A likelihood-based boosting algorithm for factor analysis models with binary data

A Likelihood-Based Boosting Algorithm for Factor Analysis Models with Binary Data

Michela Battauz[a,*], Paolo Vidoni[a]

[a] *Department of Economics and Statistics, University of Udine, via Tomadini 30/A 33100 Udine, Italy*

Abstract

Statistical boosting represents a very effective method for fitting complex models, while performing variable selection and preventing overfitting at the same time. However, the available methods are not directly applicable to factor analysis models for binary data, since any gradient descent method is not able to move from the starting point with zero loadings. The proposed algorithm, exploiting the directions of negative curvature of the log-likelihood function, is able to escape from the regions of local non-convexity. The component-wise approach followed leads to a sparse solution, which has the advantage of facilitating the interpretation without requiring a posterior rotation of the loadings. The method also performs regularization of the estimates, hence reducing their mean square error. To reduce the computational burden of the inferential procedure, a suitable pseudolikelihood, called pairwise likelihood, is exploited. In addition, a group lasso penalty is considered in order to automatically select the number of latent variables included in the model. The good performance of the proposal is illustrated through a simulation study and a real-data example.

Keywords: Latent variable models, Pairwise Likelihood, Negative curvature direction, Regularization, Sparse solution.

*Corresponding author
Email address: michela.battauz@uniud.it (Michela Battauz)

1. Introduction

Latent trait models (see, for example, Bartholomew et al., 2011) are particular cases in the general class of latent variable models, where the response variable is categorical, nominal or ordinal, and the continuous latent variables describe unobserved features of the statistical units. These models are frequently used in social sciences, where the latent variables describe attitudes, abilities, beliefs or behaviour of the subjects involved in the experiment. Namely, they refer to relevant characteristics of the subjects, called traits, hence the name commonly used for this type of models. Furthermore, the latent variables are supposed to account for the dependencies among the observed variables, so that these are assumed to be conditionally independent given the unknown value of the latent ones. In this paper we have in mind applications related to the field of educational, psychological or behavioural testing and the observed categorical variables may represent the answer to a series of questions, also called items, asked a certain group of individuals. Focusing on binary variables, two main approaches can be identified for model specification and inference.

The first approach refers to the context of the generalized linear mixed models and it is based on a direct, regression-like modelling of the probability of the response patterns. We mention the simple and the multidimensional item response theory (IRT) models where the probability of each response is described as a function of the latent traits and of some item parameters. With a specific choice of the link function, we obtain the well-known logit/normal models (such as the one-, two- and three-parameter logistic IRT models) and the probit/normal models (see, for example, Bartholomew et al., 2011; Reckase, 2009; Reise & Revicki, 2014).

The second approach for dealing with binary or ordinal variables assumes that they are generated by underlying response variables (URV) that follow a factor analysis model. More precisely, in case of binary data, each observed response is interpreted as an indicator of whether a variable, defined within a normal factor model, is below or above a suitable threshold value (Bartholomew

et al., 2011; Jöreskog & Moustaki, 2001). Although the two approaches seem quite different, in some cases, such as for the probit/normal model, it is possible to specify the same model under the URV approach (see, for example, Bartholomew et al., 2011, Section 4.4).

Various inferential procedures have been proposed for the estimation of the simple and the multidimensional IRT models (Bock & Aitkin, 1981; Schilling & Bock, 2005; Cai, 2010; Béguin & Glas, 2001) and for the estimation of the models defined within the factor analysis approach (see, for example, Katsikatsou et al., 2012, and the references therein). However, in both cases, the complexity of the model calibration techniques, with the associated computational burden, is surely considerable and it reduces the applicability of the classical inferential procedures. In particular, high-dimensional integration is required and it turns out to be computationally demanding or even infeasible as the number of observed variables, under the URV approach, or the number of latent traits, under the IRT approach, increases. A further relevant aspect, related to the complexity of the models, is the need of obtaining interpretable solutions based on a restricted number of non-zero loadings, namely the coefficients of the latent variables. To this end, different lasso-based inferential techniques are proposed by Sun et al. (2016) and Battauz (2020) in the IRT framework.

In the present paper we adopt the URV approach and we concentrate on factor analysis models for binary data. Following Katsikatsou et al. (2012), we reduce the computational burden of the inferential procedure using a simple solution based on a pseudolikelihood, called pairwise likelihood, which belongs to the general class of composite likelihoods (Lindsay, 1988; Varin et al., 2011). The novelty of our contribution consists in the integration of the pairwise likelihood objective function within a new statistical boosting procedure, which is proved to be very useful for an efficient calibration of complex latent trait models. Furthermore, a group lasso penalty is added to the pairwise log-likelihood function in order to avoid the algorithm to introduce redundant latent variables in the model. Starting from a model without latent factors, the algorithm aims at selecting the most important latent traits and a reduced number of non-zero

loadings. This automatically reduces the number of parameters to estimate, limits their variability, and provides more interpretable solutions. The method yields a unique sparse solution, so, although a posterior rotation of the loadings is always possible, such rotation would increase the number of non-zero loadings tending to make more difficult the interpretation. Due to the non-convexity of the objective function, the statistical boosting methods proposed in the literature (for example, Friedman, 2001; Tutz & Binder, 2006) are not straightforward to extend to the models of interest in this paper. Therefore, a novel boosting approach based on directions of negative curvature is proposed (Gould et al., 2000).

The paper is organized as follows. A review of the statistical boosting methods and an introduction to latent trait models for binary data and their likelihood-based inferential procedures are given in Section 2. In Section 3 a new boosting algorithm is proposed and the computational aspects of the procedure are discussed. The method is then applied to simulated and real data in Section 4. Finally, some concluding remarks are given in Section 5.

2. Preliminaries

2.1. A review on boosting procedures

Boosting algorithms were originally developed in the field of machine learning (see, for example, Freund & Schapire, 1996; Schapire & Freund, 2012), with the aim of combining weak classifiers in order to obtain, step by step, a final weighted classifier with a strong discriminating ability. These iterative methods have been successively translated into the field of statistical modeling for selecting and estimating the effect of predictors on the response variable (see Friedman et al., 2000; Friedman, 2001). In particular, the so-called componentwise boosting methods provide an iterative updating procedure for the estimates of the parameters or, more generally, of the weights of the base-learners, in order to maximize or minimize a suitable objective function. The main context of application is structured additive regression models. With a proper choice

of the number of boosting iterations regularization is automatically included in the fitting procedure, and this can be useful when the large number of model parameters does not allow the use of classical inferential techniques.

Two main statistical boosting approaches are proposed in the literature (see, for example, Mayr et al., 2014): gradient boosting and likelihood-based boosting. The first one is more general and the aim is to minimize a suitable loss function by using a gradient descent procedure (see Friedman, 2001, for an in-depth discussion). Using the component-wise version, the base-learners are fitted one by one to the current gradient of the objective function and they in turn contribute to optimize the loss function step by step. The second one, proposed by Tutz & Binder (2006), considers the negative log-likelihood, or a suitable generalization, as the loss function and the optimization involves the Fisher scoring algorithm. In the component-wise version, the base-learners are estimated step by step in order to minimize, via a single iteration of penalized Fisher scoring, the negative log-likelihood function, where the current estimates of the model parameters specify an offset. The two approaches are equivalent in the special case of a Gaussian regression model (De Bin, 2016).

2.2. Latent trait models for binary data

We consider, in particular, models for dichotomous response outcomes, which may be useful for items with two score categories, where 1 usually corresponds to a correct or endorsement response and 0 otherwise. The response variable is then a Bernoulli random variable Y_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, J$, representing the score of subject i on item j. The variables related to subject i are collected in the J-dimensional random vector $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iJ})^\top$, leading to 2^J distinct response patterns. Let $\boldsymbol{\theta}_i = (\theta_{i1}, \ldots, \theta_{iD})^\top$, $i = 1, \ldots, n$, be a D-dimensional latent random vector, with $D \geq 1$, describing traits of subject i. It is customary to assume that $\boldsymbol{\theta}_i$ follows a D-dimensional standard normal distribution, in symbols $\boldsymbol{\theta}_i \sim N_D(\mathbf{0}, \mathbf{I})$, with $\mathbf{0}$ the null (column) vector and $\mathbf{I}$ the identity matrix. Furthermore, $(\mathbf{Y}_i, \boldsymbol{\theta}_i)$, $i = 1, \ldots, n$, are supposed to be identically distributed and independent across the subjects. In the following, for simplifying

the notation, the subscript i is omitted when it is not necessary.

Adopting the URV approach (Bartholomew et al., 2011; Jöreskog & Moustaki, 2001), the vector of dichotomous observed variables $\mathbf{Y}$ is specified as a partial observation of an underlying vector of continuous variables $\mathbf{Y}^* = (Y_1^*, \ldots, Y_J^*)^\top$, defined under a suitable normal factor model. More precisely, each binary variable Y_j, $j = 1, \ldots, J$, is the indicator of whether the corresponding Y_j^* is above or below a threshold parameter τ_j, that is

$$Y_j = \begin{cases} 1 & \text{if } Y_j^* \geq \tau_j \\ 0 & \text{otherwise.} \end{cases}$$

The factor model is defined as

$$\mathbf{Y}^* = \boldsymbol{\Lambda}\boldsymbol{\theta} + \boldsymbol{\delta},$$

where $\boldsymbol{\Lambda} = [\lambda_{jd}]$ is the $J \times D$ matrix of loadings, $\boldsymbol{\theta}$ is the D-dimensional vector of latent variables defined above, also called the common factors, and $\boldsymbol{\delta} \sim N_J(\mathbf{0}, \boldsymbol{\Psi})$ is the vector of unique factors, assumed to be mutually uncorrelated and uncorrelated with $\boldsymbol{\theta}$. It is possible to generalize the model by assuming that the off-diagonal elements of the variance matrix of $\boldsymbol{\theta}$ are non-null (Katsikatsou et al., 2012), though this is possible only within the framework of confirmatory factor analysis. Finally, the variance matrix of $\boldsymbol{\delta}$ is diagonal and such that $\boldsymbol{\Psi} = \mathbf{I} - \text{diag}(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top)$. Then, we conclude that $\mathbf{Y}^* \sim N_J(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}$. Note that the mean and the variance of Y_j^*, $j = 1, \ldots, J$, are equal to 0 and 1, respectively. This is a common assumption in models for binary responses, where the mean and variance of the underlying variables are not identifiable. Moreover, we have that

$$\rho_{jl} = Cor(Y_j^*, Y_l^*) = \sum_{d=1}^{D} \lambda_{jd}\lambda_{ld}, \qquad j, l = 1, \ldots, J,\ j \neq l. \tag{1}$$

It is well-known that the matrix of loadings $\boldsymbol{\Lambda}$ is determined up to an orthogonal transformation, so that there are only $JD - D(D-1)/2$ free non-redundant parameters. The vector of unknown model parameters is $\boldsymbol{\gamma} = (\boldsymbol{\tau}^\top, \boldsymbol{\lambda}^\top)^\top$, with $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_J)^\top$ the thresholds and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_D^\top)^\top$, with $\boldsymbol{\lambda}_d = (\lambda_{1d}, \ldots, \lambda_{Jd})^\top$,

$d = 1, \ldots, D$, the loadings. In the context of educational assessment, the threshold τ_j describes, in some sense, the difficulty of item j and $\lambda_{jd} \in \mathbb{R}$, $d = 1, \ldots, D$, are called discrimination parameters, measuring how the variations in the latent traits influence the expected item score. In order to simplify the exposition, we set a vector $\boldsymbol{\gamma}$ with dimension $J(D+1)$, even if we are aware that there are only $J(D+1) - D(D-1)/2$ independent parameters. Nevertheless, in many applications, the dimension of the parameter vector can still be very large.

