PSYCHOMETRIKA—VOL. 87, NO. 4, 1473–1502
DECEMBER 2022
https://doi.org/10.1007/s11336-022-09863-9

COMPUTATION FOR LATENT VARIABLE MODEL ESTIMATION: A UNIFIED STOCHASTIC PROXIMAL FRAMEWORK

SILIANG ZHANG
EAST CHINA NORMAL UNIVERSITY

YUNXIAO CHEN
LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Latent variable models have been playing a central role in psychometrics and related fields. In many modern applications, the inference based on latent variable models involves one or several of the following features: (1) the presence of many latent variables, (2) the observed and latent variables being continuous, discrete, or a combination of both, (3) constraints on parameters, and (4) penalties on parameters to impose model parsimony. The estimation often involves maximizing an objective function based on a marginal likelihood/pseudo-likelihood, possibly with constraints and/or penalties on parameters. Solving this optimization problem is highly non-trivial, due to the complexities brought by the features mentioned above. Although several efficient algorithms have been proposed, there lacks a unified computational framework that takes all these features into account. In this paper, we fill the gap. Specifically, we provide a unified formulation for the optimization problem and then propose a quasi-Newton stochastic proximal algorithm. Theoretical properties of the proposed algorithms are established. The computational efficiency and robustness are shown by simulation studies under various settings for latent variable model estimation.

Key words: latent variable models, penalized estimator, stochastic approximation, proximal algorithm, quasi-Newton methods, Polyak–Ruppert averaging.

Latent variable models have been playing a central role in psychometrics and related fields. Commonly used latent variable models include item response theory models (Embretson & Reise 2000; Reckase 2009), latent class models (Clogg, 1995; Rupp et al., 2010; von Davier & Lee, 2019), structural equation models (Bollen, 1989), error-in-variable models (Carroll et al., 2006), random-effects models (Hsiao, 2014), and models for missing data (Little & Rubin, 1987), where latent variables have different interpretations, such as hypothetical constructs, 'true' variables measured with error, unobserved heterogeneity, and missing data. We refer the readers to Rabe-Hesketh & Skrondal (2004) and Bartholomew et al. (2011) for a comprehensive review of latent variable models.

A latent variable model contains unobserved latent variables and unknown parameters. For example, an item response theory model contains individual-specific latent traits as latent variables and item-specific parameters as model parameters. Comparing with models without latent variables, such as linear regression and generalized linear regression, the estimation of latent variable models is typically more involved. This estimation problem can be viewed from three perspectives: (1) fixed latent variables and parameters, (2) random latent variables and fixed parameters, and (3) random latent variables and parameters.

The first perspective, i.e., fixed latent variables and parameters, leads to the *joint maximum likelihood* (JML) estimator. This estimator can often be efficiently computed, for example, by an

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11336-022-09863-9.

Correspondence should be made to Yunxiao Chen, Department of Statistics, London School of Economics and Political Science, London, England. Email: y.chen186@lse.ac.uk

© 2022 The Author(s)

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

alternating minimization algorithm (Birnbaum, 1968; Chen et al., 2019; 2020). Unfortunately, however, the JML estimator is typically statistically inconsistent (Neyman & Scott, 1948; Andersen, 1973; Haberman, 1977; Ghosh, 1995), except under some high-dimensional asymptotic regime that is suitable for large-scale applications (Chen et al., 2019; 2020; Haberman, 1977; 2004). Treating both latent variables and parameters as random variables, the third perspective leads to a full Bayesian estimator, for which many Markov chain Monte Carlo (MCMC) algorithms have been developed (e.g., Béguin & Glas, 2001; Bolt & Lall, 2003; Dunson, 2000; 2003; Edwards, 2010).

The second perspective, i.e., random latent variables and fixed parameters, essentially follows an *empirical Bayes* (EB) approach (Robbins, 1956; C.-H. Zhang, 2003). This perspective is the most commonly adopted one (Rabe-Hesketh & Skrondal, 2004). Throughout the paper, we refer to estimators derived under this perspective as EB estimators. Both the full-information *marginal maximum likelihood* (MML) estimator (Bock & Aitkin, 1981) and the limited-information *composite maximum likelihood* (CML) estimator (Jöreskog & Moustaki, 2001; Vasdekis et al., 2012) can be viewed as special cases. Such estimators involve optimizing an objective function with respect to the fixed parameters, while the objective function is often intractable due to an integral with respect to the latent variables. The most commonly used algorithm for this optimization problem is the *expectation-maximization* (EM) algorithm (Dempster et al., 1977; Bock & Aitkin, 1981). This algorithm typically requires to iteratively evaluate numerical integrals with respective to the latent variables, which is often computationally unaffordable when the dimension of the latent space is high.

A high-dimensional latent space is not the only challenge to the computation of EB estimators. Penalties and constraints on parameters may also involve in the optimization, further complicating the computation. In fact, penalized estimators have become increasingly more popular in latent variable analysis for learning sparse structure, with applications to restricted latent class analysis, exploratory item factor analysis, variable selection in structural equation models, differential item functioning analysis, among others (Chen et al., 2015; Sun et al., 2016; Chen et al., 2018; Lindstrøm & Dahl, 2020; Tutz & Schauberger, 2015; Jacobucci et al., 2016; Magis et al., 2015). The penalty function is often non-smooth (e.g., Lasso penalty, Tibshirani, 1996), for which many standard optimization tools (e.g., gradient descent methods) are not applicable. In addition, complex inequality constraints are also commonly encountered in latent variable estimation, for example, in structural equation models (Van De Schoot et al., 2010) and restricted latent class models (e.g., de la Torre, 2011, Xu, 2017). Such complex constraints further complicate the optimization.

In this paper, we propose a quasi-Newton stochastic proximal algorithm that simultaneously tackles the computational challenges mentioned above. This algorithm can be viewed as an extension of the stochastic approximation (SA) method (Robbins & Monro, 1951). Comparing with SA, the proposed method converges faster and is more robust, thanks to the use of Polyak–Ruppert averaging (Polyak & Juditsky, 1992; Ruppert, 1988). The proposed method can also be viewed as a stochastic version of a proximal gradient descent algorithm (Chapter 4, Parikh & Boyd, 2014), in which constraints and penalties are handled by a proximal update. As will be illustrated by examples later, the proximal update is easy to evaluate for many commonly used penalties and constraints, making the proposed algorithm computationally efficient. Theoretical properties of the proposed method are established, showing that the proposed one is almost optimal in its convergence speed.

The proposed method is closely related to the stochastic-EM algorithm (Celeux, 1985; Ip, 2002; Nielsen, 2000; S. Zhang et al., 2020b) and the MCMC stochastic approximation algorithms (Cai, 2010a; 2010b; Gu & Kong, 1998), two popular methods for latent variable model estimation. Although these methods perform well in many problems, they are not as powerful as the proposed one. Specifically, the MCMC stochastic approximation algorithms cannot handle complex inequality constraints or non-smooth penalties, because they rely on stochastic gradients which

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

do not always exist when there are complex inequality constraints or non-smooth penalties. In addition, as will be discussed later, both the stochastic-EM algorithm and the MCMC stochastic approximation algorithms are computationally less efficient than the proposed method, even for estimation problems without complex constraints or penalties.

The proposed method is also closely related to a perturbed proximal gradient algorithm proposed in Atchadé et al. (2017). The current development improves upon that of Atchadé et al. (2017) from two aspects. First, the proposed method is a Quasi-Newton method, in which the second-order information (i.e., second derivatives) of the objective function is used in the update. Although this step may only change the asymptotic convergence speed by a constant factor (when the number of iterations grows to infinity), our simulation study suggests that the new method converges much faster than that of Atchadé et al. (2017) empirically. Second, the theoretical analysis of Atchadé et al. (2017) only considers a convex optimization setting, while we consider a non-convex setting which is typically the case for latent variable model estimation. Note that the analysis is much more involved when the objective function is non-convex. Therefore, our proof of sequence convergence is different from that of Atchadé et al. (2017). Specifically, the convergence theory is established by analyzing the convergence of a set-valued generalization of an ordinary differential equation (ODE).

The rest of the paper is organized as follows. In Sect. 1, we formulate latent variable model estimation as a general optimization problem which covers many commonly used estimators as special cases. In Sect. 2, a quasi-Newton stochastic proximal algorithm is proposed. Theoretical properties of the proposed algorithm are established in Sect. 3, suggesting that the proposed algorithm achieves the optimal convergence rate. The performance of the proposed algorithm is demonstrated and compared with other estimators by simulation studies in Sect. 4. We conclude with some discussions in Sect. 5. An R package has been developed that can be found on https://github.com/slzhang-fd/lvmcomp2.

1. Estimation of Latent Variable Models

1.1. Problem Setup

We consider the estimation of a parametric latent variable model. We adopt a general setting, followed by concrete examples in Sects. 1.2 and 1.3. Let $\mathbf{Y}$ be a random object representing observable data and let $\mathbf{y}$ be its realization. For example, in item factor analysis (IFA), $\mathbf{Y}$ represents (categorical) responses to all the items from all the respondents. A latent variable model specifies the distribution of $\mathbf{Y}$ by introducing a set of latent variables $\boldsymbol{\xi} \in \Xi$, where Ξ denotes the state space of the latent vector $\boldsymbol{\xi}$. For example, in item factor analysis, $\boldsymbol{\xi}$ consists of the latent traits of all the respondents and Ξ is a Euclidean space. Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top \in \mathcal{B}$ be a set of parameters in the model, where $\mathcal{B}$ denotes the parameter space. The goal is to estimate $\boldsymbol{\beta}$ given observed data $\mathbf{y}$.

We consider an EB estimator which takes the form

$$l(\boldsymbol{\beta}) = \log\left(\int_{\Xi} f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}) d\boldsymbol{\xi}\right), \tag{1}$$

where $f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta})$ is a complete-data likelihood/pseudo-likelihood function that has an analytic form. We assume that the objective function $l(\boldsymbol{\beta})$ is finite for any $\boldsymbol{\beta} \in \mathcal{B}$ and is also smooth in $\boldsymbol{\beta}$.

The estimator is given by solving the following optimization problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathcal{B}}{\arg\min} -l(\boldsymbol{\beta}) + R(\boldsymbol{\beta}), \tag{2}$$

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

where $R(\boldsymbol{\beta})$ is a penalty function that has an analytic form, such as Lasso, ridge, or elastic net regularization functions. Note that the penalty function often depends on tuning parameters. Throughout this paper, we assume these tuning parameters are fixed and thus do not explicitly indicate them in the objective function (2). In practice, tuning parameters are often unknown and need to be chosen by cross-validation or certain information criterion. We point out that many commonly used estimators take the form of (2), including the MML estimator, the CML estimator, and regularized estimators based on the MML and CML. We also point out that despite its general applicability to latent variable estimation problems, the proposed method is more useful for complex problems that cannot be easily solved by the classical EM algorithm. For certain problems, such as the estimation of linear factor models and simple latent class models, both the E- and M-step of the EM algorithm have closed-form solutions. In that situation, the classical EM algorithm may be computationally more efficient, though the proposed method can still be used.

1.2. High-dimensional Item Factor Analysis

Item factor analysis models are commonly used in social and behavioral sciences for analyzing categorical response data. For exposition, we focus on binary response data and point out that the extension to ordinal response data is straightforward. Consider N individuals responding to J binary-scored items. Let $Y_{ij} \in \{0, 1\}$ be a random variable denoting person i's response to item j and let y_{ij} be its realization. Thus, we have $\mathbf{Y} = (Y_{ij})_{N\times J}$ and $\mathbf{y} = (y_{ij})_{N\times J}$, where $\mathbf{Y}$ and $\mathbf{y}$ are the generic notations introduced in Sect. 1.1 for our data. A comprehensive review of IFA models and their estimation can be found in Chen & Zhang (2020a).

It is assumed that the dependence among an individual's responses is driven by a set of latent factors, denoted by $\boldsymbol{\xi} = (\xi_{ik})_{N\times K}$, where ξ_{ik} represents person i's kth factor. Recall that $\boldsymbol{\xi}$ is our generic notation for the latent variables in Sect. 1.1 and here the state space $\Xi = \mathbb{R}^{N\times K}$. Throughout this paper, we assume the number of factors K is known.

