) the LC Rasch model is interpreted and estimated as a finite mixture model.

2.2.6 Limits of the JML

We have seen how the JML estimation of the Rasch model under the fixed-effects approach is simple and straightforward, but it has several relevant drawbacks with respect to the MML estimation under the random-effects approach:

- The main drawback is the lack of consistency of the resulting estimator for J fixed as n grows to infinity. The reason is that the number of the ability parameters increases with the sample size.

- The number of different values of the ability parameters estimates is equal to the number of different values of the sufficient statistics $y_{i\cdot}$. This may represent a drawback when we desire a group structure like in LC Rasch Model, and in all those cases when we desire a number of unique values for the estimates $\hat{\boldsymbol{\theta}}$ that is lower than the number of unique values of the sufficient statistics in $\mathbf{y}_r$.

We propose to address these issues using well suited penalization techniques.

Chapter 3

An Overview on Lasso-Type Penalties

3.1 The Lasso

The lasso (least absolute shrinkage and selection operator) was first introduced by Tibshirani (1996) as a method for both shrinkage and selection in a general regression framework. It is a continuous shrinking operator, it allows automatic variable selection, and it always leads to sparse solutions. This properties gained the lasso a huge popularity, and it has become quickly a fundamental tool in statistical learning (Friedman *et al.*, 2001; James *et al.*, 2013; Hastie *et al.*, 2015). Other standard techniques, such as ridge regression (Hoerl & Kennard, 1970) and subset selection algorithms, had the drawback to perform either only shrinkage or variable selection. In particular, stepwise methods, like forward selection and backward elimination, are likely to produce unstable outputs, being discrete processes based on a certain selection criterion. On the other side, the lasso is based on the penalization of a loss function by the ℓ_1-norm of the parameter vector. It results in a quadratic programming problem with linear inequality constraints. In practice, it forces a subset of regression coefficients to be exactly equal to zero, imposing the sum of their absolute values to be less or equal than a user-specified tuning parameter.

3.1.1 Notation

In the framework of generalized linear models (GLM) (Nelder & Baker, 1972; McCullagh, 1984) we observe n values of the response variable Y, which can be either continuous or binary, and of a set of predictors X_j, with $j = 1, \ldots, J$. We can define $\mathbf{x}_i = (x_{1i}, \ldots, x_{Ji})$ as the J-dimensional vector of predictors, and each $y_i \in \mathbb{R}$ as the associated value of the response variable. The vectors $\mathbf{x}_i$ are stacked as rows of the $n \times J$ matrix of predictors $\mathbf{X}$. The model is parametrized by a vector of regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)'$ and an intercept $\beta_0 \in \mathbb{R}$.

In this chapter we make extensive use of different norms that, for the sake of clarity, we explicitly define here:

- The ℓ_1 norm, Taxicab norm or Manhattan norm of a vector $\boldsymbol{\beta}$ is defined as the sum of the absolute values of its elements, and it is indicated with the symbols $\|\cdot\|$ or $\|\cdot\|_1$:

$$\|\boldsymbol{\beta}\| = \sum_{j=1}^{J} |\beta_j|. \tag{3.1}$$

- The ℓ_2 norm or Euclidean norm of a vector $\boldsymbol{\beta}$ is defined as the square root of sum of its squared elements, and it is indicated with the symbol $\|\cdot\|_2$:

$$\|\boldsymbol{\beta}\|_2 = \sqrt{\boldsymbol{\beta}'\boldsymbol{\beta}} = \sqrt{\sum_{j=1}^{J} \beta_j^2}. \tag{3.2}$$

- The ℓ_q norm or q-norm is a generalized norm that is equal to the ℓ_1 for $q = 1$ and to the ℓ_2 for $q = 2$. It is indicated with the symbol $\|\cdot\|_q$:

$$\|\boldsymbol{\beta}\|_q = \left(\sum_{j=1}^{J} |\beta_j|^q \right)^{\frac{1}{q}}. \tag{3.3}$$

3.1.2 The Lasso for Linear Regression

Lets consider a linear regression model of the form:

$$E(Y_i|\mathbf{X}) = f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^{J} x_{ij}\beta_j, \tag{3.4}$$

for which the classic ordinary least square estimate $(\hat{\beta}_0^{\text{OLS}}, \hat{\boldsymbol{\beta}}^{\text{OLS}})$ is obtained minimizing the residual sum of squares. The lasso finds the solution $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\boldsymbol{\beta}})$ to the constrained optimization problem:

$$\min_{\beta_0\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{J} x_{ij}\beta_j \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{J} |\beta_j| \leq t, \tag{3.5}$$

which can be written more compactly just in vector notation:

$$\min_{\beta_0\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \beta_0 \mathbb{1}_n - \mathbf{X}\boldsymbol{\beta}\|_2^2 \right\} \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_1 \leq t, \tag{3.6}$$

where $\mathbf{y} = (y_1, \ldots, y_n)$ is the vector of continuous responses. The tuning parameter t is a predetermined fixed scalar, and it can be seen as a sort of budget, limiting the sum of the absolute values of the parameters estimates, and controlling the desired amount of shrinkage. The value of t is usually chosen by cross-validation, as will be discussed in Section 3.1.4. The matrix of predictors $\mathbf{X}$ is standardized so that each column is centered and has unit variance, in order to avoid biases due to different scales. If we consider the response variable to be centered too, we can omit the intercept term β_0 and rewrite the problem as:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \right\} \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_1 \leq t. \tag{3.7}$$

The optimization problem (3.7) can be expressed also in the Lagrangian form

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\} \quad \text{with} \quad \lambda \geq 0, \tag{3.8}$$

where, given the strict convexity of both the loss function and the penalty term, by the Lagrangian duality, there is a one-to-one correspondence between λ and t (Bertsekas, 1999). As already mentioned, the structure of the ℓ_1 penalty allows not only the shrinkage of the parameters, but also the automatic selection of the variables: as the value of t decreases an increasing number of parameters are forced to be exactly equal to zero. Thanks to this property the lasso always leads to sparse solutions, and it becomes more useful when working with large problems, particularly in the case of wide data, when $J \gg n$. In order to better understand the mechanism beyond the sparsity-generating process of the lasso penalty, it is useful to look at its geometrical implications, comparing its structure with another common shrinkage technique: the ridge regression. In ridge regression we optimize:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{J} \beta_j^2 \leq t^2, \tag{3.9}$$

so that, for decreasing values of t, the parameters are shrunk together, and they reach 0 only when $t = 0$. Figure 3.1 shows the profiles of the solution path for the lasso and ridge penalties applied to the estimation of the same linear regression model. We can see how the lasso solutions gradually hit the zero as t decreases, meaning that the corresponding variables can be deleted from the model, while the ridge regression solutions are shrunk together way more smoothly until they reach zero simultaneously, not allowing for variable selection. Figure 3.2 directly compares the shape of the two constraints in a

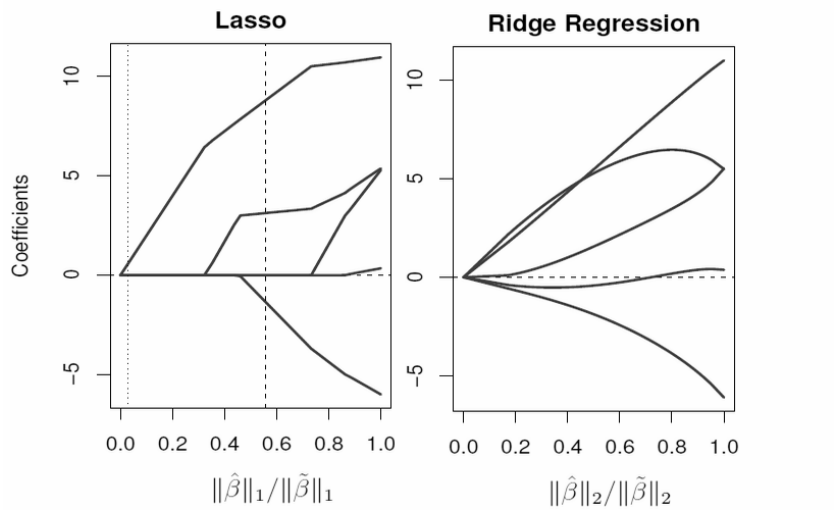

Figure 3.1: Solution paths for the lasso (left) and ridge regression (right).

linear regression model with $J = 2$. The residual sum of squares has elliptical contours, and it is centered at the full least-squares estimates. Geometrically speaking, both methods find the penalized solution where the elliptical contours first hit the constraint region. This area is a diamond $|\beta_1| + |\beta_2| \leq t$ for the lasso, and a disk $\beta_1^2 + \beta_2^2 \leq t^2$ for the ridge regression. Unlike the disk, the diamond has corners, and if the elliptical contours hits the diamond right on one of its corners, then one parameter β_j results exactly equal to zero. When $J > 2$ the diamond becomes a rhomboid, and has many more corners, increasing the number of opportunities for the estimated parameters to be zero, while the disk becomes a sphere.

Both lasso and ridge regression can be seen as special cases of a general optimization problem subject to an ℓ_q penalty, which takes the following Lagrangian form:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 - \lambda \sum_{j=1}^{J} |\beta_j|^q \right\}. \tag{3.10}$$

This problem reduces to the lasso for $q = 1$ and to ridge regression for $q = 2$. For $q = 0$ the penalty term just counts the non-zero elements of the vector $\boldsymbol{\beta}$, after performing a best-subset selection. Figure 3.3 shows the shape of the constraint regions corresponding to these penalties for the case of two predictors ($J = 2$). For values of q greater or equal to one we stay in the

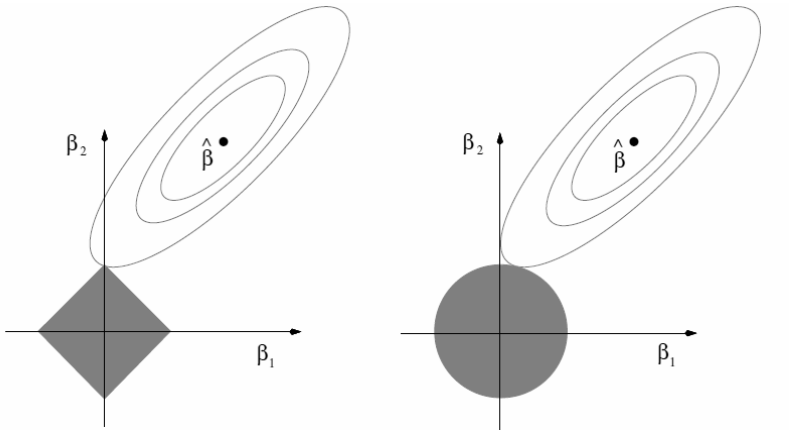

Figure 3.2: Geometrical interpretation of the lasso (left) and ridge regression (right) with $J = 2$.

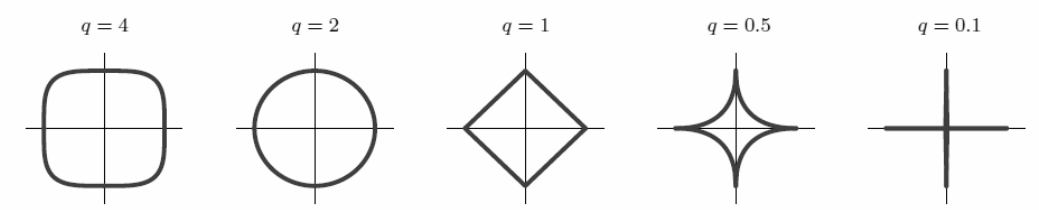

Figure 3.3: Constraint regions for a penalty of the general form $\sum_{j=1}^{J} |\beta_j|^q$ with different values of q.

framework of convex optimization, while for $q < 1$ we have a non-convex programming problem (which leads to several computational difficulties).

Computationally speaking, the lasso problem (3.8) is a convex optimization program, in particular a quadratic program (QP) with a convex constraint. As such, many sophisticated QP methods are capable of finding its solutions. However, apart from this general convex optimizers, several alternative algorithms have been designed and proposed specifically in order to find the lasso solutions. At first, in his seminal paper Tibshirani (1996) outlined a combined quadratic programming method, in which the inequality constraints were introduced sequentially, seeking a feasible solution that satisfied the optimality conditions. A few years later, his student Fu (1998) developed the shooting algorithm, a first coordinate-wise minimization procedure, derived interpreting the lasso estimator as the right limit of the bridge estimator (Frank & Friedman, 1993) when the order of the penalty norm goes

to one. An efficient alternative is represented by the least angle regression (LAR), proposed by Efron *et al.* (2004). With respect to earlier methods, LAR had the advantage to produce the entire piecewise linear solution path, instead of returning a single vector solution. This property was particularly appealing in the tuning phase of the model. LAR is sometimes referred to as the *homotopy* approach, having much in common with an earlier piecewise linear path algorithm for computing the lasso, proposed by Osborne *et al.* (2000a,b). Finally, working on the idea of Fu (1998), Friedman *et al.* (2007) developed a coordinate minimization procedure, the cyclical coordinate descent, which minimized the convex objective along a coordinate at a time, leading under mild conditions to the global optimum. Iterating the algorithm over different values of the regularization parameter, it was possible to create the entire solution path, just as for the LAR. The entire algorithm is referred to as pathwise coordinate descent and it is nowadays considered to be the fastest and simplest method to solve the basic lasso problem. For many generalizations of the lasso a correspondent coordinate descent (or ascent) algorithm has been developed. Also the algorithm that we propose to use in Chapter 4 acts coordinate-wise.

3.1.3 The Lasso for Logistic Regression

The lasso has been widely applied to generalized linear models. Tibshirani (1996) proposed its application to logistic regression, for which coordinate descent algorithm have been later developed by Friedman *et al.* (2007). In logistic regression we apply the ℓ_1 regularization to the negative log-likelihood:

$$\Lambda(\boldsymbol{\beta}) = -\frac{1}{N}\sum_{i=1}^{n}\left[y_i\left(\beta_0 + \boldsymbol{\beta}'\mathbf{x}_i\right) - \log\left(1 + e^{\beta_0+\boldsymbol{\beta}'\mathbf{x}_i}\right)\right] + \lambda\|\boldsymbol{\beta}\|_1. \qquad (3.11)$$

The objective is convex and the likelihood part is differentiable, so finding the solution is a standard task in convex optimization. Coordinate descent type algorithms can be implemented over a quadratic approximation of the likelihood.

3.1.4 Inference and Cross-Validation

The tuning parameter t in the lasso criterion controls the complexity of the model. It acts like a budget, larger values of t produce more free parameters and allow a better fit of the model to the data, smaller values induce a stronger shrinkage, reduce the number of non zero parameters, leading to sparser, more interpretable models that fit the data less closely. A value of

t that is too small can produce a poorly adapted model, while a large value can lead to overfitting. Cross-validation is the most common strategy to estimate the best value for t that strikes a good balance in the trade-off between goodness of the fit and interpretability. In practice, we first randomly divide the full dataset into some number of groups K (typically 5 or 10). We choose one group as the test set, and use the remaining $K-1$ groups as the training set. We then apply the lasso to the training data over a sequence of different t values, and we use each fitted model to predict the responses in the test set, recording the mean-squared prediction errors for each value of t. After repeating this K times, so that each group has been used once as test set, we choose the value of t that minimizes the error measure. With very large datasets this procedure can be computationally intensive and time demanding. More problems arise when the parameters involved in the penalty term are more than one, as it happens in many lasso generalizations.

Concerning inference on the lasso estimates, various standard error estimators have been proposed. Tibshirani (1996) suggested to compute the standard errors using the bootstrap (Efron, 1979; Efron & Tibshirani, 1986), arguing that a closed form approximation for the covariance matrix leads to a null estimated variance for those predictors with an estimated coefficient shrunk exactly to zero. This limitation is shared also by other sandwich formulas proposed in Fan & Li (2001) and Zou (2006). Osborne *et al.* (2000b) derived an approximation formula that yields to a positive error for all the coefficient estimates, but pointed out that the estimates may be far from normally distributed. Pötscher & Leeb (2009) showed that the finite sample distribution of the lasso (soft-thresholding) estimator, can be highly non-normal irrespective of sample size, while Knight & Fu (2000) considered its asymptotic behavior. From this findings, Kyung *et al.* (2010) proved the inconsistency of bootstrap standard errors if the true coefficient is equal to zero. Since the bootstrap can not be considered a general method of obtaining standard errors of the lasso estimates, Kyung *et al.* (2010) proposed a fully Bayesian formulation, extending the hierarchical representation suggested by Park & Casella (2008). The Bayesian interpretation of the lasso was first sketched by Tibshirani (1996), and it has been widely applied.

3.2 Generalizations of the Lasso

The good properties of the lasso stimulated much research about possible generalizations of its penalty, to overcome some of its limitations and extend the fields of application. For example, the basic lasso does not per-

form well in the presence of multicollinearity, for this reason Zou & Hastie (2005) proposed the elastic net, that combines the ℓ_1-penalty with a squared ℓ_2-penalty, tending to select (or not) the correlated features together. Another important generalization is represented by the group lasso, introduced Yuan & Lin (2007), that selects or omits groups of variables, when the groups are known. The group lasso is extremely useful for the correct treatment of polytomous categorical predictors, allowing in variable selection to include or exclude simultaneously all the dummy variables used to code the levels. But lasso-type penalties can be designed to account for many different data structures, not only groups. The fused lasso, introduced by Tibshirani *et al.* (2005), is designed for problems with features that can be ordered in some meaningful way. It encourages local constancy of the coefficient profile, penalizing the loss function by the ℓ_1-norm of both the coefficients and their successive differences. The idea of penalizing the differences of parameters, instead of penalizing the parameters themselves, stimulated much research, and it lead to what is sometimes referred to as convex clustering. This technique and the idea of using a well suited penalty to induce clustering on a set of parameters are at the basis of our proposal, and they will be covered in the last section of this chapter. Before it can be useful to have a closer look to the above mentioned generalizations of the lasso, from which the others in some way descend.

3.2.1 The Elastic Net

The lasso performs poorly when the predictors are highly correlated (Oyeyemi *et al.*, 2015). In case of multicollinearity the solution paths of the lasso tend to be erratic, expressing a wild behavior. The elastic net represents a possible solution to this problem. It consists in a compromise between the ridge regression (that was introduced as a way of handling multicollinearity problems in regression) and the lasso penalty. The linear combination that is regulated by a mixing parameter α. The elastic net problem can be expressed as:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \left[\frac{1}{2}\left(1-\alpha\right)\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1 \right] \right\}, \tag{3.12}$$

with $\lambda \geq 0$ and $\alpha \in [0,1]$. Since the penalty associated to an individual coefficient is given by:

$$\frac{\left(1-\alpha\right)\beta_j^2 + \alpha|\beta_j|}{2}, \tag{3.13}$$

it is clear that when $\alpha = 1$, the elastic net penalty reduces to the ℓ_1-norm, corresponding to the lasso, while when $\alpha = 0$ it reduces to the squared ℓ_2-norm,corresponding to the ridge penalty. Friedman *et al.* (2015) built

a system of coordinate-descent algorithms for fitting elastic-net penalized generalized linear models.

3.2.2 The Group Lasso

The group lasso encourages sparsity between natural groups of predictors, forcing the β_j inside the same group to be shrunk simultaneously to zero. We assume that the design matrix $\mathbf{X}$ is composed by G known groups of predictors, such that $\mathbf{X} = (\mathbf{X}_1|\mathbf{X}_2|\dots|\mathbf{X}_G)$, with $g = 1, \dots, G$. We indicate with J_g the size of the g-th group, corresponding to a set of J_g columns of the matrix $\mathbf{X}$. The group lasso problem can be defined as:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \sum_{g=1}^{G} X_g \boldsymbol{\beta}_g\|_2^2 \right\} \quad \text{subject to} \quad \sum_{g=1}^{G} \|\boldsymbol{\beta}_g\|_2 \leq t, \tag{3.14}$$

where $\boldsymbol{\beta}_g$ is a subvector of $\boldsymbol{\beta}$ reflecting the group structure. The group generalization of the lasso has two properties:

1. Depending on t (or λ), either the entire vector $\boldsymbol{\beta}_g$ will be zero, or all its elements will be non zero. In other words sparsity is induced only among groups, not within groups.

2. When $J_g = 1$, then we have $\|\boldsymbol{\beta}_g\|_2 = |\beta_j|$, so if all groups are singletons the group lasso problem reduces to the basic lasso.

In this formulation (3.14) all the groups are equally penalized, but in this way larger groups are more likely to be selected. A possible solution is to weight the penalties for each group according to their size, for example by a factor $\sqrt{J_g}$, as suggested by Yuan & Lin (2007).

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \sum_{g=1}^{G} X_g \boldsymbol{\beta}_g\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{p_g} \|\boldsymbol{\beta}_g\|_2 \right\} \tag{3.15}$$

To solve the problem (3.14) we apply the block coordinate descent, that consists in minimizing the Lagrangian function with respect of the k-th block of predictors (vector $\boldsymbol{\beta}_k$), cycling through the G groups while holding fixed all the other block parameter vectors to their current value $\{\hat{\boldsymbol{\beta}}_j, j \neq k\}$. The group lasso has been widely extended. Meier *et al.* (2009) adapted it to logistic regression problems, while other authores proposed variations of the penalty structure in order to allow for example sparsity within groups or the presence of overlapping groups. These are the cases of the overlap group lasso (Jacob *et al.*, 2009) and the sparse group lasso (Puig *et al.*, 2009; Simon *et al.*, 2013).

- Sparse Group Lasso, For the properties of the ℓ_2 norm, when a group is included into a group-lasso fit, all the coefficients in that group must be non-zero. Sometimes we may want to induce sparsity not only with respect to which groups are selected, but also with respect to coefficients within each group. The sparse group lasso, inspired by the elastic net, is designed to achieve within-group sparsity, augmenting the group-lasso penalty with an additional ℓ_1-penalty, through a mixing parameter $\alpha \in [0, 1]$:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \sum_{g=1}^{G} X_g \boldsymbol{\beta}_g\|_2^2 + \lambda \sum_{g=1}^{G} \left[(1-\alpha) \|\boldsymbol{\beta}_g\|_2 + \alpha \|\boldsymbol{\beta}_g\|_1 \right] \right\}, \quad (3.16)$$

  where with $\alpha = 0$ we get the group lasso and with $\alpha = 1$ the lasso.

- Overlap Group Lasso, There are cases in which some predictors may belong to more than one group. The overlap group lasso allows variables to be accounted in all the groups in which they are included. This penalty simply replicates a variable in whatever group it appears, and then fits the ordinary group lasso.

3.2.3 The Fused Lasso

The fused lasso is useful whenever we want to take into account a structural relation between the parameters object of shrinkage. The nature of this relationship can be for example temporal or spatial adjacency. In the linear regression framework a fused lasso problem in its Lagrangian form typically appears as follows:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{J} x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^{J} |\beta_j| + \lambda_2 \sum_{j=2}^{J} |\beta_j - \beta_{j-1}| \right\}, \quad (3.17)$$

where we have two tuning parameters λ_1 and λ_2, corresponding to two separate penalties: the ℓ_1 penalty of the base lasso, that shrinks the parameters towards zero, and the fusion penalty that encourages neighboring coefficients β_j to be similar, forcing many of them to be identical. This second penalty makes sense as long as the ordering of the coefficients is fixed and somewhat informative.

The fused lasso problem cannot be solved using a coordinate descent algorithm, due to the fact that the difference penalty is not a separable function of the coordinates, but it can be expressed as in Tibshirani *et al.*

(2011), where the authors propose a solution path algorithm for any lasso problem of the form:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\mathbf{D}\boldsymbol{\beta}\|_1 \right\}, \tag{3.18}$$

where $\mathbf{D}$ is a penalty matrix of size $m \times J$, being m the number of constraints built on the parameters vector of length J.