As mentioned in Section 1, this model is equivalent to the probit/normal model defined in the generalized linear mixed models setting (Bartholomew et al., 2011, Section 4.4), and then it is quite close to the well-known multidimensional two-parameter logistic IRT model.

2.3. Likelihood-based inferential procedures

Given the observed item responses $\bar{\mathbf{y}} = (\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top)^\top$, interpreted as observations of the full random vector $\bar{\mathbf{Y}} = (\mathbf{Y}_1^\top, \ldots, \mathbf{Y}_n^\top)^\top$, the log-likelihood for $\boldsymbol{\gamma}$ is

$$\ell(\boldsymbol{\gamma}; \bar{\mathbf{y}}) = \log L(\boldsymbol{\gamma}; \bar{\mathbf{y}}) = \sum_{r=1}^{R} n_r \log \pi_r(\boldsymbol{\gamma}), \tag{2}$$

where n_r and $\pi_r(\boldsymbol{\gamma})$, $r = 1, \ldots, R$, are the observed frequency and the probability of the r-th response pattern, respectively, with $R = 2^J$, $\sum_{r=1}^{R} n_r = n$, $\pi_r(\boldsymbol{\gamma}) \geq 0$ and $\sum_{r=1}^{R} \pi_r(\boldsymbol{\gamma}) = 1$.

Under the model defined above, the probability of each single response pattern can be obtained as the solution of a J-dimensional integral. For example, if the r-th response pattern corresponds to the null vector the associated probability corresponds to

$$\pi_r(\boldsymbol{\gamma}) = \int_{-\infty}^{\tau_1} \cdots \int_{-\infty}^{\tau_J} \phi_J(\mathbf{y}^*; \mathbf{0}, \boldsymbol{\Sigma})\, d\mathbf{y}^*, \tag{3}$$

with $\phi_J(\mathbf{y}^*; \mathbf{0}, \boldsymbol{\Sigma})$ denoting the density of a $N_J(\mathbf{0}, \boldsymbol{\Sigma})$ distribution.

Since the J-dimensional integral in (3) does not admit an explicit solution, suitable analytical or numerical approximations are needed. This fact makes the calculation and the direct maximization of the log-likelihood function (2) very

complex. Therefore, for moderate or large values of the number of items J, the usual likelihood-based inferential approach turns out to be too computationally demanding or even infeasible. Alternative estimation procedures have been introduced (see Jöreskog & Moustaki, 2001; Katsikatsou et al., 2012, for a brief review). However, in this paper, we adopt an alternative approach based on a suitable pseudolikelihood, called pairwise likelihood, which has the merit of reducing the computational burden of the inferential procedure (see, in particular, de Leon, 2005; Katsikatsou et al., 2012, and the references therein).

The general idea is simple and it consists in defining a surrogate of the full likelihood based on the composition of bivariate likelihood objects. Thus, the inferential procedure involves the pairwise log-likelihood as objective function

$$\begin{aligned} p\ell(\boldsymbol{\gamma}; \bar{\mathbf{y}}) &= \sum_{j<l} \log L(\boldsymbol{\gamma}; (y_j, y_l)) \\ &= \sum_{j<l} \sum_{s=0}^{1} \sum_{t=0}^{1} n_{st}^{jl} \log \pi_{st}^{jl}(\boldsymbol{\gamma}), \end{aligned} \tag{4}$$

where n_{st}^{jl} and $\pi_{st}^{jl}(\boldsymbol{\gamma})$, $s, t = 0, 1$, $j, l = 1, \ldots, J$, $j \neq l$, are the observed frequency and the probability of the response (s, t), respectively, for the pair of item score variables (Y_j, Y_l). The latter can be simply obtained by solving a bivariate integral, with a significant reduction of the computational burden. For example, if we consider a pair of zero responses, we have

$$\begin{aligned} \pi_{00}^{jl}(\boldsymbol{\gamma}) &= P(Y_j = 0, Y_l = 0; \boldsymbol{\gamma}) = \Phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl}) \\ &= \int_{-\infty}^{\tau_j} \int_{-\infty}^{\tau_l} \phi_2(y_j^*, y_l^*; \mathbf{0}, \boldsymbol{\Sigma}_{jl})\, dy_j^* dy_l^*, \end{aligned}$$

where $\Phi_2(\cdot, \cdot; \mathbf{0}, \boldsymbol{\Sigma}_{jl})$ and $\phi_2(\cdot, \cdot; \mathbf{0}, \boldsymbol{\Sigma}_{jl})$ are the distribution function and the density function, respectively, of a bivariate normal distribution with marginal mean values and variances equal to 0 and 1 and correlation ρ_{jl}; $\boldsymbol{\Sigma}_{jl}$ is the submatrix of $\boldsymbol{\Sigma}$ corresponding to the j-th and l-th components. Thus, whatever the number of observed variables is, the specification of the pairwise log-likelihood only requires the evaluation of two-dimensional normal integrals. The pairwise likelihood is a well-known pseudolikelihood which belongs to the wide class of

composite likelihoods (see, for example, Lindsay, 1988; Varin et al., 2011), specified by composing likelihood-type objects, usually related to simple marginal or conditional events.

The maximum pairwise likelihood estimator $\widehat{\boldsymbol{\gamma}}_p$, obtained maximizing the pairwise log-likelihood (4) with respect to $\boldsymbol{\gamma}$, shares the general properties of the maximum composite likelihood estimators (Varin, 2008; Varin et al., 2011). Let $\nabla p\ell(\boldsymbol{\gamma}; \bar{\mathbf{y}})$ and $\nabla^2 p\ell(\boldsymbol{\gamma}; \bar{\mathbf{y}})$ be the gradient vector and the Hessian matrix of the pairwise log-likelihood, respectively. Since the gradient is defined as the sum of the gradients of the bivariate log-likelihood components $\log L(\boldsymbol{\gamma}; (y_j, y_l))$, it is easy to see that, under suitable regularity conditions, the associated estimating equation $\nabla p\ell(\boldsymbol{\gamma}; \bar{\mathbf{y}}) = 0$ is unbiased. In regular problems, $\widehat{\boldsymbol{\gamma}}_p$ is consistent and asymptotically normal distributed with asymptotic mean $\boldsymbol{\gamma}$ and variance matrix $G(\boldsymbol{\gamma})^{-1}$, defined as the inverse of the Godambe information matrix. More precisely,

$$G(\boldsymbol{\gamma}) = I(\boldsymbol{\gamma})^\top J(\boldsymbol{\gamma})^{-1} I(\boldsymbol{\gamma}),$$

where $J(\boldsymbol{\gamma}) = Var\{\nabla p\ell(\boldsymbol{\gamma}; \mathbf{Y})\}$ is the variability matrix and $I(\boldsymbol{\gamma}) = E\{\nabla^2 p\ell(\boldsymbol{\gamma}; \mathbf{Y})\}$ is the sensitivity matrix, where the gradient vector and the Hessian matrix are now related to a single individual component $\mathbf{Y}$. Note that, in the composite likelihood setting, the information identity $I(\boldsymbol{\gamma}) = -J(\boldsymbol{\gamma})$ does not hold, and this usually causes a loss of efficiency with respect to the maximum likelihood estimator. An estimate for the Godambe matrix, useful also for specifying the estimated standard errors, can be obtained using suitable sample estimates for $J(\boldsymbol{\gamma})$ and $I(\boldsymbol{\gamma})$ (Katsikatsou et al., 2012, Section 3). However, estimating $J(\boldsymbol{\gamma})$ is particularly problematic, since the naive empirical estimator is usually numerically unstable, so that alternative inferential strategies are required. This problem also occurs in our model framework. Moreover, since the boosting algorithm, presented in the following section, produces a regularized version of the maximum pairwise likelihood estimator, which does not usually coincide with $\widehat{\boldsymbol{\gamma}}_p$, the associated asymptotic variance matrix does not correspond to $G(\boldsymbol{\gamma})$. This in fact does not allow the calculation of the estimated standard errors via

the estimated Godambe matrix and the direct use of composite likelihood information criteria (Gao & Song, 2010; Varin & Vidoni, 2005) for model selection issues.

3. Boosting in Factor Analysis Models with Binary Data

3.1. A new generalized boosting algorithm

The likelihood-based boosting algorithm introduced in this paper aims at estimating a statistical model using a stepwise maximum pairwise likelihood approach, where regularization and variable selection are automatically included. This can be useful in case of a large number of unknown parameters, as is the case of the latent trait models mentioned in the previous section. In particular, we follow a component-wise strategy where, starting with a model containing the threshold terms, only a small subset of parameters (two in our proposal) is updated in a single iteration step.

In the wide class of regression-type models, the classical likelihood-based, component-wise boosting algorithm minimizes the objective function using as offset the linear predictor obtained from the previous iteration step. Indeed, in every boosting step, each single component of the linear term is considered for optimizing the objective function using one step of Fisher scoring. Only the component which leads to the largest decrease of the objective function is selected for updating the linear term, with a suitable penalty factor in order to have small parameter updates.

Our proposal is inspired by Gould et al. (2000), where a new adaptive line-search algorithm is proposed for solving multidimensional unconstrained optimization (minimization) problems. Suitable conditions are stated in order to assure global convergence to second-order critical points, namely points where the objective function presents a null gradient and a positive semidefinite Hessian matrix. The distinctive feature of this algorithm is that it exploits any local non-convexity of the objective function and, to this end, both the information contained in the gradient vector and in the Hessian matrix are employed. More

precisely, at each iteration a pair of alternative search directions are defined: a classical Newton-type direction, involving the gradient vector, and a negative curvature direction, which is based on the Hessian matrix and enables moving away from regions of local non-convexity. Only one direction between the two is selected, and the procedure points to the one giving the largest decrease along the quadratic approximation of the objective function. Since we are looking for a component-wise procedure that updates only a subset of the model parameters at a time, the algorithm of Gould et al. (2000) is particularly suited for the task, while other proposals exploiting both directions at each step would not be helpful for our objective (see McCormick, 1977, for an early influential work). Therefore, we adapted the result of Gould et al. (2000), reviewed in Appendix A, to the boosting framework and, following a component-wise procedure, each iteration step updates a pair of parameter estimates, selected according to a Newton direction or to a negative curvature direction. As the number of iterations increases, the algorithm converges to the maximum pairwise likelihood estimates and then, for regularization purposes, a suitable stopping criterion has to be defined. In what follows, we consider the Newton direction, where the gradient is rescaled using the Hessian matrix, namely the observed information matrix. However, the Fisher scoring alternative, where the expected information matrix is considered, could be used instead.