An IFA model makes the following assumptions:

1. $\boldsymbol{\xi}_i = (\xi_{i1}, \ldots, \xi_{iK})^\top$, $i = 1, \ldots, N$, are independent and identically distributed (i.i.d.) random vectors, following a multivariate normal distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$. The diagonal terms of $\boldsymbol{\Sigma} = (\sigma_{kk'})_{K\times K}$ are set to one for model identification. As $\boldsymbol{\Sigma}$ is a positive semi-definite matrix, it is common to reparametrize $\boldsymbol{\Sigma}$ by Cholesky decomposition,

$$\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top,$$

   where $\mathbf{B} = (b_{kk'})_{K\times K}$ is a lower triangular matrix. Let $\mathbf{b}_k$ be the kth row of $\mathbf{B}$. Then $\|\mathbf{b}_k\| = 1$, $k = 1, \ldots, K$, since the diagonal terms of $\boldsymbol{\Sigma}$ are constrained to value 1.
2. Y_{ij} given $\boldsymbol{\xi}_i$ follows a Bernoulli distribution satisfying

$$\mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i) = \frac{\exp(d_j + \mathbf{a}_j^\top \boldsymbol{\xi}_i)}{1 + \exp(d_j + \mathbf{a}_j^\top \boldsymbol{\xi}_i)}, \tag{3}$$

   where d_j and $\mathbf{a}_j = (a_{j1}, \ldots, a_{jK})^\top$ are item-specific parameters. The parameters a_{jk} are often known as the loading parameters.
3. Y_{i1},..., Y_{iJ} are assumed to be conditionally independent given $\boldsymbol{\xi}_i$, which is known as the local independence assumption.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Note that we consider the most commonly used logistic model in (3). It is worth pointing out that the proposed algorithm also applies to the normal ogive (i.e., probit) model which assumes that $\mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i) = \Phi(d_j + \mathbf{a}_j^\top \boldsymbol{\xi}_i)$. Under the current setting and using the reparametrization for $\boldsymbol{\Sigma}$, our model parameters are $\boldsymbol{\beta} = \{\mathbf{B}, d_j, \mathbf{a}_j, j = 1, \ldots, J\}$. The marginal likelihood function takes the form

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{N} \log \left(\int_{\mathbf{x} \in \mathbb{R}^K} \prod_{j=1}^{J} \frac{\exp[y_{ij}(d_j + \mathbf{a}_j^\top \boldsymbol{x})]}{1 + \exp(d_j + \mathbf{a}_j^\top \boldsymbol{x})} \phi(\mathbf{x} \mid \mathbf{B}) d\mathbf{x} \right), \tag{4}$$

where $\phi(\mathbf{x} \mid \mathbf{B})$ is the density function for multivariate normal distribution $N(\mathbf{0}, \mathbf{B}\mathbf{B}^\top)$. The K-dimensional integrals involved in (4) cause a high computational burden for a relatively large K (e.g., $K \geq 5$).

IFA models are commonly used for both exploratory and confirmatory analyses. In exploratory IFA, an important problem is to learn a sparse loading matrix $(a_{ij})_{J \times K}$ from data, which facilitates the interpretation of the factors. One approach is by the L_1-regularized estimator (Sun et al., 2016) which takes the form

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathcal{B}} -l(\boldsymbol{\beta}) + R(\boldsymbol{\beta}), \tag{5}$$

where the parameter space

$$\mathcal{B} = \{\boldsymbol{\beta} \in \mathbb{R}^p : b_{kk'} = 0, 1 \leq k < k' \leq K, \sum_{k'=1}^{K} b_{kk'}^2 = 1, k = 1, \ldots, K\},$$

and the penalty term

$$R(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{J} \sum_{k=1}^{K} |a_{jk}|. \tag{6}$$

In $R(\boldsymbol{\beta})$, $\lambda > 0$ is a tuning parameter assumed to be fixed throughout this paper. This regularized estimator resolves the rotational indeterminacy issue in exploratory IFA, as the L_1 penalty term is not rotational invariant. Consequently, under mild regularity conditions, the loading matrix can be consistently estimated only up to a column swapping. Note that only the $\mathbf{B}$ matrix has constraints, as reflected by the parameter space $\mathcal{B}$. Here $b_{kk'} = 0$ is due to that $\mathbf{B}$ is a lower triangle matrix and $\sum_{k'=1}^{K} b_{kk'}^2 = 1$ is due to that the diagonal terms of $\boldsymbol{\Sigma} = \mathbf{B}\mathbf{B}^\top$ are all 1. We remark that it is possible to replace the L_1 penalty in $R(\boldsymbol{\beta})$ by other penalty functions for imposing sparsity, such as the elastic net penalty (Zou & Hastie, 2005)

$$R(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{J} \sum_{k=1}^{K} a_{jk}^2 + \lambda_2 \sum_{j=1}^{J} \sum_{k=1}^{K} |a_{jk}|, \tag{7}$$

where $\lambda_1, \lambda_2 > 0$ are two tuning parameters.

In confirmatory IFA, zero constraints are imposed on loading parameters, based on prior knowledge about the measurement design. More precisely, these zero constraints can be coded

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

by a binary matrix $\mathbf{Q} = (q_{jk})_{J\times K}$. If $q_{jk} = 0$, then item j does not load on factor k and a_{jk} is set to 0. Otherwise, a_{jk} is freely estimated. These constraints lead to parameter space $\mathcal{B} = \{\boldsymbol{\beta} : b_{kk'} = 0, 1 \le k < k' \le K; \sum_{k'=1}^{K} b_{kk'}^2 = 1, a_{jk} = 0 \text{ for } q_{jk} = 0, j = 1, \ldots, J, k = 1, \ldots, K\}$. The MML estimator for confirmatory IFA is then given by

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathcal{B}} -l(\boldsymbol{\beta}). \tag{8}$$

Besides parameter estimation, another problem of interest in confirmatory IFA is to make statistical inference, for which it is required to compute the asymptotic variance of $\hat{\boldsymbol{\beta}}$. The estimation of the asymptotic variance often requires to compute the Hessian matrix of $l(\boldsymbol{\beta})$ at $\hat{\boldsymbol{\beta}}$, which also involves intractable K-dimensional integrals. As we will see in Sect. 2.1, this Hessian matrix, as well as quantities taking a similar form, can be easily obtained as a by-product of the proposed algorithm.

1.3. Restricted Latent Class Model

Our second example is restricted latent class models which are also widely used in social and behavioral sciences. For example, they are commonly used in education for cognitive diagnosis (von Davier & Lee, 2019). These models differ from IFA models in that they assume discrete latent variables. Here, we consider a setting for cognitive diagnosis when both data and latent variables are binary. Consider data taking the same form as that for IFA, denoted by $\mathbf{Y} = (Y_{ij})_{N\times J}$ and $\mathbf{y} = (y_{ij})_{N\times J}$. In this context, $Y_{ij} = 1$ means that item j is answered correctly and $Y_{ij} = 0$ means an incorrect answer.

The restricted latent class model assumes that each individual is characterized by a K-dimensional latent vector $\boldsymbol{\xi}_i = (\xi_{i1}, \ldots, \xi_{iK})^\top$, $i = 1, \ldots, N$, where $\xi_{ik} \in \{0, 1\}$. Thus, the latent variables are $\boldsymbol{\xi} = (\xi_{ik})_{N\times K}$, whose state space $\Xi = \{0, 1\}^{N\times K}$ contains all $N \times K$ binary matrices. Each dimension of $\boldsymbol{\xi}_i$ represents a skill, and $\xi_{ik} = 1$ indicates that person i has mastered the kth skill and $\xi_{ik} = 0$ otherwise.

The restricted latent class model can be parametrized as follows.

1. The person-specific latent vectors $\boldsymbol{\xi}_i$, $i = 1, \ldots, N$, are i.i.d., following a categorical distribution satisfying

$$\mathbb{P}(\boldsymbol{\xi}_i = \boldsymbol{\alpha}) = \frac{\exp(\nu_{\boldsymbol{\alpha}})}{\sum_{\boldsymbol{\alpha}'\in\{0,1\}^K} \exp(\nu_{\boldsymbol{\alpha}'})},$$

   where $\boldsymbol{\alpha} \in \{0, 1\}^K$ represents an attribute profile representing the mastery status on all K attributes, and we set $\nu_{\boldsymbol{\alpha}'} = 0$ as the baseline, for $\boldsymbol{\alpha}' = (0, \ldots, 0)^\top$.
2. Y_{ij} given $\boldsymbol{\xi}_i$ follows a Bernoulli distribution, satisfying

$$\mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \boldsymbol{\alpha}) = \frac{\exp(\theta_{j,\boldsymbol{\alpha}})}{1 + \exp(\theta_{j,\boldsymbol{\alpha}})}, \ \boldsymbol{\alpha} \in \{0, 1\}^K.$$

3. Local independence is still assumed. That is, Y_{i1},…, Y_{iJ} are conditionally independent given $\boldsymbol{\xi}_i$.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

The above model specification leads to a marginal likelihood function

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{N} \log \left(\sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \frac{\exp(\nu_{\boldsymbol{\alpha}})}{\sum_{\boldsymbol{\alpha}' \in \{0,1\}^K} \exp(\nu_{\boldsymbol{\alpha}'})} \prod_{j=1}^{J} \frac{\exp(y_{ij}\theta_{j,\boldsymbol{\alpha}})}{1+\exp(\theta_{j,\boldsymbol{\alpha}})} \right), \tag{9}$$

where $\boldsymbol{\beta} = \{\nu_{\boldsymbol{\alpha}}, \theta_{j,\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \{0,1\}^K, j = 1, \ldots, J\}$.

We consider a confirmatory setting where there exists a design matrix, similar to the $\mathbf{Q}$-matrix in confirmatory IFA. With slight abuse of notation, we still denote $\mathbf{Q} = (q_{jk})_{J \times K}$, where $q_{jk} \in \{0, 1\}$. Here, $q_{jk} = 1$ indicates that solving item j requires the kth skill and $q_{jk} = 0$ otherwise. As will be explained below, this design matrix leads to equality and inequality constraints in model parameters.

Denote $\mathbf{q}_j = (q_{j1}, \ldots, q_{jK})^\top$ as the design vector for item j. For $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^\top$, we write

$$\boldsymbol{\alpha} \succeq \mathbf{q}_j, \text{ if } \alpha_k \geq q_{jk} \text{ for all } k \in \{1, \ldots, K\},$$

and write

$$\boldsymbol{\alpha} \not\succeq \mathbf{q}_j, \text{ if there exists } k \text{ such that } \alpha_k < q_{jk}.$$

That is, $\boldsymbol{\alpha} \succeq \mathbf{q}_j$ if profile $\boldsymbol{\alpha}$ has all the skills needed for solving item j and $\boldsymbol{\alpha} \not\succeq \mathbf{q}_j$ if not. The design information leads to the following constraints:

1. $\mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \boldsymbol{\alpha}) = \mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \boldsymbol{\alpha}')$, if both $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \succeq \mathbf{q}_j$. That is, individuals who have mastered all the required skills have the same chance of answering the item correctly.
2. $\mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \boldsymbol{\alpha}) \geq \mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \boldsymbol{\alpha}')$ if $\boldsymbol{\alpha} \succeq \mathbf{q}_j$ and $\boldsymbol{\alpha}' \not\succeq \mathbf{q}_j$. That is, students who have mastered all the required skills have a higher chance of answering the item correctly than those who do not.
3. $\mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \boldsymbol{\alpha}) \geq \mathbb{P}(Y_{ij} = 1 \mid \boldsymbol{\xi}_i = \mathbf{0})$ for all $\boldsymbol{\alpha}$. That is, students who have not mastered any skill have the lowest chance of answering correctly.

We refer the readers to Xu (2017) for more discussions on these constraints which are key to the identification of this model. Under these constraints, the MML estimator is given by

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathcal{B}} -l(\boldsymbol{\beta}), \tag{10}$$

where

$$\mathcal{B} = \{\boldsymbol{\beta} : \max_{\boldsymbol{\alpha} \succeq \mathbf{q}_j} \theta_{j,\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha} \succeq \mathbf{q}_j} \theta_{j,\boldsymbol{\alpha}} \geq \theta_{j,\boldsymbol{\alpha}'} \geq \theta_{j,\mathbf{0}}, \text{ if } \boldsymbol{\alpha}' \not\succeq \mathbf{q}_j, \ \nu_{\mathbf{0}} = 0\}.$$

When K is relatively large, the computation for solving (10) becomes challenging, due to both the summation over 2^K possible values of $\boldsymbol{\alpha}$ in $l(\boldsymbol{\beta})$, and the large number of inequality constraints.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

2. Stochastic Proximal Algorithm

In this section, we propose a quasi-Newton stochastic proximal algorithm for the computation of (2). The description in this section will focus on the computation aspect, without emphasizing the regularity conditions needed for its convergence. A rigorous theoretical treatment will be given in Sect. 3. In what follows, we describe the algorithm in its general form in Sect. 2.1, followed by details for two specific models in Sects. 2.2 and 2.3, and finally comparisons with related algorithms in Sect. 2.4.

2.1. General Algorithm

For ease of exposition, we introduce some new notation. We write the penalty function as the sum of two terms, $R(\boldsymbol{\beta}) = R_1(\boldsymbol{\beta}) + R_2(\boldsymbol{\beta})$, where $R_1(\boldsymbol{\beta})$ is a smooth function and $R_2(\boldsymbol{\beta})$ is non-smooth. In the example of regularized estimation for exploratory IFA, $R_1(\boldsymbol{\beta}) = 0$ and $R_2(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{J} \sum_{k=1}^{K} |a_{jk}|$, when $R(\boldsymbol{\beta})$ is an L_1 penalty as in (6). When an elastic net penalty is used as in (7), $R_1(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{J} \sum_{k=1}^{K} a_{jk}^2$ and $R_2(\boldsymbol{\beta}) = \lambda_2 \sum_{j=1}^{J} \sum_{k=1}^{K} |a_{jk}|$.

The optimization problem can be reexpressed as

$$\min_{\boldsymbol{\beta}} \; h(\boldsymbol{\beta}) + g(\boldsymbol{\beta}), \tag{11}$$

where $h(\boldsymbol{\beta}) = -l(\boldsymbol{\beta}) + R_1(\boldsymbol{\beta})$ and $g : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ is a generalized function taking the form $g(\boldsymbol{\beta}) = R_2(\boldsymbol{\beta}) + I_{\mathcal{B}}(\boldsymbol{\beta})$, where

$$I_{\mathcal{B}}(\boldsymbol{\beta}) = \begin{cases} 0, & \text{if } \boldsymbol{\beta} \in \mathcal{B}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{12}$$

Note that since both $l(\boldsymbol{\beta})$ and $R_1(\boldsymbol{\beta})$ are smooth in $\boldsymbol{\beta}$, $h(\boldsymbol{\beta})$ is still smooth in $\boldsymbol{\beta}$. The second term $g(\boldsymbol{\beta})$ is non-smooth in $\boldsymbol{\beta}$, unless it is degenerate (i.e., $g(\boldsymbol{\beta}) \equiv 0$). We further write

$$H(\boldsymbol{\xi}, \boldsymbol{\beta}) = -\log f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}) + R_1(\boldsymbol{\beta}), \tag{13}$$

which can be viewed as a complete-data version of $h(\boldsymbol{\beta})$ that will be used in the algorithm.