In the fused lasso example $m = n-1$ is the number of differences between consecutive elements of $\boldsymbol{\beta}$, and the matrix $\mathbf{D}$ will have rows in which the non-zero elements are always adjecent couples of -1 and 1:

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ & & \cdots & & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}. \tag{3.19}$$

In Tibshirani *et al.* (2011) the one-dimensional fused lasso is presented as a simple tool for signal approximation. In their application, since it produces a piecewise constant fit, the fused lasso is also used for smoothing purposes:

$$\min_{\boldsymbol{\theta}} \{\Lambda(\boldsymbol{\theta})\} = \min_{\boldsymbol{\theta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda_1 \sum_{i=1}^{n} |\theta_i| + \lambda_2 \sum_{i=2}^{n} |\theta_i - \theta_{i-1}| \right\}. \tag{3.20}$$

3.2.4 The Pairwise Fused Lasso

The idea of applying the shrinkage on differences among parameters instead of parameters themselves opened the path for a whole new kind of generalization aimed at reproducing specific parametric structures. In Petry *et al.* (2011) the authors introduce the pairwise fused lasso (PFL) penalty, that uses the $\ell - 1$ norm of the pairwise differences of coefficients in a generalized linear model, extending the fused lasso to the case in which it is not possible to define a fixed ordering of the predictors. The PFL penalty is defined as:

$$P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta}) = \lambda \left[\alpha \sum_{j=1}^{J} |\beta_j| + (1-\alpha) \sum_{k<j} |\beta_j - \beta_k| \right], \tag{3.21}$$

where $\lambda > 0$ and $\alpha \in [0, 1]$ are the tuning parameters. The first term in the penalty is the lasso term and accounts for variable selection, the second term

is the fusion term and accounts for grouping. Petry *et al.* (2011) proposes two optimization procedures to solve the problem

$$\min_{\boldsymbol{\beta}} \left\{ \ell(\boldsymbol{\beta}) + P_{\lambda,\alpha}^{PFL}(\boldsymbol{\beta}) \right\} \tag{3.22}$$

where $\ell(\boldsymbol{\beta})$ is the negative loglikelihood of a GLM. One approach is based on the LARS algorithm from Efron, and the other one on the local quadratic approximation (LQA) of $\ell(\boldsymbol{\beta})$.

The PFL penalty can be expressed as a generalized lasso penalty (Tibshirani *et al.*, 2011) with matrix $\mathbf{D}$ that selects in each row the j-th and k-th element of the pairwise differences, giving respectively value 1 and -1. The $\mathbf{D}$ matrix will have J columns and a number of rows $m = \binom{J}{2}$ equal to the number of 2-combinations of J elements. For $J = 4$ we get:

$$\mathbf{D}_{J=4} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{D}_{J=5} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \tag{3.23}$$

In other works similar penalties have been proposed, for example She *et al.* (2010) calls it clustered lasso

3.3 Lasso-Type Penalties for Clustering

The grouping property of the pairwise fusion has been exploited also to perform clustering of the statistical units. In this case, instead of penalizing the coefficients of a regression model, the penalty is applied on the differences among centroids. The shrinkage forces pairs of centroids to converge on the same value, producing an implicit representation of clusters through the occurrence of equal centroids. Given a data matrix $\mathbf{Y}$ of size $n \times p$, we assume each row $\mathbf{y}_i$ to be a p-dimensional centroid $\boldsymbol{\mu}_i$ of a singleton. We can express a convex clustering problem as:

$$\min_{\boldsymbol{\mu}} \left\{ \frac{1}{2} \|\mathbf{y}_i - \boldsymbol{\mu}_i\|_2^2 + \lambda \sum_{i<j} w_{ij} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_1 \right\} \tag{3.24}$$

where $i = 1, \ldots, n$, $j = i, \ldots, n$, and w_{ij} are predetermined weights. Pelckmans *et al.* (2005) first proposed convex clustering as a method to perform clustering by solving a convex optimization problem through a LAR-type algorithm. Pan *et al.* (2013) proposed a penalized regression-based clustering (PRclust), using a non-convex truncated lasso penalty (Shen *et al.*, 2012). We also cite the work of Hocking *et al.* (2011) and Lindsten *et al.* (2011). Marchetti *et al.* (2014) proposed the Solution Path Clustering (SPC), which with a blockwise coordinate approach iterates a Majorization-Minimization (MM) step over a set of data-driven tuning parameters in order to minimize an approximation of the objective function under a Minimax Concave Penalty (Zhang *et al.*, 2010). With respect to other techniques, the SPC has the advantage to select automatically the tuning parameters producing a dendrogram-like set of solutions; it does not require the input of the number of clusters, and it is capable of handling and isolating outliers.

3.3.1 The Solution Path Clustering

In Marchetti *et al.* (2014) the underlying model for each object y_i is assumed to be a multivariate Gaussian with mean parameter $\mu_i \in \mathbb{R}^J$ and constant diagonal covariance matrix $\sigma^2 \mathbb{1}_J$. The optimization problem consists in minimizing the criterion function:

$$\Lambda_K(\boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{i \in C_k} \|\mathbf{y}_i - \boldsymbol{\mu}_k\|_2^2 + \lambda \sum_{k<l} n_k n_l \rho\left(\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|_2\right), \tag{3.25}$$

where the n observations are organized in K clusters of size n_k, with centers $\mu_k \in \mathbb{R}^J$ and labels C_k, $k = 1, \ldots, K$. The penalty function $\rho(\cdot)$ is the Minimax Convex Penalty (MCP) developed by Zhang *et al.* (2010):

$$\rho(t) = \left(t - \frac{t^2}{2\lambda\delta}\right) I(t < \lambda\delta) + \left(\frac{\lambda\delta}{2}\right) I(t < \lambda\delta), \quad \text{with} \quad t \geq 0, \tag{3.26}$$

where $I(\cdot)$ is the indicator function. The MCP defines a family of penalties that are concave in $t \in [0, \infty)$, where $\lambda > 0$ controls the amount of regularization and $\delta > 0$ the degree of concavity. Such non-convex penalties promote sparser models than the ℓ_1 penalty with the same or superior prediction accuracy in regression models. MCP includes both the ℓ_1 penalty when $\delta \leftarrow \infty$ and the ℓ_0 penalty when $\delta \leftarrow 0+$, forming a continuum between the two extremes.

Figure 3.4 shows the effect of an increase of λ and δ on the shape of the penalty function $\rho(t)$. In the left panel we keep δ fixed to 1 and make λ vary between 1 and 100, while in the left panel we fix λ to 1 and make δ vary

between 0.001 and 10. Since the tuning parameters appear in the penalty always in the product $\lambda\delta$, it is not surprising to see that the overall effect on the penalty is the same. The function $\rho(t)$ grows with a rate of $1 - 1/\lambda\delta$ until $t = \lambda\delta$ is reached, then it is constant on a value of $\lambda\delta/2$. While λ has a role of controlling the amount of shrinkage, multiplying the entire penalty term, δ only affects the concavity of ρ. Figure 3.5 shows a three dimensional visualization of the penalty as a function of both the tuning parameters for a fix value of $t = 10$. Higher or lower values of t would simply produce a translation of the critical curvature point.

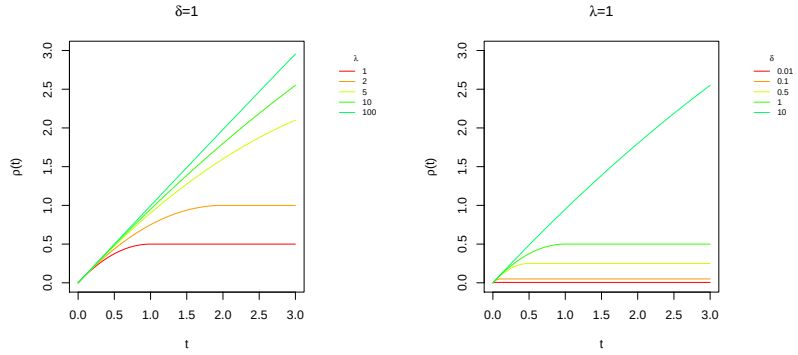

Figure 3.4: Geometrical study of the MCP penalty for growing values of λ and δ keeping the other fixed to 1.

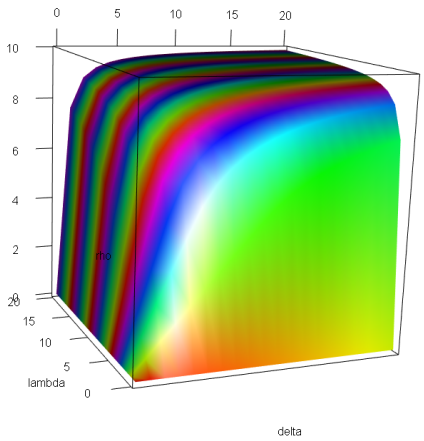

Figure 3.5: Geometrical representation of the MCP penalty in 3 dimensions.

The graphical representation of the penalty function in the space of the parameters $\boldsymbol{\mu}$ is not possible since we are working with pairwise differences.

The MCP penalty is in general non-convex, and this makes the entire criterion function not easy to minimize. Marchetti *et al.* (2014) propose to apply a blockwise MM algorithm, majorizing the penalty term by a linear function and then minimizing the majorizing the resulting function by cyclic coordinate descent. The algorithm is initialized considering all objects to be singleton clusters, $K = n$. It proceeds to gradually merge the objects into a decreasing number of clusters over a sequence of tuning parameters (δ, λ). At each step of the MM algorithm we cycle through $\boldsymbol{\mu}_k$ holding fixed all the other coordinates $\boldsymbol{\mu}_{[-k]} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{k-1}, \boldsymbol{\mu}_{k+1}, \ldots, \boldsymbol{\mu}_K)$ at their current value. We can write the objective function (3.25) isolating the k-th dimension:

$$\Lambda_K(\boldsymbol{\mu}_k) = \sum_{i \in C_k} \|\mathbf{y}_i - \boldsymbol{\mu}_k\|_2^2 + \lambda n_k \sum_{l \neq k} n_l \rho\left(\|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|_2\right). \tag{3.27}$$

Now, differentiating the majorant of Equation (3.27) at t-th step we obtain an update for the k-th cluster centroid $\boldsymbol{\mu}_k^{(t+1)}$ and a corresponding set of weights $w_{kl}^{(t)}$. This process is repeated cycling through the K blocks at each single MM iteration. The algorithm is completed by defining a data-driven fusing threshold ξ. Thresholding is necessary since the penalization proposed by Marchetti *et al.* (2014) only achieves relative sparsity, meaning that $\mu_k^{(t+1)}$ and μ_l can get very close to each other, but in general they will not have the same value, and their differences will not be exactly equal to 0. Two clusters C_k and C_l are merged when $\|\mu_k^{(t+1)} - \mu_l\|_2 < \xi$, and their centroids are set equal to their weighted mean. The algorithm is repeated until it reaches convergence in the cluster centers or until the 50th iteration.

A complete cycle of the MM algorithm returns a single solution that relies on the specification of λ and δ. Marchetti *et al.* (2014) propose the Solution Path Clustering (SPC) algorithm, that consists in repeating the MM algorithm over a set of data-driven values for the tuning parameters, until all the objects are merged into one cluster, $K = 1$. This allows to avoid cross-validation and to build a solution path similar to the one resulting from a hierarchical clustering method.

Once a solution path is produced, the final solution is selected looking at the change in the unconstrained loglikelihood corresponding to a change in the number of clusters. The idea is that as sparsity decreases, and K increases, the unpenalized loglikelihood of the data has to increase. Then, we choose a solution after which an increase in the number of clusters does not correspond to a big increase in the unpenalized log-likelihood.

Chapter 4

A Penalized Fixed-Effects Rasch Model for Clustered Abilities

4.1 The Penalized Fixed-Effects Rasch Model

4.1.1 Definition of the Optimization Problem

In Section (2.2) we pointed out that one limitation of the fixed-effects Rasch model is the direct proportionality between the number of observations and number of parameters to be estimated. The observed vector $\mathbf{y}_r$ of sufficient statistics for the ability $\boldsymbol{\theta}$ has a number m of unique values that depends both on n and J. In fact, the number m is bounded by $1 \leq m \leq J-1$, and it is intuitive to state that the probability of observing a higher variety of response patterns grows with the number of respondents. Here we present an estimation procedure, based on the Solution Path Clustering (SPC) algorithm by Marchetti *et al.* (2014), that allows to estimate a number $K < m$ of different values for $\boldsymbol{\theta}$. The method, penalizing the objective function by the pairwise differences of the abilities, allows for a natural clustering of subject-specific parameters, automatically isolating homogeneous groups of respondents. In Marchetti *et al.* (2014) the underlying model for the data is multivariate Gaussian with constant diagonal covariance matrix. In that case the object of the inference is the mean parameter, and the shrinkage is applied on the distances between group centroids $\boldsymbol{\mu}_k$. Several changes have to be implemented in terms of both theoretical premises and computational details in order to adapt the SPC algorithm to a Rasch model. We have seen that the SPC builds a solution path by minimizing a majorant of the objective function over a set of tuning parameters through cyclic block coordinate descent. In the case of the Rasch model, the object of the inference is the vector $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\beta})'$, and the shrinkage is applied on the distances between scalar parameters θ_k, which represent only a subset of the parameters space. In fact, the difficulty parameters should be invariant with respect to a shrinkage in the differences among abilities during the estimation process. Moreover, working with scalars in our formulation we could refer to differ-

ences instead of distances, and apply absolute values $|\cdot|$ instead of Euclidean norms $\|\cdot\|_2$.

We can write the optimization problem in its most general formulation as

$$\Lambda_p(\boldsymbol{\psi}) = \ell(\boldsymbol{\psi}) + \lambda P(\boldsymbol{\theta}), \tag{4.1}$$

where $\ell(\boldsymbol{\psi})$ is the loglikelihood of the Rasch model and $P(\boldsymbol{\theta})$ the penalty term, depending only on the abilities. The coordinate-wise structure of the method requires us to express the loglikelihood (2.9) of the Rasch model $\ell(\boldsymbol{\psi})$ emphasizing its group structure. So, following the Rasch formulation, the blocks on which the SPC algorithm will cycle are represented by groups C_k of individuals that share the same value θ_k of ability. We define $C_k = \{i : \theta_i = \theta_k\}$, with $k = 1, \ldots, K$, as the group label. The algorithm will start with a number of groups equal to the number of unique values of the sufficient statistics $K = m$, corresponding to the number of different values of the JML solution, and it will stop when reaching $K = 1$ by agglomeration.

We rewrite $\ell(\boldsymbol{\psi})$ as $\ell_K(\boldsymbol{\psi})$, where we separate the n elements of $\boldsymbol{\theta}$ into K blocks, as in Equation (3.25):

$$\ell_K(\boldsymbol{\psi}) = \ell_K(\boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_{k=1}^{K} \sum_{i \in C_k} y_{i\cdot}\theta_k - \sum_{j=1}^{J} y_{\cdot j}\beta_j - \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{J} \log\left(1 + e^{\theta_k - \beta_j}\right). \tag{4.2}$$

Furthermore, we isolate the k-th coordinate, on which the MM algorithm will cycle, and write the loglikelihood as a function of θ_k:

$$\ell_K(\theta_k, \boldsymbol{\beta}) = \sum_{i \in C_k} y_{i\cdot}\theta_k - \sum_{j=1}^{J} y_{\cdot j}\beta_j - n_k \sum_{j=1}^{J} \log\left(1 + e^{\theta_k - \beta_j}\right). \tag{4.3}$$

Problem (4.1) can be written as

$$\Lambda_K(\boldsymbol{\psi}) = \ell_K(\theta_k, \boldsymbol{\beta}) + \lambda P(\theta_k), \tag{4.4}$$

that is very similar to Problem (3.25), apart from the fact that the objective function is not quadratic, and that it depends on a set of parameters that is not involved in the shrinkage procedure.

In order to apply the procedure by Marchetti *et al.* (2014) to a non-quadratic objective function, here we adopt a strategy based on an adjusted dependent variable $\tilde{\theta}_k$ (McCulloch & Neuhaus, 2001). We substitute the

objective (4.3) with a quadratic function of θ_k, representing the distance with its Newton-Raphson estimate $\tilde{\theta}_k$:

$$\tilde{\theta}_k = \theta_k^{(t)} - \left[\frac{\partial^2 \ell_K\left(\theta_k^{(t)}, \boldsymbol{\beta}\right)}{\partial \theta_k^2}\right]^{-1} \frac{\partial \ell_K\left(\theta_k^{(t)}, \boldsymbol{\beta}\right)}{\partial \theta_k}, \tag{4.5}$$

where the first derivative of Equation (4.3) with respect to θ_k is:

$$\frac{\partial \ell_K\left(\theta_k, \boldsymbol{\beta}\right)}{\partial \theta_k} = \sum_{i \in C_k} y_{i\cdot} - n_k \sum_{j=1}^{J} \frac{e^{\theta_k - \beta_j}}{1 + e^{\theta_k - \beta_j}}, \tag{4.6}$$

and its second derivative is given by:

$$\frac{\partial^2 \ell_K\left(\theta_k, \boldsymbol{\beta}\right)}{\partial \theta_k^2} = -n_k \sum_{j=1}^{J} \frac{e^{\theta_k - \beta_j}}{1 + e^{\theta_k - \beta_j}} \left(1 - \frac{e^{\theta_k - \beta_j}}{1 + e^{\theta_k - \beta_j}}\right). \tag{4.7}$$

The resulting optimization problem will be

$$\tilde{\Lambda}_K(\theta_k) = \tilde{\ell}_K\left(\theta_k\right) + \lambda P(\theta_k), \tag{4.8}$$

where $\tilde{\ell}_K\left(\theta_k\right) = \sum_{i \in C_k} \|\theta_k - \tilde{\theta}_k\|_2^2$ is a surrogate function for the likelihood (4.3), and $\tilde{\theta}_k$ takes the role of y_i in the least squares formulation of the problem (3.27). We omit the dependency on $\boldsymbol{\beta}$ in $\tilde{\ell}_K\left(\theta_k\right)$, since the difficulties enter the surrogate objective function only through the Newton-Raphson step, where we compute $\tilde{\theta}_k(\boldsymbol{\beta})$ differentiating Equation (4.3). In this way, the parameter β_j does not appear explicitly in the new formulation of the problem.

We can now write the optimization problem in terms of a Lagrangian function to be majorized and then minimized, sharing the same structure of Equation (3.27):

$$\tilde{\Lambda}_K\left(\theta_k\right) := \sum_{i \in C_k} \|\theta_k - \tilde{\theta}_k\|_2^2 + \lambda n_k \sum_{l \neq k} n_l \rho\left(\|\theta_k - \theta_l\|_2\right), \tag{4.9}$$

where n_k and n_l are respectively the sizes of the k-th and l-th group of individuals with the same ability level, $k, l = 1, \ldots, K$ with $k \neq l$, and ρ is a the MCP penalty function, expressed as in Equation (3.26).

To minimize $-\tilde{\ell}_K\left(\theta_k\right)$ we cycle through θ_k, and at each step we fix $\theta_{[-k]} = \theta_l$ with $l = 1, \ldots, k-1, k+1, \ldots, m$. Being t the iteration index, we define $\theta_k^{(t)}$ as the value of θ_k before the current iteration $t+1$, so that by assumption

$\theta_k^{(t)} \neq \theta_l$. The condition $\theta_k^{(t)} = \theta_l$ is not possible, because in that case the two ability groups would have been already merged in the t-th iteration. Following the process by Marchetti *et al.* (2014) we can now majorize the penalty term in Equation (4.9) by a linear function of θ_k, writing:

$$\begin{aligned}
\tilde{\Lambda}_K(\theta_k) &\approx \sum_{i \in C_k} \|\theta_k - \tilde{\theta}_k\|_2^2 + \lambda n_k \sum_{l \neq k} n_l \left[\rho\left(\|\theta_k^{(t)} - \theta_l\|_2 \right) + \right. \\
&+ \left. \rho'\left(\|\theta_k^{(t)} - \theta_l\|_2 \right) \left(\frac{\|\theta_k - \theta_l\|_2^2 - \|\theta_k^{(t)} - \theta_l\|_2^2}{2\|\theta_k^{(t)} - \theta_l\|_2} \right) \right] = \\
&= \sum_{i \in C_k} \|\theta_k - \tilde{\theta}_k\|_2^2 + \lambda n_k \sum_{l \neq k} n_l \frac{\rho'\left(\|\theta_k^{(t)} - \theta_l\|_2 \right)}{2\|\theta_k^{(t)} - \theta_l\|_2} \|\theta_k - \theta_l\|_2^2 + C = \\
&= \sum_{i \in C_k} \|\theta_k - \tilde{\theta}_k\|_2^2 + \lambda n_k \sum_{l \neq k} w_{kl}^{(t)} \|\theta_k - \theta_l\|_2^2 + C, \qquad (4.10)
\end{aligned}$$

where C includes all the constant terms, not depending on θ_k, while $\theta_k^{(t)}$ is the value of ability for the k-th group at iteration t of the MM algorithm, and $w_{kl}^{(t)}$ can be regarded as an adaptive weight:

$$w_{kl}^{(t)} = N_l \frac{\rho'\left(\|\theta_k^{(t)} - \theta_l\|_2 \right)}{2\|\theta_k^{(t)} - \theta_l\|_2} = N_l \frac{\left(1 - \|\theta_k^{(t)} - \theta_l\|_2 / \lambda\delta \right)_+}{2\|\theta_k^{(t)} - \theta_l\|_2}, \qquad (4.11)$$

where $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ otherwise. We compute the first derivative of Equation (4.10) with respect to θ_k, and verify the first order condition to find the solution $\theta_k^{(t+1)}$ as expressed in (4.14).

$$\frac{\partial \tilde{\ell}_K(\theta_k)}{\partial \theta_k} = 2N_k \left(\theta_k - \tilde{\theta}_k \right) + 2\lambda N_k \sum_{l \neq k} N_l \left(\theta_k - \theta_l \right) w_{kl}^{(t)}, \qquad (4.12)$$

$$\frac{\partial \tilde{\ell}_K(\theta_k)}{\partial \theta_k} = 0 \Rightarrow \theta_k - \tilde{\theta}_k + \lambda \sum_{l \neq k} N_l w_{kl}^{(t)} \theta_k - \lambda \sum_{l \neq k} N_l w_{kl}^{(t)} \theta_l = 0, \qquad (4.13)$$

$$\theta_k^{(t+1)} = \frac{\tilde{\theta}_k + \lambda \sum_{l \neq k} N_l w_{kl}^{(t)} \theta_l}{1 + \lambda \sum_{l \neq k} N_l w_{kl}^{(t)}}. \qquad (4.14)$$

From Equation (4.11) and Equation (4.14) we can notice that:

- When $\|\theta_k^{(t)} - \theta_l\|_2 > \lambda\delta$ then $w_{kl}^{(t)} = 0$ and in the next iteration the ability of the k-th group $\theta_k^{(t+1)}$ will be equal to the Newton-Raphson update $\tilde{\theta}_k$;

- When $\|\theta_k^{(t)} - \theta_l\|_2 \ll \lambda\delta$ then $\theta_k^{(t+1)} \approx \tilde{\theta}_k$ and C_k and C_l will merge.

We cannot directly write $\theta_k^{(t+1)} = \tilde{\theta}_k$ because the optimization procedure only induces relative sparsity, so a thresholding step is still needed. We use as a threshold the quantity ξ, calculated as $\xi = \epsilon \cdot \sigma_{\theta_0}$, where $\epsilon = 10^{-3}$ and σ_{θ_0} is the standard deviation of the initial values of ability θ_0. In Marchetti *et al.* (2014) we have $\xi = \frac{\epsilon}{\sqrt{J}} \sum_{j=1}^{J} \sigma_j$, being σ_j the standard deviation of each column of the data matrix $\mathbf{Y}$. Our adaptation is due to the fact that the penalty is defined on a scalar parameter and not on a J-dimensional vector.