The objective function considered is the negative penalized pairwise log-likelihood

$$-p\ell^p(\boldsymbol{\gamma}) = -p\ell(\boldsymbol{\gamma}) + \eta \sum_{d=1}^{D} (\boldsymbol{\lambda}_d^\top \boldsymbol{\lambda}_d)^{1/2}, \tag{5}$$

where $p\ell(\boldsymbol{\gamma}) = p\ell(\boldsymbol{\gamma}; \bar{\mathbf{y}})$ is specified in (4), and the additive term is given by the tuning parameter $\eta \geq 0$ multiplied by a group lasso penalty (Yuan & Lin, 2006), with groups given by the loadings related to each latent variable. This penalty encourages sparsity at the group level, so that the whole vector $\boldsymbol{\lambda}_d$ is forced to zero. The choice of this penalty is discussed in Section 3.2. As suggested by Tutz & Gertheiss (2014), in order to have a differentiable function, the lasso-type penalty is approximated by adding a small constant to $\boldsymbol{\lambda}_d^\top \boldsymbol{\lambda}_d$, $d = 1, \ldots, D$, in

Equation (5). The use of penalized objective functions is not new in the boosting literature. In particular, Bühlmann & Yu (2006) used penalized residual sum of squares to obtain sparser solutions. The gradient vector $\mathbf{g}$ and Hessian matrix $\mathbf{H}$ of the objective function (5) exist and are continuous functions. We indicate with $\mathbf{g}_{bc}$ and $\mathbf{H}_{bc}$ the subvector of $\mathbf{g}$ and the submatrix of $\mathbf{H}$ related to the pair of parameters (γ_b, γ_c). Moreover, $\widehat{\boldsymbol{\tau}}$ is the J-dimensional vector containing the maximum likelihood estimates for the threshold parameters, assuming $\boldsymbol{\lambda}_d = \mathbf{0}$, $d = 1, \ldots, D$, i.e. the parameters of J univariate probit regression models.

The new likelihood-based boosting algorithm is defined as follows.

1. Set the iteration counter $m = 0$ and the number of latent variables $D = 1$. Initialize the vector of parameters $\widehat{\boldsymbol{\gamma}}^{(0)} = (\widehat{\boldsymbol{\tau}}^\top, \mathbf{0}^\top)^\top$, where $\mathbf{0}$ is a null (column) vector with dimension J.
   Iterate the following steps 2-8 until $m = m_{stop}$.
2. Set $m = m + 1$.
3. Compute the gradient vector and the Hessian matrix at $\boldsymbol{\gamma} = \widehat{\boldsymbol{\gamma}}^{(m-1)}$, that is $\widehat{\mathbf{g}}^{(m-1)} = \mathbf{g}(\widehat{\boldsymbol{\gamma}}^{(m-1)})$ and $\widehat{\mathbf{H}}^{(m-1)} = \mathbf{H}(\widehat{\boldsymbol{\gamma}}^{(m-1)})$, respectively.
4. For each pair of parameters, compute the Newton-type directions

$$\mathbf{s}_{bc}^{(m)} = (s_b^{(m)}, s_c^{(m)})^\top = -\widehat{\mathbf{H}}_{bc}^{(m-1)^{-1}} \widehat{\mathbf{g}}_{bc}^{(m-1)}, \tag{6}$$

   and the negative curvature directions

$$\mathbf{d}_{bc}^{(m)} = (d_b^{(m)}, d_c^{(m)})^\top = -sign\left\{ \left(\widehat{\mathbf{g}}_{bc}^{(m-1)} \right)^\top \widehat{\mathbf{u}}_{bc}^{(m-1)} \right\} \widehat{\mathbf{u}}_{bc}^{(m-1)}, \tag{7}$$

   with $b, c = 1, \ldots, J(D+1)$, $b < c$, where $\widehat{\mathbf{u}}_{bc}^{(m-1)}$ is the (column) eigenvector corresponding to the minimum negative eigenvalue $\lambda_{min}(\widehat{\mathbf{H}}_{bc}^{(m-1)})$ of the 2×2 submatrix $\widehat{\mathbf{H}}_{bc}^{(m-1)}$. Since $\mathbf{d}_{bc}^{(m)}$ is an eigenvector, its Euclidean norm is equal to 1, so it does not need to be rescaled. If $\widehat{\mathbf{H}}_{bc}^{(m-1)}$ has not one negative eigenvalue at least, we set $\mathbf{d}_{bc}^{(m)} = 0$.
5. Compute the rescaled variation (of the quadratic approximation) of the objective function $-p\ell^p(\widehat{\boldsymbol{\gamma}})$ when moving in each Newton direction, that

is

$$\frac{-\Delta p\ell^p_{bc}(\widehat{\boldsymbol{\gamma}}^{(m-1)})}{\parallel \mathbf{s}^{(m)}_{bc} \parallel} = \frac{\frac{1}{2}\left(\widehat{\mathbf{g}}^{(m-1)}_{bc}\right)^\top \mathbf{s}^{(m)}_{bc}}{\parallel \mathbf{s}^{(m)}_{bc} \parallel},$$

where $\parallel \cdot \parallel$ denotes the Euclidean norm, and select the pair (b^*, c^*) that leads to the lowest value.

6. Compute the variation (of the quadratic approximation) of $-p\ell^p(\widehat{\boldsymbol{\gamma}})$ when moving in each negative curvature direction, that is

$$-\Delta p\ell^p_{bc}(\widehat{\boldsymbol{\gamma}}^{(m-1)}) = \left(\widehat{\mathbf{g}}^{(m-1)}_{bc}\right)^\top \mathbf{d}^{(m)}_{bc} + \frac{1}{2}\lambda_{min}(\widehat{\mathbf{H}}^{(m-1)}_{bc}),$$

and select the pair (b^+, c^+) that leads to the lowest value.

7. Set $\widehat{\boldsymbol{\gamma}}^{(m)} = \widehat{\boldsymbol{\gamma}}^{(m-1)}$. If both $-\Delta p\ell^p_{b^*c^*}(\widehat{\boldsymbol{\gamma}}^{(m-1)}) = 0$ and $-\Delta p\ell^p_{b^+c^+}(\widehat{\boldsymbol{\gamma}}^{(m-1)}) = 0$ go to step 9. Otherwise:

   - if $-\Delta p\ell^p_{b^*c^*}(\widehat{\boldsymbol{\gamma}}^{(m-1)})/ \parallel \mathbf{s}^{(m)}_{bc} \parallel \leq -\Delta p\ell^p_{b^+c^+}(\widehat{\boldsymbol{\gamma}}^{(m-1)})$, update the components (b^*, c^*), so that

$$(\widehat{\gamma}^{(m)}_{b^*}, \widehat{\gamma}^{(m)}_{c^*})^\top = (\widehat{\gamma}^{(m-1)}_{b^*}, \widehat{\gamma}^{(m-1)}_{c^*})^\top + \nu\, \mathbf{s}^{(m)}_{b^*c^*},$$

   - if $-\Delta p\ell^p_{b^*c^*}(\widehat{\boldsymbol{\gamma}}^{(m-1)})/ \parallel \mathbf{s}^{(m)}_{bc} \parallel > -\Delta p\ell^p_{b^+c^+}(\widehat{\boldsymbol{\gamma}}^{(m-1)})$, update the components (b^+, c^+), so that

$$(\widehat{\gamma}^{(m)}_{b^+}, \widehat{\gamma}^{(m)}_{c^+})^\top = (\widehat{\gamma}^{(m-1)}_{b^+}, \widehat{\gamma}^{(m-1)}_{c^+})^\top + \nu\, \mathbf{d}^{(m)}_{b^+c^+},$$

   with $\nu \in (0, 1]$.

8. If the loadings of the last latent variable introduced, i.e. the last J components of $\widehat{\boldsymbol{\gamma}}^{(m)}$, are not null, that is $(\widehat{\gamma}^{(m)}_{JD+1}, \ldots, \widehat{\gamma}^{(m)}_{J(D+1)}) \neq \mathbf{0}^\top$, set $D = D + 1$ and $\widehat{\boldsymbol{\gamma}}^{(m)} = (\widehat{\boldsymbol{\gamma}}^{(m)\top}, \mathbf{0}^\top)^\top$.
9. Return $\widehat{\boldsymbol{\gamma}} = \widehat{\boldsymbol{\gamma}}^{(m)}$.

In Appendix A we highlight that our component-wise boosting algorithm can be viewed as a particular instance of the optimization method introduced in Gould et al. (2000), in the special case where both the Newton-type directions and the negative curvature directions are different from zero for only two parameters. Moreover, we prove that the conditions assuring the global convergence

to second-order critical points are satisfied. Note that in the first iteration, namely when $m = 1$, the gradient vector $\widehat{\mathbf{g}}^{(0)}$ is null as well as each Newton-type direction $\mathbf{s}_{bc}^{(1)}$. In such cases, the algorithm may escape from the region of possible local non-convexity, moving towards a suitable negative curvature direction. The reason for updating two parameters instead of one at each iteration is mainly due to the necessity of considering at least two parameters for moving from the starting point or for introducing a new latent variable. Furthermore, it seems a sensible choice in factor analysis models to consider two parameters at a time for capturing the correlations between items. More details on these aspects are given in Subsection 3.2. Furthermore, with regard to Steps 5 and 6, if there are more than one pair satisfying the minimum condition, a single pair is selected at random.

