


A RIEMANNIAN OPTIMIZATION ALGORITHM FOR JOINT MAXIMUM LIKELIHOOD ESTIMATION OF HIGH-DIMENSIONAL EXPLORATORY ITEM FACTOR ANALYSIS

YANG LIU 

UNIVERSITY OF MARYLAND

There has been regained interest in joint maximum likelihood (JML) estimation of item factor analysis (IFA) recently, primarily due to its efficiency in handling high-dimensional data and numerous latent factors. It has been established under mild assumptions that the JML estimator is consistent as both the numbers of respondents and items tend to infinity. The current work presents an efficient Riemannian optimization algorithm for JML estimation of exploratory IFA with dichotomous response data, which takes advantage of the differential geometry of the fixed-rank matrix manifold. The proposed algorithm takes substantially less time to converge than a benchmark method that alternates between gradient ascent steps for person and item parameters. The performance of the proposed algorithm in the recovery of latent dimensionality, response probabilities, item parameters, and factor scores is evaluated via simulations.

Key words: item response theory, item factor analysis, high-dimensional data, matrix completion, maximum likelihood, Riemannian optimization, matrix manifold, constrained optimization, penalty method.

1. Introduction

Item factor analysis (IFA; Wirth and Edwards 2007; Bartholomew et al. 2008), also known as (multidimensional) item response theory (Liu et al. 2018; Reckase 2009; Thissen and Steinberg 2009),¹ refers to a family of latent variable measurement models that have been primarily used for examining the underlying factor structure of multivariate categorical response data. IFA assumes the conditional independence among (a large number of) manifest variables given the values of (a much smaller number of) latent factors (McDonald 1981), which effectively reduces the high-dimensional item response data to the low-dimensional space spanned by the few constructs that the test is intended to measure. The present work focuses on exploratory IFA, which aims to identify the optimal number of latent factors as well as the pattern of item-factor dependency.

In an IFA model, the probability distribution of item responses is governed by two sets of unknown quantities: item parameters that characterize the association between the observed responses and latent factors, and factor scores, also known as person parameters, that position individuals in the space of latent factors. Conventionally, factor scores are treated as random effects following a certain family of distributions (e.g., multivariate normal distributions). Maximum likelihood estimation of item parameters and (co)variances of random effects is typically referred to as marginal maximum likelihood (MML; Bock and Lieberman 1970), because all random effects are integrated out when evaluating the likelihood function. MML estimation suffers from two major difficulties. First, the marginal likelihood is often intractable. Substantial effort has been made to develop efficient numerical approximations to the likelihood function and stochastic

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s11336-020-09711-8>) contains supplementary material, which is available to authorized users.

Correspondence should be made to Yang Liu, Department of Human Development and Quantitative Methodology, University of Maryland, College Park, USA. Email: yliu87@umd.edu

¹Some authors prefer to use IFA and item response theory for two different parameterizations of the same (or approximately the same) model; see Takane and de Leeuw (1987) and Wirth and Edwards (2007) for more details.

optimization algorithms for IFA models (e.g., Schilling and Bock 2005; Cai 2010; Haberman 2006; Jeon et al. 2019; Zhang et al. 2020); however, those advanced numerical techniques can still be computationally demanding if the latent dimensionality is high. Second, misspecification of the latent factor distribution may lead to inconsistent estimation of item parameters. Resorting to a flexible family of latent density functions (e.g., Monroe and Cai 2014; Woods and Lin 2009; Woods and Thissen 2006) does not fully address the issue, as the assumptions of independence and continuity of factor scores can still be violated (see Sect. 5.3 for an example).

Alternatively, joint maximum likelihood (JML) concerns simultaneous estimation of item parameters and factor scores (Baker and Kim 2004, Chapter 4; Lord 1980). Despite its chronological precedence and simple implementation compared to MML, JML is known to be inconsistent for item parameters under the conventional asymptotic setting, i.e., fixing the number of items and sending the number of respondents to infinity (Andersen 1970; Neyman and Scott 1948) and thus is often less favored than MML when applied to tests of limited lengths. However, the compelling need for integrative data analysis (e.g., Curran and Hussong 2009; Hofer and Piccinin 2009) and the recent surge of online data collection (e.g., Revelle et al. 2010) facilitate the acquisition of aggregated measures with hundreds or even thousands of items measuring tens of latent constructs and thus bring methodologists' attention back to JML. Recently, Chen et al. (2018) established under mild assumptions that a constrained JML estimator is jointly consistent for item parameters and factor scores as both dimensions of the item response matrix tend to infinity.

Chen et al.'s (2018) consistency proof follows from a non-asymptotic concentration result for one-bit matrix completion (Davenport et al. 2014; Klopp et al. 2015; Fan et al. 2019). As was pointed out by de Leeuw (2006; see also Sect. 2.1),² the cross-product of item parameter and factor score matrices in IFA is basically the low-rank parameter matrix to be recovered by one-bit matrix completion. A general matrix completion problem is typically solved by either nuclear-norm regularization³ (e.g., Davenport et al. 2014) or a direct search in the manifold of low-rank matrices (e.g., Cai and Zhou 2013; de Leeuw 2006). Adapting the latter strategy to IFA, Chen et al. (2018) proposed an alternating maximization (AM) algorithm with proximal gradient updates, which outperforms the state-of-the-art Expectation–Maximization (Bock and Aitkin 1981) algorithm and Robbins–Monro (Cai 2010) algorithm for MML estimation when both dimensions of the data matrix are large.

In the current work, a highly efficient Riemannian optimization algorithm that combines a linear-quadratic penalty method with a Riemannian conjugate gradient (CG) sub-solver is developed for JML estimation of exploratory IFA. Exploiting the differential geometry of the fixed-rank matrix manifold (Absil et al. 2008; Shalit et al. 2012; Vandereycken 2013), the Riemannian algorithm is able to find the JML solution in a small fraction of runtime needed by the Chen et al.'s (2018) AM algorithm, which makes the method more suitable for excessively large-scale problems.

The rest of the paper is organized as follows. Section 2 reviews the logistic exploratory IFA model, the JML estimation problem, and the consistency properties. Next, essential geometric elements pertaining to optimization on a fixed-rank matrix manifold (Sect. 3) and the proposed Riemannian optimization algorithm (Sect. 4) are introduced. Three simulation studies are then reported in Sect. 5, in which the performance of the proposed algorithm in computation speed and parameter recovery is evaluated. The paper is concluded with a discussion of major findings and future directions (Sect. 6).

²de Leeuw (2006) used the term binary principal component analysis instead of one-bit matrix completion.

³The nuclear norm of a matrix is the sum of its singular values.

2. Joint Maximum Likelihood Estimation

2.1. Exploratory Item Factor Analysis

Let d_1 be the number of respondents, d_2 be the number of items, and $i = 1, \dots, d_1$ and $j = 1, \dots, d_2$ be the respective indices for respondents and items. Also, let k be the number of latent factors such that $1 \leq k \ll \min\{d_1, d_2\}$. Denote by Y_{ij} the binary response to item j produced by respondent i . A logistic exploratory IFA model specifies the following conditional probability mass function (pmf) for $Y_{ij} = y_{ij} \in \{0, 1\}$, also known as the item response function:

$$p(y_{ij}|\theta_{ij}) = P\{Y_{ij} = y_{ij}|\theta_{ij}\} = \Psi(\theta_{ij})^{y_{ij}} [1 - \Psi(\theta_{ij})]^{1-y_{ij}}, \quad (1)$$

in which $\Psi(x) = [1 + \exp(-x)]^{-1}$, $x \in \mathcal{R}$, denotes the inverse logit function. In Eq. 1, the quantity θ_{ij} being conditioned on is a linear regression on the factor scores \mathbf{u}_i :

$$\theta_{ij} = w_j + \mathbf{u}_i^\top \mathbf{v}_j = w_j + u_{i1}v_{j1} + \dots + u_{ik}v_{jk}, \quad (2)$$

in which $\mathbf{u}_i = (u_{i1}, \dots, u_{ik})^\top$ denotes respondent i 's factor scores, and w_j and $\mathbf{v}_j = (v_{j1}, \dots, v_{jk})^\top$ are the intercept and slopes for item j . In the JML setting, both the factor scores \mathbf{u}_i and the item parameters w_j and \mathbf{v}_j are treated as fixed effects.

Pooling across all d_1 respondents and d_2 items, write $\mathbf{w} = (w_1, \dots, w_{d_2})^\top$, $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_{d_1})^\top$, $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{d_2})^\top$, and

$$\Theta = \mathbf{1}_{d_1} \mathbf{w}^\top + \mathbf{U} \mathbf{V}^\top = \bar{\mathbf{U}} \bar{\mathbf{V}}^\top, \quad (3)$$

in which $\mathbf{1}_{d_1}$ denotes a $d_1 \times 1$ vector of 1's, $\bar{\mathbf{U}} = (\mathbf{1}_{d_1}, \mathbf{U})$, and $\bar{\mathbf{V}} = (\mathbf{w}, \mathbf{V})$. IFA can be construed as a restricted version of one-bit matrix completion in light of Eq. 3: The parameter matrix Θ has a rank less than or equal to $k + 1$; meanwhile, it must contain $\mathbf{1}_{d_1}$ in the column span. Because one can always reduce k if either $\bar{\mathbf{U}}$ or $\bar{\mathbf{V}}$ is rank deficient, it is without loss of generality assumed that $\text{rank}(\Theta) = k + 1$. The parameter space of Θ is then given by the following matrix manifold:

$$\mathcal{M}_k(d_1, d_2) = \{\Theta \in \mathcal{R}^{d_1 \times d_2} : \Theta = \bar{\mathbf{U}} \bar{\mathbf{V}}^\top = (\mathbf{1}_{d_1}, \mathbf{U})(\mathbf{w}, \mathbf{V})^\top, \bar{\mathbf{U}} \in \mathcal{R}_*^{d_1 \times (k+1)}, \bar{\mathbf{V}} \in \mathcal{R}_*^{d_2 \times (k+1)}\}, \quad (4)$$

in which $\mathcal{R}^{n \times m}$, with integers n and m , denotes the space of $n \times m$ real matrices, and $\mathcal{R}_*^{n \times m}$ indicates the further restricted space of full-rank matrices. The decomposition $\Theta = \bar{\mathbf{U}} \bar{\mathbf{V}}^\top$ is not unique in general: $\mathbf{U} \mathbf{V}^\top = \check{\mathbf{U}} \check{\mathbf{V}}^\top$, where $\check{\mathbf{U}} = \mathbf{U} \mathbf{Q}^{-\top}$ and $\check{\mathbf{V}} = \mathbf{V} \mathbf{Q}$ for any $\mathbf{Q} \in \mathcal{R}_*^{k \times k}$.

An alternative decomposition of Θ using orthonormal basis matrices is handy for subsequent exposition. Denote by $\bar{\mathbf{U}} = \bar{\mathbf{U}}_* \bar{\mathbf{R}}_u$ the QR decomposition of $\bar{\mathbf{U}}$ via the Gram-Schmidt process (e.g., Golub and Van Loan 2013, Chapter 5). The orthonormal basis matrix $\bar{\mathbf{U}}_* = (\mathbf{1}_{d_1}/\sqrt{d_1}, \mathbf{U}_*)$, in which $\mathbf{U}_*^\top \mathbf{1}_{d_1} = \mathbf{0}_k$, $\mathbf{U}_*^\top \mathbf{U}_* = \mathbf{I}_k$, $\mathbf{0}_k$ is a $k \times 1$ vectors of 0's, and \mathbf{I}_k is a $k \times k$ identity matrix. The $(k+1) \times (k+1)$ upper-triangular matrix $\bar{\mathbf{R}}_u$ is subject to the partition

$$\bar{\mathbf{R}}_u = \begin{pmatrix} \sqrt{d_1} \mathbf{z}^\top \\ \mathbf{0}_k & \mathbf{R}_u \end{pmatrix},$$

in which \mathbf{z} is a $k \times 1$ vector, and \mathbf{R}_u is a $k \times k$ upper-triangular matrix. Similarly, denote by $\mathbf{V} = \mathbf{V}_* \mathbf{R}_v$ the QR decomposition of \mathbf{V} , in which $\mathbf{V}_*^\top \mathbf{V}_* = \mathbf{I}_k$ and \mathbf{R}_v is a $k \times k$ upper-triangular

matrix. With these additional notations, Θ can be rewritten as

$$\Theta = \bar{\mathbf{U}}\bar{\mathbf{V}}^\top = \left(\frac{\mathbf{1}_{d_1}}{\sqrt{d_1}}, \mathbf{U}_* \right) \begin{pmatrix} \sqrt{d_1} \mathbf{z}^\top \\ \mathbf{0}_k \quad \mathbf{R}_u \end{pmatrix} \begin{pmatrix} \mathbf{w}^\top \\ \mathbf{R}_v^\top \mathbf{v}_*^\top \end{pmatrix} = \frac{\mathbf{1}_{d_1} \mathbf{w}_*^\top}{\sqrt{d_1}} + \mathbf{U}_* \mathbf{R}^\top \mathbf{V}_*^\top. \quad (5)$$

in which $\mathbf{w}_* = \sqrt{d_1} \mathbf{w} + \mathbf{V}_* \mathbf{R}_v \mathbf{z}$ and $\mathbf{R} = \mathbf{R}_v \mathbf{R}_u^\top$.

2.2. Log-Likelihood and Constraints

Let $\mathbb{Y} = (Y_{ij} : i = 1, \dots, d_1, j = 1, \dots, d_2)$ be a $d_1 \times d_2$ random matrix of item responses and $\mathbf{Y} = (y_{ij} : i = 1, \dots, d_1, j = 1, \dots, d_2)$ be its realization. It is common that some of the response entries are missing: The standard Bernoulli observation model (Candès and Recht 2009; Klopp 2015) is adopted, in which the index set of all observed locations, denoted $\Omega \subset \{1, \dots, d_1\} \times \{1, \dots, d_2\}$, is independent to \mathbb{Y} , $E|\Omega| = n$, and $P\{(i, j) \in \Omega\} = n/(d_1 d_2)$ for all i and j . By the conditional independence assumption of \mathbb{Y} given Θ and the Bernoulli observation model, the sample log-likelihood function of the logistic exploratory IFA model equals to

$$\ell(\Theta; \mathbf{Y}, \Omega) = \sum_{(i,j) \in \Omega} \log p(y_{ij} | \theta_{ij}), \quad (6)$$

plus a constant that does not depend on Θ (which is omitted henceforth).