Now that all the quantities of interest have been defined, we can describe one iteration of the modified MM algorithm for a Rasch Model, that is based on Algorithm 1 in Marchetti *et al.* (2014). We add a Newton-Raphson step, to compute $\tilde{\theta}_k$, in-between the majorization step and the minimization step.

Algorithm 1 One iteration of the modified MM algorithm

for $k = 1, \ldots, K$ **do**
  Majorization step: compute weights $w_{kl}^{(t)}$ as in (4.11) for all $l \neq k$
  NR step: compute the estimate $\tilde{\theta}_k$ as in (4.5) for the k-th group
  Minimization step: update $\theta_k^{(t+1)}$ as in (4.14)
  if $\|\theta_k^{(t+1)} - \theta_l\|_2 < \xi$ for some l **then**
    set $\theta_k^{(t+1)}$ and θ_l to their weighted mean $\overline{\theta}_{kl}$
  end if
end for

The complete MM algorithm consists in repeating Algorithm 1 until it reaches convergence in the ability estimates, following the stopping rule (4.15) or until 50 iterations are reached ($t \leq 50$),

$$\max_{1 \leq k \leq K} \|\theta_k^{(t+1)} - \theta_k^{(t)}\|_2 < \xi. \tag{4.15}$$

The quantity ξ is the same used as a threshold for merging clusters. The non-convexity and non-separability of the penalty term does not allow to apply theoretical results on the convergence for coordinate descent algorithms (Hastie *et al.*, 2015), so there is no theoretical guarantee that Algorithm 1 converges to a stationary point (Marchetti *et al.*, 2014).

In order to estimate the difficulty parameters along with the shrunk $\boldsymbol{\theta}$, we compute the classic Newton-Raphson update of $\boldsymbol{\beta}$ right after Algorithm

1, plugging-in the results of the MM step. The difficulty updates are given by:

$$\hat{\beta}_j^{(t+1)} = \beta_j^{(t)} - \left[\frac{\partial^2 \ell\left(\beta_j^{(t)}; \hat{\boldsymbol{\theta}}^{(t+1)}\right)}{\partial \beta_j^2}\right]^{-1} \frac{\partial \ell\left(\beta_j^{(t)}; \hat{\boldsymbol{\theta}}^{(t+1)}\right)}{\partial \beta_j}. \tag{4.16}$$

The first and second derivatives of (4.3) with respect to β_j once $\hat{\boldsymbol{\theta}}^{(t+1)}$ are computed:

$$\frac{\partial \ell\left(\beta_j; \hat{\boldsymbol{\theta}}^{(t+1)}\right)}{\partial \beta_j} = -\left[y_{\cdot j} - \sum_{k=1}^{K^{(t+1)}} \sum_{i \in C_k} \frac{e^{\theta_k^{(t+1)} - \beta_j}}{1 + e^{\theta_k^{(t+1)} - \beta_j}}\right] \tag{4.17}$$

and

$$\frac{\partial^2 \ell\left(\beta_j; \hat{\boldsymbol{\theta}}^{(t+1)}\right)}{\partial \beta_j^2} = -\sum_{k=1}^{K^{(t+1)}} \sum_{i \in C_k} \frac{e^{\theta_k^{(t+1)} - \beta_j}}{1 + e^{\theta_k^{(t+1)} - \beta_j}} \left[1 - \frac{e^{\theta_k^{(t+1)} - \beta_j}}{1 + e^{\theta_k^{(t+1)} - \beta_j}}\right], \tag{4.18}$$

Hence, the complete MM Algorithm for a Rasch model can be written as follows.

Algorithm 2 Modified MM algorithm

1: **repeat**
2: **Algorithm 1** to obtain $\hat{K}^{(t+1)}$ and $\hat{\boldsymbol{\theta}}^{(t+1)} \left\{\hat{\theta}_k^{(t+1)}, \quad k = 1, \ldots, \hat{K}^{(t+1)}\right\}$
3: *Difficulty estimation:* compute $\hat{\boldsymbol{\beta}}^{(t+1)}$ as in (4.16)
4: **until** $\max_{1 \leq k \leq K} \|\theta_k^{(t+1)} - \theta_k^{(t)}\|_2 < \xi$ **or** $t > 50$

4.1.2 The SPC algorithm

A single solution provided by the MM algorithm relies on the specification of the tuning parameters λ and δ. Algorithm 2 is then nested into an outer loop which allows to automatically build a solution path, while computing a set of data-driven values for λ and δ.

To begin, we define a decreasing sequence $\Delta = \{\delta_1, \ldots, \delta_H\}$, with $h = 1, \ldots, H$. For each h we build an increasing sequence $\Lambda(\delta_h) = \lambda_1(\delta_h)$, ..., $\lambda_g(\delta_h)$, ..., $\lambda_G(\delta_h)$, with $g = 1, \ldots, G$. The values of H and G are determined by the algorithm, and the resulting values of λ can be arranged in a matrix that is iteratively filled:

$$\Lambda = \begin{bmatrix} \lambda_1(\delta_1) & \dots & \lambda_g(\delta_1) & \dots & \lambda_G(\delta_1) \\ \lambda_1(\delta_2) & \dots & \lambda_g(\delta_2) & \dots & \lambda_G(\delta_2) \\ \vdots & & \vdots & & \vdots \\ \lambda_1(\delta_H) & \dots & \lambda_g(\delta_H) & \dots & \lambda_G(\delta_H) \end{bmatrix}.$$

The initial values $\lambda_1(\delta_1)$ and δ_1, and the updates for $\lambda_1(\delta_h)$, $\lambda_G(\delta_h)$ and δ_h, are computed following the formulas in Marchetti *et al.* (2014), with some adaptations. Those quantities depend in fact by a series of second-order tuning parameters ω, τ, ϕ and α, all ranging in the interval $(0, 1)$, and that are defined on the basis of two theoretical lemmas. The lemmas in Marchetti *et al.* (2014) can be applied to our case without the need to change them, since we used the surrogated quadratic function, keeping the same structure of the original criterion. While ϕ and α are constant multipliers, ω and $\tau \in (0, \omega)$ are levels of the quantiles Q_ω and Q_τ of the nearest neighbor distances among the initial values of ability θ_0. In the original paper those quantiles are calculated on the distances among the rows of the data matrix.

We start computing the lower bound $\lambda_1(\delta_1)$ for $\Lambda(\delta_1)$:

$$\lambda_1(\delta_1) = \frac{2\phi Q_\omega Q_\tau}{(1-\phi)(Q_\omega - Q_\tau)}, \tag{4.19}$$

then δ_1, the initial value for δ:

$$\delta_1 = \frac{Q_\omega}{\lambda_1(\delta_1)} \tag{4.20}$$

and the upper bound $\lambda_G(\delta_1)$ for $\Lambda(\delta_1)$:

$$\lambda_G(\delta_1) = \left(1 + \delta_1^{-1}\right) \max_{i,j} \|\theta_{0i} - \theta_{0j}\|_2. \tag{4.21}$$

We build a sequence $\Lambda(\delta_1)$ of length G, that is evenly spaced in log-scale between the lower and upper bound. In Marchetti *et al.* (2014) the length G is defined as the minimum between 20 and J. In our case, being $J = 1$, we choose the arbitrary value $G = 5$. The tuning mechanism is driven by δ. Every time the SPC algorithm induces a decrease of δ, according to the product $\delta_h = \delta_{h-1}\alpha$, we also need to update $\Lambda(\delta_h)$:

$$\lambda_1(\delta_h) = \alpha^{-1/2} \lambda_{\tilde{G}}(\delta_{h-1}), \tag{4.22}$$

$$\lambda_G(\delta_h) = \left(1 + \delta_h^{-1}\right) \max_{i,j} \|\theta_{0i} - \theta_{0j}\|_2, \tag{4.23}$$

where $\lambda_{\tilde{G}}(\delta_{h-1})$ is the current value of the sequence $\Lambda(\delta_{h-1})$ in the moment when δ decreases. Finding the right value for the parameter ω is crucial, since $\lambda_1(\delta_1)$ and δ_1 depend directly on Q_ω. A small initial value of δ could lead to unclustered solutions, on the other hand a large initial value of λ could skip the right solution. The distance between ω and τ determines the scale of $\lambda_1(\delta_1)$.

The decrease of δ, together with the length H of the sequence Δ and the number of rows of the matrix Λ, is controlled by a rule on the bias-variance ratio BVR_k computed within each cluster: if the $\mathrm{BVR}_k > 1$ for any k then δ has to be decreased. The latest λ value $\lambda_{\tilde{G}}$ will then be used to compute the new $\lambda_1(\delta_h)$. We define BVR_k as:

$$\mathrm{BVR}_k = \begin{cases} \frac{\|\theta_k - \overline{\theta_{0k}}\|_2^2}{\sum_{i \in C_k} \|\theta_{0i} - \overline{\theta_{0k}}\|_2^2/(n_k-1)}, & \theta_{0i} \neq \overline{\theta_{0k}} \quad \text{for some} \quad i \in C_k \\ \\ \frac{\|\theta_k - \theta_{0i}\|_2^2}{\left(\min_{l \neq k} \|\theta_{0i} - \theta_k\|_2/2\right)^2}, & \theta_{0i} = \overline{\theta_{0k}} \forall i \in C_k \end{cases} \tag{4.24}$$

where θ_k is the estimated ability level of the k-th group, $\overline{\theta_{0k}}$ is the mean of the initial ability levels corresponding to the units in the k-th group, and θ_{0i} is the initial ability level for the i-th subject in C_k. The rule for the computation of BVR_k is different in Marchetti *et al.* (2014), and it changes on the occurrence of one-component clusters. In the Rasch model framework we are not very likely to find singletons in the initial values of ability θ_0, so a simple rule based on $n_k > 1$ is not fit anymore. Adapting the formulas, we compute the upper ratio in (4.24) when in the current cluster at least one value of the initial abilities is different from their mean (this happens when two clusters are merged), and the lower ratio when all the initial abilities in the k-th cluster are equal to their mean (no merging has occurred).

Once the tuning rule has been defined, we can write the full SPC algorithm as in the following box.

Algorithm 3 Solution Path Clustering

Inputs
Required inputs: Y, $\omega \in (0,1)$, $\boldsymbol{\theta}_0$, $\boldsymbol{\beta}_0$
Default inputs: $\tau = 0.9\omega$, $\phi = 0.5$, $\alpha = 0.9$, $G = \min(20, J)$
Initialization: $h = 1$, $K = m$, $\theta_k = \theta_{0k}, k = 1, \ldots, m$

1: **repeat**
2: compute δ_h and $\Lambda(\delta_h)$
3: **for** $g = 1, \ldots, G$ **do**
4: run the **MM Algorithm** to get $K(h,g)$, $\hat{\boldsymbol{\theta}}(h,g)$ and $\hat{\boldsymbol{\beta}}(h,g)$
5: **for** $k = 1, \ldots, K(h,g)$ **do**
6: compute BVR_k
7: **if** $\text{BVR}_k > 1$ **then**
8: $h \leftarrow h + 1$ and **go to** line 2
9: **end if**
10: **end for**
11: **end for**
12: $h \leftarrow h + 1$
13: **until** $K(h,g) = 1$

At each step h of the SPC we compute a value for δ_h and a sequence of G values for λ. The algorithm cycles over g, computing the MM solution for each combination (δ_h, λ_g). Whenever $BVR_k < 1$ we decrease the value of δ and compute a new sequence $\Lambda(\delta_h)$. Each solution is used as warm start for the next one, building gradually the solution path. The algorithm stops when all the respondents are assigned to a single ability level $K = 1$.

Once the solution path is produced, a solution selection method must be implemented. As in hierarchical clustering we want to find the optimal point where to cut the solution tree. Marchetti *et al.* (2014) suggest to use the difference ratio dr, calculated after sorting the solution path according to an increasing number of clusters. For every adjacent pair of solutions we have:

$$dr^{(s,s+1)} = \frac{\ell_{K^{(s+1)}} - \ell_{K^{(s)}}}{K^{(s+1)} - K^{(s)}}, \tag{4.25}$$

where ℓ_K is the same of Equation (3.25) and the solution are sorted so that the denominator is always greater or equal to 1. We will choose the solution indexed by

$$K^* = \max\left\{K^{(s)} : dr^{(s,s+1)} \geq a \times \max\left(dr^{(1,2),\ldots,dr^{(S-1,S)}}\right)\right\} \tag{4.26}$$

with $a = 0.05$.

4.2 An Alternative Approach: Classification Likelihood

A competing method that allows to cluster fixed effects, while estimating them, can be implemented using a classification likelihood (CL) approach (Fraley & Raftery, 1998). This method consists in computing the likelihood of the Rasch model assuming that the abilities are aggregated one at the time forming all the possible combinations of clusters. At each step we choose the combination that produces the maximum value of maximum likelihood. The current solution is used as warm start for the next step, and the algorithm stops when all the individuals are associated with the same ability level.

To define formally the CL algorithm we write the loglikelihood (2.10) as a function of a vector $\boldsymbol{\psi}^K = \left(\boldsymbol{\theta}^K, \boldsymbol{\beta}\right)'$, where $\boldsymbol{\theta}^K$ is the vector of the K unique values in $\boldsymbol{\theta}$, containing the ability levels θ_k of each group, $k = 1, \ldots, K$:

$$\ell^K\left(\boldsymbol{\psi}^K\right) = \mathbf{y}_r'\mathbf{A}\boldsymbol{\theta}^K - \mathbf{y}_c'\boldsymbol{\beta} - \mathbf{s}_K' \log\left(1 + e^{\boldsymbol{\theta}^K \mathbb{1}_J' - \mathbb{1}_K \boldsymbol{\beta}'}\right) \mathbb{1}_J, \tag{4.27}$$

where $\mathbf{A}$ is a $n \times K$ selection matrix, resulting from the dummy coding of the group structure, and $\mathbf{s}_K$ is a vector of the group sizes, $\mathbf{s}_K = (n_1, \ldots, n_K)'$. The gradient and the Hessian of the loglikelihood (4.27) are given by:

$$\begin{aligned}
\nabla_{\boldsymbol{\psi}^K} &= \begin{bmatrix} \left(\mathbf{y}_r'\mathbf{A}\right)' - \operatorname{diag}\left(\mathbf{s}_K\right)\mathbf{P}\mathbb{1}_J \\ -\mathbf{y}_c + \mathbf{P}'\mathbf{s}_K \end{bmatrix}, \qquad (4.28) \\
H_{\boldsymbol{\psi}^K} &= \begin{bmatrix} \operatorname{diag}\left\{-\operatorname{diag}\left(\mathbf{s}_K\right)\left[\mathbf{P}\left(1-\mathbf{P}\right)\right]\mathbb{1}_J\right\} & \operatorname{diag}\left(\mathbf{s}_K\right)\left[\mathbf{P}\left(1-\mathbf{P}\right)\right] \\ \left[-\operatorname{diag}\left(\mathbf{s}_K\right)\mathbf{P}\left(1-\mathbf{P}\right)\right]' & \operatorname{diag}\left\{\left[\mathbf{P}\left(1-\mathbf{P}\right)\right]'\operatorname{diag}\left(\mathbf{s}_K\right)\right\} \end{bmatrix},
\end{aligned}$$

where $\operatorname{diag}\left(\mathbf{s}_K\right)$ is a $K \times K$ diagonal matrix, with $\mathbf{s}_K$ as main diagonal, $\mathbf{P}$ is a $K \times J$ matrix of elements:

$$\mathbf{P} = \begin{pmatrix} p_1(\theta_1) & p_2(\theta_1) & \cdots & p_J(\theta_1) \\ p_1(\theta_2) & p_2(\theta_2) & \cdots & p_J(\theta_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\theta_K) & \cdots & \cdots & p_J(\theta_K) \end{pmatrix}, \tag{4.29}$$

being the ICC expressed in terms of θ_k, $p_j(\theta_k) = p(Y_{ij} = 1|\theta_k) = \frac{e^{\theta_k - \beta_j}}{1+e^{\theta_k - \beta_j}}$, and being $\left[\mathbf{P}\left(1-\mathbf{P}\right)\right]$ the matrix of the element-wise products $p_j\left(\theta_k\right)\left[1 - p_j\left(\theta_k\right)\right]$.

The $\mathbf{A}$ matrix projects the problem from a number n to a number K of units to be grouped. The algorithm merges two groups at the time, equating two values of θ_k, so that at each step we reduce the length of vector $\boldsymbol{\theta}^K$ and the number of columns of $\mathbf{A}$. For each value of K we have $\binom{K}{2}$ possible configurations of $K - 1$ groups. Each configuration, indicated by h, corresponds to a matrix $\mathbf{A}_h$, with $h = 1, \ldots, K$. We manage the labeling of groups

by defining the $K \times K$ matrix M^K of combinations for K values, and the $n \times K$ matrix $\mathbf{C}^K$ of the corresponding individual labels. The initialization of the parameters $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ are computed as in Equation (2.18). As for the SPC, we start with a value $\tilde{K} = m$ according to which we compute the initialization for the labeling matrix $\mathbf{C}^m$. We can describe the algorithm as follows:

Algorithm 4 Classification Likelihood Algorithm

Inputs
Required inputs: $\mathbf{Y}$
Initialization: $\boldsymbol{\theta}^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $K(0) = m$, $t = 0$

repeat
  compute the combination matrix $\mathbf{M}^{K(t)}$ of $K(t) - 1$ groups
  compute the corresponding class memberships matrix $\mathbf{C}^{K(t)}$
  for $h = 1, \ldots, \binom{K(t)}{2}$ **do**
    compute $\tilde{\boldsymbol{\theta}}_h^{K(t+1)}$ as the mean of $\tilde{\boldsymbol{\theta}}_h^{K(t)}$in the $K(t) - 1$ groups
    compute $\nabla_{\boldsymbol{\psi}^{K(t)}}$ and $H_{\boldsymbol{\psi}^{K(t)}}$
    compute $\hat{\boldsymbol{\theta}}_h^{K(t+1)} = \tilde{\boldsymbol{\theta}}_h^{K(t+1)} - \nabla_{\boldsymbol{\psi}^{K(t)}} \left[H_{\boldsymbol{\psi}^{K(t)}} \right]^{-1}$ and $\hat{\boldsymbol{\beta}}_h^{K(t+1)}$
    compute the maximum likelihood $\ell^K(\hat{\boldsymbol{\psi}}_h^{K(t+1)})$
  end for
  select $\hat{\boldsymbol{\theta}}_*^{K(t+1)}$ as the value of $\hat{\boldsymbol{\theta}}_h^{K(t+1)}$ for which $\ell^K(\boldsymbol{\psi}_h^{K(t+1)})$ is max
  $t = t + 1$
  update $K(t) = K(t-1) - 1$
until $K = 1$

At each iteration of the CL algorithm we cycle through the rows of the $\mathbf{M}^K$ matrix. Each row h corresponds to a different group structure and it generates a set of maximum likelihood solutions. Once that all the $\binom{K}{2}$ solutions at the current t are computed, we choose the one that generates the highest value of maximum likelihood. We update the number of clusters, use the solution as a warm start for the next iteration and repeat until $K = 1$.

This algorithm can be seen as a hierarchical unsupervised classification method with a clustering criteria based on maximum likelihood; so it is possible to arrange the solution into a solution path directly comparable with the one of the SPC.

4.3 Real Data Example: INVALSI data

In this section we present an example on real data in order to better understand the grouping properties of the Solution Path Clustering. We use data drawn from the INVALSI mathematics test, administered in June 2009, and available with Bartolucci *et al.* (2015) book. The INVALSI is the Italian Institute for the Evaluation of the Education System. Among its many research and administration activities, the INVALSI has the main purpose to measure the performance of the Italian Education System by means of standardized tests. Since the school year 2008-09, extensive questionnaires are administered to to primary, lower-middle, and high school students to investigate their proficiency in Italian language and in mathematics, with other collective and individual characteristics. The INVALSI test on Italian language includes two sections, a reading comprehension section and a grammar section. The reading comprehension skills are measured by 30 items, which require students to demonstrate a range of abilities in constructing meaning from the two written texts. The grammar section is made up of 10 items, which measure the ability of understanding the morphological and syntactic structure of sentences within a text. The INVALSI mathematics test consists of 27 items covering four main content domains: numbers, shapes and figures, algebra, and data and previsions. The number content domain consists of understanding (and operating with) whole numbers, fractions and decimals, proportions, and percentage values. The algebra domain requires students the ability to understand, among others, patterns, expressions, and first order equations, and to represent them through words, tables, and graphs. The shapes and figures domain covers topics such as geometric shapes, measurement, location, and movement. The data and previsions domain includes three main topic areas: data organization and representation (reading, organizing, and displaying data using tables and graphs), data interpretation (identifying, calculating, and comparing characteristics of datasets, including mean, median, mode), and chance (e.g., judging the chance of an outcome, using data to estimate the chance of future outcomes). All items of the reading, grammar, and mathematics dimensions of the INVALSI tests are of multiple choice type, with one correct answer and three distractors, and are dichotomously scored (assigning 1 point to correct answers and 0 otherwise). However, the mathematics test contains also two open questions for which a partial score of 1 was assigned to partially correct answers and a score of 2 was given to correct answers. For the purposes of the analyses described in the following, the open questions of the mathematics test were dichotomously rescored, giving 0 point to incorrect answers and 1 point otherwise (Bartolucci *et al.*, 2015). The dataset we are going to analyze is referred in

the book as *INVALSI reduced dataset*, and it contains the answers of 1786 male students coming from the Center of Italy to the set of 27 binary items of the test in mathematics. In Figure 4.1 we give a graphical representation of the data matrix, after the deletion of 58 observations for the students that endorsed all the items. The new data matrix is a 1728×27 binary matrix, that can be represented by an image where the gray rectangles correspond to an endorsed item $y_{ij} = 1$, the white ones to $y_{ij} = 0$. Figure 4.1 shows also the barplots of the row and column scores, measuring respectively the number $y_{i\cdot}$ of correct answers of each student at all the questionnaire items, and the number $y_{\cdot j}$ of correct answers of all the students to each item.

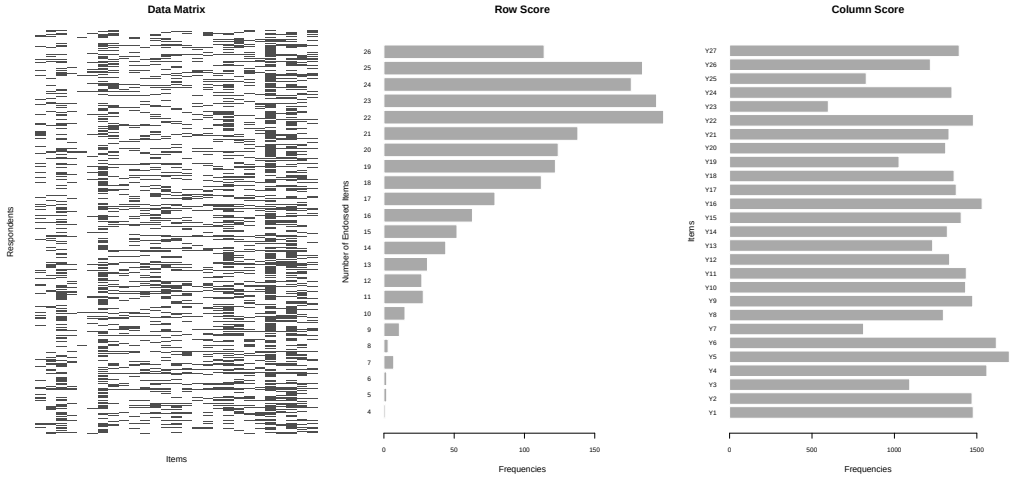

Figure 4.1: INVALSI dataset example: visualization of the data.