The algorithm relies on a scaled proximal operator (Lee et al., 2014) for the g function, defined as

$$\text{Prox}_{\gamma,g}^{\mathbf{D}}(\boldsymbol{\beta}) = \arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ g(\mathbf{x}) + \frac{1}{2\gamma} \|\mathbf{x} - \boldsymbol{\beta}\|_{\mathbf{D}}^2 \right\},$$

where $\gamma > 0$, $\mathbf{D}$ is a strictly positive definite matrix, and $\|\cdot\|_{\mathbf{D}}$ is a norm defined by $\|\mathbf{x}\|_{\mathbf{D}}^2 = \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathbf{D}} = \mathbf{x}^\top \mathbf{D} \mathbf{x}$. The choices of γ, $\mathbf{D}$, and the intuition behind the proximal operator will be explained in the sequel.

Our general algorithm is described in Algorithm 1, followed by implementation details. The proposed algorithm is an extension of a perturbed proximal gradient algorithm (Atchadé et al., 2017). The major difference is that the proposed algorithm makes use of second-order information from the smooth part of the objective function, which can substantially speed up its convergence. See Sect. 2.4 for further comparison.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Algorithm 1. (Stochastic Proximal Algorithm)

Input *Data* $\mathbf{y}$*, initial parameters* $\boldsymbol{\beta}^{(0)}$*, a sequence of step size* γ_s*,* $s = 1, 2, \ldots$*, pre-specified tuning parameters* $c_2 \geq c_1 > 0$*, and burn-in size* ϖ*.*
Update *At* t*th iteration where* $t \geq 1$*, we perform the following two steps:*

1. ***Stochastic step*** *Sample* $\boldsymbol{\xi}$ *from the conditional distribution of* $\boldsymbol{\xi}$ *given* $\mathbf{y}$*,*

$$\psi(\boldsymbol{\xi}) \propto f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}^{(t-1)}),$$

and obtain $\boldsymbol{\xi}^{(t)}$*. The sampling can be either exact or approximated by MCMC.*
2. ***Proximal step*** *Update model parameters by*

$$\boldsymbol{\beta}^{(t)} = Prox_{\gamma_t, g}^{\mathbf{D}^{(t)}}\left(\boldsymbol{\beta}^{(t-1)} - \gamma_t (\mathbf{D}^{(t)})^{-1}\mathbf{G}^{(t)}\right), \tag{14}$$

where

$$\mathbf{G}^{(t)} = \left.\frac{\partial H(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t-1)}}.$$

$\mathbf{D}^{(t)}$ *is a diagonal matrix with diagonal entries*

$$\delta_i^{(t)} = \frac{t-1}{t}\delta_i^{(t-1)} + \frac{1}{t}T\left(\tilde{\delta}_i^{(t)}; c_1, c_2\right),$$

where $T(x; c_1, c_2)$ *is a truncation function defined as*

$$T(x; c_1, c_2) = \begin{cases} c_1, & \text{if } x < c_1, \\ x, & \text{if } x \in [c_1, c_2], \\ c_2, & \text{if } x > c_2. \end{cases} \tag{15}$$

Here $\tilde{\delta}_i^{(t)} = \tilde{\delta}_{i,1}^{(t)} + (\tilde{\delta}_{i,2}^{(t)})^2$*, where*

$$\tilde{\delta}_{i,1}^{(t)} = (1-\gamma_t)\tilde{\delta}_{i,1}^{(t-1)} + \gamma_t\left(\left.\frac{\partial^2 H(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta})}{\partial \beta_i^2}\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t-1)}} - \left.\left(\frac{\partial H(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta})}{\partial \beta_i}\right)^2\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t-1)}}\right),$$

$$\tilde{\delta}_{i,2}^{(t)} = (1-\gamma_t)\tilde{\delta}_{i,2}^{(t-1)} + \gamma_t\left.\left(\frac{\partial H(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta})}{\partial \beta_i}\right)^2\right|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t-1)}}.$$

Iteratively perform these two steps until a stopping criterion is satisfied and let n *be the last iteration number.*
Output $\bar{\boldsymbol{\beta}}_n = \sum_{t=\varpi+1}^{n} \boldsymbol{\beta}^{(t)}/(n-\varpi)$.

In what follows, we make a few remarks to provide some intuitions about the algorithm.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Remark 1. (Connection with stochastic gradient descent) To provide some intuitions about the proposed method, we first make a connection between the proposed method and the stochastic gradient descent (SGD) algorithm. In fact, when the sampling of $\boldsymbol{\xi}$ is exact in the stochastic step, then $\mathbf{G}^{(t)}$ is a stochastic gradient of the smooth part of our objective function, in the sense that $\mathbb{E}(\mathbf{G}^{(t)} \mid \mathbf{y}, \boldsymbol{\beta}^{(t-1)}) = \nabla h(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t-1)}}$. If, in addition, there is no constraint or non-smooth penalty, i.e., $g(\boldsymbol{\beta}) \equiv 0$, then the proximal step degenerates to an SGD update $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \gamma_t(\mathbf{D}^{(t)})^{-1}\mathbf{G}^{(t)}$. In that case, the proposed method becomes a version of SGD.

Remark 2. (Proximal step) We provide some intuitions about the proximal step. We start with two special cases. First, as mentioned in Remark 1, if there is no constraint or non-smooth penalty, then the proximal step is nothing but a stochastic gradient descent step. This is because, the scaled proximal operator degenerates to an identity map, i.e., $\text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta}) = \boldsymbol{\beta}$. Second, when the g function involves constraints but does not contain a non-smooth penalty, then the proximal step is a projected stochastic gradient descent step. That is, one first performs a stochastic gradient descent update $\tilde{\boldsymbol{\beta}}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \gamma_t(\mathbf{D}^{(t)})^{-1}\mathbf{G}^{(t)}$. Then $\tilde{\boldsymbol{\beta}}^{(t)}$ is projected back to the feasible region $\mathcal{B}$ by the scaled proximal operator:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\mathcal{B}} \|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}^{(t)}\|_{\mathbf{D}},$$

which is a projection under the norm $\|\cdot\|_{\mathbf{D}}$. When $\mathbf{D}$ is an identity matrix as in the vanilla (i.e., non-scaled) proximal operator, then the projection is based on the Euclidean distance.

More generally, when the g function involves non-smooth penalties, then the proximal step can be viewed as minimizing the sum of $g(\boldsymbol{\beta})$ and a quadratic approximation of $h(\boldsymbol{\beta})$ at $\boldsymbol{\beta}^{(t-1)}$; see Lee et al. (2014) for more explanations. We provide an example to facilitate the understanding. Suppose that

$$g(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{p} |\beta_i|$$

is the Lasso penalty, and $\mathbf{D} = diag(\delta_1, \ldots, \delta_p)$ is a diagonal matrix, where $\lambda, \delta_i > 0$, $i = 1, \ldots, p$. Then $\text{Prox}^{\mathbf{D}}_{\gamma,g}(\tilde{\boldsymbol{\beta}}^{(t)})$ involves solving p optimization problems separately, each of which takes the form

$$\hat{\beta}_i = \arg\min_{x} \frac{1}{2}(x - \tilde{\beta}_i^{(t)})^2 + \frac{\lambda\gamma}{\delta_i}|x|. \tag{16}$$

It is well known that (16) has a closed-form solution given by soft-thresholding (see Chapter 3, Friedman et al., 2001):

$$\hat{\beta}_i = \begin{cases} \tilde{\beta}_i^{(t)} - \frac{\lambda\gamma}{\delta_i}, & \text{if } \tilde{\beta}_i^{(t)} > \frac{\lambda\gamma}{\delta_i}, \\ \tilde{\beta}_i^{(t)} + \frac{\lambda\gamma}{\delta_i}, & \text{if } \tilde{\beta}_i^{(t)} < -\frac{\lambda\gamma}{\delta_i}, \\ 0, & \text{otherwise.} \end{cases}$$

Remark 3. (Role of $\mathbf{D}^{(t)}$) Our proximal step is a quasi-Newton proximal update proposed in Lee et al. (2014) under a non-stochastic optimization setting. As shown in Lee et al. (2014), quasi-Newton proximal methods converge faster than first-order proximal methods under the

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

non-stochastic setting. Here, the diagonal matrix $\mathbf{D}^{(t)}$ is used to approximate the Hessian matrix of $h(\boldsymbol{\beta})$ at $\boldsymbol{\beta}^{(t)}$. When $\boldsymbol{\beta}^{(t)}$ converges to $\hat{\boldsymbol{\beta}}$, then $\delta_i^{(t)}$, the ith diagonal term of $\mathbf{D}^{(t)}$ converges to $T\left(\frac{\partial^2 h}{\partial \beta_i^2}|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}; c_1, c_2\right)$ where T is the truncation function defined in (15); see Remark 8 for more explanations.

In the proposed update, we choose $\mathbf{D}^{(t)}$ to be a diagonal matrix for computational convenience. Specifically, as discussed in Remark 2, the proximal step is in a closed form when $\mathbf{D}^{(t)}$ is a diagonal matrix. In addition, the proximal step requires to calculate the inverse of $\mathbf{D}^{(t)}$, whose complexity is much lower when $\mathbf{D}^{(t)}$ is diagonal.

We point out that using a diagonal matrix to approximate the Hessian matrix is a popular and effective trick in numerical optimization (e.g., Chapter 5, Bertsekas et al., 1992; Becker & Le Cun, 1988), especially for large-scale optimization problems. In principle, it is possible to allow $\mathbf{D}^{(t)}$ to be non-diagonal. In fact, it is not difficult to generalize the BFGS updating formula for $\mathbf{D}^{(t)}$ given in Lee et al. (2014) to a stochastic version.

Our choice of $\mathbf{D}^{(t)}$ guarantees its eigenvalues to be constrained in the interval $[c_1, c_2]$. It rules out the singular situation when $\mathbf{D}^{(t)}$ is not strictly positive definite. In the implementation, we set c_1's to be a sufficiently small constant and set c_2's to be a sufficiently large constant. According to simulation, the algorithm tends to be insensitive to these choices.

We further provide some remarks regarding the implementation details.

Remark 4. (Choices of step size) As will be shown in Sect. 3, the convergence of the proposed method requires the step size to satisfy $\sum_{t=1}^{\infty} \gamma_t = \infty$ and $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$. This requirement is also needed in the Robbins–Monro algorithm. Here, we choose the step size $\gamma_t = \mu t^{-\frac{1}{2}-\varepsilon}$ so that the above requirement is satisfied, where μ is a positive constant and ε is a small positive constant. As will be shown in Sect. 3, with sufficiently small ε, $\bar{\boldsymbol{\beta}}_n$ is almost optimal in terms of its convergence speed. We point out that ε is needed to prove the convergence of $\bar{\boldsymbol{\beta}}_n$, under our non-convex setting. It is not needed, if the objective function (2) is convex; see Atchadé et al. (2017). The requirement of ε may be an artifact due to our proof strategy. Simulation results show that the algorithm converges well even if we set $\varepsilon = 0$. For the numerical analysis in this paper, we set $\varepsilon = 10^{-2}$.

We point out that our choice of step size is very different from the step size in the Robbins–Monro algorithm, for which asymptotic results (Fabian, 1968) suggest that the optimal choice of step size satisfies $\gamma_t = O(1/t)$.

Remark 5. (Starting point) As the objective function (2) is typically non-convex for most latent variable models, the choice of the starting point $\boldsymbol{\beta}^{(0)}$ matters. The algorithm is more likely to converge to the global optimum given a good starting point. One strategy is to run the proposed algorithm with multiple random starting points and then choose the best-fitting solution. Alternatively, one may find a good starting point using less accurate but computationally faster estimators, such as the constrained joint maximum likelihood estimator (Chen et al., 2019; 2020) or spectral methods (H. Zhang et al., 2020a). Moreover, to further avoid convergence to local optima, one may also use multiple random starting points and choose the one with the smallest objective function value.

Remark 6. (Sampling in stochastic step) As mentioned in Remark 1, when the latent variables $\boldsymbol{\xi}$ can be sampled exactly in the stochastic step, then $\mathbf{G}^{(t)}$ is a stochastic gradient of $h(\boldsymbol{\beta})$. Unfortunately, exact sampling is only possible under some situations such as restricted latent class analysis. In most cases, we only have approximate samples from an MCMC algorithm. For example, as discussed below, the latent variables in IFA can be sampled by a block-wise Gibbs sampler. With approximate samples, $\mathbf{G}^{(t)}$ is only approximately unbiased. As we show in Sect. 3, such $\mathbf{G}^{(t)}$ may still yield convergence of $\bar{\boldsymbol{\beta}}_n$.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Remark 7 (Stopping criterion) In the implementation of Algorithm 1, we stop the iterative update by monitoring a window of successive differences in $\boldsymbol{\beta}^{(t)}$. More precisely, we stop the iteration if all differences in the window are less than a given threshold. Unless otherwise stated, the numerical analysis in this paper uses a window size 3. The same stopping criterion is also adopted by the Metropolis–Hasting Robins–Monro algorithm proposed by Cai (2010a).