Per the request of a referee, it is commented that the JML parameterization is identified (e.g., Koopmans and Reiersøl 1950). For a fixed Ω , consider two distinct parameter matrices Θ_1 and Θ_2 with $\theta_{1,ij} \neq \theta_{2,ij}$ for some $(i, j) \in \Omega$, where $\theta_{1,ij}$ and $\theta_{2,ij}$ denote the respective (i, j) th entries of Θ_1 and Θ_2 . Then, it is impossible that $\ell(\Theta_1; \mathbf{Y}, \Omega) = \ell(\Theta_2; \mathbf{Y}, \Omega)$ for all \mathbf{Y} . Suppose that $\ell(\Theta_1; \mathbf{Y}_1, \Omega) = \ell(\Theta_2; \mathbf{Y}_1, \Omega)$ for some \mathbf{Y}_1 . Then, one can construct another data matrix \mathbf{Y}_2 such that $y_{2,ij} = 1 - y_{1,ij}$ at the (i, j) th entry and $y_{2,i'j'} = y_{1,i'j'}$ for all $(i', j') \neq (i, j)$. It can be verified that $\ell(\Theta_1; \mathbf{Y}_2, \Omega) \neq \ell(\Theta_2; \mathbf{Y}_2, \Omega)$, as the log-concavity of $\Psi(\cdot)$ guarantees that $\log \Psi(\theta_{1,ij}) - \log \Psi(\theta_{2,ij}) \neq \log[1 - \Psi(\theta_{1,ij})] - \log[1 - \Psi(\theta_{2,ij})]$.

JML estimation aims to find the maximum of Eq. 6 with respect to $\Theta \in \mathcal{M}_k(d_1, d_2)$. It is known that the JML solution can be unbounded when a row or column of the observed response matrix \mathbf{Y} contains only 0's or 1's (Chen et al. 2018). Further restrictions must therefore be imposed on the parameter space to ensure finite parameter estimates. Moreover, a bounded infinity norm of Θ , i.e., $\|\Theta\|_\infty = \max_{i,j} |\theta_{ij}| \leq M$ for some $M > 0$, is a key assumption for consistent parameter recovery (see Sect. 2.3). To this end, Chen et al. (2018) proposed to enforce L^2 -norm constraints on the rows of $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$: That is,

$$1 + \|\mathbf{u}_i\|^2 \leq M, \quad i = 1, \dots, d_1, \quad \text{and} \quad w_j^2 + \|\mathbf{v}_j\|^2 \leq M, \quad j = 1, \dots, d_2, \quad (7)$$

for some $M > 0$. Equations 2 and 7, together with the Cauchy–Schwarz inequality, imply that the infinity norm of Θ is bounded by M :

$$\|\Theta\|_\infty = \max_{i,j} |w_j + \mathbf{u}_i^\top \mathbf{v}_j| \leq \max_{i,j} \left(\sqrt{1 + \|\mathbf{u}_i\|^2} \sqrt{w_j^2 + \|\mathbf{v}_j\|^2} \right) \leq M. \quad (8)$$

Equation 8 suggests that Chen et al.'s (2018) row-wise bound (Eq. 7) is sufficient but not necessary for the desired infinity-norm bound $\|\Theta\|_\infty \leq M$. The present study, on the other hand, tackles

the latter bound directly. Define $\hat{\Theta}$ as the JML estimator that solves the following constrained optimization problem:

$$\begin{aligned} & \underset{\Theta}{\text{maximize}} \quad \ell(\Theta; \mathbf{Y}, \Omega) \\ & \text{subject to} \quad \Theta \in \mathcal{M}_k(d_1, d_2), \text{ and } \|\Theta\|_\infty \leq M. \end{aligned} \quad (9)$$

The JML estimator has appealing asymptotic properties as both d_1 and d_2 tend to infinity, which is described next.

2.3. Consistency

Let Θ_0 be the \mathbb{Y} -generating parameter matrix such that $\Theta_0 \in \mathcal{M}_k(d_1, d_2)$ and $\|\Theta_0\|_\infty \leq M$. Applying Lemma A.1 of Davenport et al. (2014), Chen et al. (2018) established that

$$\frac{\|\hat{\Theta} - \Theta_0\|_F^2}{d_1 d_2} \xrightarrow{P_{\Theta_0}} 0 \quad (10)$$

as $d_1, d_2 \rightarrow \infty$, in which $\|\mathbf{X}\|_F = \sqrt{\text{tr}(\mathbf{X}^\top \mathbf{X})}$ is the Frobenius norm of matrix \mathbf{X} , and $\xrightarrow{P_{\Theta_0}}$ denotes convergence in probability under the true model with parameters Θ_0 . Equation 10 conveys that the mean squared deviation between the JML solution $\hat{\Theta}$ and the true low-rank matrix Θ_0 approaches zero with high probability as both dimensions of the data matrix increase. In addition, an intermediate step (Eq. A.7) in the proof of Chen et al. (2018) also warrants consistent recovery of response probabilities in Kullback–Leibler divergence as $d_1, d_2 \rightarrow \infty$.

An immediate follow-up question is whether the item and person parameters can also be consistently estimated given Eq. 10. It has been pointed out in Sect. 2.1 that the factorization $\Theta = \tilde{\mathbf{U}}\tilde{\mathbf{V}}^\top$ is not unique due to rotational indeterminacy. Consequently, the person and item parameters are not identified—even though the low-rank parameter matrix Θ is identified, and thereby consistency can only be established up to a rotation. Let $\Theta_0 = \tilde{\mathbf{U}}_0 \tilde{\mathbf{V}}_0^\top$ be a specific factorization of the true parameter matrix of interest, in which $\tilde{\mathbf{U}}_0 = (\mathbf{1}_{d_1}/\sqrt{d_1}, \mathbf{U}_0)^\top$ and $\tilde{\mathbf{V}}_0 = (\mathbf{w}_0, \mathbf{V}_0)^\top$ are referred to as the true person and item parameters, respectively. Without loss of generality, assume that \mathbf{U}_0 is column-wise normalized, i.e., $\text{diag}(\mathbf{U}_0^\top \mathbf{U}_0) = \mathbf{1}_k$, in which $\text{diag}(\cdot)$ extracts the diagonal elements of a square matrix. Also express the JML estimator in the form of Eq. 5, i.e., $\hat{\Theta} = \mathbf{1}_{d_1} \hat{\mathbf{w}}_*^\top / \sqrt{d_1} + \hat{\mathbf{U}}_* \hat{\mathbf{R}}^\top \hat{\mathbf{V}}_*^\top$, and write $\tilde{\mathbf{U}} = (\mathbf{1}_{d_1}/\sqrt{d_1}, \hat{\mathbf{U}}_*)^\top$ and $\tilde{\mathbf{V}} = (\hat{\mathbf{w}}_*, \hat{\mathbf{V}}_* \hat{\mathbf{R}})^\top$. By virtue of the sine-theta theorem (O’Rourke et al. 2018, Theorem 19), it can be shown that the largest principal angle between $\text{span}(\tilde{\mathbf{U}})$ and $\text{span}(\tilde{\mathbf{U}}_0)$, and symmetrically that between $\text{span}(\tilde{\mathbf{V}})$ and $\text{span}(\tilde{\mathbf{V}}_0)$, converge to 0 in P_{Θ_0} -probability (Chen et al. 2018, Appendix C; Yu et al. 2015, Theorem 3). The convergence in principal angles further implies the consistent recovery of $\tilde{\mathbf{U}}_0$ and $\tilde{\mathbf{V}}_0$ up to an oblique rotation matrix (Browne 2001, Eq. 2), providing the smallest nonzero singular values of $\tilde{\mathbf{U}}_0$ and $\tilde{\mathbf{V}}_0$, i.e., $\sigma_{k+1}(\tilde{\mathbf{U}}_0)$ and $\sigma_{k+1}(\tilde{\mathbf{V}}_0)$, are bounded away from zero at suitable rates. A formal statement of the result is presented as Proposition 1, which is a corollary of Chen et al.’s (2018) Lemma C.1. The proposition is proved in “Appendix A.1”

Proposition 1. *Suppose that the true parameter matrix Θ_0 satisfies:*

- (i) $\Theta_0 \in \mathcal{M}_k(d_1, d_2)$, and $\|\Theta_0\|_\infty \leq M$;
- (ii) $\text{diag}(\mathbf{U}_0^\top \mathbf{U}_0) = \mathbf{1}_k$;
- (iii) $\sigma_{k+1}(\tilde{\mathbf{U}}_0) \geq c_1$ and $\sigma_{k+1}(\tilde{\mathbf{V}}_0) \geq c_2 \sqrt{d_1 d_2}$ for some constants $c_1, c_2 > 0$.

Then, there exists a $(k + 1) \times (k + 1)$ oblique rotation matrix $\bar{\mathbf{Q}}$, which contains $\mathbf{e}_{k+1}^\top = (1, \mathbf{0}_k^\top)$ as its first row and satisfies $\text{diag}(\bar{\mathbf{Q}}^{-1}\bar{\mathbf{Q}}^{-\top}) = \mathbf{1}_{k+1}$, such that

$$\frac{\|\tilde{\mathbf{V}}\bar{\mathbf{Q}} - \bar{\mathbf{V}}_0\|_F^2}{d_1 d_2} \xrightarrow{P_{\Theta_0}} 0, \quad (11)$$

and

$$\|\tilde{\mathbf{U}}\bar{\mathbf{Q}}^{-\top} - \bar{\mathbf{U}}_0\|_F^2 \xrightarrow{P_{\Theta_0}} 0, \quad (12)$$

as $d_1, d_2 \rightarrow \infty$.

Remark 1. Proposition 1 closely resembles but not exactly the same as Theorem 4 of Chen et al. (2018). Unlike the latter result, $\bar{\mathbf{U}}_0$ here need not have orthogonal columns, and hence it suffices to consider an oblique rotation matrix $\bar{\mathbf{Q}}$. It should also be noted that the scaling of person parameters in Assumption ii) and that in Chen et al. (2018, Assumption A3) differ by a factor of d_1 . The Frobenius-norm loss functions in Eqs. 11 and 12 are rescaled accordingly: No additional scaling factor is needed in Eq. 12 because $\tilde{\mathbf{U}}$ and $\bar{\mathbf{U}}_0$ have normalized columns.

Remark 2. Having \mathbf{e}_{k+1}^\top as the first row in the rotation matrix $\bar{\mathbf{Q}}$ ensures that the first column of the rotated person parameter matrix $\tilde{\mathbf{U}}\bar{\mathbf{Q}}^{-\top}$ remains to be $\mathbf{1}_{d_1}$. Correspondingly, the first column of the rotated item parameter matrix $\tilde{\mathbf{V}}\bar{\mathbf{Q}}$ can still be interpreted as item intercepts.

Remark 3. The choice of $\bar{\mathbf{Q}}$ in the proof of Proposition 1 depends on Θ_0 and thus remains unknown in practice. To settle on interpretable estimates of item parameters, one needs to estimate $\bar{\mathbf{Q}}$ by minimizing a complexity function that gauges the degree to which the rotated solution $\tilde{\mathbf{V}}\bar{\mathbf{Q}}$ deviates from the assumed simple structure of $\bar{\mathbf{V}}_0$ (see Browne 2001, for a review). Deriving suitable complexity functions is beyond the scope of the current paper and therefore left for future research.

3. Geometry of the Matrix Manifold $\mathcal{M}_k(d_1, d_2)$

The proposed algorithm for solving the JML estimation problem (Eq. 9) relies on several fundamental geometric ingredients of the fixed-rank matrix manifold $\mathcal{M}_k(d_1, d_2)$, which are introduced in the current section. Readers are also referred to Absil et al. (2008) for a comprehensive overview on the differential geometry of matrix manifolds.

3.1. Embedded Manifold and Tangent Space

Let m and n be positive integers such that $m < n$. A m -dimensional manifold \mathcal{M} embedded in the ambient Euclidean space \mathcal{R}^n is essentially a subset of form $\{\mathbf{x} \in \mathcal{R}^n : \Phi(\mathbf{x}) = \mathbf{0}_{n-m}\}$, in which $\Phi : \mathcal{R}^n \rightarrow \mathcal{R}^{n-m}$ is a smooth function whose Jacobian is of full rank.⁴ For example, the unit sphere $\{\mathbf{x} \in \mathcal{R}^3 : \Phi(\mathbf{x}) = \mathbf{x}^\top \mathbf{x} - 1 = 0\}$ is a two-dimensional manifold embedded in \mathcal{R}^3 . Let \mathbf{x} be a point on the manifold \mathcal{M} and $\boldsymbol{\gamma} : \mathcal{R} \rightarrow \mathcal{M}$ be a smooth curve such that $\boldsymbol{\gamma}(0) = \mathbf{x}$. The tangent space of \mathcal{M} at \mathbf{x} , denoted $\mathcal{T}_{\mathbf{x}}\mathcal{M}$, is a linear subspace composed of tangent vectors to $\boldsymbol{\gamma}$ at the origin, i.e., $\dot{\boldsymbol{\gamma}}(0) = \lim_{t \rightarrow 0} [\boldsymbol{\gamma}(t) - \boldsymbol{\gamma}(0)]/t$. Heuristically, the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ can be conceived as a local approximation to the embedded manifold \mathcal{M} . It can also be shown that $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ is isomorphic to the Euclidean space \mathcal{R}^m (Absil et al. 2008, Sect. 3.5.7) and thus is much easier to deal with compared to \mathcal{M} itself.

Proposition 2 confirms that the parameter space of an exploratory IFA model is a $[d_1 k + (d_2 - k)(k + 1)]$ -dimensional embedded manifold and explicitly expresses the tangent vector.⁵ A proof

⁴This definition is a special case of the more general version in Absil et al. (2008, Sect. 3.3).

⁵Members of $\mathcal{T}_{\Theta}\mathcal{M}_k(d_1, d_2)$ are referred to as tangent vectors, although they are in fact matrices.

can be found in “Appendix A.2” Denote by \mathbf{U}_\perp and \mathbf{V}_\perp the respective orthogonal complements of \mathbf{U}_* and \mathbf{V}_* such that $\mathbf{U}_\perp^\top \mathbf{U}_\perp = \mathbf{I}_{d_1-k}$ and $\mathbf{V}_\perp^\top \mathbf{V}_\perp = \mathbf{I}_{d_2-k}$.

Proposition 2. *The set $\mathcal{M}_k(d_1, d_2)$ defined in Eq. 4 is a manifold of dimension $d_1k + (d_2-k)(k+1)$ embedded in $\mathcal{R}^{d_1 \times d_2}$. The tangent space $\mathcal{T}_\Theta \mathcal{M}_k(d_1, d_2)$ at $\Theta \in \mathcal{M}_k(d_1, d_2)$ is given by*

$$\mathcal{T}_\Theta \mathcal{M}_k(d_1, d_2) = \left\{ \frac{\mathbf{1}_{d_1} \mathbf{a}^\top \mathbf{V}_\perp^\top}{\sqrt{d_1}} + \mathbf{U}_* \mathbf{B} \mathbf{V}_*^\top + \mathbf{U}_\perp \mathbf{C} \mathbf{V}_*^\top + \mathbf{U}_* \mathbf{D}^\top \mathbf{V}_\perp^\top \in \mathcal{R}^{d_1 \times d_2} : \right. \\ \left. \mathbf{a} \in \mathcal{R}^{d_2-k}, \mathbf{B} \in \mathcal{R}^{k \times k}, \mathbf{C} \in \mathcal{R}^{(d_1-k) \times k}, \mathbf{D} \in \mathcal{R}^{(d_2-k) \times k} \right\}. \quad (13)$$

3.2. Riemannian Gradient

A manifold \mathcal{M} is called Riemannian if each tangent space $\mathcal{T}_\mathbf{x} \mathcal{M}$, $\mathbf{x} \in \mathcal{M}$, is equipped with an inner product, termed a Riemannian metric, which varies smoothly as a function of \mathbf{x} (Absil et al. 2008, Sect. 3.6). For an embedded manifold, a natural Riemannian metric is the Euclidean inner product inherited from the ambient space. For a Riemannian manifold \mathcal{M} embedded in \mathcal{R}^n and a differentiable function $f : \mathcal{M} \rightarrow \mathcal{R}$, the Riemannian gradient of f at $\mathbf{x} \in \mathcal{M}$, denoted $\text{grad } f(\mathbf{x})$, is the orthogonal projection of the ambient gradient $\partial f(\mathbf{x})/\partial \mathbf{x} \in \mathcal{R}^n$ onto the tangent space $\mathcal{T}_\mathbf{x} \mathcal{M}$. The notion of Riemannian gradient is important for the optimization of an objective function f since it specifies locally the direction of steepest ascent on the tangent space (Absil et al. 2008, Sect. 3.6).