We can see that the empirical distribution of the row score $y_{i\cdot}$ is strongly asymmetric and it does not exhibit a clear group structure; there are no students giving only 1, 2 or 3 correct answers, and half of the respondents endorsed more than 21 items. Concerning the column score, the empirical distribution is quite uniform, and Item 23 appears to be the most difficult, since only 35% of the students endorsed it. Given the data, we compute the estimated abilities and difficulties under three different models:

- A simple Rasch model, using the R package `ltm` (Rizopoulos, 2006), where we assume the latent ability to be normally distributed;
- A latent class Rasch model with $K = 4$, using the R package `MultiLCIRT` (Bartolucci *et al.*, 2016), where the ability is a random effect with a discrete distribution. Here we choose the optimal number of latent classes

on the basis of the BIC criterion as observed in Bartolucci *et al.* (2015) on the same data;

- A penalized fixed-effects Rasch model with the proposed SPC algorithm, where we are not making any distributional assumption on the latent ability or the number of latent classes.

The results are presented in the following figures and tables. Figure 4.2 shows the solution path of the SPC algorithm on the INVALSI data, with $\omega = 0.5$. The selection of ω has been made, at this stage, heuristically: on a grid of ω values we selected the path that better adapted to the needs of our analysis. As mentioned in the previous section, the value of ω can be seen as a measure of the amount of units that are merged in the first step of the algorithm. A low value will slow down the algorithm, while a high value produces a thinner solution paths, with a stronger initial aggregation. The algorithm starts with 23 initial values of the estimated abilities, according to the unpenalized Rasch model solutions, which are reported in the first column of Table 4.2. It reduces the abilities to a single group in 17 steps. Using the difference ratio rule, defined in Equation (4.26), the selected solution is the 16-th, with $K = 4$. Table 4.1 contains the difference ratios and the quantities for their computation, as described in Equation (4.25).

K	17	15	13	10	5	4	1
lik.	-19678.65	-19685.62	-19696.77	-19745.47	-20234.88	-20271.01	-23172.12
dr		(15 to 17)	(13 to 15)	(10 to 13)	(5 to 10)	(5 to 4)	(4 to 1)
	-	3.48	5.58	16.24	97.88	36.13	967.04

Table 4.1: INVALSI dataset example: selection of the SPC solutions with the difference ratio criterion.

In Figure 4.3 we compare the distribution of the latent ability under the normality assumption with the results from the latent class Rasch model and the SPC. All the distributions have been centered to 0.

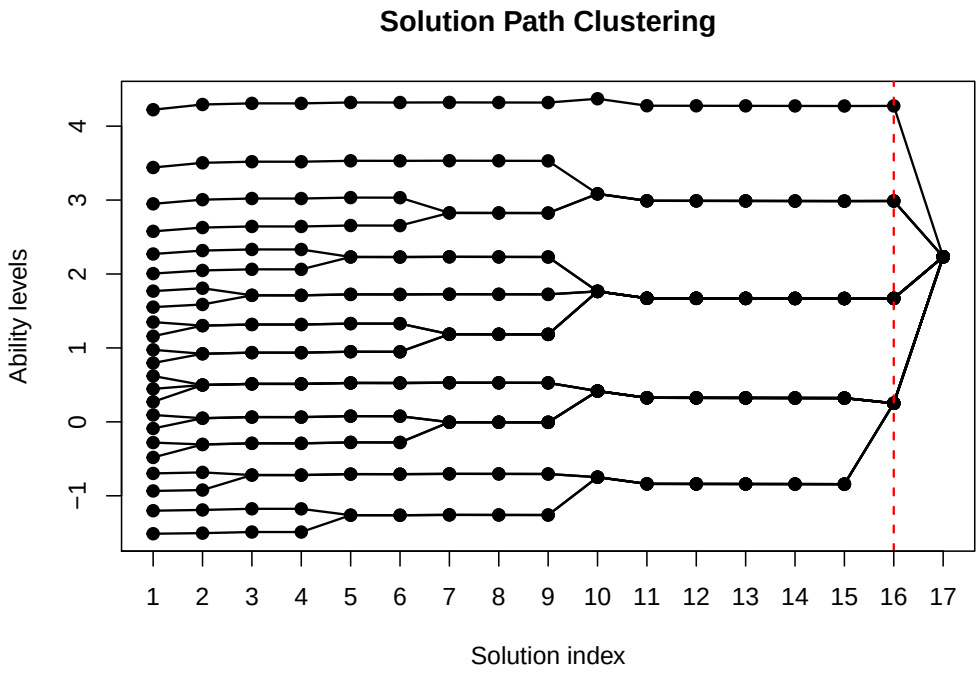

Figure 4.2: INVALSI dataset example: solution path of the SPC with $\omega = 0.5$ on the INVALSI reduced dataset.

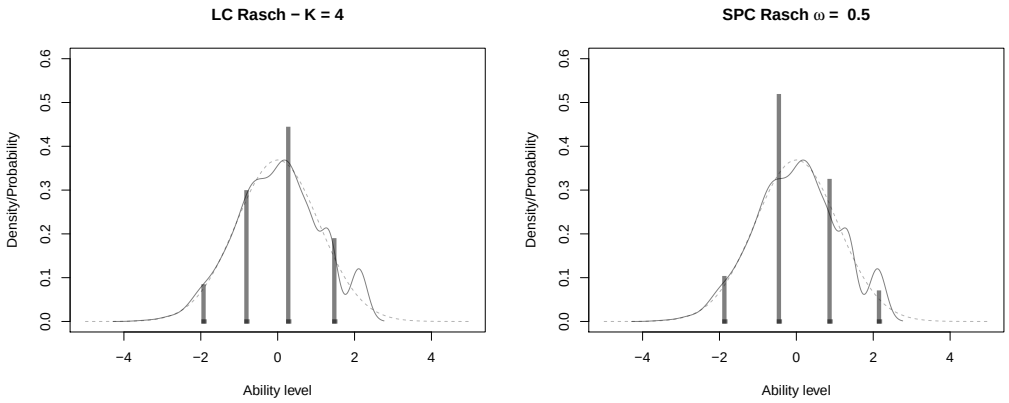

Figure 4.3: INVALSI dataset example: estimated distribution of the ability under the Rasch model with the assumption of normality (continuous lines) and discreteness (vertical bars) by latent classes (LC) or by sparsification (SPC).

Rasch			LC-Rasch			SPC-Rasch		
k	$\hat{\theta}_k$	n_k	k	$\hat{\theta}_k$	n_k	k	$\hat{\theta}_k$	n_k
1	-1.51	1	1	0.1979	138	1	0.2512	171
2	-1.20	2	2	1.3140	502	2	1.6697	889
3	-0.94	2	3	2.4020	760	3	2.9875	554
4	-0.70	7	4	3.600	328	4	4.2735	114
5	-0.48	3						
6	-0.28	11						
7	-0.09	15						
8	0.09	28						
9	0.27	27						
10	0.45	31						
11	0.62	44						
12	0.80	52						
13	0.97	63						
14	1.16	79						
15	1.35	112						
16	1.55	122						
17	1.77	124						
18	2.00	138						
19	2.27	199						
20	2.58	194						
21	2.95	176						
22	3.44	184						
23	4.22	114						

Table 4.2: INVALSI dataset example: estimated ability levels and their frequencies for the Rasch model, the LC Rasch model and the SPC with $\omega = 0.5$.

	Rasch	LC-Rasch	SPC-Rasch
j	$\hat{\beta}_j$	$\hat{\beta}_j$	$\hat{\beta}_j$
1	0.0000	0.0000	0.0000
2	0.0362	0.0362	0.0688
3	1.4204	1.4193	1.4600
4	-0.5058	-0.5056	-0.4732
5	-2.3620	-2.3618	-2.3261
6	-0.9944	-0.9941	-0.9613
7	2.1947	2.1929	2.2553
8	0.7686	0.7681	0.8028
9	0.0156	0.0156	0.0482
10	0.2254	0.2253	0.2582
11	0.2069	0.2068	0.2397
12	0.6288	0.6284	0.6625
13	0.9913	0.9906	1.0266
14	0.6812	0.6808	0.7151
15	0.3460	0.3458	0.3790
16	-0.3098	-0.3097	-0.2774
17	0.4718	0.4715	0.5051
18	0.5241	0.5238	0.5575
19	1.6068	1.6055	1.6496
20	0.7180	0.7176	0.7520
21	0.6439	0.6435	0.6776
22	-0.0052	-0.0052	0.0273
23	2.7845	2.7843	2.8797
24	0.5751	0.5748	0.6086
25	2.1515	2.1498	2.2104
26	1.0370	1.0363	1.0727
27	0.3973	0.3970	0.4303

Table 4.3: INVALSI dataset example: estimated difficulties for the Rasch model, the LC Rasch model and the SPC with $\omega = 0.5$.

Both the SPC algorithm and the latent class Rasch model produce an optimal solution characterized by 4 latent ability groups. Nevertheless, the values of the ability estimates, and the distribution of respondents inside the latent classes slightly differ in the results of the two methods. The sensitivity of the SPC to outliers leads to a solution in which the cluster of respondents endorsing 26 items stays completely separated from the rest. We can see how in Figure 4.2 the highest ability level is not aggregated to any other until step 17, when $K = 1$. In this cluster we find students characterized by a very high ability level in mathematics. On the other hand, the LC Rasch model incorporates these individuals in a larger upper latent class, associated with a lower support point, systematically underestimating their ability. Figure 4.3 compares the distribution of the ability estimates under the normality assumption with the distribution of the discrete latent ability estimated with the two methods, represented by vertical bars. Both solutions resemble the continuous one. This is quite natural for the SPC, whose solutions are the result of a hierarchical aggregation. Analyzing the frequencies in Table 4.2 we see that the values of the discrete ability estimates, under both the random and fixed effect approach, are coherent with a sequential aggregation of the solutions in the first column. Isolating the 114 individuals with initial ability level $\hat{\theta}_{23} = 4.22$, the SPC finds the modal class corresponding to a value $\hat{\theta}_2 = 1.67$ lower with respect to the mode in the LC case. Table 4.3 shows the estimates for the difficulty parameters. The βs are slightly influenced by the shrinkage of the abilities in the SPC solution, exhibiting a small constant additive increment. The effect of this increment does not change the relative differences between items, nor influence their interpretation. Item 23, item 7 and item 25 are confirmed to be the three most difficult.

However the Rasch model is too restrictive for these data, and the need for the inclusion of the discriminant indices in well justified inBartolucci *et al.* (2015).

4.4 Simulation Study

In order to test the properties of the proposed method and to compare it with alternative approaches, we performed a simulation study. We simulated data from a latent class Rasch model in 27 scenarios, obtained combining a number of respondents n equal to 250, 500 or 1000 individuals, a number of items J equal to 10, 20 or 50, and three different discrete symmetrical distributions for the true $\boldsymbol{\theta}_0$, considering values ranging in the $[-2, 2]$ interval.

We refer to:

- A 2-classes structure with $\boldsymbol{\theta}_0 = (-2, 2)$ and probability $(0.5, 0.5)$;
- A 3-classes structure with $\boldsymbol{\theta}_0 = (-2, 0, 2)$ and probability $(0.3, 0.4, 0.3)$;
- A 4-classes structure with $\boldsymbol{\theta}_0 = (-2, -0.667, 0.667, 2)$ and probability $(0.2, 0.3, 0.3, 0.2)$.

The true values $\boldsymbol{\beta}_0$ of the difficulty parameters are fixed in the same range $[-2, 2]$ of the abilities, with growing equispaced values from item 1 to item J. In each K-classes structure, we indicate scenarios with letters from A to I, corresponding to combination of n and J as in Table 4.4.

	n		
J	250	500	1000
10	A	D	G
20	B	E	H
50	C	F	I

Table 4.4: Simulation study: scenarios.

In all the above scenarios we compare the latent class Rasch model (LC in the following), the classification likelihood approach (CL in the following), as described in the previous section, and the SPC with $\omega = 0.9$. In the case of the LC model we assume to know the true number of latent classes, so this method has an a propri advantage in terms of information. We choose a high value of ω in order to reduce the computation time and to produce a more linear and simple path.

Figure 4.4, Figure 4.5 and Figure 4.6 report the boxplots of ability estimates obtained with the SPC algorithm in the 9 scenarios, respectively under the 3 latent structures. The boxplots are organized in triplets with the same

number of items and growing sample size, and they allow us to get an insight on the effect of a change in the size of the data on the estimates of the ability parameters. The case $K = 2$, in Figure 4.4, appears to be the one in which the algorithm performs better. As expected, in this case an increase of the sample size always produces a decrease in the variability of the estimates of $\hat{\theta}_1$ and $\hat{\theta}_2$. In the same way, an increase in the number of items produces a reduction of the bias: the last triplet of boxplots (for scenarios C, F and I) is closer to the horizontal red lines corresponding to the true value of the parameter.

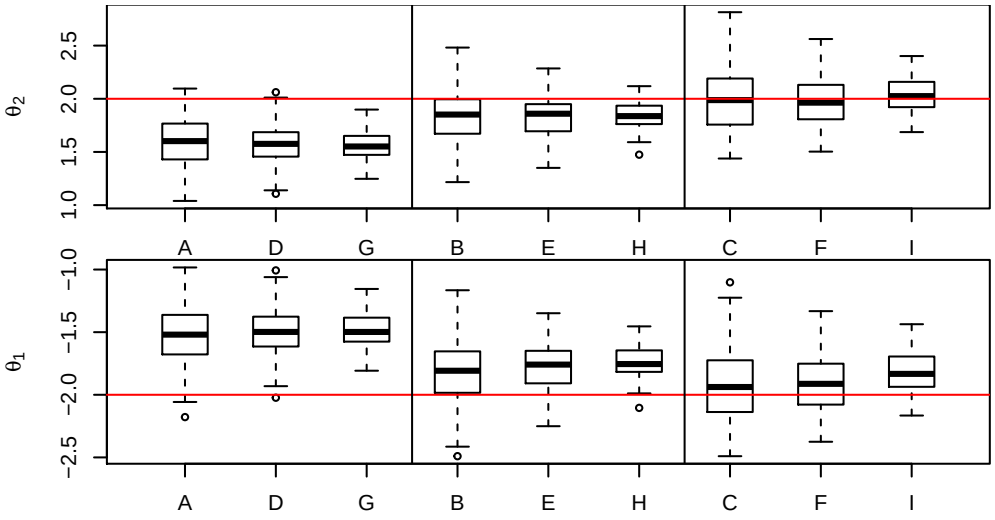

Figure 4.4: Simulation study: solutions of the SPC with $\omega = 0.9$ for $K = 2$ (boxplots in 9 scenarios grouped by number of items with increasing sample size).

Figure 4.5 shows the same configuration of boxplots for $K = 3$. Here we can see that a good behavior of the parameter estimates is attained only by the last triplets, the ones corresponding to scenarios with 50 items. In other cases we see a strong presence of outliers, extreme estimates, and globally a higher variability, in particular for scenarios A, D and H. The entity of these problems seems to intensify when the SPC algorithm deals with $K = 4$ in Figure 4.6. The estimates for θ_1 and θ_4 in this case tend to be very high in absolute value, accentuating the separation of the highest and lowest ability classes, from the ones in the middle, θ_2 and θ_3. Scenarios B, E and H show a systematic presence of outliers, while scenarios C, F and I exhibit high variability in particular for θ_1 and θ_4.

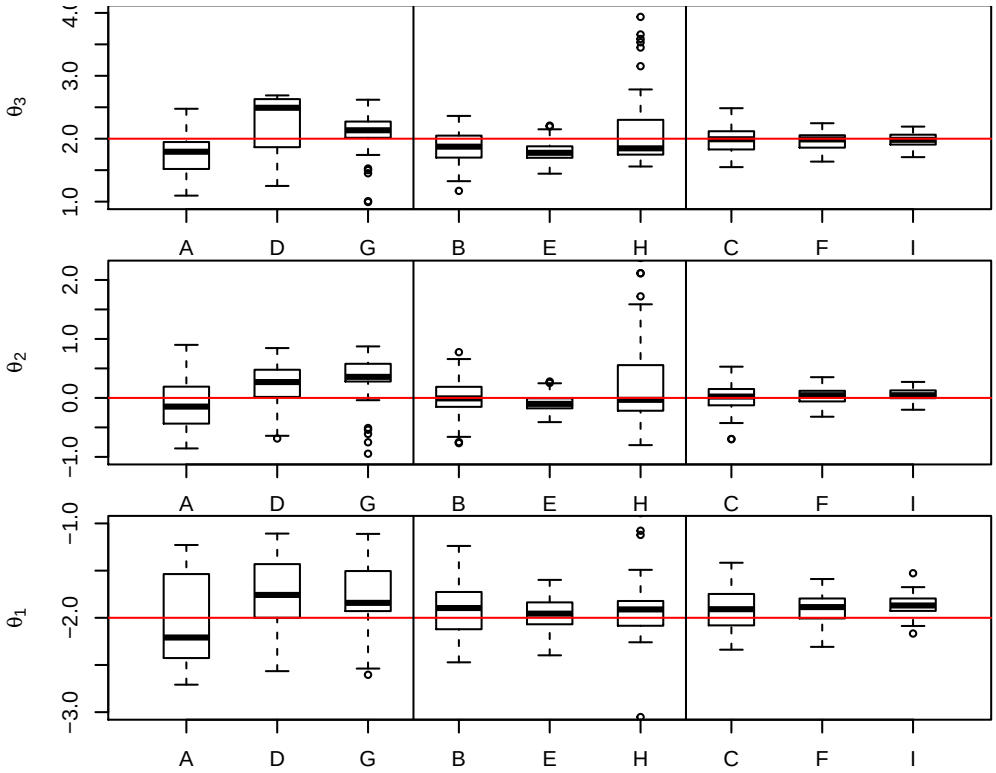

Figure 4.5: Simulation study: solutions of the SPC with $\omega = 0.9$ for $K = 3$ (boxplots in 9 scenarios grouped by number of items with increasing sample size).

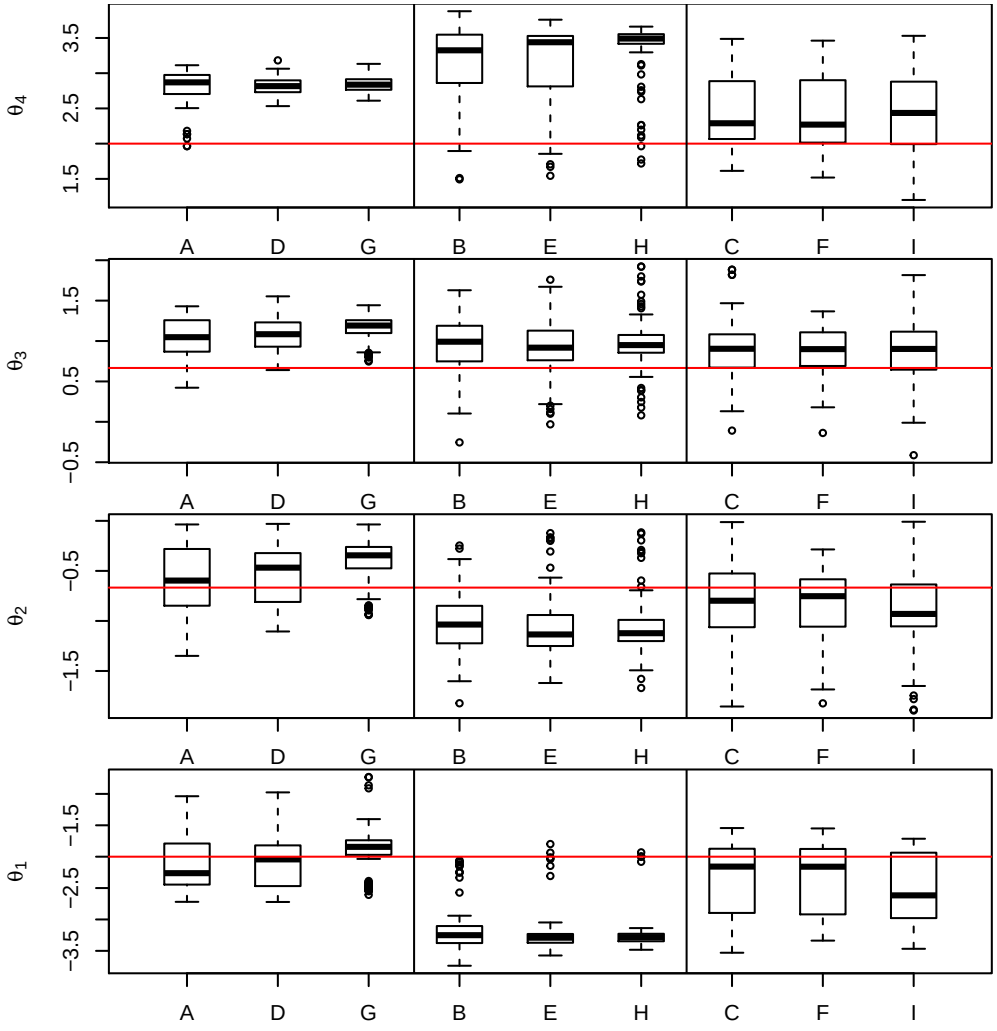

Figure 4.6: Simulation study: solutions of the SPC with $\omega = 0.9$ for $K = 4$ (boxplots in 9 scenarios grouped by number of items with increasing sample size).

These problems are connected to the selection of the tuning parameter. We used a single value of $\omega = 0.9$ throughout all the simulation study, not changing or adapting the tuning parameter in the different scenarios. Such a high value of ω induces a strong shrinkage in the first step of the algorithm, merging approximately the 90% of the initial values. This choice fastens the clustering process, but at the same time may cause it to skip the right solution, aggregating the parameters over values that are far from the true ones, without the possibility of a subsequent cluster splitting. Besides, it is not guaranteed that the solution path contains the true number of classes, for example the SPC algorithm, in a scenario with true $K = 4$, may marge directly 5 clusters into 3 ones, skipping the solution with 4 groups. A lower value of ω, producing a slower aggregation, may be more indicated when we suspect the number of true classes to be relatively large. This may explain why in the $K = 2$ case the algorithm appears to perform better than with $K = 3$ and $K = 4$. Another issue concerns the uniformity in the empirical distribution of the initial values, and the number m of different values observed for the sufficient statistics $y_{i\cdot}$. With a high value of ω and a low value m we face a high risk to skip the right solution, because of the strong initial aggregation. On the other hand, a high value of ω and a very high value of m can produce a situation in which the result of a strong aggregation at the first step may differ from replication to replication. This is due to fact that the initial values are many and very closed to each other. Different initial aggregation patterns may produce a different path and a higher variability of the final estimates. This appears to be the case of scenarios C, F and I, in the case of $K = 4$. The implications of the value of ω over the bias and the variability of the estimates has to be further inquired.

		Scenarios								
K		A	B	C	D	E	F	G	H	I
2	LC	0.2708	0.0958	0.0774	0.2603	0.0766	0.0541	0.2578	0.0588	0.0324
2	CL	0.2726	0.0960	0.0837	0.2646	0.0768	0.0620	0.2635	0.0693	0.0096
2	SPC	0.2598	0.0964	0.0821	0.2518	0.0774	0.0602	0.2483	0.0604	0.0448
3	LC	0.3267	0.0440	0.0440	0.3145	0.0458	0.0203	0.3181	0.0404	0.0124
3	CL	0.2058	0.0678	0.0563	0.3143	0.0527	0.0320	0.3214	0.0488	0.0234
3	SPC	0.1837	0.0833	0.0594	0.2233	0.0419	0.0235	0.1873	0.6520	0.0168
4	LC	0.4893	0.0898	0.0344	0.4894	0.0638	0.0201	0.4709	0.0491	0.0117
4	CL	0.3034	0.1036	0.0499	0.2830	0.1016	0.0296	0.3161	0.0556	0.0231
4	SPC	0.2960	0.8784	0.3117	0.2793	0.9402	0.2825	0.3152	1.0059	0.3309

Table 4.5: Simulation study: mean squared error of the estimated abilities of LC, CL and SPC with $\omega = 0.9$ in 27 scenarios.