Finally, as we explain in Remark 8, certain quantities, including the Hessian matrix of $l(\boldsymbol{\beta})$, can be obtained as a by-product of the proposed algorithm.

Remark 8 (By-product) It is often of interest to compute quantities of the form

$$\hat{M} = \mathbb{E}\left[m(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}) \mid \mathbf{y}, \boldsymbol{\beta}\right]\big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}, \tag{17}$$

where $m(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta})$ is a given function with an analytic form and the conditional expectation $\mathbb{E}\left[\cdot \mid \mathbf{y}, \boldsymbol{\beta}\right]$ is with respect to the conditional distribution of $\boldsymbol{\xi}$ given $\mathbf{y}$. The quantity (17) is intractable due to the high-dimensional integral with respect to $\boldsymbol{\xi}$. One such example is the Hessian matrix of $l(\boldsymbol{\beta})$ at $\hat{\boldsymbol{\beta}}$ as discussed in Sect. 1.2 that is a key quantity for the statistical inference of $\hat{\boldsymbol{\beta}}$. In fact, by Louis' formula (Louis, 1982),

$$\begin{aligned}\frac{\partial^2 l(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top} =& \mathbb{E}\left[\frac{\partial^2(\log f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}))}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\top} + \frac{\partial(\log f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}))}{\partial\boldsymbol{\beta}}\left[\frac{\partial(\log f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}))}{\partial\boldsymbol{\beta}}\right]^\top \middle| \mathbf{y}, \boldsymbol{\beta}\right] \\ &- \mathbb{E}\left[\frac{\partial(\log f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta}))}{\partial\boldsymbol{\beta}} \middle| \mathbf{y}, \boldsymbol{\beta}\right]\left(\mathbb{E}\left[\frac{\partial(\log f(\mathbf{y}, \boldsymbol{\xi} \mid \boldsymbol{\beta})))}{\partial\boldsymbol{\beta}} \middle| \mathbf{y}, \boldsymbol{\beta}\right]\right)^\top.\end{aligned}$$

The computation of (17) is a straightforward by-product of the proposed algorithm. To approximate $\hat{M}$, we only need to add the following update in each iteration

$$M^{(t)} = M^{(t-1)} + \gamma_t\big(m(\mathbf{y}, \boldsymbol{\xi}^{(t)} \mid \boldsymbol{\beta}^{(t)}) - M^{(t-1)}\big), \tag{18}$$

for $t \geq 2$, where $M^{(1)} = m(\mathbf{y}, \boldsymbol{\xi}^{(1)} \mid \boldsymbol{\beta}^{(1)})$. We approximate $\hat{M}$ by the Polyak–Ruppert averaging $\bar{M}_n = (\sum_{t=\varpi+1}^{n} M^{(t)})/(n-\varpi)$. When the sequence $\boldsymbol{\beta}^{(t)}$ converges to $\hat{\boldsymbol{\beta}}$ (see Theorem 1 for the convergence analysis), under mild conditions, Theorem 3.17 of Ben-veniste et al. (1990) suggests the convergence of $M^{(n)}$ to $\hat{M}$ with probability 1, which further implies the convergence of $\bar{M}_n$ to $\hat{M}$. Note that we use the averaged estimator $\bar{M}_n$ as it tends to converge faster than the pre-average sequence $M^{(n)}$. We point out that the updating rule for the diagonal matrix $\mathbf{D}^{(t)}$ in Algorithm 1 makes use of such an averaged estimator.

Remark 9 (Burn-in size) Like MCMC algorithms, the proposed method also has a burn-in period, where parameter updates from that period are not used in the Polyak–Ruppert averaging. The choice of the burn-in size will not affect the asymptotic property of the method, but does affect the empirical performance. This is because, the parameter updates may be far away from the solution due to the effect of the starting point. Including them in the Polyak–Ruppert averaging may introduce a high bias. In our numerical analysis, the burn-in size ϖ is fixed to be sufficiently large in each of our examples. Adaptive choice of the burn-in size is possible; see S. Zhang et al. (2020b).

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

2.2. *Example I: Item Factor Analysis*

We now explain the details of using the proposed method to solve (5) for exploratory IFA. The computation is similar when replacing the L_1 regularization by the elastic net regularization. For confirmatory IFA, the stochastic step is the same as that of exploratory IFA and the proximal update step is straightforward as no penalty is involved. Therefore, the details for the computation of confirmatory IFA are omitted here.

We first consider the stochastic step for solving (5). Note that $\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_N$ are conditionally independent given data, and thus can be sampled separately. For each $\boldsymbol{\xi}_i$, we sample its entries by Gibbs sampling. More precisely, each entry is sampled by adaptive rejection sampling (Gilks & Wild, 1992; S. Zhang et al., 2020b), as the conditional distribution of ξ_{ik} given data and the other entries of $\boldsymbol{\xi}_i$ is log-concave. We refer the readers to S. Zhang et al. (2020b) for more explanations of this sampling procedure. If a normal ogive IFA is considered instead of the logistic model above, then we can sample $\boldsymbol{\xi}_i^{(t)}$ by a similar Gibbs method with a data augmentation trick; see Chen & Zhang (2020a) for a review.

We now discuss the computation for the proximal step. Recall that $\boldsymbol{\beta} = \{\mathbf{B}, d_j, \mathbf{a}_j, j = 1,\ldots,J\}$. We denote

$$\tilde{\boldsymbol{\beta}}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \gamma_t(\mathbf{D}^{(t)})^{-1}\mathbf{G}^{(t)}$$

as the input of the scaled proximal operator. The parameter update is given by

$$\boldsymbol{\beta}^{(t)} = \text{Prox}^{\mathbf{D}}_{\gamma,g}(\tilde{\boldsymbol{\beta}}^{(t)}) = \arg\min_{\boldsymbol{\beta}} \left\{ g(\boldsymbol{\beta}) + \frac{1}{2\gamma_t}\sum_{i=1}^{p} \delta_i^{(t)}(\beta_i - \tilde{\beta}_i)^2 \right\},$$

where the parameter space

$$\mathcal{B} = \{\boldsymbol{\beta} \in \mathbb{R}^p : b_{kk'} = 0, 1 \le k < k' \le K, \sum_{k'=1}^{K} b_{kk'}^2 = 1, k = 1,\ldots,K\},$$

and $g(\boldsymbol{\beta}) = \lambda\sum_{j=1}^{J}\sum_{k=1}^{K}|a_{jk}| + I_{\mathcal{B}}(\boldsymbol{\beta})$ only involves loading parameters a_{jk} and parameters $\mathbf{B}$ for the covariance matrix.

We first look at the update for d_js. As the g function does not involve d_j, its update is simply $d_j^{(t)} = \tilde{d}_j^{(t)}$, where $\tilde{d}_j^{(t)}$ is the corresponding component in $\tilde{\boldsymbol{\beta}}^{(t)}$. We then look at the update for the loading parameters a_{jk}. Suppose that a_{jk} corresponds to the $i_{a_{jk}}$th component of $\boldsymbol{\beta}$. Then the update is given by solving the optimization

$$a_{jk}^{(t)} = \arg\min_{a_{jk}} \ \lambda|a_{jk}| + \frac{1}{2\gamma_t}\delta_{i_{a_{jk}}}^{(t)}(a_{jk} - \tilde{a}_{jk}^{(t)})^2.$$

As discussed in Remark 2, this optimization has a closed-form solution via soft-thresholding. We finally look at the update for $\mathbf{B}$. Suppose that b_{kl} corresponds to the $i_{b_{kl}}$th component of $\boldsymbol{\beta}$. Then the update of $\mathbf{b}_k$, the kth row of $\mathbf{B}$, is given by solving the following optimization problem:

$$\mathbf{b}_k^{(t)} = \arg\min_{\mathbf{b}_k:\|\mathbf{b}_k\|=1, b_{kk'}=0, k'>k} \left\{ \sum_{l=1}^{K} \delta_{i_{b_{kl}}}^{(t)}(b_{kl} - \tilde{b}_{kl}^{(t)})^2 \right\},$$

which can be easily solved by the method of Lagrangian multiplier.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

2.3. Example II: Restricted LCA

We now provide a brief discussion on the computation for the restricted LCA model. First, the stochastic step is straightforward, as the posterior distribution for each $\boldsymbol{\xi}_i$ is still a categorical distribution which can be sampled exactly. Second, the proximal step requires to solve a quadratic programming problem. Again, we denote

$$\tilde{\boldsymbol{\beta}}^{(t)} = \boldsymbol{\beta}^{(t-1)} - \gamma_t (\mathbf{D}^{(t)})^{-1} \mathbf{G}^{(t)}.$$

The proximal step requires to solve the following quadratic programming problem

$$\begin{aligned} &\min_{\boldsymbol{\beta}} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}^{(t)})^\top \mathbf{D}^{(t)} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}^{(t)}), \\ s.t.\ &\max_{\boldsymbol{\alpha} \succeq \mathbf{q}_j} \theta_{j,\boldsymbol{\alpha}} = \min_{\boldsymbol{\alpha} \succeq \mathbf{q}_j} \theta_{j,\boldsymbol{\alpha}} \geq \theta_{j,\boldsymbol{\alpha}'} \geq \theta_{j,\mathbf{0}}, \text{ for all } \boldsymbol{\alpha}' \not\succeq \boldsymbol{q}_j, \\ &\text{and } \nu_{\mathbf{0}} = 0. \end{aligned} \tag{19}$$

Quadratic programming is the most studied nonlinear convex optimization problem (Chapter 4, Boyd et al., 2004) and many efficient solvers exist. In our simulation study in Sect. 4.3, we use the dual method of Goldfarb & Idnani (1983) implemented in the R package quadprog (Turlach et al., 2019).

2.4. Comparison with Related Algorithms

We compare Algorithm 1 with several related algorithms in more details.

2.4.1. Robbins-Monro SA and Variants The proposed method is closely related to the stochastic approximation approach first proposed in Robbins & Monro (1951), and its variants given in Gu & Kong (1998) and Cai (2010a) that are specially designed for latent variable model estimation. Note that the Robbins–Monro method is the first SGD method with convergence guarantee. Both the methods of Gu & Kong (1998) and Cai (2010a) approximate the original Robbins–Monro method by using MCMC sampling to generate an approximate stochastic gradient in each iteration, when an unbiased stochastic gradient is difficult to obtain. All these methods do not handle complex constraints or non-smooth objective functions.

When there is no constraint or penalty on parameters (i.e., $g(\boldsymbol{\beta}) \equiv 0$), the proximal operator degenerates to an identity map. In this case, the proposed method is essentially the same as Gu & Kong (1998) and Cai (2010a), except for the sampling method in the stochastic step, the way the Hessian matrix is approximated, the specific choices of step size, and the averaging in the last step of the proposed method. Among these differences, the step size and the trajectory averaging are key to the advantage of the proposed method.

As pointed out in Remark 4, the Robbins–Monro procedure has the same general requirement on the step size as the proposed method. Specially, the Robbins–Monro procedure, as well as its MCMC variants (Gu & Kong 1998; Cai 2010a), typically let the step size γ_t decay in the order $1/t$ as suggested by asymptotic theory (Fabian, 1968). However, this step is often too short at the early stage of the algorithm, resulting in poor performance in practice (Sect. 4.5.3., Spall, 2003). On the other hand, the proposed method adopts a longer step size. By further adopting Polyak–Ruppert averaging (Ruppert, 1988; Polyak & Juditsky, 1992), we show in Sect. 3 that the proposed method almost achieves the optimal convergence speed.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

2.4.2. Perturbed Proximal Gradient Algorithm Proximal gradient descent algorithm (Parikh & Boyd, 2014) is a non-stochastic algorithm for solving nonsmooth and/or constrained optimization algorithms. For example, the widely used gradient projection algorithm for oblique rotation in factor analysis (Jennrich, 2002) is a special case. The vanilla proximal gradient descent algorithm does not use the second-order information of the objective function and thus sometimes converges slowly. To improve convergence speed, proximal Newton-type methods have been proposed in Lee et al. (2014) that utilize the second-order information of the smooth part of the objective function.

The perturbed proximal gradient algorithm (Atchadé et al., 2017) solves a similar optimization problem as in (2) by combining the methods of stochastic approximation, proximal gradient decent, and Polyak–Ruppert averaging. The proposed method extends Atchadé et al., (2017) by adopting a Newton-type proximal update suggested in (2014). The method of Atchadé et al., (2017) can be viewed as a special case of the proposed one with $c_1 = c_2$. As shown by simulation study in the sequel, thanks to the second-order information, the proposed method converges much faster than that of Atchadé et al., (2017). We also point out that the theoretical analysis of Atchadé et al., (2017) focuses on convex optimization, while in Sect. 3 we consider a more general setting of non-convex optimization that includes a wide range of latent variable model estimation problems as special cases.

2.4.3. Stochastic EM Algorithm The proposed method is also closely related to the stochastic-EM algorithm (Celeux, 1985; Ip, 2002; Nielsen, 2000; S. Zhang et al., 2020b). The stochastic-EM algorithm is a similar iterative algorithm, consisting of a stochastic step and a maximization step in each iteration, where the stochastic step is the same as that in the proposed algorithm. The maximization step plays a similar role as the proximal step in the proposed algorithm. More precisely, when there is no constraint or penalty, the maximization step of the stochastic-EM algorithm obtains parameter update $\boldsymbol{\beta}^{(t)}$ by minimizing the negative complete data log-likelihood function $-\log f(\mathbf{y}, \boldsymbol{\xi}^{(t)} \mid \boldsymbol{\beta})$, instead of a stochastic gradient update. It is also recommended to perform a trajectory averaging in the stochastic-EM algorithm (Nielsen, 2000; S. Zhang et al., 2020b), like the last step of the proposed algorithm. As pointed out in S. Zhang et al. (2020b), the stochastic EM algorithm can potentially handle constraints and non-smooth penalties on parameters by incorporating them into the maximization step.