Because the matrix manifold $\mathcal{M}_k(d_1, d_2)$ is embedded in $\mathcal{R}^{d_1 \times d_2}$, restricting the regular matrix inner product, i.e., $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{tr}(\mathbf{X}^\top \mathbf{Y})$ where $\mathbf{X}, \mathbf{Y} \in \mathcal{R}^{d_1 \times d_2}$, to the tangent space turns $\mathcal{M}_k(d_1, d_2)$ into a Riemannian manifold. For any differentiable objective function f defined on the manifold, an explicit formula for the Riemannian gradient of f is provided as Proposition 3, which is derived in “Appendix A.3”

Proposition 3. *Let $f : \mathcal{M}_k(d_1, d_2) \rightarrow \mathcal{R}$, $\Theta \mapsto f(\Theta)$, be a differentiable function in the ordinary sense and $\mathbf{G} = \partial f(\Theta)/\partial \Theta \in \mathcal{R}^{d_1 \times d_2}$ be the ambient Euclidean gradient. The Riemannian gradient of f at Θ , denoted $\text{grad } f(\Theta)$, is given by*

$$\text{grad } f(\Theta) = \frac{\mathbf{1}_{d_1} \mathbf{1}_{d_1}^\top \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top}{d_1} + \mathbf{U}_* \mathbf{U}_*^\top \mathbf{G} \mathbf{V}_* \mathbf{V}_*^\top + \mathbf{U}_\perp \mathbf{U}_\perp^\top \mathbf{G} \mathbf{V}_* \mathbf{V}_*^\top + \mathbf{U}_* \mathbf{U}_*^\top \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top, \quad (14)$$

which is the orthogonal projection of \mathbf{G} onto $\mathcal{T}_\Theta \mathcal{M}_k(d_1, d_2)$.

Remark 4. Equation 14 can be identified as a tangent vector (Eq. 13) by setting $\mathbf{a} = \mathbf{V}_\perp^\top \mathbf{G}^\top \mathbf{1}_{d_1} / \sqrt{d_1}$, $\mathbf{B} = \mathbf{U}_*^\top \mathbf{G} \mathbf{V}_*$, $\mathbf{C} = \mathbf{U}_\perp^\top \mathbf{G} \mathbf{V}_*$, $\mathbf{D} = \mathbf{V}_\perp^\top \mathbf{G}^\top \mathbf{U}_*$. Based on the observation that the Riemannian gradient depends on \mathbf{U}_\perp and \mathbf{V}_\perp only through $\mathbf{U}_\perp \mathbf{U}_\perp^\top = \mathbf{I}_{d_1} - \mathbf{U}_* \mathbf{U}_*^\top$ and $\mathbf{V}_\perp \mathbf{V}_\perp^\top = \mathbf{I}_{d_2} - \mathbf{V}_* \mathbf{V}_*^\top$, direct evaluation of \mathbf{U}_\perp and \mathbf{V}_\perp can be circumvented in actual computation: One can equivalently represent the Riemannian gradient using $\mathbf{V}_\perp \mathbf{a} = (\mathbf{G}^\top \mathbf{1}_{d_1} - \mathbf{V}_* \mathbf{V}_*^\top \mathbf{G}^\top \mathbf{1}_{d_1}) / \sqrt{d_1}$, $\mathbf{U}_\perp \mathbf{C} = \mathbf{G} \mathbf{V}_* - \mathbf{U}_* \mathbf{B}$, and $\mathbf{V}_\perp \mathbf{D} = \mathbf{G}^\top \mathbf{U}_* - \mathbf{V}_* \mathbf{B}^\top$ in lieu of \mathbf{a} , \mathbf{C} , and \mathbf{D} .

3.3. Retraction

At a given location \mathbf{x} on a smooth manifold \mathcal{M} embedded in \mathcal{R}^n , a retraction $R_\mathbf{x} : \mathcal{T}_\mathbf{x} \mathcal{M} \rightarrow \mathcal{M}$ satisfies

$$R_\mathbf{x}(\mathbf{0}_n) = \mathbf{x}, \quad (15)$$

$$\left. \frac{dR_{\mathbf{x}}(s\xi)}{ds} \right|_{s=0} = \xi, \quad (16)$$

in which $\xi \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$. Eqs. 15 and 16 are referred to as the centering and local rigidity conditions, respectively (Absil et al. 2008, Definition 4.1.1; Shalit et al. 2012). Retraction is an approximation to the exponential map, which takes tangent vectors back to the manifold along geodesics (Absil et al. 2008, Sect. 5.4). Such a “pull-back” operation needs to be performed at every iteration of a Riemannian optimization algorithm so as to bring a local gradient ascent update, which is a member of $\mathcal{T}_{\mathbf{x}}\mathcal{M}$, back to the manifold \mathcal{M} .

It can be shown that the orthogonal projection of $\mathbf{x} + \xi$ onto \mathcal{M} satisfies Eqs. 15 and 16 (Absil and Malick 2012),⁶ which makes it a very common choice of retraction (e.g., Vandereycken 2013; Absil et al. 2008 Sect. 4.8). However, other retractions may be favored for computational efficiency under certain circumstances (e.g., Shalit et al. 2012, and the current work). Proposition 4 describes a valid retraction operator (Eq. 17) on the matrix manifold $\mathcal{M}_k(d_1, d_2)$ that is not a projection; the centering and local rigidity conditions are verified in “Appendix A.4” In contrast to the matrix completion setting wherein the projection can be efficiently computed via singular value decomposition (Vandereycken 2013, Algorithm 6), the projection onto $\mathcal{M}_k(d_1, d_2)$ is more involved on account of the inclusion of $\mathbf{1}_{d_1}$ in the column basis and thus is not further considered.

Proposition 4. Let $\Theta \in \mathcal{M}_k(d_1, d_2)$ and $\Xi \in \mathcal{T}_{\Theta}\mathcal{M}_k(d_1, d_2)$. By construction, $\Theta = \mathbf{1}_{d_1}\mathbf{w}_*^{\top}/\sqrt{d_1} + \mathbf{U}_*\mathbf{R}^{\top}\mathbf{V}_*^{\top}$ (Eq. 5), and $\Xi = \mathbf{1}_{d_1}\mathbf{a}^{\top}\mathbf{V}_{\perp}^{\top}/\sqrt{d_1} + \mathbf{U}_*\mathbf{B}\mathbf{V}_*^{\top} + \mathbf{U}_{\perp}\mathbf{C}\mathbf{V}_*^{\top} + \mathbf{U}_*\mathbf{D}^{\top}\mathbf{V}_{\perp}^{\top}$ (Eq. 13). Then, the mapping

$$R_{\Theta}(\Xi) = \frac{\mathbf{1}_{d_1}(\mathbf{w}_* + \mathbf{V}_{\perp}\mathbf{a})^{\top}}{\sqrt{d_1}} + [\mathbf{U}_*(\mathbf{R}^{\top} + \mathbf{B}) + \mathbf{U}_{\perp}\mathbf{C}] [\mathbf{V}_* + \mathbf{V}_{\perp}\mathbf{D}\mathbf{R}^{-1}]^{\top} \quad (17)$$

is a retraction.

Remark 5. It is not difficult to see that $R_{\Theta}(\Xi)$ (Eq. 17) is a member of $\mathcal{M}_k(d_1, d_2)$ (Eq. 4). Updated values of \mathbf{w}_* , \mathbf{U}_* , \mathbf{R} , and \mathbf{V}_* corresponding to $R_{\Theta}(\Xi)$ can be obtained via the same QR decomposition argument that yields Eq. 5.

3.4. Vector Transport

Given two points \mathbf{x}_1 and \mathbf{x}_2 on a smooth manifold \mathcal{M} , a vector transport $T_{\mathbf{x}_1 \rightarrow \mathbf{x}_2} : \mathcal{T}_{\mathbf{x}_1}\mathcal{M} \rightarrow \mathcal{T}_{\mathbf{x}_2}\mathcal{M}$ maps tangent vectors from one tangent space to the other. Analogous to viewing a retraction as an approximation to the exponential map, a vector transport can be understood as an approximation to the parallel translation between a pair of tangent spaces. The exact definition of parallel translation and vector transport can be found in Absil et al. (2008, Sect. 8.1); specifically for embedded manifolds, it can be shown that the orthogonal projection onto the tangent space gives a vector transport.

At a particular iteration of the proposed Riemannian CG algorithm (Algorithm 2), the search direction is determined by the current Riemannian gradient, as well as the search direction and Riemannian gradient at the previous iteration. To make tangent vectors defined at different locations (i.e., the current and previous iterates) comparable, a vector transport must be invoked. For $\Theta^{(1)}, \Theta^{(2)} \in \mathcal{M}_k(d_1, d_2)$ and $\Xi^{(1)} \in \mathcal{T}_{\Theta^{(1)}}\mathcal{M}_k(d_1, d_2)$, the orthogonal projection of $\Xi^{(1)}$ onto

⁶The orthogonal projection typically approximates the exponential map to the second-order, which is even stronger than Eqs. 15 and 16.

$\mathcal{T}_{\Theta^{(2)}}\mathcal{M}_k(d_1, d_2)$ serves as a vector transport and has the following expression:

$$\begin{aligned} T_{\Theta^{(1)} \rightarrow \Theta^{(2)}}(\Xi^{(1)}) &= \frac{\mathbf{1}_{d_1} \mathbf{1}_{d_1}^\top \Xi^{(1)} \mathbf{V}_\perp^{(2)} (\mathbf{V}_\perp^{(2)})^\top}{d_1} + \mathbf{U}_*^{(2)} (\mathbf{U}_*^{(2)})^\top \Xi^{(1)} \mathbf{V}_*^{(2)} (\mathbf{V}_*^{(2)})^\top \\ &\quad + \mathbf{U}_\perp^{(2)} (\mathbf{U}_\perp^{(2)})^\top \Xi^{(1)} \mathbf{V}_*^{(2)} (\mathbf{V}_*^{(2)})^\top + \mathbf{U}_*^{(2)} (\mathbf{U}_*^{(2)})^\top \Xi^{(1)} \mathbf{V}_\perp^{(2)} (\mathbf{V}_\perp^{(2)})^\top. \end{aligned} \quad (18)$$

The derivation of Eq. 18 follows in essence the same steps as that of Eq. 14 and thus is not repeated.

4. Algorithm

4.1. A Linear-Quadratic Penalty Method

JML estimation of exploratory IFA poses a constrained manifold optimization problem (e.g., Liu and Boumal 2019, Eq. 1).⁷ One general strategy for constrained optimization is to incorporate penalties for violations of the constraints into the objective function and solve the resulting unconstrained, penalized problem. For example, Eq. 9 can be converted to the following penalized maximization problem on the fixed-rank matrix manifold:

$$\begin{aligned} \underset{\Theta}{\text{maximize}} \quad & \ell(\Theta; \mathbf{Y}, \Omega) - \lambda \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \max\{0, |\theta_{ij}| - M\} \\ \text{subject to} \quad & \Theta \in \mathcal{M}_k(d_1, d_2), \end{aligned} \quad (19)$$

in which $\lambda > 0$ is the penalty weight. Proposition 4.1 in Liu and Boumal (2019) guarantees that the solution of Eq. 9 necessarily solves Eq. 19 for large enough λ . However, Eq. 19 is still challenging to solve as the objective function entails the max function that is not smooth. Fortunately, it is possible to define a sequence of smooth and increasingly closer approximations to the non-smooth objective. Each of the resulting sub-problem can be handled by a Riemannian CG algorithm for smooth, unconstrained optimization on $\mathcal{M}_k(d_1, d_2)$ (Algorithm 2; see Sect. 4.2).

Pinar and Zenios (1994, Eq. 4) suggested the following piecewise linear-quadratic approximation to the function $\max\{0, x\}$:

$$\rho(x, \mu) = \begin{cases} 0, & \text{if } x \leq 0, \\ x^2/(2\mu), & \text{if } 0 < x \leq \mu, \\ x - \mu/2, & \text{if } x > \mu, \end{cases} \quad (20)$$

in which $\mu > 0$ is labeled as a smoothing constant. It is straightforward to verify that $\rho(x, \mu)$ converges to $\max\{0, x\}$ pointwise in x as $\mu \downarrow 0$, and that $\rho(x, \mu)$ is continuously differentiable in x for fixed μ . The left panel of Fig. 1 displays the graph of $\max\{0, x\}$ and the graphs of $\rho(x, \mu)$

⁷The term is used when the optimization problem involves constraints in addition to those imposed by the manifold (e.g., the infinity-norm constraint in JML estimation). Similarly, an unconstrained manifold optimization problem involves only manifold constraints.

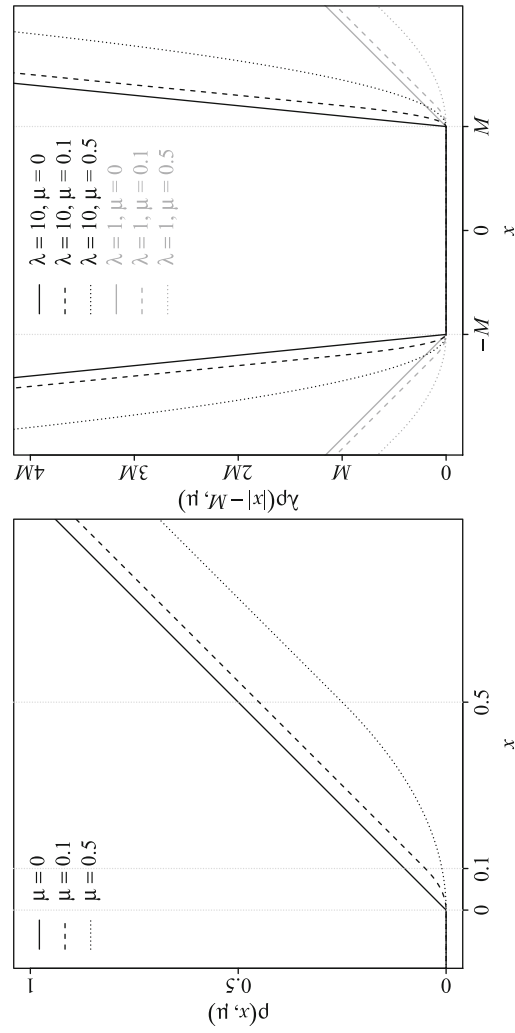


FIGURE 1.