Concerning the performance of the proposed method compared with the latent class Rasch model (LC) and the classification likelihood (CL), we show in Table 4.5 the mean squared errors computed over all the ability estimates for the 3 alternative methods in each scenario. The mean squared errors of the single θ_k parameters can be found in Table 1 of the Appendix, together with the boxplots of the estimated abilities in each scenario in Figure 1, Figure 2 and Figure 3.

Recalling some of the considerations just made for the single SPC solutions, we can see that the 3 methods perform similarly when $K = 2$. In some cases the mean squared error obtained with the SPC even dominates the other methods (in scenarios A, D, F and G). The 2-classes structure is the easiest to detect with a high value of ω, because the clusters are well separated and, independently from the initial aggregation, the algorithm convergences to the right solution.

For $K = 3$ the SPC performs well in terms of accuracy in all the scenarios apart from scenario H, that is characterized by a high variance of the estimates, in particular for θ_2. This irregular performance need to be specifically addressed. However, the SPC outmatches CL and LC in 4 scenarios (A, D, E, and G). With $K = 4$ it is clear that the SPC with $\omega = 0.9$ has a tendency to the extremization of the outer ability levels, we refer in particular to scenarios B, E and H in Figure 3 in the Appendix. The value of θ_4 is generally estimated to be greater than 2 (which is the true value the parameter). The same happens on the opposite direction for θ_1 . This tendency to the separation of the greater and lower ability levels may be seen as an advantage in those cases when the LC model tends to overlay the distributions of the latent classes, exhibiting a tendency towards the central value, as it happens in scenarios A, D and G, but in some cases it leads to an unacceptable amount of bias.

Globally, we can see that the SPC always outmatches the other methods in scenarios with $J = 10$ (A, D and G). We expected the accuracy of the SPC estimates to grow with the number of items, and this is confirmed to be always true for $K = 2$ and in all $K = 3$ scenarios apart from H.

Given the high variability of the SPC estimates in some scenarios for $K = 3$ and $K = 4$, we repeated the simulation study for selected cases considering a value of $\omega = 0.5$. We expect that such a lower value will produce a slower aggregation and a more detailed solution path, allowing the algorithm to identify solution that were skipped with $\omega = 0.9$. In particular, for $K = 3$ we selected scenarios A, D and G, with $J = 10$, to see if the accuracy can be further improved. We also selected scenario H that is the one with the worst performance. For $K = 4$ we selected scenarios B, E, H,

and I. The first 3 scenarios lead with $\omega = 0.9$ to strongly biased solutions, scenario I is instead the most variable. Table 4.6 reports the mean squared errors of the SPC with $\omega = 0.5$ over all the ability parameters in the selected scenarios, compared to the alternative methods. More detailed results can be found in Table 2, Figure 4 and Figure 5 in the Appendix.

Scenarios with $K = 3$			
	SPC	LC	CL
A	0.1184	0.3267	0.2058
D	0.1082	0.3145	0.3143
G	0.0918	0.3181	0.3214
H	0.1332	0.0124	0.0488
Scenarios with $K = 4$			
	SPC	LC	CL
B	0.1597	0.0898	0.1036
E	1.4658	0.0638	0.1016
H	1.4713	0.0491	0.0556
I	0.4113	0.0117	0.0231

Table 4.6: Simulation study: mean squared error of the estimated abilities for the SPC with $\omega = 0.5$ in 8 selected scenarios with $K = 3$ and $K = 4$.

As expected, we can appreciate a drastic improvement in the performance of the SPC for $K = 3$. The variability in the boxplots drops with respect to Figure 4.5, and they appear to be better centered on the true values. The mean squared error is always reduced almost by half, but this is not enough in scenario H. Also the selected scenarios with $K = 4$ show a strong gain in terms of mean squared error. The variability is globally lower, and scenarios B and I exhibit very nice features. Scenarios E and H are still affected by a strong bias relatively to the outer parameters, θ_1 and θ_4, confirming the tendency of the SPC to emphasize the separation of extreme clusters. The values of accuracy in these scenarios are still not acceptable. We expect that a further reduction of ω may lead to a better solution.

The focus of this work is on the abilities, the only parameters subject to shrinkage, but the difficulties have been estimated along in the 27 scenarios. The distribution of the $\hat{\beta}$s are essentially homogeneous in the three competing methods, even when the SPC estimates of the abilities present a relevant amount of outliers.

Chapter 5

A Penalized Fixed-Effects Model for Continuous Responses with Clustered Effects

5.1 The Penalized Fixed-Effects Model for Continuous Responses

In Chapter 4 we started to study properties of the Solution Path Clustering on the Rasch model for binary data, but the method ca be extended to any latent variable model with fixed effects. We expect the true potential of the SPC algorithm for fixed effects to be expressed when the number of parameters to be estimated is much higher (ideally in the longitudinal setting). In this chapter we apply the proposed algorithm to a slightly more complex framework, with a higher number of free parameters.

Depending on the response format, many possible extensions of the Rasch model have been widely investigated in the psychometric literature. From Rasch models for polythomous items, to Rasch models for items with ordered categories, or ratings (Müller, 1987), they all share a common algebraic formulation and have as basic building block the fundamental process defined by Rasch's simple logistic expression (Masters & Wright, 1984).

Here we choose to focus on continuous responses, which can be interpreted as scales, ratings or scores in the set of real numbers, under what we think as a linear adaptation of the Rasch process. Also in this case the minimal sufficient statistic for the individual parameter is a function of the row score $y_{i\cdot}$ (Andersen, 1977). Considering continuous responses, the number of unique values for $y_{i\cdot}$ is equal to the number n of individuals/respondents. In fact, even with a small number of variables/items, it is very difficult to find individuals who are assigned the same real numbered score. In other words, unlike the binary case, the number m of different values for $y_{i\cdot}$ is not bounded to $J-1$, but to n. Since the number of different values attained by the fixed effects $\theta_1, \ldots, \theta_n$ depends directly on the number of observations, we propose to apply the the Solution Path Clustering (Marchetti *et al.*, 2014) in order to obtain a smaller number K of values for θ.

5.1.1 Definition of the Optimization Problem

Lets consider a data matrix $\mathbf{Y}$ containing the continuous observations of a group of n units on a set of J variables. The generic response $y_{ij} \in \mathbb{R}$ can be modeled as follows ($i = 1, \ldots, n$ and $j = 1, \ldots, J$)

$$y_{ij} = \theta_i - \beta_j + \varepsilon_{ij}, \tag{5.1}$$

where we assume θ_i to be a fixed individual-specific effect capturing the latent attribute of the i-th respondent, β_j to be an item-dependent parameter capturing its effect of the $j-$th item on the responses, and ε_{ij} to be a Gaussian error with zero mean and a variance equal to $\pi^2/3$, the variance of the Logistic distribution.

$$\varepsilon_{ij} \sim N\left(0, \pi^2/3\right) \tag{5.2}$$

We can see this as a linear version of a Rasch model, where the Logistic link is substituted by a linear function of abilities and difficulties. Alternatively, Equation (5.1) can be viewed as a two-way ANOVA model or as a linear latent variable model with fixed-effects only.

The constrained optimization problem can be written as follows:

$$\Lambda_p(\boldsymbol{\theta}, \boldsymbol{\beta}) = \ell(\boldsymbol{\theta}, \boldsymbol{\beta}) + \lambda P(\boldsymbol{\theta}), \tag{5.3}$$

where the objective function is given by

$$\ell(\boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_{i=1}^{n} \sum_{j=1}^{J} \left(y_{ij} - \theta_i + \beta_j\right)^2, \tag{5.4}$$

and the penalty is

$$P(\boldsymbol{\theta}) = \sum_{i<j} \rho\left(\|\theta_i - \theta_h\|_2\right), \tag{5.5}$$

being $\lambda > 0$, and $\rho(\cdot)$ the Minimax Convex Penalty (MCP) as defined in Equation (3.26).

Since the objective function is quadratic, we do not need to apply the surrogation strategy seen in Equation (4.9), and we can implement an MM algorithm in the spirit of Marchetti *et al.* (2014). An important difference with both Marchetti's formulation and the Penalized Rasch Model for binary data, presented in Chapter 4, is that here in the objective function (5.4) other then $\boldsymbol{\theta}$ we keep the set of parameters $\boldsymbol{\beta}$, which need to be estimated but are not subject to shrinkage.

Being a block-wise procedure, the MM algorithm requires the objective function to be separable in its coordinates, we can then write the objective (5.4) emphasizing the group structure:

$$\ell_K(\boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{J} (y_{ij} - \theta_k + \beta_j)^2 , \tag{5.6}$$

where K is the number of different values in $\boldsymbol{\theta}$, and C_k is a label indicating a group of individuals with the same value of the attribute measured by the individual-specific parameter θ_k, $k = 1, \ldots, K$. Following the steps seen in the previous Chapter, we can now isolate the k-th coordinate (on which the algorithm will cycle), obtaining what will be our final objective function:

$$\ell_K(\theta_k, \boldsymbol{\beta}) = \sum_{i \in C_k} \sum_{j=1}^{J} (y_{ij} - \theta_k + \beta_j)^2 . \tag{5.7}$$

In the same way, the penalty term must be expressed taking into account the K blocks

$$P_K(\boldsymbol{\theta}) = \sum_{k<l} N_k N_l \rho \left(\|\theta_k - \theta_l\|_2 \right) , \tag{5.8}$$

and then, once reduced to the k-th coordinate, approximated by a majorant, just as in Equation (4.10)

$$P_K(\theta_k) = N_k \sum_{l \neq k} N_l \rho \left(\|\theta_k - \theta_l\|_2 \right) \approx N_k \sum_{l \neq k} w_{kl}^{(t)} \|\theta_k - \theta_l\|_2^2 + C, \tag{5.9}$$

where both the weight $w_{kl}^{(t)}$ and the constant C do not depend on θ_k, and the expression of $w_{kl}^{(t)}$ is identical to the original in Marchetti *et al.* (2014).

The final optimization problem can be written in Lagrangian form

$$\Lambda_p(\theta_k, \boldsymbol{\beta}) = \ell_K(\theta_k, \boldsymbol{\beta}) + \lambda P_K(\theta_k), \tag{5.10}$$

$$\Lambda_p(\theta_k, \boldsymbol{\beta}) = \sum_{i \in C_k} \sum_{j=1}^{J} (y_{ij} - \theta_k + \beta_j)^2 + \lambda N_k \sum_{l \neq k} w_{kl}^{(t)} \|\theta_k - \theta_l\|_2^2 + C, \tag{5.11}$$

and differentiated with respect to the parameter of interest θ_k

$$\frac{\partial \Lambda_p(\theta_k, \boldsymbol{\beta})}{\partial \theta_k} = -2 \sum_{i \in C_k} \sum_{j=1}^{J} (y_{ij} - \theta_k + \beta_j) + 2\lambda N_k \sum_{l \neq k} w_{kl}^{(t)} (\theta_k - \theta_l) . \tag{5.12}$$

In order to find the update for the clustered individual-specific parameters we verify the first order condition

$$-\sum_{i\in C_k}\sum_{j=1}^{J}(y_{ij}-\theta_k+\beta_j)+\lambda N_k\sum_{l\neq k}w_{kl}^{(t)}(\theta_k-\theta_l)=0 \tag{5.13}$$

$$-\sum_{i\in C_k}\sum_{j=1}^{J}(y_{ij}+\beta_j)+N_kJ\theta_k+\lambda N_k\theta_k\sum_{l\neq k}w_{kl}^{(t)}-\lambda N_k\sum_{l\neq k}w_{kl}^{(t)}\theta_l=0 \tag{5.14}$$

$$\theta_kN_k\left(J+\lambda\sum_{l\neq k}w_{kl}^{(t)}\right)=\sum_{i\in C_k}\sum_{j=1}^{J}(y_{ij}+\beta_j)+\lambda N_k\sum_{l\neq k}w_{kl}^{(t)}\theta_l, \tag{5.15}$$

so that

$$\theta_k^{(t+1)}=\frac{\frac{1}{N_k}\sum_{i\in C_k}\sum_{j=1}^{J}\left(y_{ij}+\beta_j^{(t)}\right)+\lambda\sum_{l\neq k}w_{kl}^{(t)}\theta_l}{J+\lambda\sum_{l\neq k}w_{kl}^{(t)}}. \tag{5.16}$$

We can see that unlike in the original paper and in the binary formulation, here the update of θ_k directly involves the β_j parameters. Comparing the above expression with corresponding one in the binary Rasch model, we can see that the adjusted dependent variable $\tilde{\theta}_k$ for the k-th group is substituted by the mean of the raw score corrected by the difficulty parameter.

The estimation of $\boldsymbol{\beta}$ appears outside the block structure of the MM algorithm, so we can compute the update for β_j starting from Equation (5.4), ignoring group and penalty part of Equation (5.6):

$$\frac{\partial\Lambda_p(\boldsymbol{\theta},\boldsymbol{\beta})}{\partial\beta_j}=2\sum_{i=1}^{n}(y_{ij}-\theta_i+\beta_j)=2\left[\sum_{i=1}^{n}(y_{ij}-\theta_i)+n\beta_j\right], \tag{5.17}$$

from which

$$\beta_j^{(t+1)}=\frac{1}{n}\sum_{i=1}^{n}\left(\theta_i^{(t+1)}-y_{ij}\right). \tag{5.18}$$

The update for β_j relies on the previously estimated values of $\boldsymbol{\theta}$, so it must be computed right after $\theta_k^{(t+1)}$. The resulting MM algorithm is:

5.1.2 The SPC algorithm

We have seen that the SPC algorithm is a tuning framework in which the MM is nested. Its structure does not change in the continuous case, but several adjustments are needed in order to initialize the tuning parameters λ and δ.

Algorithm 5 Modified MM algorithm for continuous data

repeat
  for $k = 1, \ldots, K$ **do**
    Majorization step: compute weights $w_{kl}^{(t)}$ for all $l \neq k$
    Minimization step: update $\theta_k^{(t+1)}$
    if $\|\theta_k^{(t+1)} - \theta_l\|_2 < \xi$ for some l **then**
      set $\theta_k^{(t+1)}$ and θ_l to their weighted mean $\overline{\theta}_{kl}$
    end if
  end for
  Item-specific step: compute $\hat{\boldsymbol{\beta}}^{(t+1)}$ as in (5.18)
until $\max_{1 \leq k \leq K} \|\theta_k^{(t+1)} - \theta_k^{(t)}\|_2 < \xi$ **or** $t > 50$

The surrogate objective function used in the binary case allowed us to leave unaltered the initialization mechanism for the tuning parameters, that in Marchetti *et al.* (2014) is based on two lemmas. The presence of the βs in the objective function in the present case changes the way we define the first value of λ.

Following the proof of lemma 1 in the appendix of Marchetti *et al.* (2014), we set

$$\theta_1^{(t+1)} - \theta_2^{(t)} = \frac{y_1 + \sum_{j=1}^{J} \beta_j + \lambda w_{12}^{(t)} \theta_2^{(t)}}{J + \lambda w_{12}^{(t)}} - \theta_2^{(t)} = (1 - \phi) \left(\theta_1^{(t)} - \theta_2^{(t)} \right) \quad (5.19)$$

and, simply rearranging the terms, we end up with the following definition of λ

$$\lambda = \frac{2\eta \| \left(J\theta_1 - y_1 - \sum_{j=1}^{J} \beta_j \right) + J\phi \left(\theta_1^{(t)} - \theta_2^{(t)} \right) \|_2}{(1 - \phi) (\eta - \|\theta_1 + \theta_2\|_2)}. \quad (5.20)$$

To ensure identifiability we set $\sum_{j=1}^{J} \beta_j = 0$, so that

$$\lambda = \frac{2\eta \| \left(J\theta_1 - y_1 \right) + J\phi \left(\theta_1^{(t)} - \theta_2^{(t)} \right) \|_2}{(1 - \phi) (\eta - \|\theta_1 + \theta_2\|_2)}. \quad (5.21)$$

In Equation (5.21) the values of θ_i, with $i = 1, 2$, are substituted with the corresponding sufficient statistics. In particular we use the mean score on the J items y_i/J, with $i = 1, 2$. The result is

$$\lambda = \frac{2\eta\varphi \|y_1 - y_2\|_2}{(1 - \varphi) (\eta - \|y_1 + y_2\|_2 / J)} = \frac{2\eta\varphi d}{(1 - \varphi) (\eta - d/J)}. \quad (5.22)$$

As in Marchetti *et al.* (2014)we set $d = Q_\tau$ and $\eta = Q_\omega$, where $\tau \in (0, \omega)$, and Q_ω and Q_τ are quantiles of the distribution of the nearest neighbor distances between the initial values of the sufficient statistics:

$$\lambda = \frac{2\varphi Q_\omega Q_\tau}{(1-\varphi)(Q_\omega - Q_\tau/J)}. \tag{5.23}$$

With respect to the binary case in Equation (4.19) we can see that the value of Q_τ is divided by J, resulting in smaller values of λ_1. The nearest neighbor distances can be computed either on the values of the sufficient statistics or on the row vectors of the respondent profiles, this choice does not affect the results of the algorithm. The expressions for λ_G and δ_1 are not affected by the change in the update and are set equal to Equation (4.21) and Equation (4.20).

The SPC algorithm starts with $K = m$ and usually $m = n$ and it ends when $K = 1$ with the exact formulation of Algorithm 3, presented in Section 4.1.2.

5.2 Real Data Example

In Section 4.3 we tested the SPC algorithm for a binary Rasch model on a dataset from the INVALSI mathematics test. As we have already seen, INVALSI tests are performed annually to assess the proficiency of Italian students in the fields of mathematics and Italian language. Ability scores for the two disciplines are then systematically estimated by the Institute through a simple binary Rasch model. Here we use the estimated ability scores in mathematics and Italian to compute an overall categorical ability level for each student. In other words we want to find latent classes of global ability starting by the separated scores in mathematics and Italian.

The INVALSI test are performed at a population level, and they are compulsory in every school in Italy. The Institute then draws a sample of students with a two stages strategy: at a first step they sample schools, and in the second one they select two classes per school. We work on the INVALSI sample of 2015, made by 33687 observations for which we have the scores in Italian and mathematics, resulting respectively from a Rasch model on 40 and 27 items. We randomly sample 1000 students from this dataset, and represent the resulting distributions of the scores WLE_ITA and WLE_MAT in Figure 5.1. We can't see a clear group structure, and the boxplots do not exhibit strong differences in the mean, while the score in mathematics is more variable.

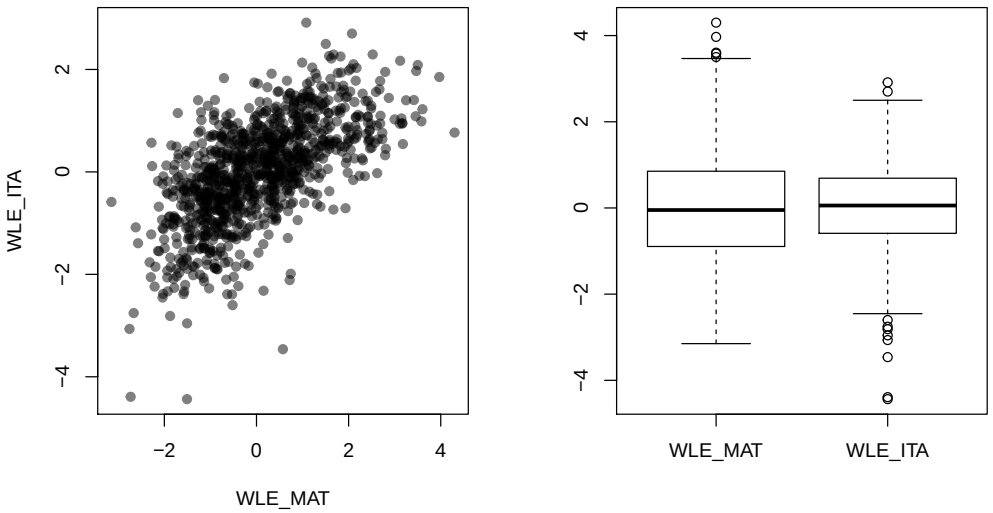

Figure 5.1: INVALSI scores data.

We run the SPC algorithm for the estimation of clustered fixed effects in a model with continuous responses, as presented in the previous section, with $\omega = 0.85$, together with two classic agglomerative hierarchical clustering algorithms: complete-linkage clustering (CLINK) and Ward's minimum variance method (Ward Jr, 1963). In Figure 5.2 we visualize the results of the SPC algorithm, where the solution is selected applying the usual difference ratio rule, as in Equation (4.26), while Figure 5.3 we show the dendrograms resulting from the hierarchical cluster analysis. The SPC solution identifies 5 groups, and it adapts pretty well to the empirical distribution of the sufficient statistics, whose density is overlayed in Figure 5.2. The two central groups of this solution include the 82.3 % of the observation in the sample, while there is a group made by only 2 outlier observations, indicating the two students with the highest global ability estimates. As it was in the binary case, the solution path in Figure 5.2 tends to isolate outliers, rather than aggregating them with larger groups. The 6 groups solution (one step before the one selected) presents two outlier groups, one corresponding to the highest ability level and the other one to the lowest. Concerning the hierarchical clustering, we can see that the dendrograms lead both to a 2 groups solution. In Table 5.1 we report, together with the selected SPC solution in the first column, the mean values of the sufficient statistics computed within the clusters found with CLINK and the Ward method, for $K = 2$ and $K = 4$. The number of

SPC	Hierarchical Clustering			
	CLINK		Ward	
θ_k	$\bar{y}_k$	$\bar{y}_k$	$\bar{y}_k$	$\bar{y}_k$
2.9515 (2)				
1.8277 (94)		1.7076 (127)		1.6494 (130)
0.6363 (375)	0.5803 (651)	0.3071 (524)	1.0064 (408)	0.7057 (278)
-0.5373 (448)	-0.9912 (349)	-0.7139 (170)	-0.6398 (592)	-0.1616 (284)
-1.7193 (81)		-1.2546 (179)		-1.0807 (308)

Table 5.1: INVALSI scores example: SPC estimates for θ_k, with $k = 1, \ldots, 4$ (in brackets the size of each group), empirical means $\bar{y}_k$ of the sufficient statistics in K groups defined by a hierarchical agglomeration (for Complete link and Ward method) for $K = 2$ and $K = 4$.

groups to select in this exploratory analysis is often context-dependent, and in our case we want to check if the mean values of the sufficient statistics in such clusters are comparable with the SPC estimates.

Producing confusion matrices between the SPC and the 4 cluster analysis solutions, we assessed that the most similar result in terms of group label assignment is between the SPC and the 2 cluster solution with the Ward method. The solution path of the SPC with $\omega = 0.85$ skips the 2 groups solution, aggregating the 5 groups into 1, and this is due to the tendency of the proposed method to produce classes that reflect the shape of the original distribution. Concerning the item-specific parameters, we get $\beta_{MAT} = 0.0063$ and $\beta_{ITA} = -0.0063$. These values are very close to 0 because of the strong similarity in the distribution of the variables, but the opposite sign indicates a relative higher difficulty connected with the mathematics scores.

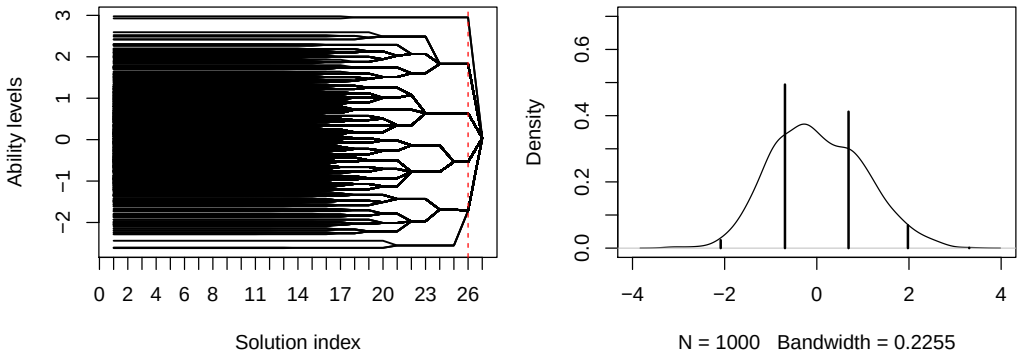

Figure 5.2: INVALSI scores example: SPC algorithm.