The stochastic-EM algorithm is typically not as fast as the proposed method, which is revealed by simulation studies below. This is because, it requires to solve an optimization problem completely in each iteration, which is time consuming, especially when constraints and non-smooth penalties are involved. On the other hand, the proximal step of the proposed algorithm can often be efficiently performed.

3. Theoretical Properties

In what follows, we establish the asymptotic properties of the proposed algorithm, under suitable technical conditions. For readers who are not interested in the asymptotic theory, this section can be skipped without affecting the reading of the rest of the paper. Note that in this section, we view data as fixed and the randomness comes from sampling of the latent variables. The following expectation is taken with respect to latent variable $\boldsymbol{\xi}$ given data $\boldsymbol{y}$ and parameters $\boldsymbol{\beta}$, denoted by $\mathbb{E}(\cdot \mid \boldsymbol{\beta}) = \int \cdot \pi_{\boldsymbol{\beta}}(\boldsymbol{\xi}) d\boldsymbol{\xi}$, where $\pi_{\boldsymbol{\beta}}$ is the posterior distribution for $\boldsymbol{\xi}$ given $\boldsymbol{y}$ and $\boldsymbol{\beta}$. Let $\|\cdot\|$ denote the vector l_2-norm. Following the typical convergence analysis of non-convex optimization (e.g., Chapter 3, Floudas, 1995), we will first discuss the convergence of the sequence $\boldsymbol{\beta}^{(t)}$ to a stationary point of the objective function $h(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$ in Theorem 1, which follows the theoretical development in Duchi & Ruan (2018). Then with some additional assumptions on the

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

local geometry of the objective function at the stationary point being converged to, we will show the convergence rate of the Polyak–Ruppert averaged sequence $\bar{\boldsymbol{\beta}}_n$ in Theorem 2 which extends the results of Atchadé et al. (2017) to the setting of non-convex optimization.

For a function $f : \mathbb{R}^d \mapsto \mathbb{R} \cup +\infty$, denote the Fréchet subdifferential (Chapter 8.B Rockafellar & Wets, 1998) of f at the point $\boldsymbol{x}$ by $\partial f(\boldsymbol{x})$,

$$\partial f(\boldsymbol{x}) = \left\{ \boldsymbol{z} \in \mathbb{R}^p : f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{z}^\top (\boldsymbol{y} - \boldsymbol{x}) + o(\|\boldsymbol{y} - \boldsymbol{x}\|) \text{ as } \boldsymbol{y} \to \boldsymbol{x} \right\}.$$

Define the set of stationary points of the objective function as

$$\mathcal{B}^* = \{\boldsymbol{\beta} \in \mathcal{B} : \exists\, \mathbf{x} \in \partial h(\boldsymbol{\beta}) + \partial g(\boldsymbol{\beta}) \text{ with } \mathbf{x}^\top (\mathbf{y} - \boldsymbol{\beta}) \geq 0, \text{ for all } \mathbf{y} \in \mathcal{B}\}.$$

Note that the global minimum $\hat{\boldsymbol{\beta}}$ is a stationary point, i.e., $\hat{\boldsymbol{\beta}} \in \mathcal{B}^*$. In addition, when the objective function is smooth, i.e., $g(\boldsymbol{\beta}) \equiv 0$, then $\mathcal{B}^* = \{\boldsymbol{\beta} \in \mathcal{B} : \nabla h(\boldsymbol{\beta}) = 0\}$, which is the standard definition of stationary points set for a smooth function.

The following assumptions are assumed for our objective function.

H1. $\mathcal{B}$ is compact and contains finite stationary points. For stationary points $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathcal{B}^*$, $h(\boldsymbol{\beta}) + g(\boldsymbol{\beta}) = h(\boldsymbol{\beta}') + g(\boldsymbol{\beta}')$ if and only if $\boldsymbol{\beta} = \boldsymbol{\beta}'$.

H2. $H(\boldsymbol{\xi}, \boldsymbol{\beta})$ is a differentiable function with respect to $\boldsymbol{\beta}$ for given $\boldsymbol{\xi}$ and let $\mathbf{G}_{\boldsymbol{\beta}}(\boldsymbol{\xi}) = \partial H(\boldsymbol{\xi}, \boldsymbol{\beta})/\partial \boldsymbol{\beta}$. Define function $M_\epsilon \colon \Theta \times \Xi \to \mathbb{R}_+$ as

$$M_\epsilon(\boldsymbol{\beta}; \boldsymbol{\xi}) = \sup_{\boldsymbol{\beta}' \in \mathcal{B}, \|\boldsymbol{\beta}' - \boldsymbol{\beta}\| < \epsilon} \|\mathbf{G}_{\boldsymbol{\beta}'}(\boldsymbol{\xi})\|.$$

There exists $\epsilon_0 > 0$ such that for all $0 < \epsilon < \epsilon_0$,

$$\mathbb{E}[M_\epsilon(\boldsymbol{\beta}; \boldsymbol{\xi})^2 \mid \boldsymbol{\beta}] < \infty \text{ for all } \boldsymbol{\beta} \in \mathcal{B}.$$

H3. There exists $\epsilon_0 > 0$ such that for all $\boldsymbol{\beta}' \in \mathcal{B}$, there exists $\lambda(\boldsymbol{\xi}, \boldsymbol{\beta}') \geq 0$ such that

$$\boldsymbol{\beta} \mapsto H(\boldsymbol{\xi}, \boldsymbol{\beta}) + \frac{\lambda(\boldsymbol{\xi}, \boldsymbol{\beta}')}{2} \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|^2$$

is convex on the set $\{\boldsymbol{\beta} : \|\boldsymbol{\beta} - \boldsymbol{\beta}'\| \leq \epsilon_0\}$ for any $\boldsymbol{\beta}_0$, and $\mathbb{E}[\lambda(\boldsymbol{\xi}, \boldsymbol{\beta}') \mid \boldsymbol{\beta}] < \infty$.

H4. The stochastic gradient $\mathbf{G}_{\boldsymbol{\beta}^{(t-1)}}(\boldsymbol{\xi}^{(t)})$ is a Monte Carlo approximation of $\nabla h(\boldsymbol{\beta}^{(t-1)})$. That is, if computationally feasible, we take $\boldsymbol{\xi}^{(t)}$ as an exact sample from $\pi_{\boldsymbol{\beta}^{(t-1)}}$, where, as defined earlier, $\pi_{\boldsymbol{\beta}^{(t-1)}}$ is the posterior distribution of $\boldsymbol{\xi}$ given $\boldsymbol{y}$ and $\boldsymbol{\beta}^{(t-1)}$. If not, we sample $\boldsymbol{\xi}^{(t)}$ from a Markov kernel $P_{\boldsymbol{\beta}^{(t-1)}}$ with invariant distribution $\pi_{\boldsymbol{\beta}^{(t-1)}}$.

H5. Define

$$\begin{aligned} \boldsymbol{\beta}_\gamma^+(\boldsymbol{\xi}) &= \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{B}} \left\{ [\boldsymbol{G}_{\boldsymbol{\beta}}(\boldsymbol{\xi})]^\top (\boldsymbol{x} - \boldsymbol{\beta}) + \mathbf{g}(\boldsymbol{x}) + \frac{1}{2\gamma} \|\boldsymbol{x} - \boldsymbol{\beta}\|_{\mathbf{D}}^2 \right\}, \\ \boldsymbol{U}_\gamma(\boldsymbol{\xi}; \boldsymbol{\beta}) &= \frac{1}{\gamma}(\boldsymbol{\beta} - \boldsymbol{\beta}_\gamma^+(\boldsymbol{\xi})), \end{aligned} \tag{20}$$

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

where step size satisfy $\sum_{t=1}^{\infty} \gamma_t = \infty$, $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$. Then with probability 1,

$$\lim_{n\to\infty} \sum_{t=1}^{n} \gamma_t \left(\boldsymbol{U}_{\gamma_t}(\boldsymbol{\xi}^{(t)}; \boldsymbol{\beta}^{(t-1)}) - \mathbb{E}[\boldsymbol{U}_{\gamma_t}(\boldsymbol{\xi}^{(t)}; \boldsymbol{\beta}^{(t-1)}) \mid \boldsymbol{\beta}^{(t-1)}] \right)$$

exists and is finite.

We remark that conditions H1 through H5 are quite mild. Condition H1 imposes mild requirements on the compactness of the parameter space and the properties of the stationary points of the objective function. Specifically, the compactness of the parameter space is often assumed when analyzing stochastic optimization problems without assuming convexity, see e.g., Gu & Kong (1998), Nielsen (2000), Cai (2010b), and Duchi & Ruan (2018). It also requires that the objective function has different values at different stationary points. Conditions H2 and H3 require the complete-data log-likelihood function $H(\boldsymbol{\xi}, \cdot)$ is locally Lipschitzian and weakly convex, respectively. These conditions hold when the complete-data log-likelihood function $H(\boldsymbol{\xi}, \cdot)$ is Lipschitzian and convex on the entire parameter space. Requiring locally Lipschitzian and weakly convex enables our theory to be applicable to a wider range of problems. Similar conditions are imposed in Duchi & Ruan (2018). For the examples that we consider in Sects. 1.2 and 1.3, these two conditions are satisfied because $H(\boldsymbol{\xi}, \boldsymbol{\beta})$ is smooth and convex in $\boldsymbol{\beta}$. Condition H4 is automatically satisfied according to the way the latent variables are sampled in Algorithm 1. Finally, H5 is a key condition for the convergence of the sequence $\boldsymbol{\beta}^{(t)}$. When exact samples from the posterior distribution are used, Lemma 1 guarantees that H5 is satisfied. With approximate samples from an MCMC algorithm, H5 may still hold when the bias from the MCMC samples is small.

Lemma 1 *Define the filtration of σ-algebra $\mathcal{F}_{t-1} = \sigma\left(\boldsymbol{\beta}^{(0)}, \boldsymbol{\xi}^{(k)}, 0 \le k \le t-1\right)$. $\boldsymbol{\xi}$ is a sample from $\pi_{\boldsymbol{\beta}}$. Let*

$$\boldsymbol{\epsilon}_{\gamma}(\boldsymbol{\xi}; \boldsymbol{\beta}) = \boldsymbol{U}_{\gamma}(\boldsymbol{\xi}; \boldsymbol{\beta}) - \mathbb{E}[\boldsymbol{U}_{\gamma}(\boldsymbol{\xi}; \boldsymbol{\beta}) \mid \boldsymbol{\beta}],$$

then $\gamma_t \boldsymbol{\epsilon}_{\gamma_t}(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta}^{(t-1)})$ is a square-integrable martingale difference sequence adapted to $\mathcal{F}_{t-1}$, and with probability 1, $\lim_n \sum_{t=1}^{n} \gamma_t \boldsymbol{\epsilon}_{\gamma_t}(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta}^{(t-1)})$ exists and is finite.

Theorem 1 *Apply Algorithm 1 to optimization problem (11) with step size $\gamma_t = t^{-\frac{1}{2}-\epsilon}$, $\epsilon \in (0, \frac{1}{2}]$, for which conditions H1-H5 hold. Then with probability 1, the sequence $\boldsymbol{\beta}^{(n)}$ converges to a stationary point in $\mathcal{B}^*$.*

We remark that the convergence of the proposed method is similar to that of the EM algorithm. In fact, for marginal maximum likelihood estimation that is non-convex, the EM algorithm also only guarantees the convergence to a stationary point (Wu, 1983). Moreover, when the objective function has a single stationary point (e.g., when the objective function is strictly convex), then Theorem 1 guarantees global convergence.

The convergence of $\boldsymbol{\beta}^{(n)}$ guarantees the convergence of the Polyak–Ruppert averaging sequence $\bar{\boldsymbol{\beta}}_n$. However, Theorem 1 does not provide information on the convergence speed. In what follows, we establish the convergence speed of $\bar{\boldsymbol{\beta}}_n$. Without loss of generality, by Theorem 1, we assume that $\boldsymbol{\beta}^{(n)}$ converges to $\boldsymbol{\beta}_* \in \mathcal{B}^*$.

- H6. There exists $\delta > 0$, such that $h(\boldsymbol{\beta})$ is strongly convex in $\mathcal{B}_1 = \{\boldsymbol{\beta} \in \mathcal{B} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_*\| \le \delta\}$ and $\nabla h(\boldsymbol{\beta})$ is Lipschitz in $\mathcal{B}_1$ with Lipschitz constant L.
- H7. For $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathcal{B}_1$, any $\gamma > 0$, and diagonal matrix $\mathbf{D}$ with diagonal entries $\delta_i \in [c_1, c_2]$, the following conditions hold.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

(i) $g\left(\text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta})\right) - g\left(\boldsymbol{\beta}'\right) \leq -\frac{1}{\gamma}\left\langle \text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta}) - \boldsymbol{\beta}', \text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta}) - \boldsymbol{\beta}\right\rangle_{\mathbf{D}}$.