Linear-quadratic penalty function. Left panel: Function $\rho(x, \mu)$ with different choices of μ . As $\mu \downarrow 0$, $\rho(x, \mu)$ approaches $\rho(0, \mu) = \max\{0, x\}$ (solid line) which is not differentiable at $x = 0$. Right panel: Function $\lambda \rho(|x| - M, \mu)$ with different choices of λ and μ .

Algorithm 1

A Riemannian smoothed penalty method

Require: Data \mathbf{Y} , observation indices Ω , starting values of the parameter matrix $\Theta^{[0]} \in \mathcal{M}_k(d_1, d_2)$, infinity-norm bound $M > 0$, initial penalty weight $\lambda^{[0]} > 0$, initial smoothing constant $\mu^{[0]} > 0$, initial tolerance for the sub-problem $\varepsilon^{[0]} > 0$, final tolerance for the sub-problem $\varepsilon > 0$, final tolerance for the smoothing constant $\mu > 0$, tolerance for maximum parameter change $\delta > 0$, tolerance for infeasible solutions $\tau > 0$, multipliers $\kappa_\lambda, \kappa_\mu, \kappa_\varepsilon > 0$, and additional tuning parameters for the Riemannian CG algorithm (Algorithm 2)

- 1: **for** $r = 0, 1, \dots$ **do**
- 2: Solve the sub-problem (Eq. 21) $\Theta^{[r+1]} = \operatorname{argmax}_{\Theta \in \mathcal{M}_k(d_1, d_2)} f(\Theta; \lambda^{[r]}, \mu^{[r]}; \mathbf{Y}, \Omega)$ with tolerance $\varepsilon^{[r]}$ and warm start $\Theta^{[r]}$ (Algorithm 2);
- 3: **if** $\|\Theta^{[r+1]} - \Theta^{[r]}\|_\infty \leq \delta$, $\varepsilon^{[r]} \leq \varepsilon$, and $\mu^{[r]} \leq \mu$ **then**
- 4: **return** $\Theta^{[r+1]}$
- 5: **end if**
- 6: Set $\varepsilon^{[r+1]} = \max\{\varepsilon, \kappa_\varepsilon \varepsilon^{[r]}\}$ and $\mu^{[r+1]} = \max\{\mu, \kappa_\mu \mu^{[r]}\}$
- 7: **if** $\|\Theta^{[r+1]}\|_\infty - M > \tau$ **then**
- 8: Set $\lambda^{[r+1]} = \kappa_\lambda \lambda^{[r]}$
- 9: **else**
- 10: Set $\lambda^{[r+1]} = \lambda^{[r]}$
- 11: **end if**
- 12: **end for**

with two different values of μ . Substituting $\rho(\cdot, \mu)$ for $\max\{0, \cdot\}$ in Eq. 19 yields the following smoothed sub-problem of JML estimation:

$$\begin{aligned}
 & \underset{\Theta}{\text{maximize}} \quad f(\Theta; \lambda, \mu; \mathbf{Y}, \Omega) = \ell(\Theta; \mathbf{Y}, \Omega) - \lambda \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \rho(|\theta_{ij}| - M, \mu) \\
 & \text{subject to} \quad \Theta \in \mathcal{M}_k(d_1, d_2).
 \end{aligned} \tag{21}$$

It is observed from the right panel of Fig. 1 that the function $\lambda\rho(|x| - M, \mu)$ diverges as x extends beyond $[-M, M]$: At any fixed x outside the interval, increasing λ or decreasing μ results in a larger penalty.

Ideally, one prefers to work on a sub-problem (Eq. 21) with a sufficiently large λ and a sufficiently small μ so that the solution is as close to the JML estimator (i.e., the solution to Eq. 9) as possible. Such a sub-problem, however, is likely to be ill-conditioned and incurs convergence issues for numerical search algorithms. A practically more robust scheme is to form a sequence of sub-problems with increasing λ 's and decreasing μ 's, each of which is solved numerically using the solution of the preceding sub-problem as a warm start.

A particular implementation of the iterative strategy is summarized as Algorithm 1, which is an adaptation of Liu and Boumal's (2019) Algorithm 2. Convergence of Algorithm 1 is determined by four pre-specified tolerances: δ for maximum parameter change, ε for the Riemannian gradient (see Sect. 3 for its definition) in the final sub-problem, μ for the final smoothing constant in the linear-quadratic penalty, and τ for violating the infinity-norm constraint. As the algorithm iterates (along r), new sub-problems are defined with non-decreasing $\lambda^{[r]}$'s and decreasing $\mu^{[r]}$'s; meanwhile, a decreasing sequence of $\varepsilon^{[r]}$'s is used as convergence tolerances when solving the sub-problems, as it is unnecessary to obtain precise solutions at the few initial iterations. The optimal values of tuning constants are likely to be problem-specific, but the following default

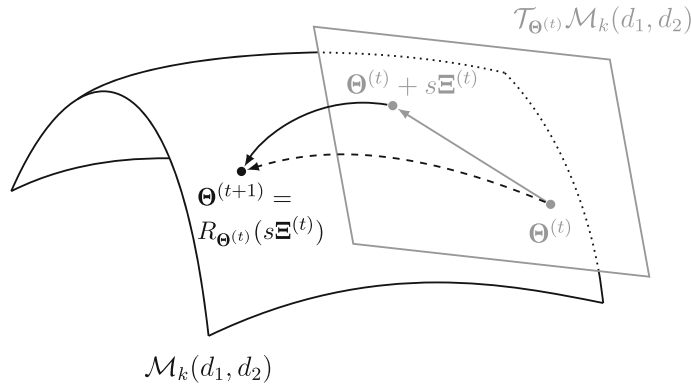


FIGURE 2.

Illustration of a Riemannian gradient step $\Theta^{(t)} \mapsto \Theta^{(t+1)}$ (dashed arrow) on the matrix manifold $\mathcal{M}_k(d_1, d_2)$. The update comprises two steps. First, a search direction $\Xi^{(t)}$ is determined based on the current iterate $\Theta^{(t)}$. An Euclidean gradient ascent step (gray solid arrow) is then carried out within the tangent space $\mathcal{T}_{\Theta^{(t)}}\mathcal{M}_k(d_1, d_2)$, resulting in an updated location $\Theta^{(t)} + s\Xi^{(t)}$, in which $s > 0$ is the step size. Second, applying the retraction $R_{\Theta^{(t)}}(s\Xi^{(t)})$ brings the tangent vector $\Theta^{(t)} + s\Xi^{(t)}$ back to the manifold (black solid arrow), leading to the new iterate $\Theta^{(t+1)}$.

Algorithm 2

A Riemannian conjugate gradient algorithm

Require: Data \mathbf{Y} , observation indices Ω , starting values of the low-rank parameter matrix $\Theta^{(0)} \in \mathcal{M}_k(d_1, d_2)$, convergence tolerance $\varepsilon > 0$, and additional tuning parameters for the line search

- 1: **for** $t = 0, 1, \dots$ **do**
 - 2: Evaluate the Riemannian gradient $\text{grad } f(\Theta^{(t)})$ (Eq. 14)
 - 3: **if** $\|\text{grad } f(\Theta^{(t)})\|_F \leq \varepsilon$ **then**
 - 4: **return** $\Theta^{(t)}$
 - 5: **end if**
 - 6: Compute the conjugate direction $\Xi^{(t)}$ by the Polak-Ribière rule (Eq. 25)
 - 7: Determine the step size $s^{(t)}$ via an approximate line search (Eq. 27)
 - 8: Update the parameters by retraction $\Theta^{(t+1)} = R_{\Theta^{(t)}}(s^{(t)}\Xi^{(t)})$ (Eq. 17)
 - 9: **end for**
-

configuration works reasonably well in the reported Monte Carlo studies (Sect. 5): $\lambda^{[0]} = 1$, $\mu^{[0]} = 0.1$, $\varepsilon^{[0]} = 0.1$, $\kappa_\lambda = 2.5$, $\kappa_\mu = (\mu/\mu^{[0]})^{1/10}$, and $\kappa_\varepsilon = (\varepsilon/\varepsilon^{[0]})^{1/10}$.

4.2. A Riemannian Conjugate Gradient Sub-solver

The sub-problem (Eq. 21) appertains to unconstrained maximization of a smooth objective function $f(\Theta; \lambda, \mu; \mathbf{Y}, \Omega)$ on the smooth manifold $\mathcal{M}_k(d_1, d_2)$ (Eq. 4). For notational simplicity, the objective function is abbreviated to $f(\Theta)$ in the sequel when other dependencies need not be highlighted. Numerical search problems of such kind can often be solved efficiently via Riemannian optimization (e.g., Shalit et al. 2012; Vandereycken 2013), which capitalizes on the differential geometry of the manifold. Heuristically, a single Riemannian gradient ascent iteration on a such an embedded manifold is composed of 1) traveling from the current iterate along an ascent direction within the tangent space, which amounts to an Euclidean gradient step and 2) performing retraction onto the manifold. A graphical illustration of a Riemannian gradient step can be found in Fig. 2.

The proposed Riemannian CG algorithm, whose main steps are summarized as Algorithm 2, is deduced from Algorithm 13 in Absil et al. (2008); a similar proposal can be found in Vandereycken (2013, Algorithm 1) for the purpose of matrix completion with noisy continuous data. Riemannian CG generalizes nonlinear CG (Golub and Van Loan 2013, Chapter 10) for unconstrained optimization on Euclidean spaces, which is known to converge faster than the basic gradient descent/ascent method. For JML estimation, the Riemannian CG algorithm also fares more time- and memory-efficient compared to Riemannian (quasi-)Newton algorithms (Absil et al. 2008, Algorithm 4; Huang et al. 2015): The high-dimensionality of the fixed-rank matrix manifold (see Proposition 2) renders solving the Newton equation prohibitively expensive, although the rate of convergence for (quasi-)Newton methods is superior in theory.

The Riemannian CG algorithm and the AM algorithm (Chen et al. 2018) have the same order of computational complexity per iteration. To illustrate, consider fixed k and $d_1 = d_2 = O(d)$. An AM iteration computes the gradient of the log-likelihood for each person and each item, which amounts to $O(d^2)$ flops. Meanwhile, a Riemannian CG iteration encompasses not only the same gradient calculations but also various sorts of matrix operations (e.g., matrix multiplication and QR decomposition): The leading order of computational complexity is determined by multiplying an $O(d) \times O(d)$ matrix by an $O(d) \times k$ matrix, which is $O(d^2)$ as well. Even though the computational cost is the same in order within each iteration, the Riemannian algorithm often converges much faster than the AM algorithm does (see Sect. 5.1).

It is also remarked that the Riemannian gradient method is in general not the same as the gradient projection method (which is originally developed for convex optimization; see Bertsekas 1999, Sect. 2.3). Both algorithms perform gradient ascent/descent updates and then return to the manifold by projection (or retraction, which is treated momentarily as a synonym for projection for ease of exposition). However, the projected gradient method uses the ambient gradient $\partial f(\Theta)/\partial \Theta$ for the update, whereas the Riemannian gradient method uses $\text{grad } f(\Theta)$, which is the orthogonal projection of the ambient gradient onto the tangent space $T_{\Theta}\mathcal{M}_k(d_1, d_2)$. Vandereycken (2013, p. 1222) demonstrated in the context of matrix completion that retracting a tangent vector onto a fixed-rank matrix manifold is of lower-order complexity than projecting a general full-rank matrix onto the same manifold. It is then inferred that a Riemannian gradient update, which involves retracting the tangent vector $\text{grad } f(\Theta)$, is computationally more favorable than a projected gradient update, which includes projecting a general full-rank matrix $\Theta + \partial f(\Theta)/\partial \Theta$.

Miscellaneous implementation details of Algorithm 2 are provided as follows.

4.2.1. Euclidean Derivatives Differentiating the objective function $f(\Theta; \lambda, \mu; \mathbf{Y}, \Omega)$ with respect to the parameter matrix Θ yields the ambient Euclidean gradient

$$\mathbf{G} = \frac{\partial f(\Theta; \lambda, \mu; \mathbf{Y}, \Omega)}{\partial \Theta} = \mathbf{Y}_{\Omega} - \Psi(\Theta_{\Omega}) - \lambda \rho'_{M,\mu}(\Theta_{\Omega}), \quad (22)$$

in which the subscript Ω indicates the projection onto the set of observed entries, i.e., $\mathbf{X}_{\Omega} = (Z_{ij} : Z_{ij} = X_{ij} \text{ if } (i, j) \in \Omega; Z_{ij} = 0 \text{ if } (i, j) \notin \Omega)$ for any $\mathbf{X} \in \mathcal{R}^{d_1 \times d_2}$. $\rho'_{M,\mu}$ is a scalar function

$$\rho'_{M,\mu}(x) = \begin{cases} 0, & \text{if } |x| - M \leq 0, \\ \text{sgn}(x)(|x| - M)/\mu, & \text{if } 0 < |x| - M \leq \mu, \\ \text{sgn}(x), & \text{if } |x| - M > \mu, \end{cases} \quad (23)$$

in which $\text{sgn}(\cdot)$ denotes the sign function. Both Ψ and $\rho'_{M,\mu}$ are applied element-wise to Θ_{Ω} in Eq. 22.

4.2.2. Conjugate Gradient Update A Riemannian gradient ascent step is performed when $t = 0$: i.e., $\Xi^{(0)} = \text{grad } f(\Theta^{(0)})$. At iteration $t > 0$, the conjugate search direction $\Xi^{(t)}$ (Line 6 of Algorithm 2) is determined by

$$\Xi^{(t)} = \text{grad } f(\Theta^{(t)}) + \beta_t T_{\Theta^{(t-1)} \rightarrow \Theta^{(t)}}(\Xi^{(t-1)}), \quad (24)$$

in which β_t , the weight assigned to the (vector transport of the) previous search direction is computed by generalizing the Polak–Ribière rule for a Euclidean nonlinear conjugate gradient update (Polak and Ribière 1969):

$$\beta_t = \frac{\langle \text{grad } f(\Theta^{(t)}) - T_{\Theta^{(t-1)} \rightarrow \Theta^{(t)}}(\text{grad } f(\Theta^{(t-1)})), \text{grad } f(\Theta^{(t)}) \rangle}{\langle \text{grad } f(\Theta^{(t-1)}), \text{grad } f(\Theta^{(t-1)}) \rangle}. \quad (25)$$

In Eqs. 24 and 25, the vector transport $T_{\Theta^{(t-1)} \rightarrow \Theta^{(t)}}$ is applied to translate $\Xi^{(t-1)}$ and $\text{grad } f(\Theta^{(t-1)})$ to the current tangent space. To ensure that the search direction is gradient-related (Absil et al. 2008, Definition 4.2.1), it is necessary to monitor the cosine angle between $\Xi^{(t)}$ and the Riemannian gradient $\text{grad } f(\Theta^{(t)})$:

$$\vartheta^{(t)} = \frac{\langle \Xi^{(t)}, \text{grad } f(\Theta^{(t)}) \rangle}{\sqrt{\langle \Xi^{(t)}, \Xi^{(t)} \rangle \langle \text{grad } f(\Theta^{(t)}), \text{grad } f(\Theta^{(t)}) \rangle}}. \quad (26)$$

In the current implementation, the search direction is reset to $\text{grad } f(\Theta^{(t)})$ whenever $\vartheta^{(t)} < 0.1$.