As noted above, the decrease of the objective function $-p\ell^p(\boldsymbol{\gamma})$ is evaluated by considering its quadratic approximation in a neighbourhood of the current value for $\boldsymbol{\gamma}$. More precisely, the variation produced when we move from $\boldsymbol{\gamma}$ to $\boldsymbol{\gamma} + \mathbf{w}$, with $\mathbf{w} \in \mathbb{R}^{J(D+1)}$, is

$$-\Delta p\ell^p(\boldsymbol{\gamma}) = -\{p\ell^p(\boldsymbol{\gamma} + \mathbf{w}) - p\ell^p(\boldsymbol{\gamma})\} = \mathbf{g}^\top \mathbf{w} + \frac{1}{2}\mathbf{w}^\top \mathbf{H}\mathbf{w}.$$

Then, the rate of decrease in Step 5 is obtained with

$$\mathbf{w} = (0, \ldots, s_b, 0, \ldots, 0, s_c, \ldots, 0)^\top,$$

and that one in Step 6 with

$$\mathbf{w} = (0, \ldots, d_b, 0, \ldots, 0, d_c, \ldots, 0)^\top,$$

so that we attain, as expected,

$$-\Delta p\ell_{bc}^p(\boldsymbol{\gamma}) = \frac{1}{2}\widehat{\mathbf{g}}_{bc}\mathbf{s}_{bc}$$

and

$$-\Delta p\ell_{bc}^p(\boldsymbol{\gamma}) = \widehat{\mathbf{g}}_{bc}\mathbf{d}_{bc} + \frac{1}{2}\lambda_{min}(\widehat{\mathbf{H}}_{bc}).$$

The use of the quadratic approximation enables a more general evaluation of the convergence of the algorithm which accounts for the contribution of

both gradient-related directions and Hessian-based negative curvature directions. The analytical derivation of the gradient vector and the Hessian matrix for the latent trait model is presented in Appendix B. Note also that comparing these two quantities, as done in Step 7 of the algorithm, corresponds to set the constant $\tau = 2$ in the procedure explained in Gould et al. (2000, Equation (2.1)). Such value constitutes a natural choice when exactly the Newton direction is assumed (see Gould et al., 2000, Equation (2.7)).

Finally, we highlight that the term $\nu \in (0, 1]$, considered in Step 7, is a penalty factor that weakens the change of the parameter estimates produced in each iteration step. Empirical evidence supports the fact that, with a suitable penalization, the stability of the procedure improves even if an increased number of iterations, and thus more computing time, are required. Taking this into account, an alternative version of the algorithm can be defined by considering a small, fixed step length $\varepsilon > 0$. A further tuning parameter, which is far more influential for regularization purposes, is the maximum number m_{stop} of iterations. Thus, a suitable stopping criterion has to be defined in order to prevent overfitting and, in the following Section 3.2, we discuss the use of cross-validation procedures for specifying a convenient value for m_{stop}.

3.2. Computational aspects of the algorithm

It is interesting to note that the starting point of the algorithm, with loadings all equal to zero, corresponds to a point with null gradient, as can be seen from Equation (B.1) in Appendix B. Hence, the first direction chosen by the algorithm will always be a negative curvature direction. Furthermore, when the loadings of one latent variable $\boldsymbol{\lambda}_d$ are null, the gradient with respect to them is also null (see Equations (B.1) and (B.2)). So, the only possible direction for introducing a new latent variable in the model is a negative curvature direction. In fact, the Hessian matrix is not null even if $\boldsymbol{\lambda}_d = \mathbf{0}$ (see Equation (B.3) and the comment after Equation (B.4)). It is also interesting to observe that the second derivative of the log-likelihood function with respect to one zero loading is instead null. This explains why it is necessary to consider at least two

loadings for obtaining a negative curvature direction, and so introducing a new latent variable. Although it would be feasible to update only one parameter in the Newton direction, the algorithm updates two of them for two reasons. First, the negative curvature direction could have been favored by updating two parameters, whereas the Newton direction updates only one, leading to choose the negative curvature direction more often. Second, as mentioned before, updating two parameters is a sensible choice since the factor analysis model captures correlations between manifest variables that are expressed as the sum of products between two loadings related to one latent variable and couples of items (see Equation (1)).

The choice of the number of iterations of the algorithm m_{stop} is fundamental to obtain a sparse solution and to regularize the estimates. To this end, in this paper we adopt K-fold cross-validation, which is a very effective and general procedure for model validation. The method requires to randomly assign the observations to K groups and to use $K-1$ groups in turn for estimating the parameters and the remaining group for evaluating the prediction error. In this paper, as prediction error we considered the negative pairwise log-likelihood, similarly to Van Houwelingen & Le Cessie (1990). The number of iterations m showing the smallest average of the prediction errors over the K groups is then chosen, determining m_{stop}. Since the tuning parameter η needs to be selected too, it is first necessary to determine m_{stop} for each η in a set of candidate values and then compare the cross-validation error obtained for each η.

The choice of including a penalty in the objective function is related to the identifiability of the parameters in factor analysis models. Consider, for example, the introduction of a new latent variable with loadings proportional to the loadings of a latent variable already present in the model. In this case, the new latent variable is redundant. Specifically, suppose that one column in $\boldsymbol{\Lambda}$ is proportional to another one, that is $\boldsymbol{\lambda}_d = \alpha\boldsymbol{\lambda}_{d'}$. In this case, the model can be reduced to a model with $D-1$ dimensions, where $\boldsymbol{\lambda}_d$ are the loadings of a new latent variable $\theta_d + \alpha\theta_{d'}$ that substitutes θ_d and $\theta_{d'}$. However, this new latent variable is a normal with zero mean and variance equal to $1+\alpha^2$. In order to

obtain a factor model with the usual standard normal distribution for the latent variables, the loadings should be transformed to $\sqrt{1+\alpha^2}\boldsymbol{\lambda}_d$. The two solutions, however, are perfectly equivalent from the point of view of the fit. Hence, without penalization, updating $\boldsymbol{\lambda}_d$ or $\boldsymbol{\lambda}_{d'}$ turns out to be equivalent. The group lasso penalty used in Equation (5), always leads to prefer to update the loadings of a latent variable already included in the model, thus solving this indeterminacy issue. Consider now the case of the introduction of a new latent variable that is not perfectly redundant, so that it is not possible to find an equivalent solution updating the parameters of the latent variables already included in the model. For a suitable choice of the tuning parameter, the algorithm will update the parameters of the latent variables already included in the model if the solution is nearly equivalent. This is particularly important also because the choice of the direction to take is based on a quadratic approximation of the loss function. Using the unpenalized pairwise log-likelihood function, in case of equivalent or nearly equivalent solutions, the algorithm may prefer the negative curvature directions and add many new latent variables in the model, thus leading to a final solution with a large number of latent variables. Although this solution could be the best in terms of model fit, it does not serve the purpose of dimensionality reduction and interpretability of the solution.

Estimation of factor analysis models can encounter the so called Heywood case, which occurs when the diagonal elements of $\boldsymbol{\Psi}$ are negative (see Bartholomew et al., 2011, Section 3.12.3). To deal with this issue, when an Heywood case occurs, the step length ν is reduced progressively till the diagonal elements of $\boldsymbol{\Psi}$ are all positive. If reducing ν does not solve the problem, then the algorithm takes a different direction. If the direction chosen in the first place was a negative curvature direction, then the best Newton direction is selected. If the direction chosen in the first place was a Netwon direction, then the second best Newton direction is taken. In our experiments, this strategy always allowed to escape the Heywood case.

Our algorithm was implemented in the R (R Core Team, 2020) package BoostingLVM, publicly available on GitHub (`https://github.com/micbtz/`

`BoostingLVM`). Since the procedure is quite computationally demanding, due to the necessity to compute the Hessian matrix at each iteration, large parts of the code were written in C++, using the packages Rcpp (Eddelbuettel & François, 2011) and RcppArmadillo (Eddelbuettel & Sanderson, 2014), in order to reduce the computational time.

4. Applications

4.1. A simulation study

The performance of the proposal of this paper was assessed through a simulation study. Various settings were considered. The sample size was set to n = 200, 500, 1000, 2000, 5000, while the number of items was taken equal to J = 10, 20, 30. The data were generated from a three-factor model. One factor is related to all the items, while the other two factors contribute to a subset composed of half of the items. The loadings were generated from a uniform distribution in the $[0.3, 0.6]$ range. Taking the case of 10 items as example, the matrix of item loadings is as follows:

$$\Lambda = \begin{pmatrix} 0.58 & 0.44 & \\ 0.36 & 0.48 & \\ 0.50 & 0.45 & \\ 0.34 & 0.36 & \\ 0.38 & 0.55 & \\ 0.42 & & 0.50 \\ 0.30 & & 0.54 \\ 0.41 & & 0.33 \\ 0.56 & & 0.52 \\ 0.40 & & 0.42 \end{pmatrix}.$$

The thresholds were generated from a uniform distribution in the $[-0.2, 0.2]$ range. All results are based on 200 replications for each combination of sample size and number of items. All analyses were performed in R (R Core Team, 2020). The selection of the number of iterations of the boosting algorithm as

well as the tuning parameter η was performed by 5-fold cross-validation. The set of possible values for η was chosen in order to limit the number of latent variable that the algorithm can introduce, on the basis of some preliminary simulations. The value of ν was fixed at 0.05. The results obtained with our proposal were compared to the complete likelihood method, the pairwise likelihood method, both implemented in our code, and to the factor analysis model fitted to the estimated tetrachoric correlation matrix (Kirk, 1973) implemented in the Psych package (Revelle, 2020). The complete likelihood method was implemented under the IRT approach. For all these methods a 3-factor model was fitted, while the boosting algorithm selects the number of latent variables within the procedure. The solution with 2 factors was the most frequent and it was selected in 78% of the cases. The case of 3 factors was chosen in 17% of the cases, while 4 factors occurred in 4% of the cases and 5 factors in less than 1%. It could seem surprising that the more frequent solution is not the one with 3 latent variables, while the true parameters refer to 3 factors. However, this is related to the rotational indeterminacy of the matrix of the loadings, since the true one can also be rotated to obtain infinite equivalent ones. Figure 1 represents the number of estimated loadings obtained with the boosting algorithm as a function of n, showing that the procedure leads to a number of loadings just slightly above the actual number of loadings different from zero, which is $2J$. Instead, the other two methods estimate $3J-3$ loadings for each data set, hence providing a less parsimonious solution.

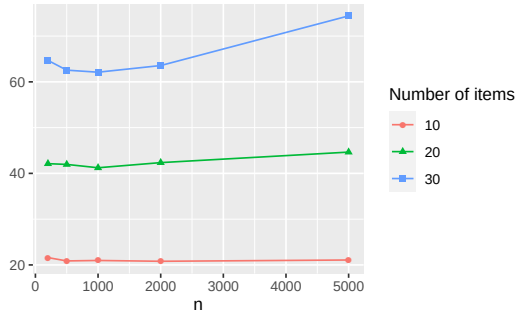


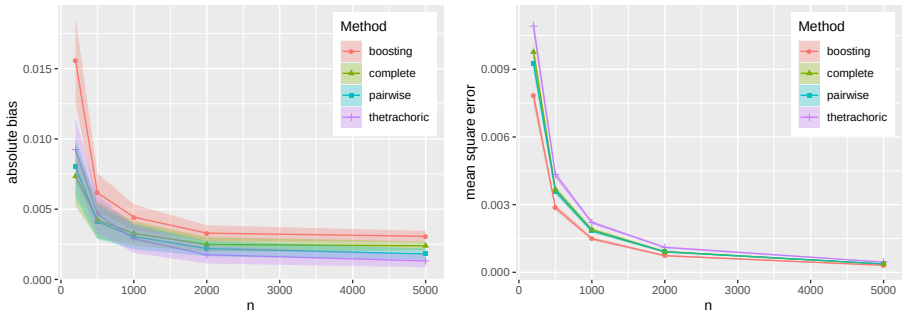
Figure 1: Number of estimated loadings with the boosting algorithm.