(ii) $\left\|\text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta}) - \text{Prox}^{\mathbf{D}}_{\gamma,g}\left(\boldsymbol{\beta}'\right)\right\|^2_{\mathbf{D}} + \left\|\left(\text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta}) - \boldsymbol{\beta}\right) - \left(\text{Prox}^{\mathbf{D}}_{\gamma,g}\left(\boldsymbol{\beta}'\right) - \boldsymbol{\beta}'\right)\right\|^2_{\mathbf{D}} \leq \left\|\boldsymbol{\beta} - \boldsymbol{\beta}'\right\|^2_{\mathbf{D}}$.

(iii) $\sup_{\gamma\in(0,c_1/L]} \sup_{\boldsymbol{\beta}\in\mathcal{B}_1} \gamma^{-1}\left\|\text{Prox}^{\mathbf{D}}_{\gamma,g}(\boldsymbol{\beta}) - \boldsymbol{\beta}\right\| < \infty$.

H8. For a measurable function $V : \Xi \to [1, +\infty)$, a signed measure μ on the σ-field of Ξ, and a function $f : \Xi \to \mathbb{R}$, define

$$|f|_V \stackrel{\text{def}}{=} \sup_{\boldsymbol{\xi}\in\Xi} \frac{|f(\boldsymbol{\xi})|}{V(\boldsymbol{\xi})}, \quad \|\mu\|_V \stackrel{\text{def}}{=} \sup_{f,|f|_V\leq 1} \left|\int f \mathrm{d}\mu\right|.$$

There exist $\lambda \in (0, 1)$, $b < \infty$, $m \geq 4$ and a measurable function $W : \Xi \to [1, +\infty)$ such that

$$\sup_{\boldsymbol{\beta}\in\mathcal{B}_1} \left|\mathbf{G}_{\boldsymbol{\beta}}\right|_W < \infty, \quad \sup_{\boldsymbol{\beta}\in\mathcal{B}_1} P_{\boldsymbol{\beta}} W^m \leq \lambda W^m + b,$$

where $\mathbf{G}_{\boldsymbol{\beta}}(\boldsymbol{\xi}) = \partial H(\boldsymbol{\xi}, \boldsymbol{\beta})/\partial\boldsymbol{\beta}$ and $P_{\boldsymbol{\beta}}$ is the Markov kernel defined in condition H4. In addition, for any $\ell \in (0, m]$, there exists $C < \infty$, $\rho \in (0, 1)$ such that for any $\boldsymbol{\xi} \in \Xi$,

$$\sup_{\boldsymbol{\beta}\in\mathcal{B}_1} \left\| P^n_{\boldsymbol{\beta}}(\boldsymbol{\xi}, \cdot) - \pi_{\boldsymbol{\beta}}\right\|_{W^\ell} \leq C\rho^n W^\ell(\boldsymbol{\xi}).$$

H9. There exists a constant C such that for any $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathcal{B}_1$,

$$\left|\mathbf{G}_{\boldsymbol{\beta}} - \mathbf{G}_{\boldsymbol{\beta}'}\right|_W + \sup_{\boldsymbol{\xi}\in\Xi} \frac{\left\| P_{\boldsymbol{\beta}}(\boldsymbol{\xi}, \cdot) - P_{\boldsymbol{\beta}'}(\boldsymbol{\xi}, \cdot)\right\|_W}{W(\boldsymbol{\xi})} + \left\|\pi_{\boldsymbol{\beta}} - \pi_{\boldsymbol{\beta}'}\right\|_W \leq C\left\|\boldsymbol{\beta} - \boldsymbol{\beta}'\right\|.$$

We provide a few remarks on conditions H6-H9, which are needed for establishing the convergence speed in addition to conditions H1–H5. Condition H6 requires that the smooth part of the objective function is strongly convex and its derivative is Lipschitz continuous in a small neighborhood of $\boldsymbol{\beta}_*$. Specifically, $h(\boldsymbol{\beta})$ being strongly convex in $\mathcal{B}_1$ means that there exists a positive constant C, such that $(\nabla h(\boldsymbol{\beta}) - \nabla h(\boldsymbol{\beta}'))^\top(\boldsymbol{\beta} - \boldsymbol{\beta}') \geq C\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|^2$, for any $\boldsymbol{\beta}$ and $\boldsymbol{\beta}' \in \mathcal{B}_1$. Condition H7 imposes some requirements on the non-smooth part of the objective function, with regard to the proximal operator. As verified in Lemma C.1, H7 holds when g is a generalized function that indicates constraints or when g is locally Lipschitz continuous and convex that holds when g is a L_1 regularization function. Thus, H7 holds for the examples we consider in Sects. 2.2 and 2.3. Conditions H8 and H9 impose mild regularity conditions on the stochastic gradient in a local neighborhood of $\boldsymbol{\beta}_*$, especially when the stochastic gradients are generated by a Markov kernel. These conditions are used to control the bias caused by MCMC sampling. H8 is essentially a uniform-in-$\boldsymbol{\beta}$ ergodic condition and H9 is a local Lipschitzian condition on the Markov kernel. These regularity conditions are commonly adopted in the stochastic approximation literature (Benveniste et al., 1990; Andrieu et al., 2005; Fort et al., 2016) and have been shown to hold for general families of MCMC kernels including Metropolis–Hastings and Gibbs samplers (Andrieu & Moulines, 2006; Fort et al., 2011; Schmidt et al., 2011).

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Theorem 2. *Suppose that H1–H9 hold. Then there exists a constant C, such that for the Polyak–Ruppert averaging sequence* $\bar{\boldsymbol{\beta}}_n = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{\beta}^{(t)}$ *from Algorithm 1,*

$$\mathbb{E}\|\bar{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_\star\|^2 \leq Cn^{-\frac{1}{2}+\varepsilon}. \tag{21}$$

Note that the expectation is taken with respect to $\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(t)}$ *given* $\boldsymbol{\beta}^{(0)}$ *and* $\boldsymbol{\xi}^{(0)}$.

We now provide a few remarks regarding the convergence speed (21). First, the small positive constant ε comes from the requirement on step size that $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ in H5. Since $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ is satisfied when $\gamma_t = \mu t^{-\frac{1}{2}-\varepsilon}$, for any $\mu, \varepsilon > 0$, the convergence speed of $\mathbb{E}\|\bar{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_\star\|^2$ can be arbitrarily close to $O(n^{-\frac{1}{2}})$ by choosing an arbitrarily small ε. Second, this ε might be an artifact due to our proof strategy to overcome the non-convexity of the problem. In fact, if the objective function is convex, similar to Atchadé et al. (2017), we can choose $\epsilon = 0$ and then prove under similar conditions that $\mathbb{E}\|\bar{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_\star\|^2 \leq Cn^{-\frac{1}{2}}$. Lastly, it is well-known that for non-smooth convex optimization, the minimax optimal convergence rate is $O(n^{-\frac{1}{2}})$; see Chapter 3, Nesterov (2004). In this sense, our algorithm is almost minimax optimal, when ε is very close to zero. It is well-known that Polyak–Ruppert averaging typically improves the convergence speed of a slowly convergent sequence (Ruppert, 1988, Bonnabel, 2013).

4. Simulation Study

4.1. Study I: Confirmatory IFA

In the first study, we compare the performance of four variants of the proposed method and the stochastic EM (StEM) algorithm. The five methods, including their abbreviations, are given in Table 1. For a fair comparison, the same Gibbs sampling method is used. We further explain the differences below.

1. USP is the method that we recommend. It has a step size γ_t close to $t^{-1/2}$, applies Polyak–Ruppert averaging, and uses a quasi-Newton update in the proximal step.
2. The USP-PPG method is the perturbed proximal gradient method that is implemented the same as the USP method except that $c_1 = c_2$ so that it does not involve a quasi-Newton update. c_1 is set to be 1 without tuning in this study.
3. The USP-RM1 method is implemented the same as the USP method, except that $\boldsymbol{\beta}^{(n)}$ from the last iteration is taken as the estimator instead of applying Polyak–Ruppert averaging. This method is very similar to a Robbins–Monro algorithm, except for the update of parameters **B** for the covariance matrix where constraints involve.

TABLE 1.
Comparison of five stochastic algorithms.

Estimator	Step size	Averaging	Quasi-Newton	MCMC
USP	$\gamma_t = t^{-0.51}$	Yes	Yes	Gibbs
USP-PPG	$\gamma_t = t^{-0.51}$	Yes	No	Gibbs
USP-RM1	$\gamma_t = t^{-0.51}$	No	Yes	Gibbs
USP-RM2	$\gamma_t = t^{-1}$	No	Yes	Gibbs
StEM	NA	Yes	NA	Gibbs

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

4. The USP-RM2 method is the same as USP-RM1, except that we set the step size $\gamma_t = 1/t$ which is the asymptotic optimal step size for the Robbins–Monro algorithm (Fabian, 1968).
5. The implementation of the StEM algorithm is the same as USP, except for the proximal step. Instead of making stochastic gradient update, StEM obtains $\boldsymbol{\beta}^{(t)}$ by completely solving an optimization problem

$$\boldsymbol{\beta}^{(t)} = \arg\max_{\boldsymbol{\beta}\in\mathcal{B}} H(\boldsymbol{\xi}^{(t)}, \boldsymbol{\beta}) + g(\boldsymbol{\beta}).$$

In our implementation, this optimization problem is solved by making the quasi-Newton proximal update (14) iteratively until convergence.

We consider a confirmatory IFA setting with only two factors (i.e., $K = 2$), so that an EM algorithm with sufficient numbers of quadrature points and EM steps can be used to obtain a more accurate approximation of $\hat{\boldsymbol{\beta}}$ that will be used as the standard when comparing the five methods. We emphasize that it is important to compare the convergence speed of difference algorithms based on $\hat{\boldsymbol{\beta}}$ rather than the true model parameters. This is because, under suitable conditions, these algorithms converge to $\hat{\boldsymbol{\beta}}$ rather than the true model parameters. If we compare the algorithms based on the true model parameters, the difference in the convergence speed cannot be observed clearly, as the statistical error (i.e., the difference between $\hat{\boldsymbol{\beta}}$ and the true model parameters) tends to dominate the computational errors (i.e., the difference between $\hat{\boldsymbol{\beta}}$ and the results given by the stochastic algorithms).

More precisely, we consider sample size $N = 1000$ and the number of items $J = 20$. The design matrix $\mathbf{Q}$ is specified by the assumptions that items 1 through 5 only measure the first factor, items 6 through 10 only measure the second factor, and items 11 through 20 measure both. The intercept parameters d_j are drawn i.i.d. from the standard normal distribution, and the non-zero loading parameters are drawn i.i.d. from a uniform distribution over the interval (0.5, 1.5). The variances of the two factors are set to be 1 and the covariance is set to be 0.4. Under these parameters, 100 independent datasets are generated, based on which the five methods are compared. To ensure a fair comparison, the true parameters are used as the starting point for all the methods. In addition, 1000 iterations are run (i.e., $n = 1000$) for each method, instead of using an adaptive stopping criterion. For USP, USP-PPG, and StEM, the burn-in size ϖ is chosen to be 500. All algorithms are implemented in C++ and run on the same platform[1] using a single core.

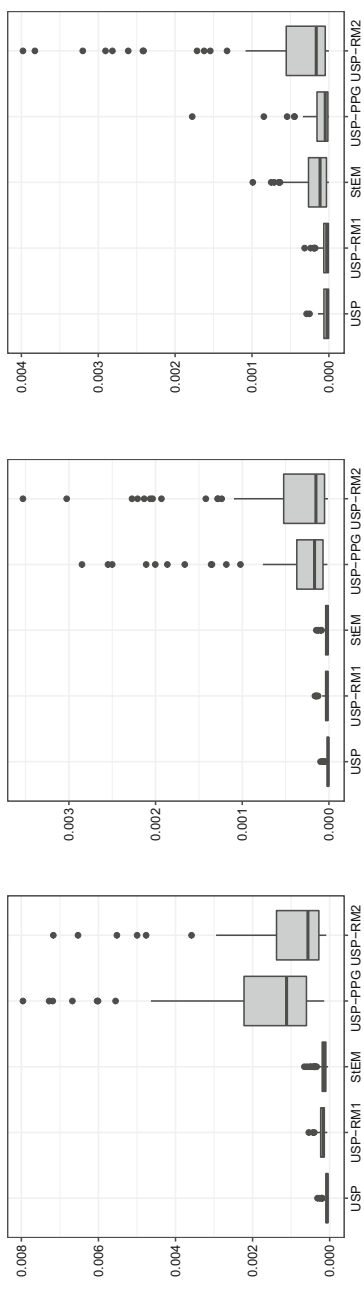
The results regarding the accuracy of the proposed methods are given in Figs. 1 and 2 that are based on the following performance metrics. Specifically, for the intercept parameters d_j, the following mean squared error (MSE) is calculated for each simulated dataset and each method,

$$\frac{1}{J}\sum_{j=1}^{J}\left(\tilde{d}_j - \hat{d}_j\right)^2,$$

where $\hat{d}_j$, which is treated as the global optimum, is obtained by an EM algorithm with 31 Gaussian–Hermite quadrature points per dimension, and $\tilde{d}_j$ is given by one of the five stochastic methods after 1000 iterations. Similarly, the MSEs for the loading parameters and for the correlation σ_{12} between the factors are calculated, where the MSE for the loading parameters is calculated for the unrestricted ones, i.e.,

[1] CPU: 2.6 GHz 6-Core Intel Core i7; RAM: 16 GB 2400 MHz DDR4.