4.2.3. Line Search The step size $s^{(t)}$ at iteration t (Line 7 of Algorithm 2) is decided from an Armijo-type backtracking line search (Absil et al. 2008, Sect. 4.6.3). Given an initial guess of the step size $s_0^{(t)} > 0$ and constants $\alpha_1, \alpha_2 \in (0, 1)$, backtracking is performed to find the smallest non-negative integer m such that

$$f(R_{\Theta^{(t)}}(s_0^{(t)} \alpha_2^m \Xi^{(t)})) - f(\Theta^{(t)}) > \alpha_1 \langle s_0^{(t)} \alpha_2^m \Xi^{(t)}, \text{grad } f(\Theta^{(t)}) \rangle. \quad (27)$$

By default, α_1 and α_2 are fixed at 10^{-4} and 0.5, respectively, in the proposed algorithm. The initial step size is determined by a simple procedure suggested by Borckmans et al. (2014, Sect. 5.3): Set $s_0^{(0)} = 1$ and $s_0^{(t)} = \omega s^{(t-1)}$, in which $\omega > 1$ is a pre-specified enlarging factor. The default value $\omega = 2.5$ is used in the subsequent Monte Carlo studies (Sect. 5).

4.3. Convergence

For a sufficiently large λ and $\delta = \mu = \varepsilon = 0$, the (infinite) sequence of sub-problem solutions $\Theta^{[r]}$ produced by the Riemannian penalty method (Algorithm 1) converges and the limit point satisfies the first-order necessary conditions of the constrained problem (C. Liu and Boumal 2019, Proposition 4.2; see also Definition 2.3 for a formal statement of the first-order necessary conditions). To assure termination in finitely many steps, positive δ , μ , and ε values need to be used. Besides, the appropriate magnitude of λ is unlikely to be known in practice; the heuristic of increasing $\lambda^{[r]}$ contingent upon a check of feasibility (Line 7 of Algorithm 1) was originally suggested by Pinar and Zenios (1994) for constrained optimization on Euclidean spaces.

As for the sub-solver, it is noted that the limit point for a sequence of fixed-rank matrices may be rank deficient and thus no longer reside in the manifold. Therefore, the local convergence of rank-constrained optimization is typically established upon modifying the objective function to fence off iterates with lower ranks (Vandereycken 2013, Sect. 4). For instance, an additional penalty term $\zeta \|\Theta^+\|_F^2/2$ can be appended to the objective function $f(\Theta)$, in which Θ^+ stands for the Moore–Penrose pseudoinverse of Θ , and ζ is the penalty weight. Because $\|\Theta^+\|_F^2$ equals to the sum of inverse squares of the first $k+1$ singular values of Θ when $\Theta \in \mathcal{M}_k(d_1, d_2)$, the penalty functions as a barrier to matrices with ranks lower than $k+1$. The local convergence of Algorithm 2 with the modified objective function can be shown using essentially the same argument as the proof of Propositions 4.1 and 4.2 in Vandereycken (2013). The proof hinges upon the fact that the backtracing line search (Eq. 27) always increases the objective function as the algorithm iterates.

In addition, there is in general no guarantee that the critical point being converged to is the global maximum, a drawback shared by gradient-type methods for unconstrained optimization (e.g., Absil et al. 2008). Fortunately, numerical experiments suggest that the proposed algorithm almost always converges to a stable solution regardless of starting values, as long as the fitted dimensionality does not exceed the truth (see Sects. 5.1 and 5.2). Over-fitting k , on the other hand, typically incurs non-convergence. The simulation results in Sect. 5.2 imply that the convergence issue is likely a problem inherent to the optimization problem (Eq. 9), not just a feature of the proposed algorithm. Future investigation is encouraged to explore alternative methods that are robust to over-specification of latent dimensionality.

4.4. Selection of k

Seeing that an important goal of exploratory IFA is to determine the number of underlying factors, one often has to select the rank k empirically by cross-validation. To this end, a simple split-data scheme is considered: The observation set Ω is partitioned into a calibration subset $\Omega_{(c)}$ and a validation subset $\Omega_{(v)}$ such that $\Omega_{(c)} \cap \Omega_{(v)} = \emptyset$ and $\Omega_{(c)} \cup \Omega_{(v)} = \Omega$. The root mean squared cross-validation error

$$e(k; \mathbf{Y}, \Omega_{(c)}, \Omega_{(v)}) = |\Omega_{(v)}|^{-1/2} \left\| \left[\mathbf{Y} - \Psi(\hat{\Theta}_{(c)}) \right]_{\Omega_{(v)}} \right\|_F \quad (28)$$

is then used to identify the optimal k from a pre-determined grid, in which $\hat{\Theta}_{(c)}$ denotes the JML estimate obtained from the calibration sub-sample $\mathbf{Y}_{\Omega_{(c)}}$. If desired, one could also split the data into multiple disjoint subsets and perform a multifold cross-validation as suggested by Chen et al. (2018).

5. Simulation Study

Three Monte Carlo studies were conducted to evaluate the empirical performance of the proposed Riemannian optimization algorithm (Algorithm 1). Study 1 (Sect. 5.1) evaluates the comparative performance of the proposed algorithm and the competing AM algorithm (Chen et al. 2018) in computational efficiency. The next study (Sect. 5.2) concerns the impact of over- and under-fitting latent dimensionality, especially on the convergence of both algorithms (Sect. 4.4). Parameter recovery under various combinations of missing proportions and noise levels is examined in Study 3 (Sect. 5.3).

5.1. Study 1: Computational Cost

5.1.1. Simulation Setup In the first study, the number of respondents and items are fixed to $d_1 = 5000$ and $d_2 = 500$, respectively, and there is no missing data ($n = d_1 d_2$). 100 replications were performed under each of the seven latent dimensionality conditions: $k = 3, 5, \dots, 15$. In each replication, a distinct true parameter matrix Θ_0 was generated in steps resembling those of Chen et al. (2018, Sect. 4.1), which is briefly summarized here. The unnormalized true factor score matrix $\mathbf{U}_1 = (\mathbf{u}_{1,1}, \dots, \mathbf{u}_{1,d_1})^\top$ is constituted of independent and identically distributed (i.i.d.) $\mathcal{N}(0, 1)$ variates truncated such that $\|\mathbf{u}_{1,i}\| \leq 4\sqrt{k}$, $i = 1, \dots, d_1$. The item intercepts \mathbf{w}_1 were i.i.d. $\mathcal{U}(-2, 2)$ variates. Each item's slope parameters $\mathbf{v}_{1,j} \in \mathcal{R}^k$, $j = 1, \dots, d_2$, were determined in three steps: 1) Generate k i.i.d. $\mathcal{U}(-2, 2)$ variates, collectively denoted $\check{\mathbf{v}}_j$, 2) Generate a k -dimensional binary vector $\mathbf{p}_j \in \{0, 1\}^k \setminus \{\mathbf{0}_k, \mathbf{1}_k\}$ uniformly from its support, and 3) set $\mathbf{v}_{1,j} = \mathbf{p}_j \circ \check{\mathbf{v}}_j$, in which \circ denotes the element-wise product. It follows that $\mathbf{v}_{1,j}$ is sparse and contains at least one nonzero entry. Finally, compute the true parameter matrix $\Theta_0 = \bar{\mathbf{U}}_1 \bar{\mathbf{V}}_1^\top$, in which $\bar{\mathbf{U}}_1 = (\mathbf{1}_{d_1}, \mathbf{U}_1)$, $\bar{\mathbf{V}}_1 = (\mathbf{w}_1, \mathbf{V}_1)$, and $\mathbf{V}_1 = (\mathbf{v}_{1,1}, \dots, \mathbf{v}_{1,d_2})^\top$.

Algorithm 2 was implemented in the statistical computing environment R (R Core Team 2018); the tuning parameters were fixed to the default values outlined in Sect. 4. The source code is provided as supplementary material of the article. The R package `mirtjml`, which implements the AM algorithm (Chen et al. 2018) in C++, was used as a benchmark for comparison. For the AM algorithm, the bound on the person and item parameters was set to $M = 25k$ —the default configuration of the `mirtjml` package; the same value was used as the infinity-norm bound for the Riemannian optimization algorithm. For a fair comparison, the source code of `mirtjml` was modified to remove the analytic rotation step that involves extra computations; the parallel processing feature of the package was also disabled. Both algorithms were initiated with the same set of starting values determined by an SVD-based algorithm (Chen et al. 2018, Supplementary Material, Section F).

To obtain solutions at different numerical precision, three convergence tolerance levels were considered for each algorithm. For the Riemannian penalty method (Algorithm 1), set $\varepsilon = \mu = \delta = \tau = 10^{-2}$, 10^{-3} , and 10^{-4} . The AM algorithm declares convergence if a further iteration does not alter the log-likelihood function more than a pre-specified threshold $\varphi > 0$. A pilot study suggested that, corresponding to the three levels of ε specified for Algorithm 1, setting the tolerance levels $\varphi = 10^{-3}$, 10^{-5} , and 10^{-7} for the absolute log-likelihood change yielded comparable solutions.

5.1.2. Result A single core of an Intel® Xeon® E5-2683 v4 central processing unit (CPU) was used to execute both algorithms. The runtime and number of iterations⁸ until convergence are summarized in Fig. 3.

To attain similar numerical precision, the Riemannian optimization algorithm takes at least 78% less CPU time (0.66 difference in the based-10 logarithm of CPU time) compared to the AM algorithm in the majority of replications. For instance, the median runtime of the Riemannian algorithm ranges from 25.02 seconds for three-dimensional models to 43.41 seconds for 15-dimensional models using tolerance $\varepsilon = 10^{-2}$; in contrast, the median runtime for the AM algorithm ranges from 153.19 seconds for nine-dimensional models to 315.15 seconds for 15-dimensional models with $\varphi = 10^{-3}$. When the desired precision is high ($\varepsilon = 10^{-4}$ and $\varphi = 10^{-7}$), the Riemannian algorithm can be more than ten times faster than the AM algorithm. The computational cost of the Riemannian algorithm increases as the true dimensionality k grows; meanwhile, the AM algorithm converges the fastest when $k = 7$ and 9.

The AM algorithm also needs more iterations than the Riemannian algorithm does prior to meeting the same level of precision. With $\varepsilon = 10^{-2}$, for example, the median number of iterations

⁸For the Riemannian optimization algorithm, the total number of inner (Riemannian CG) iterations was reported.

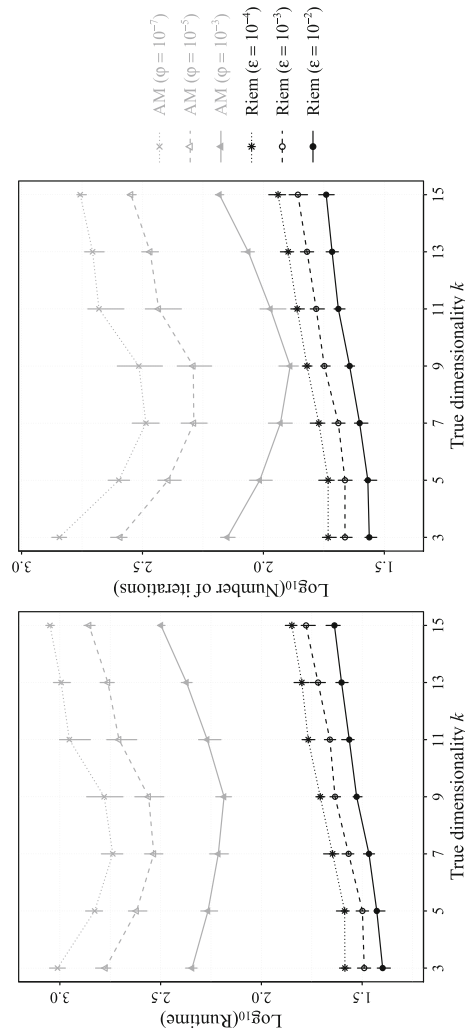


FIGURE 3.

Median computational time (in the base-10 logarithm scale; left panel) and median number of iterations completed (in the base-10 logarithm scale; right panel) for the Riemannian penalty method (Riem, black) and the alternating maximization algorithm (AM, gray). Results based on different convergence tolerances are shown in distinct line and symbol types. Vertical bars indicate interquartile ranges of summary statistics.

performed by the Riemannian algorithm ranges from 30 ($k = 3$) to 51 ($k = 15$); correspondingly with $\varphi = 10^{-3}$, the median number of iterations performed by the AM algorithm ranges from 77 ($k = 9$) to 152 ($k = 15$). The advantage of Riemannian optimization is more salient in the high-precision condition, a pattern resembling the comparison of CPU time. It is inferred that each Riemannian CG iteration traverses a longer distance toward the final solution than each gradient ascent step in the AM algorithm does.

5.1.3. Discussion It should be noted that the AM algorithm was granted a somewhat favorable position in the Monte Carlo experiment since its source code was written in C++. Therefore, it is safe to conclude that the proposed Riemannian algorithm accomplishes a better computational efficiency: It not only takes a shorter runtime on a single core but also fewer steps to reach convergence. However, one advantage of the AM algorithm is the ease of parallelization. In spite of its consumption of a longer CPU time, the AM algorithm may be faster than the Riemannian algorithm when running on a large parallel computing cluster.

5.2. Study 2: Dimensionality

5.2.1. Simulation Setup Study 2 examines the extent to which the computational speed and convergence of both the Riemannian and AM algorithms are affected by under- and over-fitting the latent dimensionality as well as whether the true dimensionality of latent factors can be identified by cross-validation (Sect. 4.4). The following design aspects are identical to Study 1: 1) the dimension of the data matrix $d_1 = 5000$ and $d_2 = 500$, 2) the true numbers of latent factors k ranging from 3 to 15 at an interval of 2, 3) no missing data, and 4) generating mechanism of the true parameter matrix Θ_0 .

On each set of simulated data, three IFA models with dimensionality $\tilde{k} = k - 2$, k , and $k + 2$ were fitted to a randomly selected calibration subset (90% of the observed entries). Respective convergence tolerances $\varepsilon = \mu = \tau = 10^{-2}$ and $\varphi = 10^{-3}$ were specified for the Riemannian penalty method and the AM algorithm. The cross-validation error (Eq. 28) was calculated from the validation subset (the remaining 10% data entries). The numbers of iterations were capped at 2000 for both algorithms, and the proportions of non-converging replications were recorded. The medians and interquartile ranges of CPU time and numbers of iterations until convergence or reaching the maximum 2000 were also reported.

5.2.2. Result Both algorithms never fail to converge within 2000 replications when the fitted dimensionality \tilde{k} is less than or equal to the true dimensionality k . In stark contrast, convergence rarely happens if the dimensionality is over-specified (see Table 1). In cases of over-fitting, the final iterates of the Riemannian algorithm are often close to the infinity-norm bound M , which is proportional to k ($M = 25k$). It is then anticipated that convergence becomes more difficult when k is large, because the iterates concentrate in the vicinity of the bound where the likelihood function tends to be flat. On the other hand, the AM algorithm converges slightly better for large k 's, which could be resulted from the fact that the row-wise L^2 -norm constraint is more restrictive.