Because of the rotational indeterminacy of the solution of any factor analysis model, it is not possible to compare the estimates obtained with the three methods to the true parameters (note also that even the true parameters can be rotated without changing the covariance matrix between variables). Since the only invariant quantities are the correlations between variables, the bias and the mean square error were computed for these parameters. Since there are $J(J-1)/2$ correlations between J variables, we report the mean over all the pair of items.

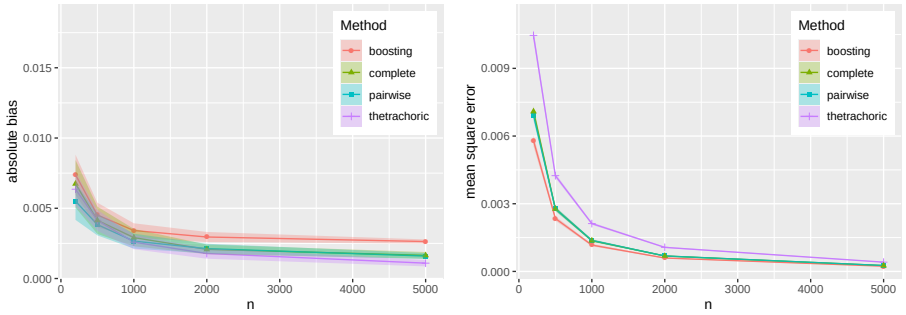
Figure 2 shows the bias in absolute value and the mean square error as a function of the sample size. The bands represent the 95% normal confidence intervals obtained with non-parametric bootstrap standard errors, which were computed by resampling over the 200 replications. The boosting method presents values of absolute bias slightly larger than those obtained with the other methods, though the confidence intervals generally overlap. At any rate, it should be noted that all the values are rather small and decrease as the sample size increases. The bias is slightly larger in the case of 10 items and $n = 200$ using all the methods, while the number of items is not influential for larger sample sizes.

Considering the mean square error, the boosting method is the most efficient in all settings, while the method based on the tetrachoric correlations leads to the largest values. The complete likelihood is very similar to the pairwise likelihood and they are located between the other two methods. The root mean square error tends to be larger when the number of variables is smaller, but this effect disappears for large sample sizes. However, the mean square error using the tetrachoric correlations is not influenced by the number of items.

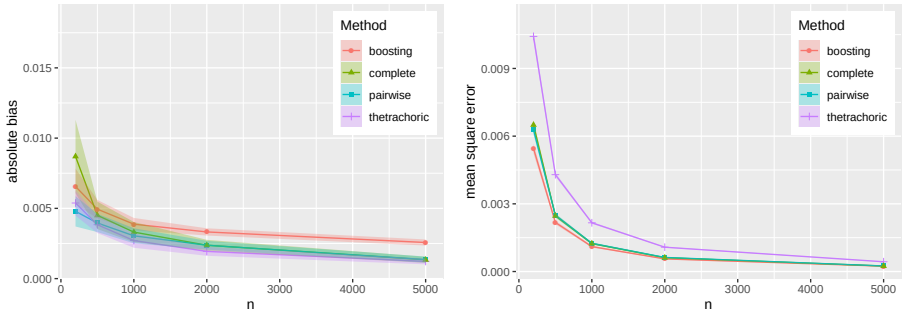
The computational time mainly depends on the number of items. On an Intel Core i7 at 2.7 GHz, estimating the parameters for one value of η and running 5-fold cross-validation to determine m_{stop} requires less than 1 minute in the case of 10 items, about 12 minutes in the case of 20 items, and about 1 hour in the case of 30 items.



(a) 10 items



(b) 20 items



(c) 30 items

Figure 2: Absolute bias of estimated correlations (average over all pairs of items). The bands represent the 95% confidence intervals.

4.2. A real-data example

The method proposed in this paper was then applied the 2017 Eurobarometer survey (European Commission and European Parliament, Brussels, 2018), focusing on a subset of items related to the attitudes of European citizens towards the environment, which are reported in Appendix C, and selecting the 1027 respondents from Italy. The dataset is publicly available at `https://search.gesis.org/research_data/ZA6925`.

For this analysis, we set $\nu = 0.05$ and we used 10-fold cross-validation to select the number of iterations and the tuning parameter η. The reason for using 10 folds here, instead of 5 as in the simulation study, is given to the computational time, which is not an issue for a single data set. In order to deal with the randomness involved in this procedure, cross-validation was repeated 20 times for each value of the tuning parameter η, and the average over these replications is considered as cross-validation error. The tuning parameters considered are all the integer values between 2 and 20. The value $\eta = 1$, included in our preliminary analyses, was then discarded because this corresponds to a too small penalization for these data, which leads the algorithm to include in the model more parameters than the maximum number of parameters that can be estimated (that is $J(J-1)/2$). Figure 3 shows the cross-validation error as a function of the number of iterations, with each line representing a different value of η. The minimum value is reached for $\eta = 4$ and $m = 540$, which correspond to a model with 3 latent variables. Since some lines are not distinguishable, the panel on the right zooms on the part where the minimum is attained. Figure 4a shows the minimum cross validation error for each η. Both these figures show that the differences are very small for $\eta = 2, \ldots, 6$. Figure 4b represents the number of latent variables selected by cross-validation as a function of η. We can observe that the penalty leads to the expected effect of reducing the number of latent variables for increasing values of the tuning parameter. The solution for $\eta = 2$ corresponds to a small penalization which leads to 8 latent variables. For $\eta = 3$, the number of latent variables is 4, while it persists at 3 for $\eta = 4, \ldots, 8$. For higher values of the tuning parameter the number of latent

variables becomes 2 and then 1.

Besides employing the boosting algorithm presented in this paper, the loadings were estimated with the pairwise maximum likelihood method and the tetrachoric correlations, both using the quartimax rotation. The loadings are reported in Table 1, where the values in bold indicate absolute values above 0.1. The boosting method led to a more parsimonious solution with 33 loadings different from zero, while the other two methods required the estimation of 42 loadings (which results from 15 item $\times$ 3 factors $-$ 3 constrains).

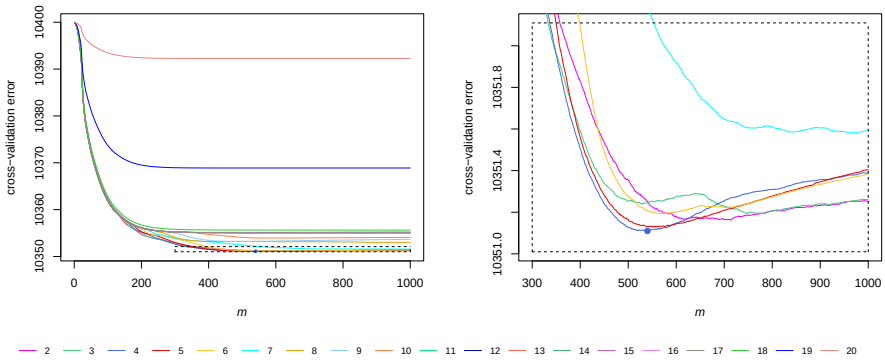
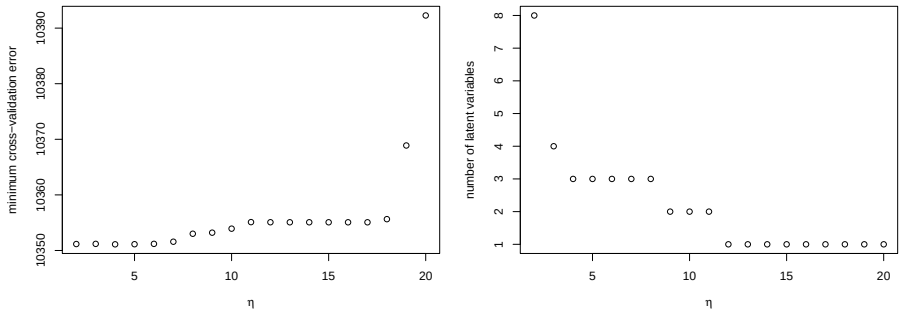


Figure 3: Cross-validation error as a function of the number of iterations for different values of the tuning parameter η. The right panel zooms on the region with the dashed border. The point shows the minimum cross-validation error.



(a) (b)

Figure 4: The left panel shows the minimim cross-validation errors for each η. The right panel shows the number of latent variables selected by cross-validation for each η.

Table 1: Parameter estimates of the real-data application.

	boosting			pairwise			tetrachoric		
	λ_{i1}	λ_{i2}	λ_{i3}	λ_{i1}	λ_{i2}	λ_{i3}	λ_{i1}	λ_{i2}	λ_{i3}
QD4.1	**0.60**	**0.44**	-0.06	**0.44**	**0.54**	-0.08	**0.35**	**0.60**	**0.13**
QD4.2	**0.40**	-0.05	0.09	**0.40**	0.07	**0.19**	**0.44**	0.10	0.02
QD4.3	**0.59**	0.00	0.00	**0.56**	**0.18**	0.08	**0.52**	**0.25**	0.06
QD4.4	**0.14**	0.00	**0.32**	0.06	**0.12**	**0.34**	**0.12**	**0.16**	**-0.39**
QD4.5	**0.19**	0.00	**0.24**	**0.16**	0.06	**0.30**	**0.23**	0.05	**-0.18**
QD4.6	**0.54**	0.00	0.00	**0.49**	**0.22**	0.07	**0.44**	**0.29**	0.02
QD4.7	**0.40**	**-0.12**	0.00	**0.44**	-0.01	**0.11**	**0.50**	-0.01	0.10
QD4.8	**0.25**	0.00	**0.57**	**0.14**	**0.12**	**0.60**	**0.27**	**0.14**	**-0.40**
QD4.9	**0.25**	**0.15**	-0.04	**0.19**	**0.24**	-0.08	**0.15**	**0.25**	0.10
QD19.1	**0.53**	0.00	**-0.29**	**0.57**	**0.14**	**-0.27**	**0.38**	**0.27**	**0.40**
QD19.2	**0.69**	-0.06	0.02	**0.66**	**0.17**	**0.16**	**0.66**	**0.24**	-0.04
QD19.3	**0.42**	**0.69**	0.08	0.10	**0.97**	0.05	0.07	**0.84**	-0.04
QD19.4	**0.22**	0.06	0.00	**0.20**	**0.18**	-0.06	**0.11**	**0.25**	**0.20**
QD19.5	**0.36**	**-0.27**	0.00	**0.46**	**-0.18**	**0.15**	**0.55**	**-0.19**	-0.07
QD19.6	**0.38**	0.00	**-0.24**	**0.45**	0.06	**-0.24**	**0.36**	**0.11**	**0.49**

Focusing on the boosting method, it is possible to observe a common factor loading on all items, which could be interpreted as a general attitude toward protecting the environment. The second factor identified by the algorithm is related to the travelling behavior and is positively correlated with items related to travelling avoiding the car or using an electric one (items QD4.1, QD4.9, QD19.3), and it is negative correlated with buying a low emission-car (item QD19.5). The other item that loads on this factor (QD4.7) is more difficult to interpret. The third factor is related to consumption and waste (items QD4.4, QD4.5, QD4.8) in contrast to the use of low-emission products for heating or barbecue (QD19.1 and QD19.6). This could be because the last two items are related to economic factors other than environmental. The other two methods

provide a solution which is more difficult to interpret, due to the large number of loadings taking high values. Although the solutions could appear different, Figure 5 shows that the estimated correlations between the variables $\mathbf{Y}^*$ obtained with the boosting method and the pairwise likelihood are very similar, while the factor analysis applied to the tetrachoric correlation provides slightly different values.