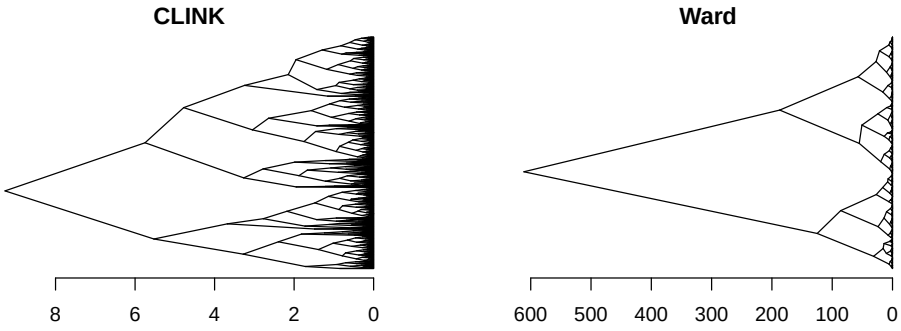

Figure 5.3: INVALSI scores example: hierarchical clustering.

5.3 Simulation Study

We run a simulation study to test the SPC algorithm in the continuous setting, using the same structure and labeling for the scenarios used in the binary data setting, and presented in Table 4.4.

In this case, within each group we simulated n_k continuous response profiles from a J-variate normal distribution with mean $\mathbb{1}_J\theta_k - \boldsymbol{\beta}$ and variance $\text{diag}(\mathbb{1}_J)\pi^2/3$

$$\mathbf{y}_i \sim N_J\left(\mathbb{1}_J\theta_k - \boldsymbol{\beta}, \text{diag}(\mathbb{1}_J)\frac{\pi^2}{3}\right), \tag{5.24}$$

with $i = 1, \ldots, n_k$, $k = 1, \ldots, K$ and n_k is fixed in a way that $\sum_{k=1}^{K} n_k = n$ and $n_1 = n_2 = \ldots \approx n_K, \forall K$.

A sensitivity analysis has been performed to assess the value of ω. In each scenario we performed the SPC on the same dataset with 9 values of ω from 0.25 to 0.95. The analysis showed that in the continuous case, when the algorithm converges, the value of ω does not affect the computational time. The time is affected only by the number of respondents n and consequently the number of unique values of the sufficient statistics m. The algorithm always leads to a high value for λ and a low one for δ, corresponding to a strong regularization effect on a concave penalty, since λ controls the amount of shrinkage and δ the degree of concavity (when $\delta \to 0$ the penalty approaches the ℓ_0). However, since the tuning parameters are multiplied together in the MCP formulation and in Equation (4.11), we verify that the product $\lambda\delta$ is stable on values between 2 and 2.9. Concerning the solutions, we can say that within each scenario the value of ω has a scarce impact, leading always to the same solution. Between scenarios we can see that problems arise for $J = 10$ when n is high. In these cases the algorithm jumps directly from

a number of clusters greater than the true number K^0 to 1. This can be due to the fact that when J is too small for these particular scenarios, the empirical distribution of the sufficient statistics is too homogeneous and the algorithm fails to separate the right number of groups. The best scenario is the one with a high number of variables/items with respect to the number of individuals/respondents.

The sensitivity analysis gave us an insight on the algorithm limitations, and showed that in the continuous case the value of ω is not relevant, leading always to the same final values of λ and ω. We choose to perform the simulations with $\omega = 0.85$ for scenarios with $J = 20$ and $J = 50$. Figure 5.4, Figure 5.5 and Figure 5.6 show the boxplots of the estimated ability $\hat{\theta}_k$,with $k = 1, \ldots, K$, in the 6 scenarios respectively with $K = 2$, $K = 3$ and $K = 4$. The red line represents the true value θ_k^0. We can see how the boxplots are always well centered on the true values of the ability. The variability is influenced both by n and J, since in scenarios with high J the empirical distribution of the sufficient statistics shows naturally groups that are better separated.

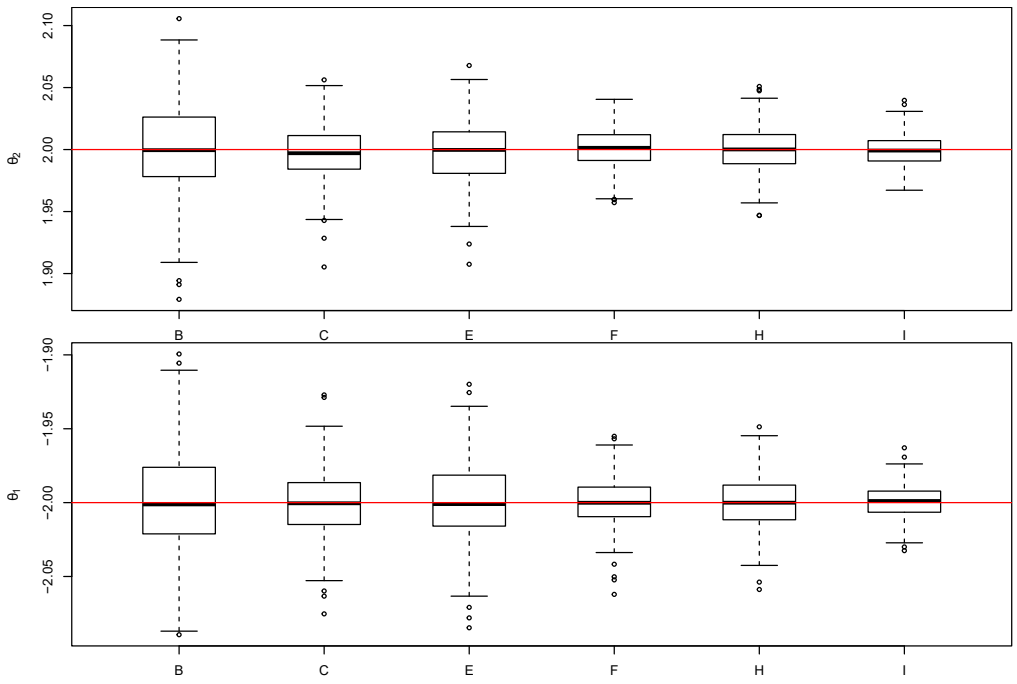

Figure 5.4: Simulation study: solutions of SPC with $\omega = 0.85$ for $K = 2$ (boxplots in 6 scenarios).

Table 5.2, Table 5.3 and Table 5.4 report the values of the mean squared error in the selected scenarios. As expected, scenarios where $J = 50$ (C, F and I) are always characterized by a lower mean squared error for $\hat{\theta}_k$ with respect to those where $J = 20$ (B, E and H). Comparing the results among different assumption for the underlying group structure, we can notice how the mean squared errors of the estimates grow with growing values of K. Similarly to what we have seen before, this is due to a lower separation of the clusters in the distribution of the sufficient statistics. Concerning $K = 4$ we point out that the variability of the central ability levels estimates $\hat{\theta}_2$ and $\hat{\theta}_3$ is higher than for the extremes $\hat{\theta}_1$ and $\hat{\theta}_4$.

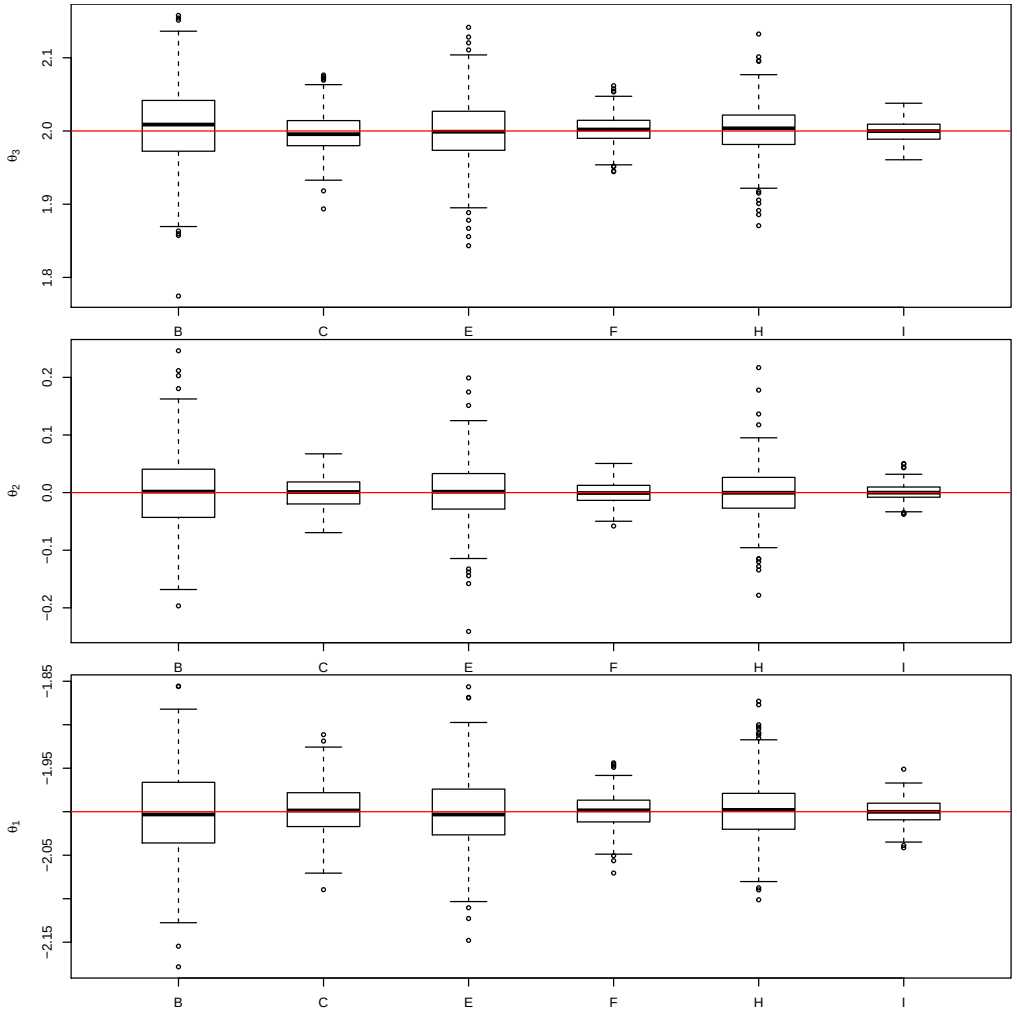

Figure 5.5: Simulation study: solutions of SPC with $\omega = 0.85$ for $K = 3$ (boxplots in 6 scenarios).

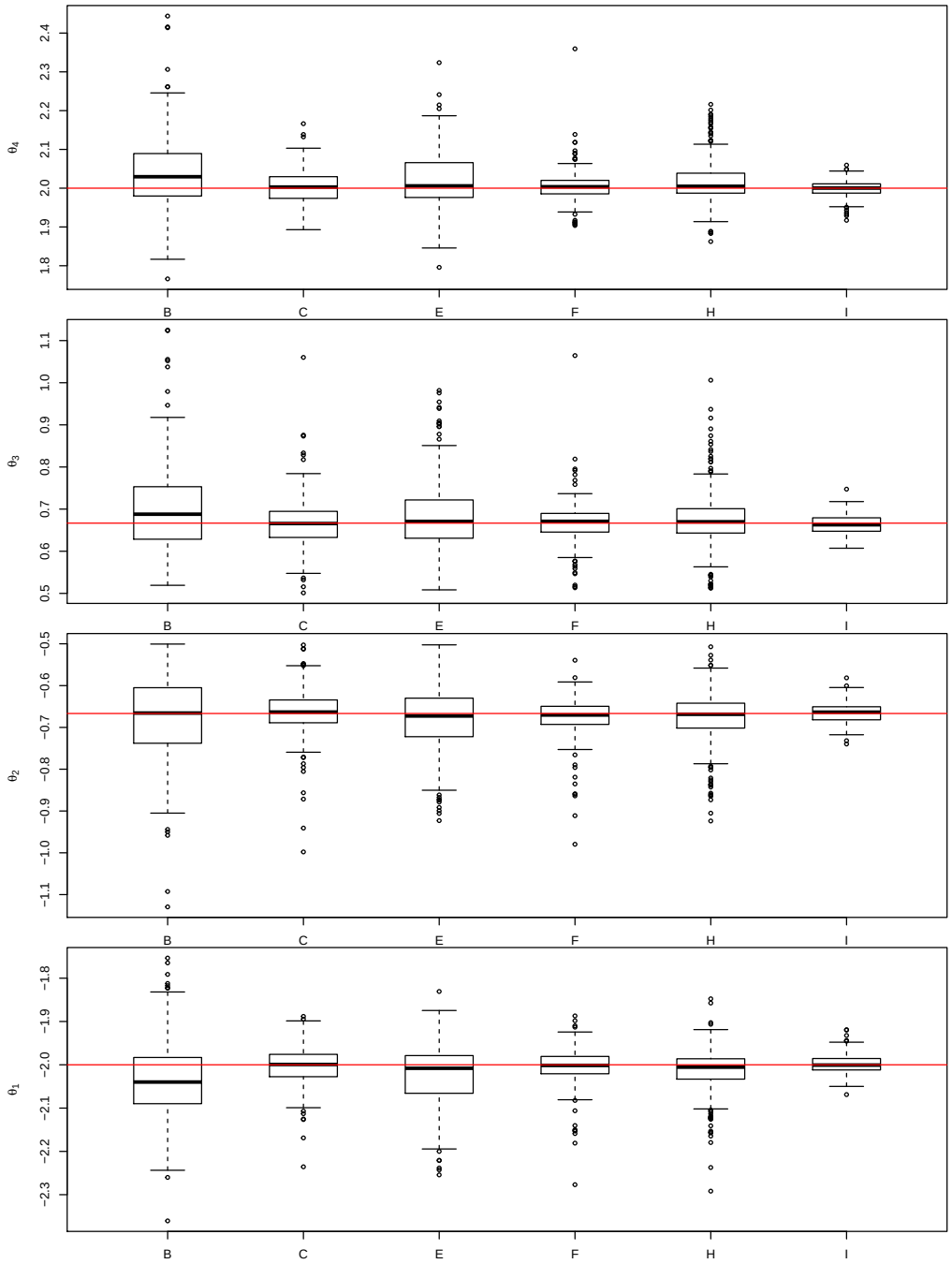

Figure 5.6: Simulation study: solutions of SPC with $\omega = 0.85$ for $K = 4$ (boxplots in 6 scenarios).

The SPC algorithm shows a good performance in the continuous case. The simulation confirmed that it performs best when the number of items

Scenario	$\hat{\theta}_1$	$\hat{\theta}_2$
B	0.00117	0.00127
C	0.00048	0.00051
E	0.00065	0.00060
F	0.00024	0.00024
H	0.00030	0.00030
I	0.00012	0.00015

Table 5.2: Simulation study: mean squared error of the estimated abilities of SPC with $\omega = 0.85$ in 6 scenarios with $K = 2$.

Scenario	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$
B	0.00284	0.00444	0.00299
C	0.00080	0.00074	0.00081
E	0.00155	0.00268	0.00191
F	0.00035	0.00034	0.00039
H	0.00126	0.00195	0.00119
I	0.00019	0.00019	0.00021

Table 5.3: Simulation study: mean squared error of the estimated abilities of SPC with $\omega = 0.85$ in 6 scenarios with $K = 3$.

Scenario	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$
B	0.00987	0.01118	0.01251	0.01043
C	0.00179	0.00282	0.00304	0.00178
E	0.00538	0.00731	0.00838	0.00567
F	0.00168	0.00230	0.00220	0.00154
H	0.00333	0.00457	0.00550	0.00389
I	0.00047	0.00059	0.00051	0.00047

Table 5.4: Simulation study: mean squared error of the estimated abilities of SPC with $\omega = 0.85$ in 6 scenarios with $K = 4$.

and respondents is very large. With respect to the binary case, here we can clearly see that, when the amount of initial information is high and the complexity of the model grows, the proposed method behaves regularly, exactly as expected.

Unfortunately, the classification likelihood algorithm, presented in Section 4.2 is not applicable to the continuous case. Having to perform $\binom{K}{2}$ optimization at each step, it becomes unfeasible with $K = 1000$.

The main interest of our analysis is focused on subject-specific parameters, since the estimation of item-specific parameters is not influenced by the shrinkage. Table 5.5 report the mean values and the mean squared errors of β_j for $j = 1, \ldots, J$ for scenarios with $J = 20$. In Table 3 of the Appendix we show the same results for scenarios with $J = 50$, together with boxplots in Figure 6 and Figure 7. For what concerns the mean squared errors, we can see that their values remain almost constant in the order of 0.007, for growing values of both K and J. Looking at the boxplots, we can see how the βs span between the theoretical extreme values of -2 and 2 and how the number of outliers grows, in general, with K and with J. This depends on the fact that the range from -2 to 2 stay fixed while the amount of points to be estimated is more than doubled from 20 to 50; but also on the fact that with $K = 4$ the variability of θ_k is higher, and θ_k appears in the update of β_j, as shown in Equation (5.18).

	$K=2$		$K=3$		$K=4$	All K
	mean	mse	mean	mse	mean	mse
$\hat{\beta}_1$	-1.98	0.0086	-2.00	0.0067	-2.00	0.0081
$\hat{\beta}_2$	-1.80	0.0084	-1.78	0.0083	-1.79	0.0068
$\hat{\beta}_3$	-1.57	0.0060	-1.57	0.0075	-1.58	0.0067
$\hat{\beta}_4$	-1.37	0.0075	-1.37	0.0057	-1.36	0.0067
$\hat{\beta}_5$	-1.16	0.0065	-1.16	0.0073	-1.16	0.0069
$\hat{\beta}_6$	-0.96	0.0069	-0.94	0.0076	-0.95	0.0074
$\hat{\beta}_7$	-0.74	0.0082	-0.74	0.0070	-0.74	0.0072
$\hat{\beta}_8$	-0.53	0.0074	-0.52	0.0072	-0.52	0.0070
$\hat{\beta}_9$	-0.31	0.0071	-0.32	0.0081	-0.32	0.0063
$\hat{\beta}_{10}$	-0.11	0.0107	-0.10	0.0068	-0.11	0.0080
$\hat{\beta}_{11}$	0.12	0.0097	0.10	0.0070	0.11	0.0058
$\hat{\beta}_{12}$	0.30	0.0082	0.32	0.0084	0.32	0.0075
$\hat{\beta}_{13}$	0.52	0.0080	0.52	0.0070	0.52	0.0076
$\hat{\beta}_{14}$	0.75	0.0070	0.72	0.0072	0.74	0.0066
$\hat{\beta}_{15}$	0.93	0.0062	0.94	0.0089	0.95	0.0060
$\hat{\beta}_{16}$	1.17	0.0068	1.16	0.0073	1.15	0.0063
$\hat{\beta}_{17}$	1.37	0.0072	1.36	0.0078	1.37	0.0075
$\hat{\beta}_{18}$	1.57	0.0071	1.57	0.0088	1.58	0.0075
$\hat{\beta}_{19}$	1.79	0.0074	1.80	0.0076	1.79	0.0070
$\hat{\beta}_{20}$	1.99	0.0064	2.01	0.0072	2.00	0.0070
All j		0.0076		0.0075		0.0070

Table 5.5: SPC estimates of the β parameters (mean values) and relative mean squared errors in the scenarios with $J = 20$.

Chapter 6

Final Remarks

Simplicity is always a desirable attribute in a statistical model. The more complex a model is, the more urgent is the need for statistical methods to reduce such complexity. We have defined the complexity of a model as the number of free parameters to be estimated. Such number can easily become very large in latent variable models under the fixed-effects approach, where it directly grows with the number of observations. In this framework, the values attained by the latent variables are interpreted as fixed parameters that can be for example subject-specific, item-specific, time-specific, or even simultaneously varying across more than one of these dimensions.

In this work we have proposed to address the problem of complexity in Fixed-effect latent variable models by means of lasso-type regularization. The idea is to promote simplicity, reducing the number of different values attained by the fixed effects, while forcing them to group into different clusters. We started from the Solution Path Clustering (SPC) algorithm, proposed by Marchetti *et al.* (2014), we have developed a general procedure that can be applied to any latent variable model.

At first we adapted the SPC to the case of the binary Fixed-effects Rasch model, where the latent ability is treated as a subject-specific parameter, and the number of estimated ability levels is equal to the number of different values attained by the sufficient statistics in the sample. If we wish to estimate a smaller number of ability levels we can apply the random-effects approach estimating a latent class Rasch model. The proposed method offers an alternative, allowing for the estimation of latent classes of ability even under the fixed-effects approach, without the need of specifying any distributional assumptions. The two methods have been compared in a real data example on INVALSI mathematics tests. Both approaches have lead to a solution with four latent classes of ability. The example showed the tendency of the SPC algorithm to isolate groups of outliers, a unique property that distin-

guishes the method from the latent class Rasch model. Other than the SPC we have presented a second technique to cluster fixed effects in the Rasch model: the classification likelihood algorithm. This technique aggregates the ability levels following a maximum likelihood criteria. The three competing methods have been compared in a simulation study across 27 scenarios, with a number of true ability classes equal to 2, 3 and 4. The overall performance of the SPC in terms of accuracy of the estimates appeared to be equal or better than the competing methods when the number of groups was $K = 2$. For $K = 3$ and $K = 4$ the proposed algorithm outmatched the competing methods in all the scenarios with a low number of items, and it performed relatively well in the others, with a few exception that need to be further investigated in terms of sensitivity of the solution to the tuning parameter.

As a step towards higher complexity, we implemented the SPC algorithm on a Fixed-effects latent variable model for continuous data. We have considered an example on INVALSI scores, where we have defined latent classes of the overall ability level of the students, starting from the individual scores in mathematics and Italian. We have performed a simulation study to evaluate the behavior of the proposed algorithm in the continuous setting. The simulation was performed over the same structure of scenarios as in the binary case, and the performance appeared to be considerably improved. A sensitivity analysis showed that, in the continuous setting, the tuning process does not have a strong impact on the solutions. With respect to the binary case, the SPC estimates resulted to be characterized by smaller mean squared errors. In this case, the higher accuracy level corresponds to a lower bias of the estimates, and a variability that decreases regularly with an increase in the number of observations and variables. The simulation confirmed that the SPC for continuous data performs best when the number of items and respondents is very large. These good results encourage the extension of the method to more complex models.

The work on the SPC can be improved in many ways. We need to implement a better strategy to select the tuning parameter, especially in the binary case. In the Rasch setting we miss a solid criterion to link the data structure to a value of the tuning parameter ω. We need to further inquiry the tuning strategy with a more comprehensive approach and a dedicated simulation study. The performance of the SPC has to be compared with further competing methods, in particular those based on similar regularization strategies. In terms of computational efficiency the R code has to be improved, both for the SPC and the classification likelihood algorithm.

Apart from these critical aspects, the work stimulated much future research goals. Future research can be devoted to:

- extending the method to the case of a double penalization, that is penalizing simultaneously subject-specific and item-specific parameters. This may be interesting in the case of datasets with many items/variables, allowing for example to identify at the same time classes of respondents with the same level of ability and groups of items with the same difficulty.

- extending the method to more complex IRT models (2PL, 3PL, ...) generalizing the results obtained for the Rasch model, after a clear assessment of the sensitivity issues.

- extending the method to longitudinal latent variable models, where the latent constructs varies across time and we need to estimate a number of parameters equal to the product between the number of individuals and the number of times. Such a number can be very large, making the fixed-effects approach a less common choice with respect to the random effects. In this context we expect the SPC could lead to the most interesting results.

Appendix

This appendix contains some additional numerical results concerning the simulation studies in Chapter 4 and Chapter 5. Its content can be useful to get a better insight on the performance of the Solution Path Clsutering algorithm in the various scenarios.