The root mean squared cross-validation error (Eq. 28) is evaluated at the final iterate regardless of convergence status. It turns out that fitting the correct number of factors ($\tilde{k} = k$) always results in the smallest cross-validation error. In addition to the convergence rate, Table 1 also collects the median log-likelihood values, after dividing by a factor of $-n$, as well as the median root mean squared cross-validation error when $\tilde{k} = k + 2$. It is worth noting that the two algorithms reach different parameter estimates (at the final iterate) when $\tilde{k} = k + 2$: Solutions obtained from the AM algorithm are often associated with slightly less optimal log-likelihood but slightly better cross-validation error. The different constraints imposed by the two algorithms are again speculated as the cause of the pattern.

TABLE 1.

Percentages of convergence, log-likelihood (divided by $-n$), and root mean squared cross-validation error when over-fitting the latent dimensionality ($\tilde{k} = k + 2$).

Statistic	Method	True dimensionality k						
		3	5	7	9	11	13	15
Convergence	AM	2	1	0	1	0	12	13
percentage	Riem	11	2	0	0	0	0	0
Log-likelihood	AM	0.4321	0.3967	0.3687	0.3437	0.3216	0.3027	0.2856
(scaled by $-n$)	Riem	0.4318	0.3966	0.3685	0.3434	0.3215	0.3026	0.2853
Root mean squared	AM	0.4054	0.3901	0.3775	0.3666	0.3575	0.3501	0.3437
Cross-validation error	Riem	0.4057	0.3900	0.3776	0.3667	0.3575	0.3501	0.3438

n : Total number of observed entries. k : True dimensionality. \tilde{k} : Fitted dimensionality. AM: Alternating maximization. Riem: Riemannian penalty method.

Finally, the computational costs of the two algorithms are contrasted in Fig. 4. Under-fitting the latent dimensionality does not incur convergence issues but typically slows down the numerical search to some degree. Similar to what has been observed in Study 1, the Riemannian algorithm consumes less CPU time and numbers of iterations. The maximum number of iterations is reached in a vast majority of replications when the dimensionality is over-fitted; yet the Riemannian algorithm tends to finish 2000 iterations faster than the AM algorithm.

5.2.3. Discussion Study 2 reveals that over-specifying the number of factors leads to serious convergence problems no matter which algorithm is in use, which further implies that the optimization problem itself is likely to be poorly conditioned. The split-data cross-validation procedure is able to identify the correct dimensionality in the setup of Study 2; however, Chen et al. (2018) warned that the efficacy of the cross-validation procedure may drop when the dimension of the data matrix is small.

5.3. Study 3: Parameter Recovery

5.3.1. Simulation Setup The design of Study 3 consists of three fully crossed factors: the true latent dimensionality, the size of the data matrix, and the proportion of missing data. 100 data sets were simulated under each condition. The true numbers of latent factors k were set to 2, 5, and 8. The number of items d_2 ranges from 200 to 600 at an interval of 100, and the number of respondents $d_1 = 10d_2$. Observed response entries were generated based on the Bernoulli observation model with $n = \nu d_1 d_2$, where $\nu = 0.5, 0.75$, and 1; $1 - \nu$ gives the expected proportion of missing data.

As suggested by a referee, more extreme item parameter values were considered in Study 3: The generating mechanism of $\tilde{\mathbf{V}}_1$ remains the same as was described in Study 1 (Sect. 5.1), with the exception that $\mathcal{U}(-3, 3)$ was used in place of $\mathcal{U}(-2, 2)$. Unnormalized factor scores \mathbf{U}_1 were sampled based on a multilevel model (e.g., Fox 2005). The (i, l) th entry of \mathbf{U}_1 , denoted $u_{1,il}$, $i = 1, \dots, d_1$, $l = 1, \dots, k$, is subject to the decomposition $u_{1,il} = \chi_{c(i),l} + \iota_{il}$, in which $\chi_{c(i),l}$ and ι_{il} are cluster- and individual-level scores at dimension l . Every 50 individuals form a cluster, indexed by $c(i) = \lfloor (i - 1)/50 \rfloor + 1$, in which $\lfloor x \rfloor$ denotes the greatest integer that is less than or equal to x . At each level, the factor scores are i.i.d. normal: In particular, $\chi_{c(i),l} \sim \mathcal{N}(0, 0.3)$ and $\iota_{il} \sim \mathcal{N}(0, 0.7)$ for all i and l , resulting in an intraclass correlation of 0.3. Only random draws satisfying $\|\Theta_0\|_\infty \leq 50$ are kept.

The default tuning of Algorithm 1 was adopted with $\varepsilon = \mu = \tau = 10^{-3}$; similar to Study 2, the maximum number of iterations was set to 2000. Convergence rate was recorded under

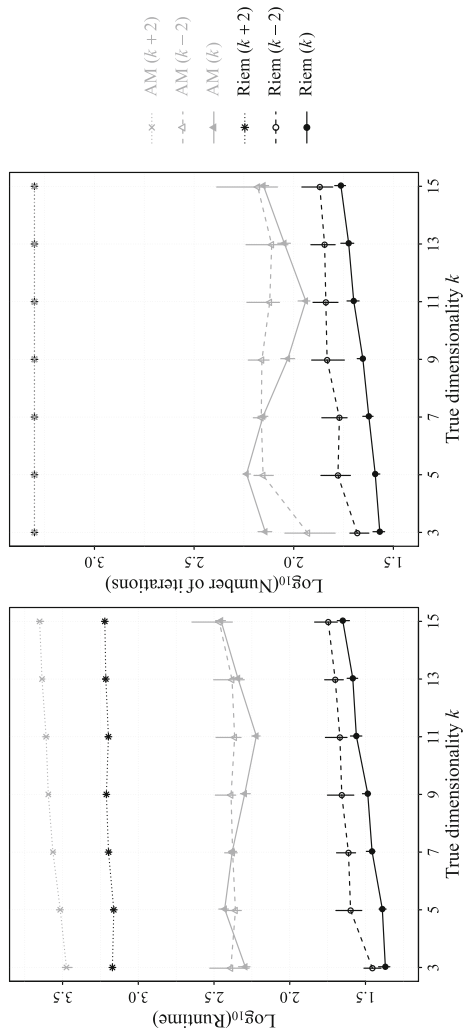


FIGURE 4.

Median computational time (in the base-10 logarithm scale; left panel) and median number of iterations completed (in the base-10 logarithm scale; right panel) for the Riemannian penalty method (Riem, black) and the alternating maximization algorithm (AM, gray). Results for correctly fitting ($\tilde{k} = k$), under-fitting ($\tilde{k} = k - 2$), and over-fitting ($\tilde{k} = k + 2$) the latent dimensionality are shown in distinct line and symbol types. Vertical bars indicate interquartile ranges of summary statistics.

TABLE 2.
Percentages of convergence for the Riemannian optimization algorithm in Study 3

True k	Missing %	Number of items d_2				
		200	300	400	500	600
2	50	67	98	100	100	100
	25	98	100	100	100	100
	0	100	100	100	100	100
5	50	0	51	100	100	100
	25	55	100	100	100	100
	0	98	100	100	100	100
8	50	0	0	6	75	96
	25	0	49	98	100	100
	0	21	100	100	100	100

Numbers less than 50 are highlighted in bold.

each simulation condition. Overall successfulness of parameter estimation was gauged by the relative Frobenius-norm error $\|\hat{\Theta} - \Theta_0\|_F / \|\Theta_0\|_F$, i.e., the overall magnitude of estimation error normalized by the size of the true parameter matrix. It is also of interest to investigate the accuracy in recovering the true item parameters $\tilde{\mathbf{V}}_0$ and factor scores $\tilde{\mathbf{U}}_0$, in which $\tilde{\mathbf{U}}_0$ is obtained by normalizing each column of $\tilde{\mathbf{U}}_1$, and $\tilde{\mathbf{V}}_0$ by rescaling $\tilde{\mathbf{V}}_1$ so that $\Theta_0 = \tilde{\mathbf{U}}_0 \tilde{\mathbf{V}}_0^\top = \tilde{\mathbf{U}}_1 \tilde{\mathbf{V}}_1^\top$. In this regards, the relative Frobenius-norm error $\|\tilde{\mathbf{V}}\tilde{\mathbf{Q}} - \tilde{\mathbf{V}}_0\|_F / \|\tilde{\mathbf{V}}_0\|_F$ and $\|\tilde{\mathbf{U}}\tilde{\mathbf{Q}}^{-\top} - \tilde{\mathbf{U}}_0\|_F / \|\tilde{\mathbf{U}}_0\|_F$ are computed, in which $\tilde{\mathbf{Q}}$ is defined by Eq. A2. It is also remarked that JML estimation does not produce estimates for cluster-level factor scores, which differs from scoring after fitting a multilevel IRT model using MML.

5.3.2. Result The convergence rates of the proposed algorithm under various simulation conditions are tabulated in Table 2. It is concluded that non-convergence is more likely to happen when the latent dimensionality is high, the proportion of missingness is high, and/or the size of the data matrix is small. Regardless of the data size, convergence occurs more than 50% of the time when $k = 2$. As k increases to 8, attaining convergence in a majority of replications requires $d_2 \geq 300$ when there is no missing data, $d_2 \geq 400$ when there are 25% missing data, and $d_2 \geq 500$ when there are 50% missing data.

It is observed from Fig. 5 that the recovery of the true parameter matrix Θ_0 was also impaired as the number of factors increases. On the other hand, increasing the dimension of the data matrix or decreasing the proportion of missing data yields better estimates, which parallels the asymptotic theory. Take $\text{median}(\|\hat{\Theta} - \Theta_0\|_F / \|\Theta_0\|_F) = 15\%$ as a benchmark for satisfactory parameter recovery. For $k = 2$, the desired accuracy is attained whenever $d_2 \geq 300$ without missing data, $d_2 \geq 400$ with 25% missing data, and $d_2 \geq 500$ with 50% missing data. As k increases to 8, the benchmark is barely met except when $d_2 = 600$ with no missing data.

Figures 6 and 7 display the median relative estimation error for item parameters and factor scores, respectively: Both plots exhibit a pattern analogous to that of Fig. 5. It is also observed that, given the current setup that the number of respondents is ten times the number of items ($d_1 = 10d_2$), the item parameters are better estimated compared to the factor scores.

5.3.3. Discussion Accuracy of the JML estimator is influenced by all the factors being manipulated in Study 3. For each fixed k , overall parameter recovery is improved as the dimension of the data matrix and the proportion of observed entries increases. It is also remarked that the reported estimation error for item and person parameters should be treated as a lower bound as the optimal

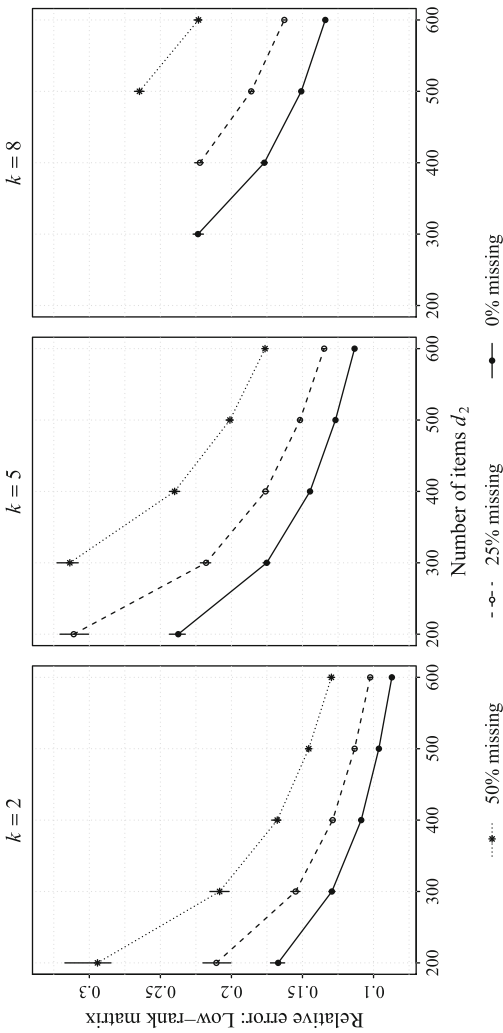


FIGURE 5.

Median relative error in the low-rank parameter matrix. Results for three dimensionality conditions ($k = 2, 5$, and 8) are shown in separate panels. Line and symbol types indicate expected proportions of missing entries (dotted line + asterisk = 50% , dashed line + circle = 75% , and solid line + dot = 0%). Summary statistics are not shown when the convergence rate is lower than 50% .

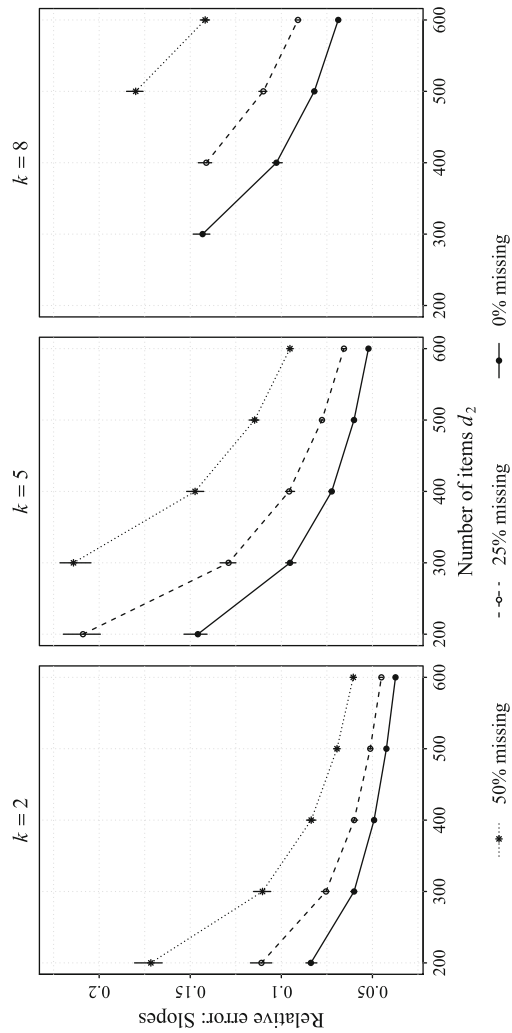


FIGURE 6. Median relative error in item parameters. Results for three dimensionality conditions ($k = 2, 5$, and 8) are shown in separate panels. Line and symbol types indicate expected proportions of missing entries (dotted line + asterisk = 50%, dashed line + circle = 75%, and solid line + dot = 0%). Summary statistics are not shown when the convergence rate is lower than 50%.