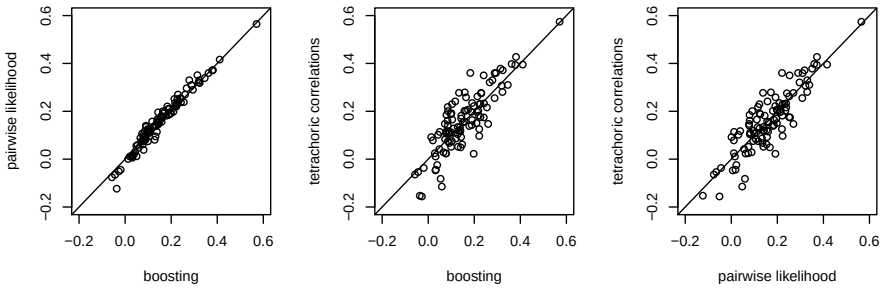


Figure 5: Comparison of the estimated correlations between variables in the real-data example.

5. Conclusions

This paper proposes a boosting algorithm for the factor analysis model with binary data, following a likelihood-based approach. The model under study is particularly challenging, since the starting point with zero loadings presents a null gradient, making the boosting methods proposed in the literature (for example Friedman, 2001; Tutz & Binder, 2006) not directly applicable. Hence, our proposal employs the Hessian matrix for moving along the directions of negative curvature. More specifically, the algorithm chooses a Newton direction (which is the common strategy of statistical boosting methods) or a negative curvature direction on the basis of the decrement of a quadratic approximation of the objective function. The algorithm so obtained can be seen as a particular instance of the method proposed in Gould et al. (2000), with the peculiarity that only two parameters are updated at each iteration. Such choice makes the procedure component-wise, hence leading to a sparse solution of the loadings estimation

task. The advantage lies in the interpretability of the solution attained, without requiring a posterior rotation of the factor loadings. Since the procedure requires the computation of the Hessian matrix at each step, it is computationally quite demanding. This makes necessary an efficient approach for the evaluation of the objective function, and the pairwise likelihood (Katsikatsou et al., 2012) is particularly suited for this task.

A proper calibration of the number of iterations m_{stop} and of the tuning parameter η is fundamental for performing regularization and providing a sparse solution. In this paper, the selection of these values was performed by cross-validation. As pointed out in Seibold et al. (2018) for the case of Cox regression, and confirmed in our real-data analysis, the results obtained with cross-validation could be highly variable. Therefore, repeated cross-validation would be preferable to limit this variability. Other methods commonly used for this task are based on information criteria, such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). However, they require the knowledge of the effective degrees of freedom, which are not straightforward to obtain in the present context. So, a further interesting point is to discuss the possibility of deriving the effective degrees of freedom analytically or using composite likelihood information criteria (Gao & Song, 2010; Varin & Vidoni, 2005) in the boosting inferential framework. All these issues can be a matter for future research.

Another approach for obtaining a sparse solution in this framework is the lasso penalization proposed in Sun et al. (2016). However, their approach requires some restrictions on the matrix of the loadings to assure the identifiability of the parameters and the specification of the number of latent variables. Instead, in our proposal these elements are automatically determined by the algorithm.

Although the algorithm was proposed in this paper for the estimation of factor analysis models with binary responses, it represents a quite general procedure that could potentially be employed in many other contexts. In particular, we see a lot of potential for dealing with any non-convex loss function, especially

when the starting point presents a null gradient.

Appendix A: The linesearch algorithm

Gould et al. (2000) propose an efficient linesearch algorithm for solving unconstrained optimization problems specified as

$$\min_{\mathbf{x}\in\mathbb{R}^m} f(\mathbf{x}),$$

where $f(\cdot)$ is a real valued, twice continuously differentiable function defined on $\mathbb{R}^m$, $m \geq 1$, with gradient $\mathbf{g} = \mathbf{g}(\mathbf{x})$ and Hessian matrix $\mathbf{H} = \mathbf{H}(\mathbf{x})$. This algorithm belongs to the class of linesearch procedures that use the additional information given by the Hessian matrix and are proved to converge to a second-order critical point, namely a point $\mathbf{x}_*$ such that $\mathbf{g}(\mathbf{x}_*) = \mathbf{0}$ and $\mathbf{H}(\mathbf{x}_*)$ is positive semidefinite. Its peculiarity, useful for the specification of boosting algorithms, is that only the most promising direction, between that one based on the Newton method and that one related to a negative curvature, is employed at each iteration step. The approach of Gould et al. (2000) is extended, for example, by Olivares et al. (2008), where a suitable combination of these two directions is also considered. As mentioned in Section 3, in the m-th iteration a pair of directions $(\mathbf{s}_m, \mathbf{d}_m)$ is computed. The first vector defines a Newton-type direction that guarantees convergence under convexity assumptions, namely under a positive curvature given by the Hessian matrix. The second one specifies, if any, a negative curvature direction (that is, $\mathbf{d}$ such that $\mathbf{d}^\top\mathbf{H}\mathbf{d} < 0$), which allows moving away from regions of local non-convexity. More precisely, in order to ensure the convergence of the algorithm, the following conditions are required:

C1. There exist constants $c_1, c_2 > 0$ such that

$$\mathbf{s}_m^\top \mathbf{g}_{m-1} \leq -c_1 \| \mathbf{g}_{m-1} \|^2, \quad \| \mathbf{s}_m \| \leq c_2 \| \mathbf{g}_{m-1} \|, \tag{A.1}$$

with $\mathbf{g}_{m-1} = \mathbf{g}(\mathbf{x}_{m-1})$, $\mathbf{H}_{m-1} = \mathbf{H}(\mathbf{x}_{m-1})$, where $\mathbf{x}_{m-1}$ is the value given by the $(m-1)$-th iteration;

C2. For some $\theta \in (0,1)$,

$$\mathbf{d}_m^\top \mathbf{g}_{m-1} \leq 0, \qquad \mathbf{d}_m^\top \mathbf{H}_{m-1} \mathbf{d}_m \leq 0,$$
$$\frac{\mathbf{d}_m^\top \mathbf{H}_{m-1} \mathbf{d}_m}{\| \mathbf{d}_m \|^2} \leq \theta \lambda_{min}(\mathbf{H}_{m-1}) + \eta(\mathbf{g}_{m-1}), \tag{A.2}$$

where $\eta : \mathbb{R}^+ \to \mathbb{R}^+$ is such that $\eta(t) \to 0$ as $t \to 0$ and $\lambda_{min}(\mathbf{H}_{m-1})$ is the minimum negative eigenvalue of $\mathbf{H}_{m-1}$.

If $\mathbf{H}_{m-1}$ has at least one negative eigenvalue, $(\mathbf{s}_m, \mathbf{d}_m)$ defines a pair of descent directions. Condition C1. is a usual requirement for Newton-type directions, while condition C2. assures that $\mathbf{d}_m$ is a negative curvature direction which is, in some sense, related to the eigenvector of $\mathbf{H}_{m-1}$ corresponding to the minimum negative eigenvalue. Indeed, the third requirement in (A.2) emphasizes that the contribution of $\mathbf{d}_m$ is essential when the gradient is small. A possible choice for the pair $(\mathbf{s}_m, \mathbf{d}_m)$ would be $\mathbf{s}_m = -\mathbf{g}_{m-1}$ and $\mathbf{d}_m = -sign(\mathbf{g}_{m-1}^\top \mathbf{u}_m)\mathbf{u}_m$, where $\mathbf{u}_m$ is (column) eigenvector associated with the minimum negative eigenvalue of the Hessian matrix. Note that the sign of $\mathbf{d}_m$ is defined in order to verify that $\mathbf{d}_m^\top \mathbf{g}_{m-1} \leq 0$. On the other hand, whenever $\mathbf{H}_{m-1}$ has no negative eigenvalue and $\mathbf{g}_{m-1} \neq \mathbf{0}$, the second-order information is not relevant and we set $\mathbf{d}_m = \mathbf{0}$.

Furthermore, the rate of decrease of the objective function $f(\cdot)$ is evaluated along the associated quadratic approximation obtained from a Taylor series expansion around $\mathbf{x}_{m-1}$, and this approximation determines the choice between $\mathbf{s}_m$ and $\mathbf{d}_m$. For each selected direction, the adaptive algorithm proposed by Gould et al. (2000) defines a specific linesearch step which improves the efficiency of the optimization procedure.

The pairwise likelihood-based boosting algorithm introduced in Section 3 follows a component-wise approach and, for this reason, it can be viewed as special case of the algorithm proposed by Gould et al. (2000), with a Newton-type direction $\mathbf{s}_m$ taking values different from zero for only two component parameters, namely the (b,c)-th elements defined in (6), and a negative curvature direction $\mathbf{d}_m$ taking values different from zero for only two component parameters, namely the (b,c)-th elements defined in (7). It is easy to verify that

these directions satisfy the conditions C1. and C2., and hence the convergence to a second-order stationary point is assured. More specifically, concerning the first condition, since in our case $\mathbf{s}_m^\top \mathbf{g}_{m-1} = -(\mathbf{g}_{bc}^{(m-1)})^\top \widehat{\mathbf{H}}_{bc}^{(m-1)^{-1}} \mathbf{g}_{bc}^{(m-1)}$, it is always possible to find a positive constant c_1 small enough to satisfy the first inequality in (A.1), provided that $\widehat{\mathbf{H}}_{bc}^{(m-1)}$ is positive definite, namely that the function is convex in this direction. Furthermore, if we consider $\| \mathbf{s}_m \|$ it is always possible to find a positive constant c_2 large enough to satisfy the second inequality in (A.1). Finally, with regard to the second condition, the first inequality in (A.2) is always satisfied by taking the appropriate sign for the eigenvector as done in (7). Since, in our case, $\mathbf{d}_m^\top \mathbf{H}_{m-1} \mathbf{d}_m = \lambda_{min}(\widehat{\mathbf{H}}_{bc}^{(m-1)})$, the second inequality in (A.2) is also verified because this direction is chosen only in such a case. Furthermore, it is always possible to find θ small enough to fulfill the third inequality in (A.2).