Simulation Study: Binary Case

Table 1 reports the mean squared errors for the single ability parameter estimates in all the 27 scenarios for the 3 competing methods: the Solution Path Clustering (SPC) with $\omega = 0.9$, the latent class Rasch model (LC) and the classification likelihood (CL). Figure 1, Figure 2 and Figure 3 show a graphical comparison of the 3 methods in terms of boxplots of the estimates for 100 replications. The grids of plots, one for each latent structure assumption ($K = 2$, $K = 3$, $K = 4$), are organized in 9 panels, corresponding to scenarios from A to I. The single element of the grid is separated in 3 areas, one for each method (LC, CL and SPC, in order). Table 2 reports the mean squared errors for the single ability parameter estimates in 8 selected scenarios for the SPC with $\omega = 0.5$, while Figure 4 and Figure 5 show the corresponding boxplots of the estimates.

		Scenarios								
K		A	B	C	D	E	F	G	H	I
2	$\hat{\theta}_1^{(LC)}$	0.3044	0.0913	0.0758	0.3121	0.0761	0.0530	0.3065	0.0700	0.0395
2	$\hat{\theta}_2^{(LC)}$	0.2371	0.1002	0.0791	0.2085	0.0772	0.0552	0.2090	0.0477	0.0252
2	$\hat{\theta}_1^{(CL)}$	0.3242	0.0916	0.0853	0.3496	0.0763	0.0616	0.3640	0.0828	0.0108
2	$\hat{\theta}_2^{(CL)}$	0.2209	0.1004	0.0822	0.1797	0.0772	0.0623	0.1630	0.0558	0.0083
2	$\hat{\theta}_1^{(SPC)}$	0.2838	0.0991	0.0845	0.2878	0.0862	0.0652	0.2845	0.0806	0.0610
2	$\hat{\theta}_2^{(SPC)}$	0.2357	0.0938	0.0796	0.2157	0.0685	0.0552	0.2121	0.0403	0.0286
3	$\hat{\theta}_1^{(LC)}$	0.5730	0.0494	0.0494	0.5311	0.0578	0.0217	0.5444	0.0560	0.0148
3	$\hat{\theta}_2^{(LC)}$	0.0539	0.0398	0.0398	0.0325	0.0176	0.0162	0.0206	0.0089	0.0080
3	$\hat{\theta}_3^{(LC)}$	0.3531	0.0427	0.0427	0.3799	0.0620	0.0231	0.3894	0.0562	0.0143
3	$\hat{\theta}_1^{(CL)}$	0.2915	0.0739	0.0622	0.5352	0.0653	0.0357	0.5512	0.0684	0.0297
3	$\hat{\theta}_2^{(CL)}$	0.1661	0.0640	0.0541	0.0403	0.0271	0.0259	0.0316	0.0177	0.0182
3	$\hat{\theta}_3^{(CL)}$	0.1598	0.0656	0.0527	0.3674	0.0658	0.0343	0.3814	0.0604	0.0222
3	$\hat{\theta}_1^{(SPC)}$	0.2198	0.0794	0.0870	0.2285	0.0305	0.0313	0.2102	0.4172	0.0282
3	$\hat{\theta}_2^{(SPC)}$	0.1657	0.0833	0.0512	0.1899	0.0295	0.0190	0.2521	1.2895	0.0121
3	$\hat{\theta}_3^{(SPC)}$	0.1657	0.0872	0.0401	0.2516	0.0656	0.0202	0.0995	0.2493	0.0100
4	$\hat{\theta}_1^{(LC)}$	0.9426	0.1300	0.0401	0.8770	0.0993	0.0193	0.9171	0.0920	0.0152
4	$\hat{\theta}_2^{(LC)}$	0.2118	0.0746	0.0371	0.2486	0.0404	0.0201	0.1995	0.0273	0.0095
4	$\hat{\theta}_3^{(LC)}$	0.2953	0.0703	0.0292	0.2878	0.0419	0.0213	0.2490	0.0216	0.0087
4	$\hat{\theta}_4^{(LC)}$	0.5076	0.0844	0.0311	0.5444	0.0735	0.0198	0.5181	0.0554	0.0134
4	$\hat{\theta}_1^{(CL)}$	0.3530	0.1517	0.0539	0.3428	0.1005	0.0323	0.2974	0.1040	0.0276
4	$\hat{\theta}_2^{(CL)}$	0.2864	0.0873	0.0412	0.2551	0.1105	0.0262	0.3539	0.0338	0.0244
4	$\hat{\theta}_3^{(CL)}$	0.2785	0.0821	0.0593	0.2468	0.0961	0.0314	0.2853	0.0345	0.0217
4	$\hat{\theta}_4^{(CL)}$	0.2956	0.0934	0.0451	0.2876	0.0992	0.0283	0.3278	0.0502	0.0187
4	$\hat{\theta}_1^{(SPC)}$	0.1686	1.4505	0.4454	0.1380	1.6127	0.4378	0.1421	1.6066	0.5419
4	$\hat{\theta}_2^{(SPC)}$	0.1002	0.2162	0.1692	0.0869	0.2603	0.1120	0.1195	0.2343	0.1495
4	$\hat{\theta}_3^{(SPC)}$	0.1927	0.1969	0.1812	0.2004	0.1952	0.1245	0.2709	0.1952	0.1527
4	$\hat{\theta}_4^{(SPC)}$	0.7227	1.6499	0.4509	0.6919	1.6928	0.4556	0.7284	1.9874	0.4795

Table 1: Simulation study: mean squared error of the estimated abilities of LC, CL and SPC with $\omega = 0.9$ in 27 scenarios.

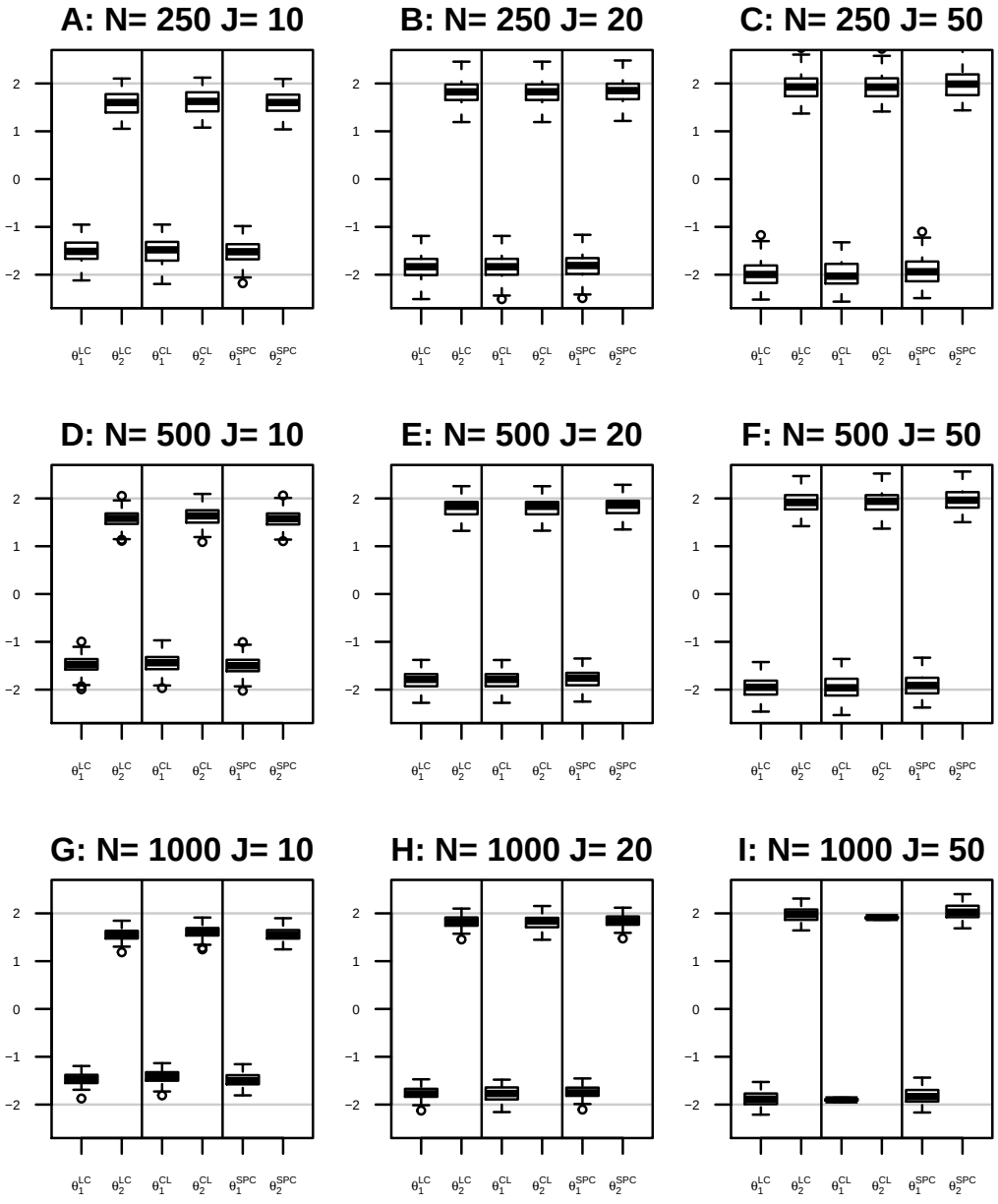

Figure 1: Simulation study: solutions of LC, CL and SPC with $\omega = 0.9$ for $K = 2$ (boxplots in 9 scenarios).

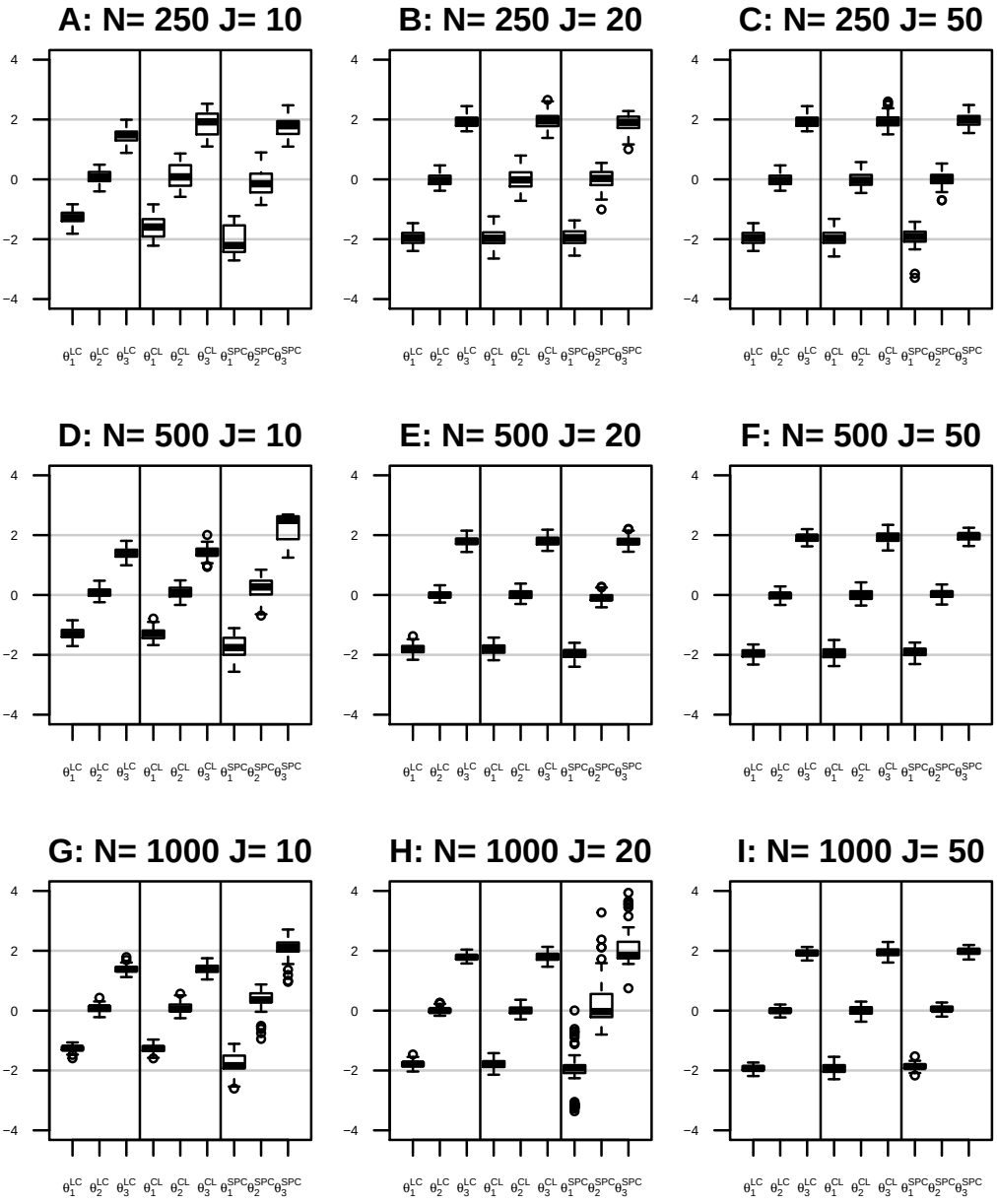

Figure 2: Simulation study: solutions of LC, CL and SPC with $\omega = 0.9$ for $K = 3$ (boxplots in 9 scenarios).

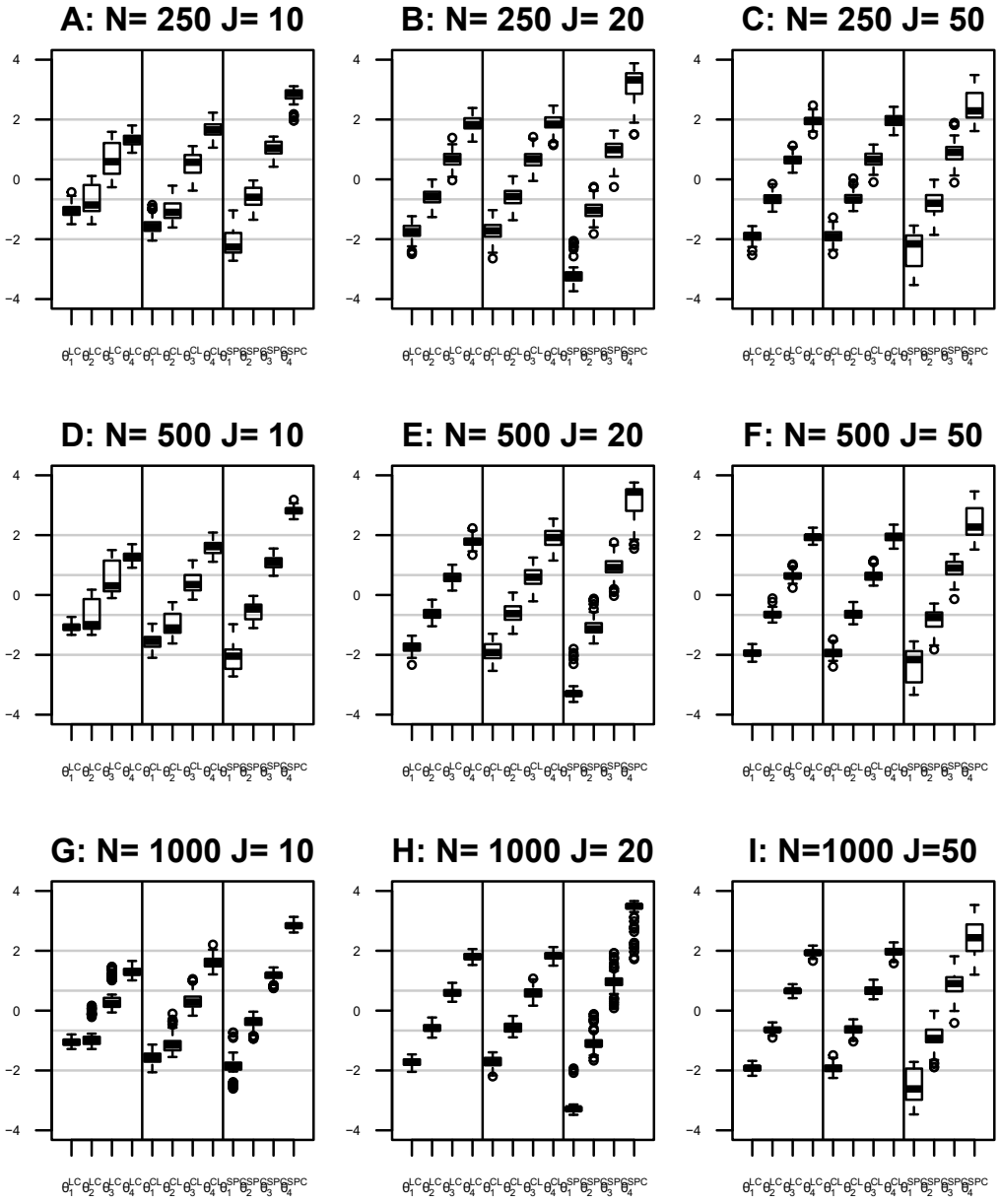

Figure 3: Simulation study: solutions of LC, CL and SPC with $\omega = 0.9$ for $K = 4$ (boxplots in 9 scenarios).

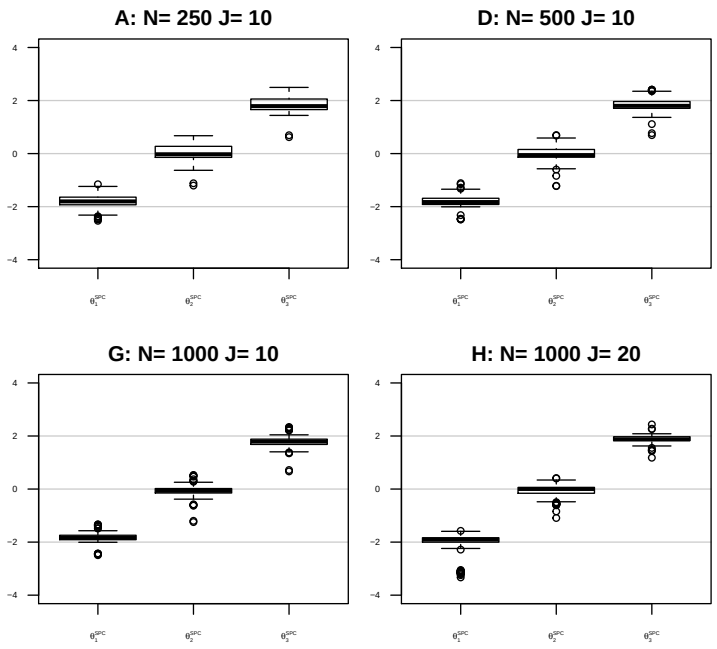

Figure 4: Simulation study: solutions of the SPC with $\omega = 0.5$ in 4 selected scenarios for $K = 3$.

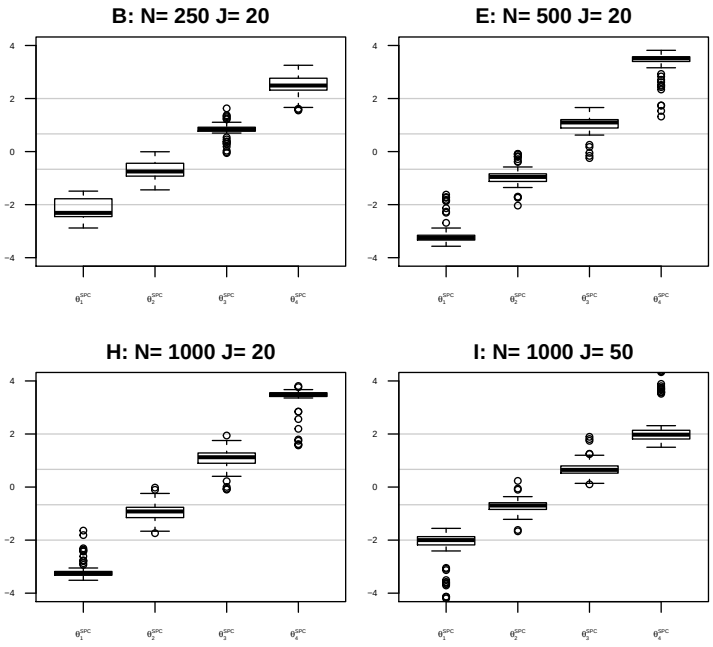

Figure 5: Simulation study: solutions of the SPC with $\omega = 0.5$ in 4 selected scenarios for $K = 4$.

	Scenarios with $K = 3$			
	$\theta_1^{(SPC)}$	$\theta_2^{(SPC)}$	$\theta_3^{(SPC)}$	
A	0.1245	0.1002	0.1304	-
D	0.1068	0.0980	0.1199	-
G	0.0780	0.0807	0.1169	-
H	0.2763	0.0840	0.0394	-
	Scenarios with $K = 4$			
	$\theta_1^{(SPC)}$	$\theta_2^{(SPC)}$	$\theta_3^{(SPC)}$	$\theta_4^{(SPC)}$
B	0.1597	0.1026	0.0878	0.3531
E	1.4658	0.1857	0.2523	2.0874
H	1.4713	0.1702	0.2816	2.1070
I	0.4113	0.0723	0.0951	0.5089

Table 2: Simulation study: mean squared error of the estimated abilities for the SPC with $\omega = 0.5$ in 8 selected scenarios with $K = 3$ and $K = 4$.

Simulation Study: Continuous Case

Figure 6 and Figure 7 show the boxplots of the estimates of the difficulty parameters in the simulated scenarios with $J = 20$ and $J = 50$. Table 3 reports the mean values and relative mean squared errors of the difficulties in the scenarios with $J = 50$.

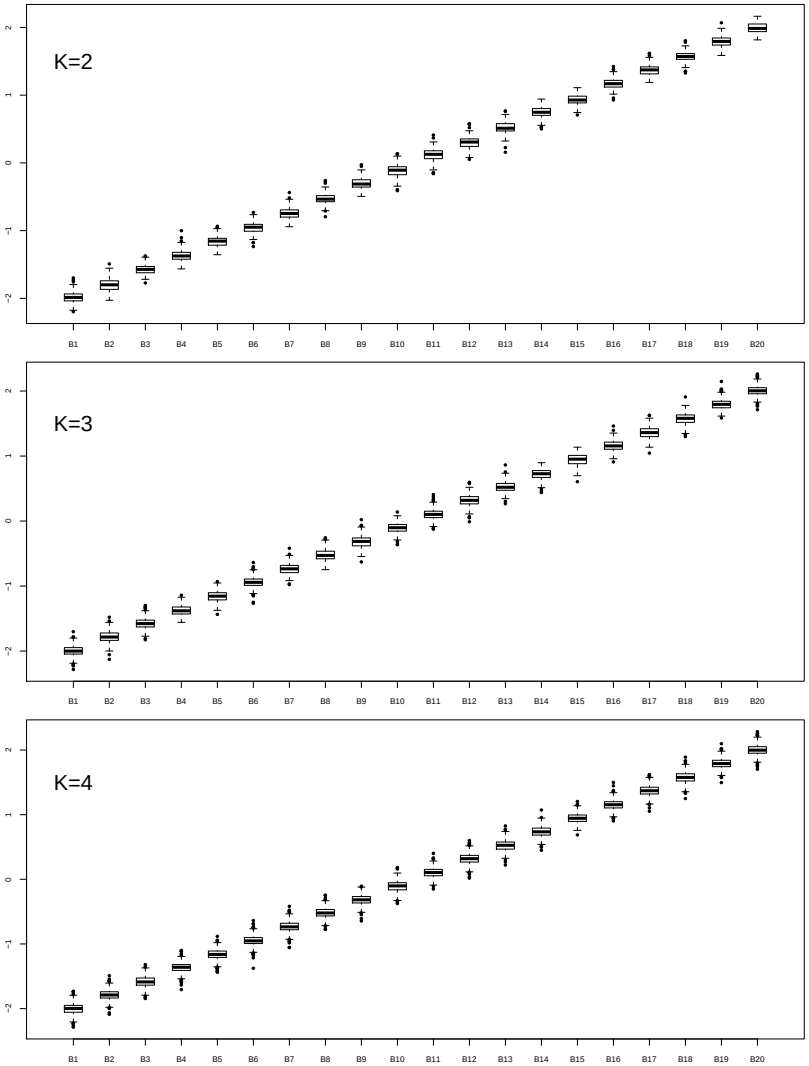

Figure 6: Simulation study: SPC estimates of the β parameters in the scenarios with $J = 20$.