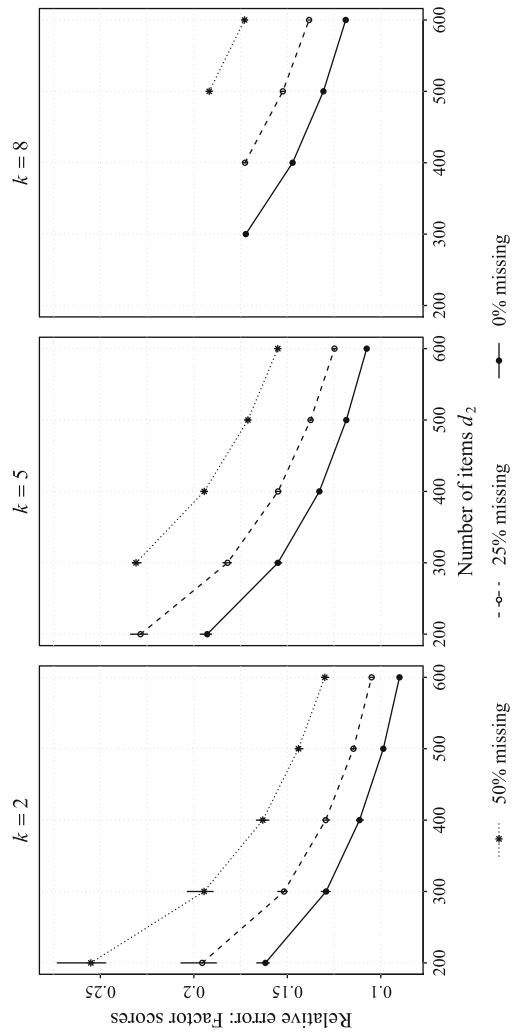


FIGURE 7.
Median relative error in person parameters. Results for three dimensionality conditions ($k = 2, 5$, and 8) are shown in separate panels. Line and symbol types indicate expected proportions of missing entries (dotted line + asterisk = 50%, dashed line + circle = 75%, and solid line + dot = 0%). Summary statistics are not shown when the convergence rate is lower than 50%.

rotation matrix (Eq. A2) is used. In practice, analytic rotation must be performed to minimize a suitable criterion function describing the complexity (or lack of sparsity) of the rotated solution; readers are referred to Browne (2001) for a comprehensive survey of the topic.

6. Discussion

The current paper recasts JML estimation of exploratory IFA as a manifold optimization problem and proposes an efficient Riemannian penalty method (Algorithm 1) with a Riemannian subsolver for parameter estimation. Monte Carlo experiments suggest that the proposed algorithm is superior in computation speed to the existing AM algorithm, which was proposed earlier for essentially the same purpose (Chen et al. 2018). It is also illustrated with simulated data that the parameter recovery performance is contingent upon the size of the data matrix and the proportion of missingness. When no *a priori* information about the number of latent factors is available, which is common in real applications of exploratory IFA, a simple split-data cross-validation procedure suffices to correctly identify the true dimensionality. However, over-fitting the number of factors almost always results in non-convergence.

There are several limitations and extensions to be addressed by future research.

First, Study 2 in the present work suggests that the optimization problem is poorly conditioned when the fitted dimensionality exceeds the true dimensionality. A systematic investigation on convergence is needed to better understand the phenomenon. As the iterations often terminate nearby the infinity-norm boundary in the case of over-fitting, it is worth investigating whether alternative methods for constrained manifold optimization, such as sub-gradient-based methods (e.g., Borckmans et al. 2014), can be more efficient when the solution falls exactly on the boundary. In addition, algorithms with known global convergence properties, such as the Riemannian trust region approach (Absil et al. 2008, Sect. 7.2), may serve as the sub-solver in place of the Riemannian CG algorithm. The trust region approach draws on more involved second-order geometry of the manifold and demands more careful tuning.

Second, while consistency guarantees the quality of point estimation when the dimension of the data matrix is sufficiently large, it is often desirable to report confidence regions for model parameters to quantify sampling error and better answer the question of “how large is large enough.” Characterizing the sampling distribution of the JML estimator in the high-dimensional asymptotic setting proves to be challenging. Conventional proofs of asymptotic normality count on the assumption that the sample size grows at a much faster rate compared to the dimension of the parameter space (e.g., Fan and Peng 2004), which unfortunately does not apply to JML estimation of IFA. Recently, concentration results for empirical processes have been applied to construct honest and adaptive confidence regions in the context of matrix completion with noisy continuous data (Carpentier et al. 2018). It is conjectured that a similar construction is possible for one-bit matrix completion and exploratory IFA.

Third, the proposed Riemannian optimization algorithm can be modified to accommodate continuous, polytomous, count, or mixed-format response data: The corresponding log-likelihood function of each data entry can be substituted for the Bernoulli log-likelihood in Eq. 6. The flexibility and efficiency of JML estimation may facilitate the use of collateral information, such as physiological measures and response times, in large-scale assessment in order to generate a more comprehensive profile of test taking behavior.

Fourth, efficient estimation algorithms of confirmatory IFA, which involves additional constraints on the item slope matrix and is suitable for validating a fully specified factor structure driven by substantive theory, are in demand to handle large and complex testing/survey data. Pioneering work on the estimability and identifiability of confirmatory IFA models under JML can be found in Chen et al. (2019). Because the parameter space of a confirmatory model is

essentially a constrained fixed-rank matrix manifold, and it is conjectured that an extension of the proposed Riemannian optimization technique can be derived for the confirmatory IFA setting using sub-manifold geometry.

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

Appendix A: Proofs

A.1 Proof of Proposition 1

Let $\bar{\mathbf{U}}_0 = \bar{\mathbf{U}}_{0*} \bar{\mathbf{R}}_{0u}$ be the QR factorization of the true person parameter matrix: $\bar{\mathbf{U}}_{0*} = (\mathbf{1}_{d_1}/\sqrt{d_1}, \mathbf{U}_{0*})^\top$ is orthonormal, i.e., $\bar{\mathbf{U}}_{0*}^\top \bar{\mathbf{U}}_{0*} = \mathbf{I}_{k+1}$, and $\bar{\mathbf{R}}_{0u} = (\mathbf{e}_{k+1}, \mathbf{R}_{0u})$ is upper-triangular with normalized columns by Assumption ii). The block matrix inversion formula (Magnus and Neudecker 1999, p. 12) implies that $\bar{\mathbf{R}}_{0u}^{-1} = (\mathbf{e}_{k+1}^-, \mathbf{R}_{0u}^-)$, i.e., the first column remains to be \mathbf{e}_{k+1} . By Lemma C.1 of Chen et al. (2018), there exists constants $C_1, C_2 > 0$ and a $(k+1) \times (k+1)$ orthogonal rotation matrix of form $\bar{\mathbf{Q}}_* = \mathbf{1} \oplus \mathbf{Q}_*$, where \oplus denotes the (matrix) direct sum and $\mathbf{Q}_*^\top \mathbf{Q}_* = \mathbf{I}_k$, such that

$$\frac{\|\tilde{\mathbf{V}}\bar{\mathbf{Q}}_* - \bar{\mathbf{V}}_0 \bar{\mathbf{R}}_{0u}^\top\|_F^2}{d_1 d_2} \leq C_1 \sqrt{\frac{d_1 + d_2}{n}} \quad (\text{A1})$$

with probability greater than $1 - C_2/(d_1 + d_2)$. Now, let

$$\bar{\mathbf{Q}} = \bar{\mathbf{Q}}_* \bar{\mathbf{R}}_{0u}^{-\top}. \quad (\text{A2})$$

Note that the first row of $\bar{\mathbf{Q}}$ is \mathbf{e}_{k+1}^\top due to the partitioned structure of $\bar{\mathbf{Q}}_*$ and $\bar{\mathbf{R}}_{0u}^{-1}$. $\bar{\mathbf{Q}}$ is indeed an oblique rotation matrix, because $\bar{\mathbf{Q}}^{-1} = \bar{\mathbf{R}}_{0u}^\top \bar{\mathbf{Q}}_*^\top$, and thus $\text{diag}(\bar{\mathbf{Q}}^{-1} \bar{\mathbf{Q}}^{-\top}) = \text{diag}(\bar{\mathbf{R}}_{0u}^\top \bar{\mathbf{R}}_{0u}) = \mathbf{1}_{k+1}$. Finally, the Cauchy–Schwarz inequality implies that

$$\|\tilde{\mathbf{V}}\bar{\mathbf{Q}} - \bar{\mathbf{V}}_0\|_F^2 = \|(\tilde{\mathbf{V}}\bar{\mathbf{Q}}_* - \bar{\mathbf{V}}_0 \bar{\mathbf{R}}_{0u}^\top) \bar{\mathbf{R}}_{0u}^{-\top}\|_F^2 \leq \|\tilde{\mathbf{V}}\bar{\mathbf{Q}}_* - \bar{\mathbf{V}}_0 \bar{\mathbf{R}}_{0u}^\top\|_F^2 \|\bar{\mathbf{R}}_{0u}^{-\top}\|_F^2. \quad (\text{A3})$$

As $\bar{\mathbf{R}}_{0u}$ and $\bar{\mathbf{U}}_0$ share the same set of singular values, $\|\bar{\mathbf{R}}_{0u}^{-\top}\|_F^2 \leq (k+1)/\sigma_{k+1}^2(\bar{\mathbf{U}}_0) \leq (k+1)/c_1^2$ by Assumption iii. Equation 11 then follows from Eqs. A1 and A3.

To establish Eq. 12, notice that

$$\|\tilde{\mathbf{U}}\bar{\mathbf{Q}}^{-\top} - \bar{\mathbf{U}}_0\|_F^2 = \|(\tilde{\mathbf{U}}\bar{\mathbf{Q}}_* - \bar{\mathbf{U}}_{0*}) \bar{\mathbf{R}}_{0u}\|_F^2 \leq \|\tilde{\mathbf{U}}\bar{\mathbf{Q}}_* - \bar{\mathbf{U}}_{0*}\|_F^2 \|\bar{\mathbf{R}}_{0u}\|_F^2. \quad (\text{A4})$$

Because $\bar{\mathbf{R}}_{0u}$ is column-wise normalized, $\|\bar{\mathbf{R}}_{0u}\|_F^2 = k+1$. The remaining task is to bound $\|\tilde{\mathbf{U}}\bar{\mathbf{Q}}_* - \bar{\mathbf{U}}_{0*}\|_F^2$; because the leading columns of $\tilde{\mathbf{U}}$ and $\bar{\mathbf{U}}_{0*}$ are identical, it further suffices to bound $\|\hat{\mathbf{U}}_* \mathbf{Q}_* - \mathbf{U}_{0*}\|_F^2$. By the choice of $\mathbf{Q}_* = \hat{\mathbf{U}}_*^\top \mathbf{U}_{0*} (\mathbf{U}_{0*}^\top \hat{\mathbf{U}}_* \hat{\mathbf{U}}_*^\top \mathbf{U}_{0*})^{-1/2}$ in Chen et al. (2018, Eq. C.12),

$$\begin{aligned} \|\hat{\mathbf{U}}_* \mathbf{Q}_* - \mathbf{U}_{0*}\|_F^2 &= \|\hat{\mathbf{U}}_* \mathbf{Q}_*\|_F^2 + \|\mathbf{U}_{0*}\|_F^2 - 2\text{tr}(\mathbf{Q}_*^\top \hat{\mathbf{U}}_*^\top \mathbf{U}_{0*}) = 2 \left[k - \sum_{l=1}^k \sigma_l(\hat{\mathbf{U}}_*^\top \mathbf{U}_{0*}) \right] \\ &\leq 2 \left[k - \sum_{l=1}^k \sigma_l^2(\hat{\mathbf{U}}_*^\top \mathbf{U}_{0*}) \right] = 2 \sum_{l=1}^k \sin^2 \angle_l(\hat{\mathbf{U}}_*, \mathbf{U}_{0*}), \end{aligned} \quad (\text{A5})$$

in which $\angle_l(\hat{\mathbf{U}}_*, \mathbf{U}_{0*})$, $l = 1, \dots, k$, denotes the principal angles between $\text{span}(\hat{\mathbf{U}}_*)$ and $\text{span}(\mathbf{U}_{0*})$, and the inequality follows from the fact that $\sigma_l(\hat{\mathbf{U}}_*^\top \mathbf{U}_{0*}) = \cos \angle_l(\hat{\mathbf{U}}_*, \mathbf{U}_{0*})$ (Björck and Golub 1973). The right-hand side of Eq. A5 converges to 0 in P_{Θ_0} -probability by Equation C.10 in Chen et al. (2018). The proof is now complete.

A.2 Proof of Proposition 2

Let $\Theta \in \mathcal{M}_k(d_1, d_2)$ and $\gamma : \mathcal{R} \rightarrow \mathcal{M}_k(d_1, d_2)$ be a smooth curve such that $\gamma(0) = \Theta$. There exists $\mathbf{w}(t) \in \mathcal{R}^{d_2}$, $\mathbf{U}(t) \in \mathcal{R}_*^{d_1 \times k}$, and $\mathbf{V}(t) \in \mathcal{R}_*^{d_2 \times k}$ such that

$$\gamma(t) = \mathbf{1}_{d_1} \mathbf{w}(t)^\top + \mathbf{U}(t) \mathbf{V}(t)^\top \quad (\text{A6})$$

for t in some neighborhood of 0. Differentiating Eq. A6 with respect to t and evaluating at $t = 0$ yield

$$\dot{\gamma}(0) = \mathbf{1}_{d_1} \dot{\mathbf{w}}(0)^\top + \dot{\mathbf{U}}(0) \mathbf{V}(0)^\top + \mathbf{U}(0) \dot{\mathbf{V}}(0)^\top. \quad (\text{A7})$$

Given a choice of orthonormal basis matrices $(\mathbf{1}_{d_1}/\sqrt{d_1}, \mathbf{U}_*)$ and \mathbf{V}_* corresponding to $(\mathbf{1}_{d_1}, \mathbf{U}(0))$ and $\mathbf{V}(0)$, there exist fixed $\mathbf{m} \in \mathcal{R}^k$ and $\mathbf{M}, \mathbf{N} \in \mathcal{R}_*^{k \times k}$ such that

$$\mathbf{U}(0) = \mathbf{1}_{d_1} \mathbf{m}^\top + \mathbf{U}_* \mathbf{M}, \text{ and } \mathbf{V}(0) = \mathbf{V}_* \mathbf{N}. \quad (\text{A8})$$