Appendix B: Gradient and Hessian matrix of the objective function

This appendix provides the gradient and the Hessian matrix of the objective function, given by the negative penalized pairwise log-likelihood defined in Equation (5). The gradient and the Hessian matrix are then given by:

$$\mathbf{g} = -\frac{\partial p\ell(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} + \eta \frac{\partial \sum_{d=1}^{D} (\boldsymbol{\lambda}_d^\top \boldsymbol{\lambda}_d)^{1/2}}{\partial \boldsymbol{\gamma}}$$

and

$$\mathbf{H} = -\frac{\partial^2 p\ell(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^\top} + \eta \frac{\partial^2 \sum_{d=1}^{D} (\boldsymbol{\lambda}_d^\top \boldsymbol{\lambda}_d)^{1/2}}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^\top}.$$

The gradient of the pairwise log-likelihood function can be found in Katsikatsou et al. (2012) in the more general case of ordinal data, and it is also given in the following for the case of binary responses for the sake of completeness.

In order to obtain the derivatives it is useful to recall that

$$\pi_{00}^{jl}(\boldsymbol{\gamma}) = \Phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl}),$$

while

$$\pi_{01}^{jl}(\boldsymbol{\gamma}) = \Phi(\tau_j) - \Phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl}),$$

$$\pi_{10}^{jl}(\boldsymbol{\gamma}) = \Phi(\tau_l) - \Phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl}),$$

and

$$\pi_{11}^{jl}(\boldsymbol{\gamma}) = 1 - \Phi(\tau_j) - \Phi(\tau_l) + \Phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl}),$$

where $\Phi(\cdot)$ is the distribution function of a univariate standard normal distribution. The gradient of the pairwise log-likelihood function given in Equation (4) can be written as follows:

$$\frac{\partial p\ell(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} = \sum_{j<l} \sum_{s=0}^{1} \sum_{t=0}^{1} n_{st}^{jl} \frac{\partial \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}.$$

Since $\boldsymbol{\gamma} = (\boldsymbol{\tau}, \boldsymbol{\lambda})$ we have:

$$\frac{\partial \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\tau}} = \frac{1}{\pi_{st}^{jl}(\boldsymbol{\gamma})} \frac{\partial \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\tau}},$$

where

$$\frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j} = \phi(\tau_j)\Phi\left\{\frac{\tau_l - \rho_{jl} \cdot \tau_j}{(1-\rho_{jl}^2)^{1/2}}\right\},$$

$$\frac{\partial \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j} = \phi(\tau_j) - \frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j},$$

$$\frac{\partial \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j} = -\frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j},$$

$$\frac{\partial \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j} = -\phi(\tau_j) + \frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j},$$

$$\frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l} = \phi(\tau_l)\Phi\left\{\frac{\tau_j - \rho_{jl} \cdot \tau_l}{(1-\rho_{jl}^2)^{1/2}}\right\},$$

$$\frac{\partial \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l} = -\frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l},$$

$$\frac{\partial \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l} = \phi(\tau_l) - \frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l},$$

$$\frac{\partial \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l} = -\phi(\tau_l) + \frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l},$$

and $\phi(\cdot)$ denotes the density of the standard normal distribution. The chain rule is exploited to obtain the second one:

$$\frac{\partial \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\lambda}} = \frac{\partial \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} \frac{\partial \rho_{jl}}{\partial \boldsymbol{\lambda}} = \frac{1}{\pi_{st}^{jl}(\boldsymbol{\gamma})} \frac{\partial \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} \frac{\partial \rho_{jl}}{\partial \boldsymbol{\lambda}},$$

where

$$\frac{\partial \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} = \frac{\partial \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} = \phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl}),$$

$$\frac{\partial \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} = \frac{\partial \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} = -\phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl})$$

and

$$\frac{\partial \rho_{jl}}{\partial \boldsymbol{\lambda}} = (\mathbf{X}_j^\top \mathbf{X}_l + \mathbf{X}_l^\top \mathbf{X}_j)\boldsymbol{\lambda}, \tag{B.1}$$

with $\mathbf{X}_j, j = 1, \ldots, J$, being a $D \times JD$ design matrix such that the j-th row of $\boldsymbol{\Lambda}$, can be written as $\mathbf{X}_j\boldsymbol{\lambda}$, and then

$$\rho_{jl} = \sum_{d=1}^{D} \lambda_{jd}\lambda_{ld} = (\mathbf{X}_j\boldsymbol{\lambda})^\top(\mathbf{X}_l\boldsymbol{\lambda}) = \boldsymbol{\lambda}^\top \mathbf{X}_j^\top \mathbf{X}_l \boldsymbol{\lambda}. \tag{B.2}$$

Note that when the loadings of one latent variable $\boldsymbol{\lambda}_d$ are null, the derivative of ρ_{jl} with respect to $\boldsymbol{\lambda}_d$ is null too, and so is the derivative of the log-likelihood with respect to $\boldsymbol{\lambda}_d$.

The associated Hessian matrix is specified as:

$$\frac{\partial^2 p\ell(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^\top} = \sum_{j<l} \sum_{s=0}^{1} \sum_{t=0}^{1} n_{st}^{jl} \frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^\top}.$$

The second derivative with respect to $\boldsymbol{\tau}$ is given by:

$$\frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\tau} \partial \boldsymbol{\tau}^\top} = -\frac{1}{\left(\pi_{st}^{jl}(\boldsymbol{\gamma})\right)^2} \frac{\partial \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\tau}} \frac{\partial \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\tau}^\top} + \frac{1}{\pi_{st}^{jl}(\boldsymbol{\gamma})} \frac{\partial^2 \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\tau} \partial \boldsymbol{\tau}^\top},$$

with

$$\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2} = -\tau_j \phi(\tau_j) \Phi\left\{\frac{\tau_l - \rho_{jl} \cdot \tau_j}{(1-\rho_{jl}^2)^{1/2}}\right\} - \phi(\tau_j)\phi\left\{\frac{\tau_l - \rho_{jl} \cdot \tau_j}{(1-\rho_{jl}^2)^{1/2}}\right\} \frac{\rho_{jl}}{(1-\rho_{jl}^2)^{1/2}},$$

$$\frac{\partial^2 \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2} = -\tau_j \phi(\tau_j) - \frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2},$$

$$\frac{\partial^2 \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2} = -\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2},$$

$$\frac{\partial^2 \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2} = \tau_j \phi(\tau_j) - \frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j^2},$$

$$\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2} = -\tau_l \phi(\tau_l) \Phi \left\{ \frac{\tau_j - \rho_{jl} \cdot \tau_l}{(1-\rho_{jl}^2)^{1/2}} \right\} - \phi(\tau_l) \phi \left\{ \frac{\tau_j - \rho_{jl} \cdot \tau_l}{(1-\rho_{jl}^2)^{1/2}} \right\} \frac{\rho_{jl}}{(1-\rho_{jl}^2)^{1/2}},$$

$$\frac{\partial^2 \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2} = -\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2},$$

$$\frac{\partial^2 \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2} = -\tau_l \phi(\tau_l) - \frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2},$$

$$\frac{\partial^2 \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2} = \tau_l \phi(\tau_l) - \frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_l^2},$$

$$\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j \partial \tau_l} = \phi(\tau_j) \phi \left\{ \frac{\tau_l - \rho_{jl} \cdot \tau_j}{(1-\rho_{jl}^2)^{1/2}} \right\} (1-\rho_{jl}^2)^{-1/2},$$

$$\frac{\partial^2 \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j \partial \tau_l} = \frac{\partial^2 \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j \partial \tau_l} = -\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j \partial \tau_l}$$

and

$$\frac{\partial^2 \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j \partial \tau_l} = \frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \tau_j \partial \tau_l}.$$

The second derivative with respect to $\boldsymbol{\lambda}$ is again obtained using the chain rule:

$$\frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \boldsymbol{\lambda} \partial \boldsymbol{\lambda}^\top} = \frac{\partial \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} \frac{\partial^2 \rho_{jl}}{\partial \boldsymbol{\lambda} \boldsymbol{\lambda}^\top} + \frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2} \frac{\partial \rho_{jl}}{\partial \boldsymbol{\lambda}} \frac{\partial \rho_{jl}}{\partial \boldsymbol{\lambda}^\top}, \tag{B.3}$$

with

$$\frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2} = -\frac{1}{(\pi_{st}^{jl}(\boldsymbol{\gamma}))^2} \left(\frac{\partial \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}} \right)^2 + \frac{1}{\pi_{st}^{jl}(\boldsymbol{\gamma})} \frac{\partial^2 \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2},$$

where

$$\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2} = \phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl})(1-\rho_{jl}^2)^{-1} \rho_{jl} + \phi_2(\tau_j, \tau_l; \mathbf{0}, \boldsymbol{\Sigma}_{jl})(\tau_j \tau_l (1-\rho_{jl}^2) - z_{jl} \rho_{jl})(1-\rho_{jl}^2)^{-2},$$

$$z_{jl} = \tau_j^2 - 2\rho_{jl} \tau_j \tau_l + \tau_l^2,$$

$$\frac{\partial^2 \pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2} = \frac{\partial^2 \pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2} = -\frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2},$$

$$\frac{\partial^2 \pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2} = \frac{\partial^2 \pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial \rho_{jl}^2},$$

and

$$\frac{\partial^2 \rho_{jl}}{\partial \boldsymbol{\lambda} \boldsymbol{\lambda}^\top} = \mathbf{X}_j^\top \mathbf{X}_l + \mathbf{X}_l^\top \mathbf{X}_j. \tag{B.4}$$

Note that even if $\boldsymbol{\lambda}_d = \mathbf{0}$, the second derivative in (B.3) is not null because the first term is not null, while $\partial\rho_{jl}/\partial\boldsymbol{\lambda}_d = \mathbf{0}$. However, since the diagonal of (B.4) is null, the second derivative of the log-likelihood function with respect to one loading is null.