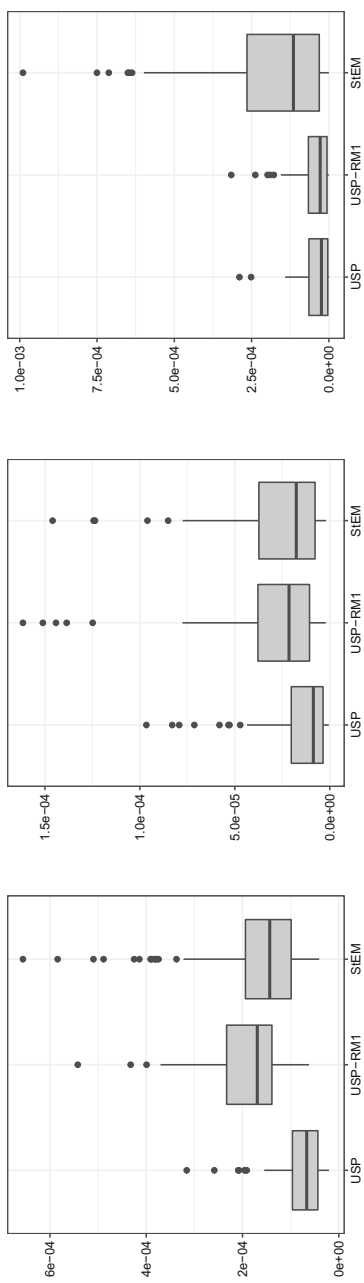
https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press



(a) MSE for unrestricted loading parameters. (b) MSE for intercept parameters. (c) MSE for correlation parameter σ_{12}.

FIGURE 1.
The boxplot of mean squared errors for estimated parameters from the five methods.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press



(a) MSE for unrestricted loading parameters.

(b) MSE for intercept parameters.

(c) MSE for correlation parameter σ_{12}.

FIGURE 2.
The boxplot of mean squared errors for estimated parameters from 'USP,' 'USP-RM1,' and 'StEM' method.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

TABLE 2.
The elapsed time (seconds) for the five methods in confirmatory IFA.

Elapsed time	USP	USP-RM1	StEM	USP-PPG	USP-RM2
25% quantile	12.2	12.2	20.3	12.2	12.1
median	12.3	12.3	20.4	12.3	12.2
75% quantile	12.3	12.4	20.5	12.3	12.3

$$\frac{\sum_{j,k} 1_{\{q_{jk}\neq 0\}}(\tilde{a}_{jk}-\hat{a}_{jk})^2}{\sum_{j,k} 1_{\{q_{jk}\neq 0\}}}.$$

Again, $\hat{a}_{jk}$ is given by the EM algorithm, and $\tilde{a}_{jk}$ is given by one of the five methods.

Figure 1 compares the accuracy of all the five methods. As we can see, the USP, USP-RM1, and StEM methods have much smaller MSEs than the USP-PPG and USP-USP-RM2 methods. Since the USP-PPG method only differs from the USP method by whether using a quasi-Newton update, the inferior performance of USP-PPG implies the importance of the second-order information in the stochastic proximal gradient update. As the USP-RM2 method only differs from USP-RM1 by their step sizes, the inferior performance of USP-RM2 is mainly due to the use of short step size.

In Fig. 2, we zoom in to further compare the USP, USP-RM1, and StEM methods. First, we see that the USP method performs the best among the three, for all the parameters. As the USP-RM1 method is the same as the USP method except for not applying Polyak–Ruppert averaging, this result suggests that averaging does improve accuracy. Moreover, the USP method and the StEM method only differ by the way the parameters are updated, where the USP method takes a quasi-Newton proximal update, while the StEM method completely solves an optimization problem. It is likely that the way parameters are updated in the USP method yields more smoothing (i.e., averaging) than the StEM, which leads to the outperformance of the USP method.

On the computational efficiency, we show in Table 2 the elapsed time for the five methods. 'USP', 'USP-RM1', 'USP-PPG', and 'USP-RM2' share similar computation time since their floating point operations per iteration are at the same level. 'StEM' is most time consuming because an inner loop of optimization is involved in each iteration. In summary, the proposed USP algorithm is computationally the most efficient among the five algorithms, in the sense that it achieves the highest accuracy (see Figs. 1 and 2), within a similar or smaller amount of time (see Table 2).

4.2. Study II: Exploratory IFA by Regularization

In the second study, we apply the proposed method to regularized estimation for exploratory IFA as discussed in Sect. 1.2. We consider increasing sample size $N = 1000, 2000, 4000$, eighty items and five correlated latent factors (i.e., $J = 80$, $K = 5$). The true loading matrix is sparse, where the items each factor loads on are given in Table 3. Similar to Study I, the intercept parameters d_j are drawn i.i.d. from the standard normal distribution, and the non-zero loading parameters a_{jk} are drawn i.i.d. from a uniform distribution over the interval (0.5, 1.5). The elements of covariance matrix $\Sigma = (\sigma_{k,k})_{5\times 5}$ are set to be $\sigma_{k,k'} = 1$, for $k = k'$ and $\sigma_{k,k'} = 0.4$ for $k \neq k'$.

For each sample size, 50 independent datasets are generated. In the proposed algorithm, we adopt a burn-in size $\varpi = 50$ and stop based on the criterion discussed in Sect. 2, where the stopping threshold is set to be 10^{-3}. A decreasing penalty parameter $\lambda_N = \sqrt{\log J/N}$ is used to ensure estimation consistency (Chapter 6, Bühlmann & van de Geer, 2011). Other implementation

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

TABLE 3.
The sparse loading structure in the data generation IFA model.

Factor	Items
1	1-10, 51, 52, 54, 57, 61, 62, 64, 67, 71, 72, 73, 75, 76, 78
2	11-20, 51, 53, 55, 58, 61, 63, 65, 68, 71, 72, 74, 75, 77, 79
3	21-30, 52, 53, 56, 59, 62, 63, 66, 69, 71, 73, 74, 76, 77, 80
4	31-40, 54, 55, 56, 60, 64, 65, 66, 70, 72, 73, 74, 78, 79, 80
5	41-50, 57, 58, 59, 60, 67, 68, 69, 70, 75, 76, 77, 78, 79, 80

TABLE 4.
The mean squared errors for estimated loading parameters in exploratory IFA with L_1 regularization.

MSE of $\tilde{\mathbf{A}}$	$N = 1000$	$N = 2000$	$N = 4000$
25% quantile	0.032	0.026	0.017
median	0.034	0.027	0.018
75% quantile	0.036	0.027	0.019

details can be found in Sect. 2.2. The algorithm in this example is implemented in C++ and is run on the same platform as in Study I. Although a regularized EM algorithm (Sun et al., 2016) can also solve this problem, it suffers from a very high computational cost. Due to the five-dimensional numerical integrals involved, it takes a few hours to fit one dataset. We thus do not consider it here.

We focus on the accuracy in the estimation of the loading matrix $\mathbf{A} = (a_{jk})_{J\times K}$. Note that although the rotational indeterminacy issue is resolved in this regularized estimator, the loading matrix can still only be identified up to column swapping. That is, two estimates of the loading matrix have the same objective function value, if one can be obtained by swapping the columns of the other. The following mean-squared-error measure is used that takes into account column swapping of the loading matrix

$$\min_{\mathbf{A}'\in\mathcal{P}(\tilde{\mathbf{A}})}\left\{\frac{1}{JK}\|\mathbf{A}'-\mathbf{A}\|_F^2\right\}, \tag{22}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathbf{A}$ is the true loading matrix, $\tilde{\mathbf{A}}$ is the output of Algorithm 1, and $\mathcal{P}(\tilde{\mathbf{A}})$ denotes the set of $J \times K$ matrices that can be obtained by swapping the columns of $\tilde{\mathbf{A}}$.

Results are given in Tables 4 and 5. In Table 4, we see that the MSE for the loading matrix is quite small and decreases as the sample size grows, suggesting that consistency of the regularized estimator. In Table 5, the quantiles of time consumption under different sample sizes are given, which suggests the computational efficiency of the proposed method.

4.3. Study III: Restricted LCA

In this study, we apply the proposed method to the estimation of a restricted latent class model as discussed in Sect. 1.3, where the optimization involves complex inequality constraints. Specifically, data are from a Deterministic Input, Noisy 'And' gate (DINA) model (Junker & Sijtsma, 2001) that is a special restricted latent class model. Note that the DINA assumptions are only used in the data generation. We solve optimization (10) which is based on a general restricted

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

TABLE 5.
The elapsed time (seconds) for exploratory IFA with L_1 regularization.

Elapsed time	N=1000	N=2000	N=4000
25% quantile	9.2	14.8	25.8
median	9.7	15.1	26.6
75% quantile	10.4	15.7	27.6

TABLE 6.
The design matrix $\mathbf{Q}$ for the restricted LCA model.

Attribute	Items																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
α_1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	0	1	1	1	1	0
α_2	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1	1	0	1
α_3	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	1	0	1	1
α_4	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1

latent class model considered in Xu (2017) instead of the DINA model, mimicking the practical situation when the parametric form is unknown.

We consider a test consisting of twenty items (i.e., $J = 20$) that measure four binary attributes (i.e., $K = 4$). Three sample sizes are considered, including $N = 1000$, 2000, and 4000. The design matrix $\mathbf{Q}$ is given in Table 6. In addition, the guessing and slipping parameters s_j and g_j of the DINA model are drawn i.i.d. from a uniform distribution over the interval (0.05, 0.2), which gives the values of $\theta_{j,\boldsymbol{\alpha}}$. That is,

$$\theta_{j,\boldsymbol{\alpha}} = \begin{cases} \log((1-s_j)/s_j), & \text{if } \boldsymbol{\alpha} \succeq \mathbf{q}_j, \\ \log(g_j/(1-g_j)), & \text{otherwise.} \end{cases}$$

Finally, we let $\nu_{\boldsymbol{\alpha}} = 0$, for all $\boldsymbol{\alpha} \in \{0, 1\}^K$, so that $\mathbb{P}(\boldsymbol{\xi} = \boldsymbol{\alpha}) = 1/2^K$. According to the results in Xu (2017), the model parameters are identifiable, given the $\mathbf{Q}$-matrix in Table 6.

For each sample size, 50 independent datasets are generated. The proposed algorithm adopts a burn-in size $\varpi = 50$ and stops based on the criterion discussed in Sect. 2, where the stopping threshold is set to be 10^{-3}. Other implementation details can be found in Sect. 2.3. The following metrics are used to evaluate the estimation accuracy. For item parameters $\theta_{j,\boldsymbol{\alpha}}$, the MSE is calculated as

$$\frac{1}{J \times 2^K} \sum_{j=1}^{J} \sum_{\boldsymbol{\alpha} \in \{0,1\}^K} \left(\tilde{\theta}_{j,\boldsymbol{\alpha}} - \theta_{j,\boldsymbol{\alpha}}\right)^2.$$

For structural parameters $\nu_{\boldsymbol{\alpha}}$, the MSE is calculated as

$$\frac{1}{2^K - 1} \sum_{\boldsymbol{\alpha} \in \{0,1\}^K,\ \boldsymbol{\alpha} \neq \mathbf{0}} (\tilde{\nu}_{\boldsymbol{\alpha}} - \nu_{\boldsymbol{\alpha}})^2.$$

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

TABLE 7.
The MSE for item parameters $\theta_{j,\boldsymbol{\alpha}}$ in the restricted latent class model.

MSE of $\tilde{\boldsymbol{\theta}}$	$N = 1000$	$N = 2000$	$N = 4000$
25% quantile	0.150	0.062	0.028
median	0.182	0.070	0.031
75% quantile	0.252	0.077	0.033

TABLE 8.
The MSE for structural parameters $\nu_{\boldsymbol{\alpha}}$ in the restricted latent class model.

MSE of $\tilde{\boldsymbol{\nu}}_{\boldsymbol{\alpha}}$	$N = 1000$	$N = 2000$	$N = 4000$
25% quantile	0.028	0.012	0.005
median	0.045	0.018	0.007
75% quantile	0.085	0.028	0.009

Our results are given in Tables 7 and 8. As we can see, the estimation becomes more accurate as the sample size increases for both sets of parameters. It confirms that the current model is identifiable as suggested by Xu (2017) and thus can be consistently estimated.

5. Concluding Remarks

In this paper, a unified stochastic proximal optimization framework is proposed for the computation of latent variable model estimation. This framework is very general that applies to a wide range of estimators for almost all commonly used latent variable models. Comparing with existing stochastic optimization methods, the proposed method not only solves a wider range of problems including regularized and constrained estimators, but also is computationally more efficient. Theoretical properties of the proposed method are established. These results suggest that the convergence speed of the proposed method is almost optimal in the minimax sense.

The power of the proposed method is shown via three examples, including confirmatory IFA, exploratory IFA by regularized estimation, and restricted latent class analysis. Specifically, the proposed method is compared with several stochastic optimization algorithms, including a stochastic-EM algorithm and a Robbin–Monro algorithm with MCMC sampling, in the simulation study of confirmatory IFA, where there is no complex constraint or penalty. Using the same starting point and the same number of iterations, the proposed one is always more accurate than its competitors. The simulation studies on exploratory IFA and restricted latent class analysis further show the power of the proposed method for handling optimization problems with non-smooth penalties and complex inequality constraints.