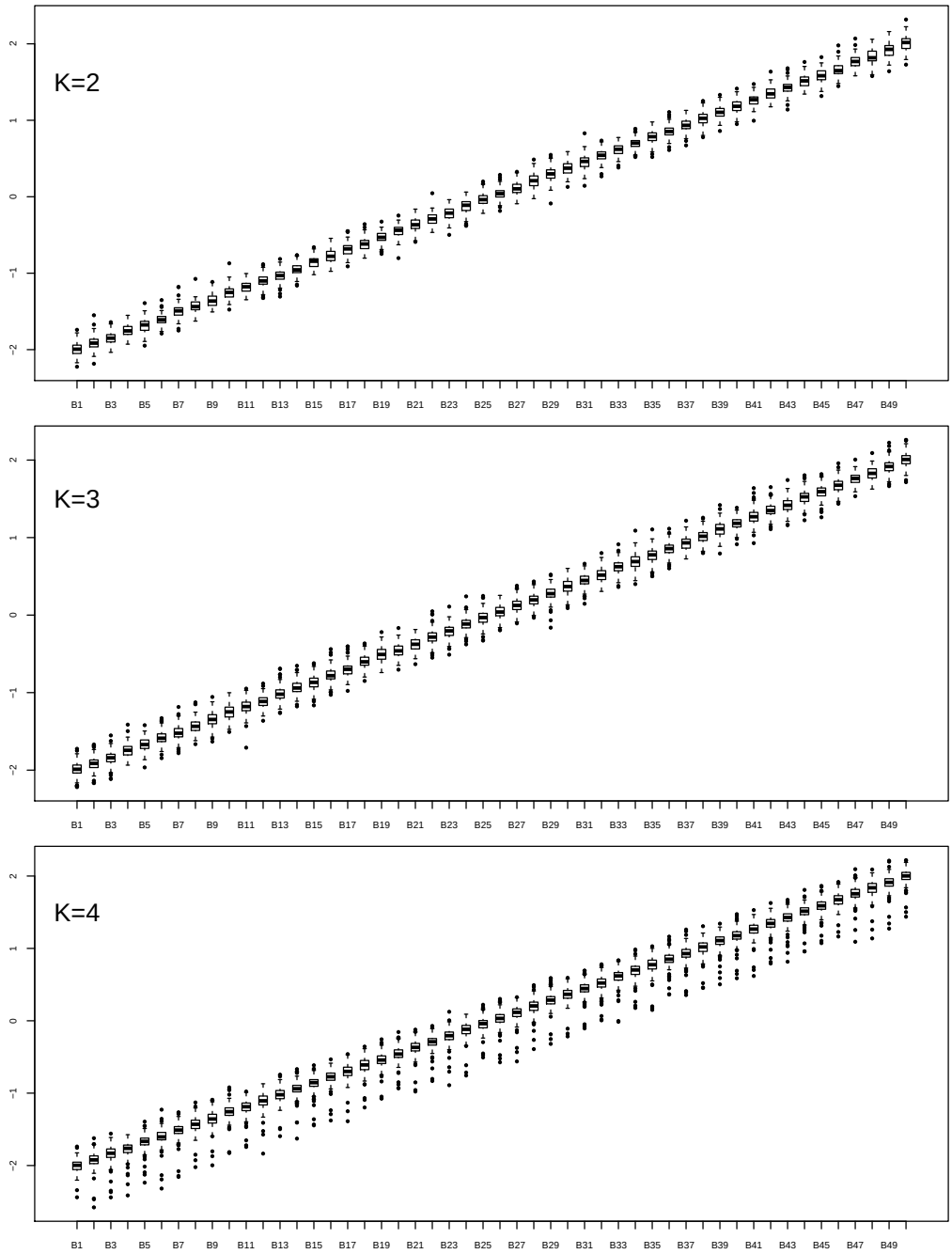

Figure 7: Simulation study: SPC estimates of the β parameters in the scenarios with $J = 50$.

	$K=2$		$K=3$		$K=4$	All K
	mean	mse	mean	mse	mean	mse
$\hat{\beta}_1$	-1.99	0.0088	-1.99	0.0080	-2.00	0.0077
$\hat{\beta}_2$	-1.91	0.0086	-1.92	0.0079	-1.91	0.0070
$\hat{\beta}_3$	-1.85	0.0071	-1.85	0.0075	-1.83	0.0080
$\hat{\beta}_4$	-1.75	0.0065	-1.75	0.0070	-1.77	0.0067
$\hat{\beta}_5$	-1.69	0.0088	-1.67	0.0068	-1.67	0.0069
$\hat{\beta}_6$	-1.61	0.0059	-1.58	0.0070	-1.60	0.0071
$\hat{\beta}_7$	-1.50	0.0089	-1.52	0.0081	-1.51	0.0062
$\hat{\beta}_8$	-1.44	0.0069	-1.43	0.0083	-1.43	0.0075
$\hat{\beta}_9$	-1.35	0.0073	-1.35	0.0089	-1.35	0.0084
$\hat{\beta}_{10}$	-1.25	0.0075	-1.25	0.0092	-1.25	0.0078
$\hat{\beta}_{11}$	-1.18	0.0060	-1.18	0.0089	-1.19	0.0076
$\hat{\beta}_{12}$	-1.10	0.0070	-1.12	0.0061	-1.10	0.0075
$\hat{\beta}_{13}$	-1.03	0.0072	-1.01	0.0085	-1.02	0.0063
$\hat{\beta}_{14}$	-0.95	0.0067	-0.93	0.0078	-0.94	0.0069
$\hat{\beta}_{15}$	-0.86	0.0057	-0.87	0.0079	-0.86	0.0063
$\hat{\beta}_{16}$	-0.77	0.0074	-0.78	0.0079	-0.77	0.0054
$\hat{\beta}_{17}$	-0.69	0.0083	-0.70	0.0075	-0.69	0.0076
$\hat{\beta}_{18}$	-0.62	0.0073	-0.60	0.0079	-0.61	0.0086
$\hat{\beta}_{19}$	-0.53	0.0061	-0.51	0.0090	-0.54	0.0076
$\hat{\beta}_{20}$	-0.45	0.0069	-0.46	0.0070	-0.45	0.0082
$\hat{\beta}_{21}$	-0.37	0.0075	-0.38	0.0076	-0.36	0.0075
$\hat{\beta}_{22}$	-0.29	0.0064	-0.28	0.0084	-0.29	0.0056
$\hat{\beta}_{23}$	-0.22	0.0075	-0.21	0.0074	-0.21	0.0069
$\hat{\beta}_{24}$	-0.13	0.0088	-0.12	0.0072	-0.12	0.0069
$\hat{\beta}_{25}$	-0.04	0.0058	-0.04	0.0098	-0.04	0.0069
$\hat{\beta}_{26}$	0.04	0.0067	0.04	0.0075	0.04	0.0071
$\hat{\beta}_{27}$	0.11	0.0072	0.12	0.0069	0.11	0.0066
$\hat{\beta}_{28}$	0.21	0.0099	0.20	0.0061	0.20	0.0082
$\hat{\beta}_{29}$	0.30	0.0096	0.27	0.0084	0.29	0.0078
$\hat{\beta}_{30}$	0.37	0.0077	0.37	0.0086	0.37	0.0068
$\hat{\beta}_{31}$	0.46	0.0085	0.45	0.0069	0.45	0.0076
$\hat{\beta}_{32}$	0.54	0.0066	0.52	0.0072	0.52	0.0082
$\hat{\beta}_{33}$	0.61	0.0057	0.62	0.0074	0.62	0.0066
$\hat{\beta}_{34}$	0.70	0.0062	0.69	0.0090	0.70	0.0075
$\hat{\beta}_{35}$	0.78	0.0087	0.77	0.0089	0.78	0.0078
$\hat{\beta}_{36}$	0.85	0.0074	0.86	0.0066	0.86	0.0071
$\hat{\beta}_{37}$	0.94	0.0067	0.93	0.0075	0.93	0.0085
$\hat{\beta}_{38}$	1.02	0.0087	1.02	0.0068	1.02	0.0078
$\hat{\beta}_{39}$	1.11	0.0074	1.11	0.0088	1.11	0.0070
$\hat{\beta}_{40}$	1.18	0.0070	1.18	0.0061	1.18	0.0080
$\hat{\beta}_{41}$	1.26	0.0061	1.27	0.0099	1.27	0.0068
$\hat{\beta}_{42}$	1.35	0.0073	1.35	0.0077	1.35	0.0070
$\hat{\beta}_{43}$	1.42	0.0075	1.42	0.0086	1.43	0.0082
$\hat{\beta}_{44}$	1.51	0.0082	1.52	0.0087	1.51	0.0063
$\hat{\beta}_{45}$	1.58	0.0085	1.59	0.0080	1.59	0.0071
$\hat{\beta}_{46}$	1.66	0.0071	1.67	0.0068	1.67	0.0070
$\hat{\beta}_{47}$	1.77	0.0071	1.76	0.0051	1.76	0.0083
$\hat{\beta}_{48}$	1.83	0.0084	1.83	0.0070	1.83	0.0077
$\hat{\beta}_{49}$	1.92	0.0086	1.92	0.0081	1.92	0.0064
$\hat{\beta}_{50}$	2.01	0.0102	2.01	0.0078	2.00	0.0067
		0.0075		0.0078		0.0073

Table 3: SPC estimates of the β parameters (mean values) and relative mean squared errors (with $J = 50$).

Bibliography

Agresti, Alan, Caffo, Brian, & Ohman-Strickland, Pamela. 2004. Examples in which misspecification of a random effects distribution reduces efficiency, and possible remedies. *Computational statistics & data analysis*, **47**(3), 639–653.

Aitkin, Murray. 1995. Npml estimation of the mixing distribution in general statistical models with unobserved random effects. *Pages 1–9 of: Statistical modelling*. Springer.

Andersen, Erling B. 1977. Sufficient statistics and latent trait models. *Psychometrika*, **42**(1), 69–81.

Baker, Frank B. 2001. *The basics of Item Response Theory*. ERIC.

Bartholomew, David J, Knott, Martin, & Moustaki, Irini. 2011. *Latent variable models and factor analysis: A unified approach*. Vol. 904. John Wiley & Sons.

Bartolucci, Francesco. 2007. A class of multidimensional irt models for testing unidimensionality and clustering items. *Psychometrika*, **72**(2), 141–157.

Bartolucci, Francesco, Farcomeni, Alessio, & Pennoni, Fulvia. 2012. *Latent markov models for longitudinal data*. CRC Press.

Bartolucci, Francesco, Bacci, Silvia, & Gnaldi, Michela. 2015. *Statistical analysis of questionnaires: a unified approach based on R and Stata*. Vol. 34. CRC Press.

Bartolucci, Francesco, Bacci, Silvia, & of Perugia, Michela Gnaldi University. 2016. *MultiLCIRT: Multidimensional Latent Class Item Response Theory Models*. R package version 2.10.

Berger, Moritz, & Tutz, Gerhard. 2018. Tree-structured clustering in fixed effects models. *Journal of computational and graphical statistics*, **0**(0), 1–23.

Bertsekas, Dimitri P. 1999. *Nonlinear programming.* Athena scientific Belmont.

Birnbaum, Allan. 1968. Some latent trait models and their use in inferring an examinee's ability. *Statistical theories of mental test scores.*

Borsboom, Denny, Mellenbergh, Gideon J, & Van Heerden, Jaap. 2003. The theoretical status of latent variables. *Psychological review*, **110**(2), 203.

Child, Dennis. 2006. *The essentials of factor analysis.* A&C Black.

Clarke, Paul, Crawford, Claire, Steele, Fiona, & Vignoles, Anna F. 2010. The choice between fixed and random effects models: some considerations for educational research.

Crocker, Linda, & Algina, James. 1986. *Introduction to classical and modern test theory.* ERIC.

Dempster, Arthur P, Laird, Nan M, & Rubin, Donald B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. series b (methodological)*, 1–38.

Efron, Bradley. 1979. Bootstrap methods: another look at the jackknife. *Annals of statistics*, **7**, 1–26.

Efron, Bradley, & Tibshirani, Robert. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, 54–75.

Efron, Bradley, Hastie, Trevor, Johnstone, Iain, Tibshirani, Robert, *et al.* 2004. Least angle regression. *The annals of statistics*, **32**(2), 407–499.

Fan, Jianqing, & Li, Runze. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the american statistical association*, **96**(456), 1348–1360.

Fischer, Gerhard H. 1981. On the existence and uniqueness of maximum-likelihood estimates in the Rasch model. *Psychometrika*, **46**(1), 59–77.

Fischer, Gerhard H, & Molenaar, Ivo W. 2012. *Rasch models: Foundations, recent developments, and applications.* Springer Science & Business Media.

Formann, Anton K. 1995. Linear logistic latent class analysis and the Rasch model. *Pages 239–255 of: Rasch models*. Springer.

Fraley, Chris, & Raftery, Adrian E. 1998. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, **41**(8), 578–588.

Frank, LLdiko E, & Friedman, Jerome H. 1993. A statistical view of some chemometrics regression tools. *Technometrics*, **35**(2), 109–135.

Friedman, Jerome, Hastie, Trevor, & Tibshirani, Robert. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

Friedman, Jerome, Hastie, Trevor, Höfling, Holger, Tibshirani, Robert, *et al.* 2007. Pathwise coordinate optimization. *The annals of applied statistics*, **1**(2), 302–332.

Friedman, Jerome, Hastie, Trevor, Simon, S., & Tibshirani, Rob. 2015. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, **2.0**.

Fu, Wenjiang J. 1998. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, **7**(3), 397–416.

Gardiner, Joseph C, Luo, Zhehui, & Roman, Lee Anne. 2009. Fixed effects, random effects and gee: what are the differences? *Statistics in medicine*, **28**(2), 221–239.

Goodman, Leo A. 1974. Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, **61**(2), 215–231.

Hambleton, Ronald K, & Swaminathan, Hariharan. 1985. *Item Response Theory: Principles and applications*. Vol. 7. Springer Science & Business Media.

Hambleton, Ronald K, & Swaminathan, Hariharan. 2013. *Item Response Theory: Principles and applications*. Springer Science & Business Media.

Hastie, Trevor, Tibshirani, Robert, & Wainwright, Martin. 2015. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press.

Heagerty, Patrick J, & Kurland, Brenda F. 2001. Misspecified maximum likelihood estimates and generalised linear mixed models. *Biometrika*, 973–985.

Hocking, Toby Dylan, Joulin, Armand, Bach, Francis, & Vert, Jean-Philippe. 2011. Clusterpath: an algorithm for clustering using convex fusion penalties. *Page 1 of: 28th international conference on machine learning.*

Hoerl, Arthur E, & Kennard, Robert W. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67.

Jacob, Laurent, Obozinski, Guillaume, & Vert, Jean-Philippe. 2009. Group lasso with overlap and graph lasso. *Pages 433–440 of: Proceedings of the 26th annual international conference on machine learning.* ACM.

James, Gareth, Witten, Daniela, Hastie, Trevor, & Tibshirani, Robert. 2013. *An introduction to statistical learning.* Vol. 6. Springer.

Kelava, Augustin, Kohler, Michael, KrzyÅijak, Adam, & Schafland, Tim Fabian. 2017. Nonparametric estimation of a latent variable model. *Journal of multivariate analysis*, **154**, 112 – 134.

Knight, Keith, & Fu, Wenjiang. 2000. Asymptotics for lasso-type estimators. *Annals of statistics*, 1356–1378.

Kyung, Minjung, Gill, Jeff, Ghosh, Malay, Casella, George, *et al.* 2010. Penalized regression, standard errors, and bayesian lassos. *Bayesian analysis*, **5**(2), 369–411.

Laird, Nan. 1978. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the american statistical association*, **73**(364), 805–811.

Lazarsfeld, Paul Felix, Henry, Neil W, & Anderson, Theodore Wilbur. 1968. *Latent structure analysis.* Houghton Mifflin Boston.

Lindsay, Bruce, Clogg, Clifford C, & Grego, John. 1991. Semiparametric estimation in the rasch model and related exponential response models, including a simple latent class model for item analysis. *Journal of the american statistical association*, **86**(413), 96–107.

Lindsay, Bruce G. 1995. Mixture models: theory, geometry and applications. *Pages i–163 of: Nsf-cbms regional conference series in probability and statistics*. JSTOR.

Lindsten, Fredrik, Ohlsson, Henrik, & Ljung, Lennart. 2011. *Just relax and come clustering!: A convexification of k-means clustering*. Linköping University Electronic Press.

Litière, Saskia, Alonso, Ariel, & Molenberghs, Geert. 2008. The impact of a misspecified random-effects distribution on the estimation and the performance of inferential procedures in generalized linear mixed models. *Statistics in medicine*, **27**(16), 3125–3144.

Marchetti, Yuliya, Zhou, Qing, *et al.* 2014. Solution path clustering with adaptive concave penalty. *Electronic journal of statistics*, **8**(1), 1569–1603.

Masters, Geofferey N, & Wright, Benjamin D. 1984. The essential process in a family of measurement models. *Psychometrika*, **49**(4), 529–544.

McCullagh, Peter. 1984. Generalized linear models. *European journal of operational research*, **16**(3), 285–292.

McCulloch, Charles E, & Neuhaus, John M. 2001. *Generalized linear mixed models*. Wiley Online Library.

McCulloch, Charles E, & Neuhaus, John M. 2011. Misspecifying the shape of a random effects distribution: why getting it wrong may not matter. *Statistical science*, 388–402.

McLachlan, Geoffrey, & Peel, David. 2004. *Finite mixture models*. John Wiley & Sons.

Meier, Lukas, Van De Geer, Sara, & Bühlmann, Peter. 2009. The group lasso for logistic regression. *Journal of the royal statistical society: Series b (statistical methodology)*, **70**(1), 53–71.

Müller, Hans. 1987. A rasch model for continuous ratings. *Psychometrika*, **52**(2), 165–181.

Nelder, John Ashworth, & Baker, R Jacob. 1972. *Generalized linear models*. Wiley Online Library.

Osborne, Michael R, Presnell, Brett, & Turlach, Berwin A. 2000a. A new approach to variable selection in least squares problems. *Ima journal of numerical analysis*, **20**(3), 389–403.

Osborne, Michael R, Presnell, Brett, & Turlach, Berwin A. 2000b. On the lasso and its dual. *Journal of computational and graphical statistics*, **9**(2), 319–337.

Oyeyemi, Gafar Matanmi, Ogunjobi, Eyitayo Oluwole, & Folorunsho, Adeyinka Idowu. 2015. On performance of shrinkage methods–a monte carlo study. *International journal of statistics and applications*, **5**(2), 72–76.

Pan, Wei, Shen, Xiaotong, & Liu, Binghui. 2013. Cluster analysis: unsupervised learning via supervised learning with a non-convex penalty. *The journal of machine learning research*, **14**(1), 1865–1889.

Park, Trevor, & Casella, George. 2008. The bayesian lasso. *Journal of the american statistical association*, **103**(482), 681–686.

Pelckmans, Kristiaan, De Brabanter, Joseph, Suykens, JAK, & De Moor, B. 2005. Convex clustering shrinkage. *In: Pascal workshop on statistics and optimization of clustering workshop*.

Petry, Sebastian, Flexeder, Claudia, & Tutz, Gerhard. 2011. *Pairwise fused lasso*.

Pötscher, Benedikt M, & Leeb, Hannes. 2009. On the distribution of penalized maximum likelihood estimators: The lasso, scad, and thresholding. *Journal of multivariate analysis*, **100**(9), 2065–2082.

Puig, Arnau Tibau, Wiesel, Ami, & Hero III, Alfred O. 2009. A multidimensional shrinkage-thresholding operator. *Pages 113–116 of: Statistical signal processing, 2009. ssp'09. ieee/sp 15th workshop on*. IEEE.

Rasch, Georg. 1960. Probabilistic models for some intelligence and achievement tests. *Copenhagen: Danish institute for educational research*.

Rasch, Georg. 1961. On general laws and the meaning of measurement in psychology. *Pages 321–333 of: Proceedings of the fourth berkeley symposium on mathematical statistics and probability*, vol. 4.

Raykov, Tenko, & Marcoulides, George A. 2011. *Introduction to psychometric theory*. Routledge.

Rizopoulos, Dimitris. 2006. ltm: An R package for latent variable modelling and Item Response Theory analyses. *Journal of statistical software*, **17**(5), 1–25.

Rost, Jürgen. 1990. Rasch models in latent classes: An integration of two approaches to item analysis. *Applied psychological measurement*, **14**(3), 271–282.

She, Yiyuan, *et al.* 2010. Sparse regression with exact clustering. *Electronic journal of statistics*, **4**, 1055–1096.

Shen, Xiaotong, Pan, Wei, & Zhu, Yunzhang. 2012. Likelihood-based selection and sharp parameter estimation. *Journal of the american statistical association*, **107**(497), 223–232.

Simon, Noah, Friedman, Jerome, Hastie, Trevor, & Tibshirani, Robert. 2013. A sparse-group lasso. *Journal of computational and graphical statistics*, **22**(2), 231–245.

Skrondal, Anders, & Rabe-Hesketh, Sophia. 2004. *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models.* Crc Press.

Skrondal, Anders, & Rabe-Hesketh, Sophia. 2007. Latent variable modelling: a survey. *Scandinavian journal of statistics*, **34**(4), 712–745.

Spearman, Charles. 1904. General Intelligence, objectively determined and measured. *The american journal of psychology*, **15**(2), 201–292.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the royal statistical society. series b (methodological)*, 267–288.

Tibshirani, Robert, Saunders, Michael, Rosset, Saharon, Zhu, Ji, & Knight, Keith. 2005. Sparsity and smoothness via the fused lasso. *Journal of the royal statistical society: Series b (statistical methodology)*, **67**(1), 91–108.

Tibshirani, Ryan Joseph, Taylor, Jonathan E, Candes, Emmanuel Jean, & Hastie, Trevor. 2011. The solution path of the generalized lasso. *Annals of statistics*, **39**(3), 1335–1371.

Townsend, Zac, Buckley, Jack, Harada, Masataka, & Scott, Marc A. 2013. The choice between fixed and random effects. *The sage handbook of multilevel modeling. SAGE.*

Tutz, Gerhard, & Oelker, Margret-Ruth. 2017. Modelling clustered heterogeneity: Fixed effects, random effects and mixtures. *International statistical review*, **85**(2), 204–227.

van der Linden, Wim J, & Hambleton, Ronald K. 2013. *Handbook of modern Item Response Theory*. Springer Science & Business Media.

Ward Jr, Joe H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the american statistical association*, **58**(301), 236–244.

Yuan, Ming, & Lin, Yi. 2007. Model selection and estimation in regression with grouped variables. *Journal of the royal statistical society: Series b (statistical methodology)*, **68**(1), 49–67.

Zhang, Cun-Hui, *et al.* 2010. Nearly unbiased variable selection under minimax concave penalty. *The annals of statistics*, **38**(2), 894–942.

Zou, Hui. 2006. The adaptive lasso and its oracle properties. *Journal of the american statistical association*, **101**(476), 1418–1429.

Zou, Hui, & Hastie, Trevor. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: Series b (statistical methodology)*, **67**(2), 301–320.

Zucchini, Walter, & MacDonald, Iain L. 2009. *Hidden markov models for time series: an introduction using R*. CRC press.