The second derivative with respect to $\boldsymbol{\tau}$ and $\boldsymbol{\lambda}$ is given by:

$$\frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial\boldsymbol{\tau}\partial\boldsymbol{\lambda}^\top} = \frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\boldsymbol{\tau}}\frac{\partial\rho_{jl}}{\partial\boldsymbol{\lambda}},$$

where

$$\frac{\partial^2 \log \pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\boldsymbol{\tau}} = -\frac{1}{(\pi_{st}^{jl}(\boldsymbol{\gamma}))^2}\left(\frac{\partial\pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}}\frac{\partial\pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial\boldsymbol{\tau}}\right) + \frac{1}{\pi_{st}^{jl}(\boldsymbol{\gamma})}\frac{\partial^2\pi_{st}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\boldsymbol{\tau}},$$

$$\frac{\partial^2\pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_j} = \phi_2(\tau_j,\tau_l;\mathbf{0},\boldsymbol{\Sigma}_{jl})(\tau_j - \rho_{jl}\tau_l)(1-\rho_{jl}^2)^{-1},$$

$$\frac{\partial^2\pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_j} = \frac{\partial^2\pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_j} = -\frac{\partial^2\pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_j},$$

$$\frac{\partial^2\pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_j} = \frac{\partial^2\pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_j},$$

$$\frac{\partial^2\pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_l} = \phi_2(\tau_j,\tau_l;\mathbf{0},\boldsymbol{\Sigma}_{jl})(\tau_l - \rho_{jl}\tau_j)(1-\rho_{jl}^2)^{-1},$$

$$\frac{\partial^2\pi_{01}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_l} = \frac{\partial^2\pi_{10}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_l} = -\frac{\partial^2\pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_l}$$

and

$$\frac{\partial^2\pi_{11}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_l} = \frac{\partial^2\pi_{00}^{jl}(\boldsymbol{\gamma})}{\partial\rho_{jl}\partial\tau_l}.$$

Finally, the non-zero derivatives of the penalty term are

$$\frac{\partial\sum_{d=1}^D(\boldsymbol{\lambda}_d^\top\boldsymbol{\lambda}_d)^{1/2}}{\partial\lambda_{jd'}} = (\boldsymbol{\lambda}_{d'}^\top\boldsymbol{\lambda}_{d'})^{-1/2}\lambda_{jd'},\ \ d' = 1,\ldots,D,\ j = 1,\ldots,J,$$

$$\frac{\partial^2\sum_{d=1}^D(\boldsymbol{\lambda}_d^\top\boldsymbol{\lambda}_d)^{1/2}}{\partial\lambda_{jd'}^2} = -(\boldsymbol{\lambda}_{d'}^\top\boldsymbol{\lambda}_{d'})^{-3/2}\lambda_{jd'}^2 + (\boldsymbol{\lambda}_{d'}^\top\boldsymbol{\lambda}_{d'})^{-1/2},$$

and

$$\frac{\partial^2\sum_{d=1}^D(\boldsymbol{\lambda}_d^\top\boldsymbol{\lambda}_d)^{1/2}}{\partial\lambda_{jd'}\partial\lambda_{ld'}} = -(\boldsymbol{\lambda}_{d'}^\top\boldsymbol{\lambda}_{d'})^{-3/2}\lambda_{jd'}\lambda_{ld'}.$$

As mentioned in Section 3.1, in order to assure differentiability a small constant is added to $\boldsymbol{\lambda}_d^\top\boldsymbol{\lambda}_d$ as in Tutz & Gertheiss (2014, Section 4.1).

Appendix C: Items included in the real-data analysis

Table 2: Items of the Eurobarometer survey included in the real-data analysis

QD4		Have you done any of the following in the past six months?
	QD4.1	Chosen a more environmentally-friendly way of travelling (walk, bicycle, public transport, electric car)
	QD4.2	Avoided buying over-packaged products
	QD4.3	Avoided single-use plastic goods other than plastic bags (e.g. plastic cutlery, cups, plates, etc.) or bought reusable plastic products
	QD4.4	Separated most of your waste for recycling
	QD4.5	Cut down your water consumption
	QD4.6	Cut down your energy consumption (e.g. by turning down air conditioning or heating, not leaving appliances on stand-by, buying energy-efficient appliances)
	QD4.7	Bought products marked with an environmental label
	QD4.8	Bought local products
	QD4.9	Used your car less by avoiding unnecessary trips, working from home (teleworking), etc.
QD19		There are different ways to reduce harmful emissions into the air. In order to reduce these problems have you done any of the following in the last two years?
	QD19.1	You have changed your home heating system from a higher- emission system (e.g. coal, oil or wood-fired) to a lower one (natural gas, pellets, electricity, solar, etc.)
	QD19.2	You have replaced older energy-intensive equipment (hot water boiler, oven, dishwasher, etc.) with newer equipment with a better energy efficiency rating (for instance products labelled A+++)
	QD19.3	You have frequently used public transport or a bicycle, or chosen to walk instead of taking your car
	QD19.4	You have bought an electric vehicle (car, motorbike, bicycle)
	QD19.5	You have bought a low emission-car (for example an hybrid car)
	QD19.6	You have bought low-emission products to fuel your open fire or barbecue (e.g. briquettes instead of coal)

References

Bartholomew, D. J., Knott, M., & Moustaki, I. (2011). *Latent Variable Models and Factor Analysis: a Unified Approach*. Wiley, London.

Battauz, M. (2020). Regularized estimation of the nominal response model. *Multivariate Behavioral Research*, *55*, 811–824. doi:10.1080/00273171.2019.1681252.

Béguin, A. A., & Glas, C. A. (2001). MCMC estimation and some model-fit analysis of multidimensional IRT models. *Psychometrika*, *66*, 541–561. doi:10.1007/BF02296195.

Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, *46*, 443–459. doi:10.1007/BF02293801.

Bühlmann, P., & Yu, B. (2006). Sparse boosting. *Journal of Machine Learning Research*, *7*, 1001–1024.

Cai, L. (2010). High-dimensional exploratory item factor analysis by a Metropolis–Hastings Robbins–Monro algorithm. *Psychometrika*, *75*, 33–57. doi:10.1007/s11336-009-9136-x.

De Bin, R. (2016). Boosting in Cox regression: A comparison between the likelihood-based and the model-based approaches with focus on the R-packages CoxBoost and mboost. *Computational Statistics*, *31*, 513–531. doi:10.1007/s00180-015-0642-2.

de Leon, A. (2005). Pairwise likelihood approach to grouped continuous model and its extension. *Statistics & Probability Letters*, *75*, 49–57. doi:10.1016/j.spl.2005.05.017.

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, *40*, 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel, D., & Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics & Data Analysis*, *71*, 1054–1063. doi:10.1016/j.csda.2013.02.005.

European Commission and European Parliament, Brussels (2018). Eurobarometer 88.1 (2017). GESIS Data Archive, Cologne. ZA6925 Data file Version 1.0.0. doi:10.4232/1.12959.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference* (p. 148–156). Morgan Kaufmann, San Francisco, CA, USA.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, *28*, 337–407. doi:10.1214/aos/1016218223.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*, 1189–1232. doi:10.1214/aos/1013203451.

Gao, X., & Song, P. X.-K. (2010). Composite likelihood Bayesian information criteria for model selection in high dimensional data. *Journal of the American Statistical Association*, *105*, 1303–1325. doi:10.1198/jasa.2010.tm09414.

Gould, N. I. M., Lucidi, S., Roma, M., & Toint, P. L. (2000). Exploiting negative curvature directions in linesearch methods for unconstrained optimization. *Optimization Methods and Software*, *14*, 75–98. doi:10.1080/10556780008805794.

Jöreskog, K. G., & Moustaki, I. (2001). Factor analysis of ordinal variables: A comparison of three approaches. *Multivariate Behavioral Research*, *36*, 347–387. doi:10.1207/S15327906347-387.

Katsikatsou, M., Moustaki, I., Yang-Wallentin, F., & Jöreskog, K. G. (2012). Pairwise likelihood estimation for factor analysis models with ordinal data.

Computational Statistics & Data Analysis, *56*, 4243–4258. doi:10.1016/j.csda.2012.04.010.

Kirk, D. B. (1973). On the numerical approximation of the bivariate normal (tetrachoric) correlation coefficient. *Psychometrika*, *38*, 259–268. doi:10.1007/BF02291118.

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, *80*, 221–239. doi:10.1090/conm/080.

Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014). The evolution of boosting algorithms. *Methods of Information in Medicine*, *53*, 419–427. doi:10.3414/ME13-01-0122.

McCormick, G. P. (1977). A modification of Armijo's step-size rule for negative curvature. *Mathematical Programming*, *13*, 111–115. doi:10.1007/BF01584328.

Olivares, A., Moguerza, J. M., & Prieto, F. J. (2008). Nonconvex optimization using negative curvature within a modified linesearch. *European Journal of Operational Research.*, *189*, 706–722. doi:10.1016/j.ejor.2006.09.097.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing Vienna, Austria.

Reckase, M. D. (2009). *Multidimensional Item Response Theory Models*. Springer, New York.

Reise, S. P., & Revicki, D. A. (2014). *Handbook of Item Response Theory Modeling: Applications to Typical Performance Assessment*. Routledge, New York.

Revelle, W. (2020). *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University Evanston, Illinois. URL: https://CRAN.R-project.org/package=psych R package version 2.0.9.

Schapire, R. E., & Freund, Y. (2012). *Boosting: Foundations and Algorithms*. MIT Press, Cambridge.

Schilling, S., & Bock, R. D. (2005). High-dimensional maximum marginal likelihood item factor analysis by adaptive quadrature. *Psychometrika*, *70*, 533–555. doi:10.1007/s11336-003-1141-x.

Seibold, H., Bernau, C., Boulesteix, A.-L., & De Bin, R. (2018). On the choice and influence of the number of boosting steps for high-dimensional linear Cox-models. *Computational Statistics*, *33*, 1195–1215. doi:10.1007/s00180-017-0773-8.

Sun, J., Chen, Y., Liu, J., Ying, Z., & Xin, T. (2016). Latent variable selection for multidimensional item response theory models via L_1 regularization. *Psychometrika*, *81*, 921–939. doi:10.1007/s11336-016-9529-6.

Tutz, G., & Binder, H. (2006). Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics*, *62*, 961–971. doi:10.1111/j.1541-0420.2006.00578.x.

Tutz, G., & Gertheiss, J. (2014). Rating scales as predictors—the old question of scale level and some answers. *Psychometrika*, *79*, 357–376. doi:10.1007/s11336-013-9343-3. Doi: 10.1007/s11336-013-9343-3.

Van Houwelingen, J. C., & Le Cessie, S. (1990). Predictive value of statistical models. *Statistics in Medicine*, *9*, 1303–1325. doi:10.1002/sim.4780091109.

Varin, C. (2008). On composite marginal likelihoods. *Advances in Statistical Analysis*, *92*, 1–28. doi:10.1007/s10182-008-0060-7.

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, *21*, 5–42. doi:10.2307/24309261.

Varin, C., & Vidoni, P. (2005). A note on composite likelihood inference and model selection. *Biometrika*, *92*, 519–528. doi:10.1093/biomet/92.3.519.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*, 49–67. doi:10.1111/j.1467-9868.2005.00532.x.