The implementation of the proposed algorithm involves several tuning parameters. First, we need to choose a step size γ_t. Our theoretical results suggest that $\gamma_t = t^{-0.5-\epsilon}$ for any $\epsilon \in (0, 0.5]$, and a smaller ϵ leads to faster convergence. In practice, we suggest to set $\gamma_t = t^{-0.51}$ that performs well in all our simulations. This choice of step size is very different from the choice of $\gamma_t = t^{-1}$ in the MCMC stochastic approximation algorithms. Second, a burn-in size ϖ is needed. The burn-in in the proposed algorithm is similar to the burn-in in MCMC algorithms. It does not affect the asymptotic convergence of the algorithm but improves the finite sample performance. In practice, the burn-in size can be decided similarly as in MCMC algorithms by monitoring the parameter updates using trace plots. Third, two positive constraints c_1 and c_2 are needed to regularize the

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

second-order matrix in the scaled proximal update. Depending on the scale of each particular problem, we suggest to choose c_1 to be sufficiently small and c_2 to be sufficiently large. It is found that the performance of our algorithm is not sensitive to their choices. Finally, a stopping criterion is needed. We suggest to stop the iterative update by monitoring a window of successive differences in parameter updates.

The proposed framework may be improved from several aspects that are left for future investigation. First, the sampling strategy in the stochastic step needs further investigation. Although in theory any reasonable MCMC sampler can yield the convergence of the algorithm, a good sampler will lead to superior finite sample performance. More sophisticated MCMC algorithms need to be investigated regarding their performance under the proposed framework. Second, methods for parallel and distributed computing need to be developed. As we can see, many steps of Algorithm 1 can be performed independently. This enables us to design parallel and/or distributed computing systems for solving large-scale and/or distributed versions of latent variable model estimation problems (e.g., fitting models for assessment data from online learning platforms and large-scale mental health records). Finally, the performance of the proposed method under other latent variable models needs to be investigated. For example, the proposed method can also be applied to latent stochastic process models (e.g., Chow et al., 2016; Chen & Zhang, 2020) that are useful for analyzing intensive longitudinal data. These models bring additional challenges, as stochastic processes need to be sampled in the stochastic step of our algorithm.

In summary, the proposed method is computationally efficient, theoretically solid, and applicable to a broad range of latent variable model inference problems. Like the EM algorithm as the standard tool for low-dimensional latent variable models, we believe that the proposed method may potentially serve as the standard approach to the estimation of high-dimensional latent variable models.

Acknowledgments

We thank the editor, associate editor, and three anonymous referees for their careful review and valuable comments. Zhang's research is partially supported by Shanghai Science and Technology Commission Science and Technology Project (22YF1411100).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

Andersen, E. B. (1973). *Conditional inference and models for measuring*. Copenhagen, Denmark: Mentalhygiejnisk Forlag.

Andrieu, C., & Moulines, É. (2006). On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability, 16*, 1462–1505.

Andrieu, C., Moulines, É., & Priouret, P. (2005). Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization, 44*, 283–312.

Atchadé, Y. F., Fort, G., & Moulines, E. (2017). On perturbed proximal gradient algorithms. *The Journal of Machine Learning Research, 18*, 310–342.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Bartholomew, D. J., Knott, M., & Moustaki, I. (2011). *Latent variable models and factor analysis: A unified approach*. Hoboken, NJ: Wiley.
Becker, S., & Le Cun, Y. (1988). Improving the convergence of back-propagation learning with second order methods. In T. S. D. Touretzky, G. Hinton (Ed.), *Proceedings of the 1988 connectionist models summer school* (pp. 29–37). San Mateo: Morgan Kaufmann.
Béguin, A. A., & Glas, C. A. (2001). MCMC estimation and some model-fit analysis of multidimensional IRT models. *Psychometrika, 66*, 541–561.
Benveniste, A., Priouret, P., & Métivier, M. (1990). *Adaptive algorithms and stochastic approximations*. New York, NY: Springer.
Bertsekas, D. P., Gallager, R. G., & Humblet, P. (1992). *Data networks*. Upper Saddle River, NJ: Prentice-Hall International.
Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. Lord & M. Novick (Eds.), *Statistical Theories of Mental Test Scores* (pp. 397–479). Boston, MA: Addison-Wesley.
Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika, 46*, 443–459.
Bollen, K. A. (1989). *Structural equations with latent variables*. Hoboken, NJ: Wiley.
Bolt, D. M., & Lall, V. F. (2003). Estimation of compensatory and noncompensatory multidimensional item response models using Markov chain Monte Carlo. *Applied Psychological Measurement, 27*, 395–414.
Bonnabel, S. (2013). Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control, 58*, 2217–2229.
Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge, UK: Cambridge University Press.
Bühlmann, P., & van de Geer, S. (2011). *Statistics for High-Dimensional Data*. Berlin, Heidelberg: Springer, Berlin Heidelberg.
Cai, L. (2010). High-dimensional exploratory item factor analysis by a Metropolis-Hastings Robbins-Monro algorithm. *Psychometrika, 75*, 33–57.
Cai, L. (2010). Metropolis-Hastings Robbins-Monro algorithm for confirmatory item factor analysis. *Journal of Educational and Behavioral Statistics, 35*, 307–335.
Carroll, R. J., Ruppert, D., Stefanski, L. A., & Crainiceanu, C. M. (2006). *Measurement error in nonlinear models: A modern perspective*. Boca Raton, FL: CRC Press.
Celeux, G. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly, 2*, 73–82.
Chen, Y., Li, X., Liu, J., & Ying, Z. (2018). Robust measurement via a fused latent and graphical item response theory model. *Psychometrika, 83*, 538–562.
Chen, Y., Li, X., & Zhang, S. (2019). Joint maximum likelihood estimation for high-dimensional exploratory item factor analysis. *Psychometrika, 84*, 124–146.
Chen, Y., Li, X., & Zhang, S. (2020). Structured latent factor analysis for large-scale data: Identifiability, estimability, and their implications. *Journal of the American Statistical Association, 115*, 1756–1770.
Chen, Y., Liu, J., Xu, G., & Ying, Z. (2015). Statistical analysis of Q-matrix based diagnostic classification models. *Journal of the American Statistical Association, 110*, 850–866.
Chen, Y., & Zhang, S. (2020a). Estimation methods for item factor analysis: An overview. In Y. Zhao & D. G. Chen (Eds.), *Modern statistical methods for health research.* New York, NY: Springer.
Chen, Y., & Zhang, S. (2020). A latent Gaussian process model for analysing intensive longitudinal data. *British Journal of Mathematical and Statistical Psychology, 73*, 237–260.
Chow, S. M., Lu, Z., Sherwood, A., & Zhu, H. (2016). Fitting nonlinear ordinary differential equation models with random effects and unknown initial conditions using the stochastic approximation expectation-maximization (saem) algorithm. *Psychometrika, 81*, 102–134.
Clogg, C. C. (1995). Latent class models. In G. Arminger, C. C. Clogg, & M. E. Sobel (Eds.), *Handbook of Statistical Modeling for the Social and Behavioral Sciences* (pp. 311–359). Boston, MA: Springer, US.
de la Torre, J. (2011). The generalized DINA model framework. *Psychometrika, 76*, 179–199.
Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 39*, 1–38.
Duchi, J. C., & Ruan, F. (2018). Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization, 28*, 3229–3259.
Dunson, D. B. (2000). Bayesian latent variable models for clustered mixed outcomes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 62*, 355–366.
Dunson, D. B. (2003). Dynamic latent trait models for multidimensional longitudinal data. *Journal of the American Statistical Association, 98*, 555–563.
Edwards, M. C. (2010). A Markov chain Monte Carlo approach to confirmatory item factor analysis. *Psychometrika, 75*, 474–497.
Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.
Fabian, V. (1968). On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics, 39*, 1327–1332.
Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: Fundamentals and applications*. Oxford, UK: Oxford University Press.
Fort, G., Moulines, E., & Priouret, P. (2011). Convergence of adaptive and interacting Markov chain Monte Carlo algorithms. *Annals of Statistics, 39*, 3262–3289.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Fort, G., Moulines, É., Schreck, A., & Vihola, M. (2016). Convergence of Markovian stochastic approximation with discontinuous dynamics. *SIAM Journal on Control and Optimization, 54*, 866–893.
Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*. New York, NY: Springer.
Ghosh, M. (1995). Inconsistent maximum likelihood estimators for the Rasch model. *Statistics and Probability Letters, 23*, 165–170.
Gilks, W. R., & Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. Journal of the Royal Statistical Society. *Series C (Applied Statistics), 41*, 337–348.
Goldfarb, D., & Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming, 27*, 1–33.
Gu, M. G., & Kong, F. H. (1998). A stochastic approximation algorithm with Markov chain Monte-Carlo method for incomplete data estimation problems. *Proceedings of the National Academy of Sciences of the United States of America, 95*, 7270–7274.
Haberman, S. J. (1977). Maximum likelihood estimates in exponential response models. *The Annals of Statistics, 5*, 815–841.
Haberman, S. J. (2004). Joint and conditional maximum likelihood estimation for the Rasch model for binary responses. *ETS Research Report Series RR-04-20*.
Hsiao, C. (2014). *Analysis of panel data*. Cambridge, UK: Cambridge University Press.
Ip, E. H. (2002). On single versus multiple imputation for a class of stochastic algorithms estimating maximum likelihood. *Computational Statistics, 17*, 517–524.
Jacobucci, R., Grimm, K. J., & McArdle, J. J. (2016). Regularized structural equation modeling. *Structural Equation Modeling: A Multidisciplinary Journal, 23*, 555–566.
Jennrich, R. I. (2002). A simple general method for oblique rotation. *Psychometrika, 67*, 7–19.
Jöreskog, K. G., & Moustaki, I. (2001). Factor analysis of ordinal variables: A comparison of three approaches. *Multivariate Behavioral Research, 36*, 347–387.
Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement, 25*, 258–272.
Lee, J. D., Sun, Y., & Saunders, M. A. (2014). Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization, 24*, 1420–1443.
Lindstrøm, J. C., & Dahl, F. A. (2020). Model selection with Lasso in multi-group structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal, 27*, 33–42.
Little, R. J., & Rubin, D. B. (1987). *Statistical analysis with missing data*. Hoboken, NJ: Wiley.
Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 44*, 226–233.
Magis, D., Tuerlinckx, F., & De Boeck, P. (2015). Detection of differential item functioning using the Lasso approach. *Journal of Educational and Behavioral Statistics, 40*, 111–135.
Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*. Boston, MA: Kluwer Academic Publishers.
Neyman, J., & Scott, E. L. (1948). Consistent estimates based on partially consistent observations. *Econometrica, 16*, 1–32.
Nielsen, S. F. (2000). The stochastic EM algorithm: Estimation and asymptotic results. *Bernoulli, 6*, 457–489.
Parikh, N., & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends® in Optimization, 1*, 127–239.
Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization, 30*, 838–855.
Rabe-Hesketh, S., & Skrondal, A. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models*. New York, NY: Chapman and Hall/CRC.
Reckase, M. (2009). *Multidimensional item response theory*. New York, NY: Springer.
Robbins, H. (1956). An empirical Bayes approach to statistics. In J. Neyman (Ed.), *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability* (pp. 157–163). Berkeley, CA: University of California Press.
Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics, 22*, 400–407.
Rockafellar, R. T., & Wets, R. J. (1998). *Variational analysis*. New York, NY: Springer.
Rupp, A. A., Templin, J., & Henson, R. A. (2010). *Diagnostic measurement: Theory, methods, and applications*. New York, NY: Guilford Press.
Ruppert, D. (1988). *Efficient estimators from a slowly convergent Robbins-Monro procedure* (p. 781). School of Oper: Res and Ind Eng, Cornell Univ, Ithaca, NY, Tech Rep No.
Schmidt, M., Roux, N. L., & Bach, F. R. (2011). Convergence rates of inexact proximal-gradient methods for convex optimization. *Advances in Neural Information Processing Systems 24* (pp. 1458–1466). Red Hook, NY: Curran Associates Inc.
Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization*. Hoboken, NJ: Wiley.
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.
Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 58*, 267–288.
Turlach, B. , Weingessel, A., & Moler, C. (2019). quadprog: Functions to solve quadratic programming problems [Computer software manual]. (R package version 1.5-8).
Tutz, G., & Schauberger, G. (2015). A penalty approach to differential item functioning in Rasch models. *Psychometrika, 80*, 21–43.

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press

Van De Schoot, R., Hoijtink, H., & Deković, M. (2010). Testing inequality constrained hypotheses in SEM models. *Structural Equation Modeling, 17*, 443–463.
Vasdekis, V. G., Cagnone, S., & Moustaki, I. (2012). A composite likelihood inference in latent variable models for ordinal longitudinal responses. *Psychometrika, 77*, 425–441.
von Davier, M., & Lee, Y. S. (2019). *Handbook of Diagnostic Classification Models*. New York, NY: Springer.
Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics, 11*, 95–103.
Xu, G. (2017). Identifiability of restricted latent class models with binary responses. *The Annals of Statistics, 45*, 675–707.
Zhang, C. H. (2003). Compound decision theory and empirical Bayes methods. *The Annals of Statistics, 31*, 379–390.
Zhang, H., Chen, Y., & Li, X. (2020). A note on exploratory item factor analysis by singular value decomposition. *Psychometrika, 85*, 358–372.
Zhang, S., Chen, Y., & Liu, Y. (2020). An improved stochastic EM algorithm for large-scale full-information item factor analysis. *British Journal of Mathematical and Statistical Psychology, 73*, 44–71.
Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*, 301–320.

Manuscript Received: 17 AUG 2020
Final Version Received: 13 APR 2021
Published Online Date: 7 MAY 2022

https://doi.org/10.1007/s11336-022-09863-9 Published online by Cambridge University Press