Because $\mathbf{1}_{d_1}$ is perpendicular to \mathbf{U}_* , it is possible to select $\mathbf{U}_\perp = (\mathbf{1}_{d_1}/\sqrt{d_1}, \mathbf{U}_\dagger)$ where $\mathbf{U}_\dagger^\top \mathbf{1}_{d_1} = \mathbf{0}_{d_1-k-1}$, $\mathbf{U}_\dagger^\top \mathbf{U}_* = \mathbf{0}_{(d_1-k-1) \times k}$, and $\mathbf{U}_\dagger^\top \mathbf{U}_\dagger = \mathbf{I}_{d_1-k-1}$. The vector/matrix derivatives in Eq. A7 can then be expanded on the orthonormal bases:

$$\dot{\mathbf{w}}(0) = \mathbf{V}_* \mathbf{a}_1 + \mathbf{V}_\perp \mathbf{a}_2, \quad \dot{\mathbf{U}}(0) = \mathbf{U}_* \mathbf{B}_1 + \mathbf{1}_{d_1} \mathbf{b}_2^\top + \mathbf{U}_\dagger \mathbf{B}_3, \text{ and } \dot{\mathbf{V}}(0) = \mathbf{V}_* \mathbf{C}_1 + \mathbf{V}_\perp \mathbf{C}_2, \quad (\text{A9})$$

in which $\mathbf{a}_1, \mathbf{b}_2 \in \mathcal{R}^k$, $\mathbf{a}_2 \in \mathcal{R}^{d_2-k}$, $\mathbf{B}_1, \mathbf{C}_1 \in \mathcal{R}^{k \times k}$, $\mathbf{B}_3 \in \mathcal{R}^{(d_1-k-1) \times k}$, and $\mathbf{C}_2 \in \mathcal{R}^{(d_2-k) \times k}$. Plugging Eqs. A8 and A9 into Eq. A7 gives

$$\begin{aligned} \dot{\gamma}(0) &= \mathbf{1}_{d_1} (\mathbf{a}_1^\top \mathbf{V}_*^\top + \mathbf{a}_2^\top \mathbf{V}_\perp^\top) + (\mathbf{U}_* \mathbf{B}_1 + \mathbf{1}_{d_1} \mathbf{b}_2^\top + \mathbf{U}_\dagger \mathbf{B}_3) \mathbf{N}^\top \mathbf{V}_*^\top \\ &\quad + (\mathbf{1}_{d_1} \mathbf{m}^\top + \mathbf{U}_* \mathbf{M}) (\mathbf{C}_1^\top \mathbf{V}_*^\top + \mathbf{C}_2^\top \mathbf{V}_\perp^\top) \end{aligned} \quad (\text{A10})$$

which reduces to Eq. 13 upon identifying $\mathbf{a} = \sqrt{d_1}(\mathbf{a}_2 + \mathbf{C}_2 \mathbf{m})$, $\mathbf{B} = \mathbf{B}_1 \mathbf{N}^\top + \mathbf{M} \mathbf{C}_1^\top$, $\mathbf{C} = (\mathbf{a}_1^\top + \mathbf{b}_2^\top \mathbf{N}^\top + \mathbf{m}^\top \mathbf{C}_1^\top, \mathbf{B}_3 \mathbf{N}^\top)$, and $\mathbf{D} = \mathbf{C}_2 \mathbf{M}^\top$. Therefore, every tangent vector can be expressed as a member of $\mathcal{T}_\Theta \mathcal{M}_k(d_1, d_2)$ (Eq. 13). Conversely, let Ξ be a member of Eq. 13. Equation A14 in ‘‘Appendix A.4’’ implies that the curve $\gamma : t \mapsto R_\Theta(t \Xi)$ passes through Θ at $t = 0$ and $\dot{\gamma}(0) = \Xi$, so Eq. 13 is a subset of the tangent space. In conclusion, Eq. 13 gives a representation of the tangent space of $\mathcal{M}_k(d_1, d_2)$ at Θ .

A.3 Proof of Proposition 3

It suffices to verify that $\langle \mathbf{G} - \Xi, \Xi \rangle = 0$. Note that $\mathbf{I}_{d_1} = \mathbf{U}_* \mathbf{U}_*^\top + (\mathbf{I}_{d_1} - \mathbf{U}_* \mathbf{U}_*^\top) = \mathbf{U}_* \mathbf{U}_*^\top + \mathbf{U}_\perp \mathbf{U}_\perp^\top$ and $\mathbf{I}_{d_2} = \mathbf{V}_* \mathbf{V}_*^\top + (\mathbf{I}_{d_2} - \mathbf{V}_* \mathbf{V}_*^\top) = \mathbf{V}_* \mathbf{V}_*^\top + \mathbf{V}_\perp \mathbf{V}_\perp^\top$, which admits the following decomposition of $\mathbf{G} \in \mathcal{R}^{d_1 \times d_2}$:

$$\mathbf{G} = \mathbf{U}_* \mathbf{U}_*^\top \mathbf{G} \mathbf{V}_* \mathbf{V}_*^\top + \mathbf{U}_* \mathbf{U}_*^\top \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top + \mathbf{U}_\perp \mathbf{U}_\perp^\top \mathbf{G} \mathbf{V}_* \mathbf{V}_*^\top + \mathbf{U}_\perp \mathbf{U}_\perp^\top \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top. \quad (\text{A11})$$

Partition $\mathbf{U}_\perp = (\mathbf{1}_{d_1}/\sqrt{d_1}, \mathbf{U}_\dagger)$ as in Sect. A.2. The orthogonal decomposition

$$\mathbf{U}_\perp \mathbf{U}_\perp^\top \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top = \frac{\mathbf{1}_{d_1} \mathbf{1}_{d_1}^\top \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top}{d_1} + \mathbf{U}_\dagger^\top \mathbf{U}_\dagger \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top \quad (\text{A12})$$

implies that $\mathbf{G} - \mathbf{\Xi} = \mathbf{U}_\dagger^\top \mathbf{U}_\dagger \mathbf{G} \mathbf{V}_\perp \mathbf{V}_\perp^\top$; therefore, $\langle \mathbf{G} - \mathbf{\Xi}, \mathbf{\Xi} \rangle = 0$.

A.4 Proof of Proposition 4

Expand and rearrange the right-hand side of Eq. 17:

$$\begin{aligned} R_\Theta(\mathbf{\Xi}) &= \left[\frac{\mathbf{1}_{d_1} \mathbf{w}_*^\top}{d_1} + \mathbf{U}_* \mathbf{R}^\top \mathbf{V}_*^\top \right] + \left[\frac{\mathbf{1}_{d_1} \mathbf{a}^\top \mathbf{V}_\perp^\top}{\sqrt{d_1}} + \mathbf{U}_* \mathbf{B} \mathbf{V}_*^\top + \mathbf{U}_\perp \mathbf{C} \mathbf{V}_*^\top + \mathbf{U}_* \mathbf{D}^\top \mathbf{V}_\perp^\top \right] \\ &\quad + (\mathbf{U}_* \mathbf{B} + \mathbf{U}_\perp \mathbf{C}) \mathbf{R}^{-\top} \mathbf{D}^\top \mathbf{V}_\perp^\top \\ &= \mathbf{\Theta} + \mathbf{\Xi} + (\mathbf{U}_* \mathbf{B} + \mathbf{U}_\perp \mathbf{C}) \mathbf{R}^{-\top} \mathbf{D}^\top \mathbf{V}_\perp^\top. \end{aligned} \quad (\text{A13})$$

For s belonging to some neighborhood of 0, it follows that

$$R_\Theta(s\mathbf{\Xi}) = \mathbf{\Theta} + s\mathbf{\Xi} + s^2(\mathbf{U}_* \mathbf{B} + \mathbf{U}_\perp \mathbf{C}) \mathbf{R}^{-\top} \mathbf{D}^\top \mathbf{V}_\perp^\top. \quad (\text{A14})$$

The centering condition (Eq. 15) follows by setting $s = 0$ in Eq. A14. Equation A14 also suggests that $R_\Theta(s\mathbf{\Xi})$ is quadratic in s ; therefore, differentiating with respect to s and evaluating at $s = 0$ yield the local rigidity condition (Eq. 16). It is then concluded that $R_\Theta(\mathbf{\Xi})$ is a valid retraction.

References

- Absil, P.-A., Mahony, R., & Sepulchre, R. (2008). *Optimization algorithms on matrix manifolds*. Princeton: Princeton University Press.
- Absil, P.-A., & Mallick, J. (2012). Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1), 135–158.
- Andersen, E. B. (1970). Asymptotic properties of conditional maximum-likelihood estimators. *Journal of the Royal Statistical Society: Series B (Methodological)*, 32(2), 283–301.
- Baker, F. B., & Kim, S.-H. (2004). *Item response theory: Parameter estimation techniques*. Boca Raton: CRC Press.
- Bartholomew, D. J., Steele, F., Galbraith, J., & Moustaki, I. (2008). *Analysis of multivariate social science data*. Boca Raton: CRC Press.
- Bertsekas, D. P. (1999). *Nonlinear programming*. Belmont: Athena Scientific.
- Björck, A., & Golub, G. H. (1973). Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123), 579–594.
- Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, 46(4), 443–459.
- Bock, R. D., & Lieberman, M. (1970). Fitting a response model for n dichotomously scored items. *Psychometrika*, 35(2), 179–197.
- Borckmans, P. B., Selvan, S. E., Boumal, N., & Absil, P.-A. (2014). A Riemannian subgradient algorithm for economic dispatch with valve-point effect. *Journal of Computational and Applied Mathematics*, 255, 848–866.
- Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research*, 36(1), 111–150.
- Cai, L. (2010). High-dimensional exploratory item factor analysis by a Metropolis–Hastings Robbins–Monro algorithm. *Psychometrika*, 75(1), 33–57.
- Cai, T., & Zhou, W. X. (2013). A max-norm constrained minimization approach to 1-bit matrix completion. *The Journal of Machine Learning Research*, 14(1), 3619–3647.
- Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6), 717–772.
- Carpentier, A., Klopp, O., Löffler, M., & Nickl, R. (2018). Adaptive confidence sets for matrix completion. *Bernoulli*, 24(4), 2429–2460.

- Chen, Y., Li, X., & Zhang, S. (2018). Joint maximum likelihood estimation for high-dimensional exploratory item factor analysis. *Psychometrika* (Advance Online Publication). <https://doi.org/10.1007/s11336-018-9646-5>.
- Chen, Y., Li, X., & Zhang, S. (2019). Structured latent factor analysis for large-scale data: Identifiability, estimability, and their implications. *Journal of the American Statistical Association*. (Advance Online Publication) <https://doi.org/10.1080/01621459.2019.1635485>
- Curran, P. J., & Hussong, A. M. (2009). Integrative data analysis: The simultaneous analysis of multiple data sets. *Psychological Methods*, 142, 81–100.
- Davenport, M. A., Plan, Y., Van Den Berg, E., & Wootters, M. (2014). 1-bit matrix completion. *Information and Inference: A Journal of the IMA*, 33, 189–223.
- de Leeuw, J. (2006). Principal component analysis of binary data by iterated singular value decomposition. *Computational Statistics & Data Analysis*, 501, 21–39.
- Fan, J., Gong, W., & Zhu, Z. (2019). Generalized high-dimensional trace regression via nuclear norm regularization. *Journal of Econometrics*, 212(1), 177–202.
- Fan, J., & Peng, H. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 323, 928–961.
- Fox, J.-P. (2005). Multilevel IRT using dichotomous and polytomous response data. *British Journal of Mathematical and Statistical Psychology*, 581, 145–172.
- Golub, G., & Van Loan, C. (2013). *Matrix computations* (4th ed.). Baltimore: Johns Hopkins University Press.
- Haberman, S. J. (2006). *Adaptive quadrature for item response models* (Tech. Rep. No. RR-06-29). Princeton: ETS.
- Hofer, S. M., & Piccinin, A. M. (2009). Integrative data analysis through coordination of measurement and analysis protocol across independent longitudinal studies. *Psychological Methods*, 142, 150.
- Huang, W., Gallivan, K. A., & Absil, P.-A. (2015). A Broyden class of quasi-newton methods for Riemannian optimization. *SIAM Journal on Optimization*, 253, 1660–1685.
- Jeon, M., Kaufman, C., & Rabe-Hesketh, S. (2019). Monte Carlo local likelihood approximation. *Biostatistics*, 201, 164–179.
- Klopp, O. (2015). Matrix completion by singular value thresholding: Sharp bounds. *Electronic Journal of Statistics*, 92, 2348–2369.
- Klopp, O., Lafond, J., Moulines, É., & Salmon, J. (2015). Adaptive multinomial matrix completion. *Electronic Journal of Statistics*, 92, 2950–2975.
- Koopmans, T. C., & Reiersøl, O. (1950). The identification of structural characteristics. *The Annals of Mathematical Statistics*, 212, 165–181.
- Liu, C., & Boumal, N. (2019). Simple algorithms for optimization on Riemannian manifolds with constraints. *Applied Mathematics & Optimization*, <https://doi.org/10.1007/s00245-019-09564-3>.
- Liu, Y., Magnus, B., Quinn, H., & Thissen, D. (2018). Multidimensional item response theory. In D. Hughes, P. Irwing, & T. Booth (Eds.), *Handbook of psychometric testing*. Hoboken: Wiley.
- Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Mahwah: Routledge.
- Magnus, J., & Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*. New York: Wiley.
- McDonald, R. P. (1981). The dimensionality of tests and items. *British Journal of Mathematical and Statistical Psychology*, 341, 100–117.
- Monroe, S., & Cai, L. (2014). Estimation of a Ramsay-curve item response theory model by the Metropolis–Hastings Robbins–Monro algorithm. *Educational and Psychological Measurement*, 742, 343–369.
- Neyman, J., & Scott, E. L. (1948). Consistent estimates based on partially consistent observations. *Econometrica: Journal of the Econometric Society*, 16(1), 1–32.
- O’Rourke, S., Vu, V., & Wang, K. (2018). Random perturbation of low rank matrices: Improving classical bounds. *Linear Algebra and its Applications*, 540, 26–59.
- Pinar, M. Ç., & Zenios, S. A. (1994). On smoothing exact penalty functions for convex constrained optimization. *SIAM Journal on Optimization*, 43, 486–511.
- Polak, E., & Ribière, G. (1969). Note sur la convergence de méthodes de directions conjuguées. *Revue Française d’Informatique et de Recherche Opérationnelle. Série Rouge*, 316, 35–43.
- R Core Team. (2018). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. <https://www.R-project.org/>
- Reckase, M. (2009). *Multidimensional item response theory*. New York: Springer.
- Revelle, W., Wilt, J., & Rosenthal, A. (2010). Individual differences in cognition: New methods for examining the personality-cognition link. In A. Gruszka, G. Matthews, & B. Szymura (Eds.), *Handbook of individual differences in cognition* (pp. 27–49). Berlin: Springer.
- Schilling, S., & Bock, R. D. (2005). High-dimensional maximum marginal likelihood item factor analysis by adaptive quadrature. *Psychometrika*, 703, 533–555.
- Shalit, U., Weinshall, D., & Chechik, G. (2012). Online learning in the embedded manifold of low-rank matrices. *Journal of Machine Learning Research*, 13(Feb), 429–458.
- Takane, Y., & de Leeuw, J. (1987). On the relationship between item response theory and factor analysis of discretized variables. *Psychometrika*, 523, 393–408.
- Thissen, D., & Steinberg, L. (2009). Item response theory. In R. Millsap & A. Maydeu-Olivares (Eds.), *The sage handbook of quantitative methods in psychology* (pp. 148–177). London: Sage Publications.
- Vandereycken, B. (2013). Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 232, 1214–1236.

- Wirth, R., & Edwards, M. C. (2007). Item factor analysis: Current approaches and future directions. *Psychological methods*, 12(1), 58–79.
- Woods, C. M., & Lin, N. (2009). Item response theory with estimation of the latent density using Davidian curves. *Applied Psychological Measurement*, 33(2), 102–117.
- Woods, C. M., & Thissen, D. (2006). Item response theory with estimation of the latent population distribution using spline-based densities. *Psychometrika*, 71(2), 281–301.
- Yu, Y., Wang, T., & Samworth, R. J. (2015). A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2), 315–323.
- 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(1), 44–71.

Manuscript Received: 27 FEB 2019

Final Version Received: 3 APR 2020

Published Online Date: 15 JUL 2020