

# Certifiably Optimal Low Rank Factor Analysis

**Dimitris Bertsimas**

DBERTSIM@MIT.EDU

*Sloan School of Management*

*MIT*

*Cambridge, MA 02139, USA*

**Martin S. Copenhaver**

MCOPEN@MIT.EDU

*Operations Research Center*

*MIT*

*Cambridge, MA 02139, USA*

**Rahul Mazumder**

RAHULMAZ@MIT.EDU

*Sloan School of Management*

*MIT*

*Cambridge, MA 02139, USA*

**Editor:** Benjamin Recht

## Abstract

Factor Analysis (FA) is a technique of fundamental importance that is widely used in classical and modern multivariate statistics, psychometrics, and econometrics. In this paper, we revisit the classical rank-constrained FA problem which seeks to approximate an observed covariance matrix ( $\Sigma$ ) by the sum of a Positive Semidefinite (PSD) low-rank component ( $\Theta$ ) and a diagonal matrix ( $\Phi$ ) (with nonnegative entries) subject to  $\Sigma - \Phi$  being PSD. We propose a flexible family of rank-constrained, nonlinear Semidefinite Optimization based formulations for this task. We introduce a reformulation of the problem as a smooth optimization problem with convex, compact constraints and propose a unified algorithmic framework, utilizing state of the art techniques in nonlinear optimization to obtain high-quality feasible solutions for our proposed formulation. At the same time, by using a variety of techniques from discrete and global optimization, we show that these solutions are *certifiably optimal* in many cases, even for problems with thousands of variables. Our techniques are general and make *no* assumption on the underlying problem data. The estimator proposed herein aids statistical interpretability and provides computational scalability and significantly improved accuracy when compared to current, publicly available popular methods for rank-constrained FA. We demonstrate the effectiveness of our proposal on an array of synthetic and real-life datasets. To our knowledge, this is the first paper that demonstrates how a previously intractable rank-constrained optimization problem can be solved to provable optimality by coupling developments in convex analysis and in global and discrete optimization.

**Keywords:** factor analysis, rank minimization, semidefinite optimization, first order methods, nonlinear optimization, global optimization, discrete optimization

## 1. Introduction

Factor Analysis (FA) (Anderson, 2003; Bartholomew et al., 2011; Mardia et al., 1979), a widely used methodology in classical and modern multivariate statistics, is used as a

tool to obtain a parsimonious representation of the correlation structure among a set of variables in terms of a smaller number of common hidden factors. A basic FA model is of the form  $\mathbf{x} = \mathbf{L}\mathbf{f} + \boldsymbol{\epsilon}$ , where  $\mathbf{x}_{p \times 1}$  is the observed random vector,  $\mathbf{f}_{r_1 \times 1}$  (with  $r_1 \leq p$ ; note that we do not necessarily restrict  $r_1$  to be small) is a random vector of common factor variables or scores,  $\mathbf{L}_{p \times r_1}$  is a matrix of factor loadings and  $\boldsymbol{\epsilon}_{p \times 1}$  is a vector of uncorrelated random variables. We assume that the variables are mean-centered,  $\mathbf{f}$  and  $\boldsymbol{\epsilon}$  are uncorrelated and without loss of generality, the covariance of  $\mathbf{f}$  is the identity matrix. We will denote  $\text{Cov}(\boldsymbol{\epsilon}) = \boldsymbol{\Phi} = \text{diag}(\Phi_1, \dots, \Phi_p)$ . It follows that

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_c + \boldsymbol{\Phi}, \quad (1)$$

where  $\boldsymbol{\Sigma}$  is the covariance matrix of  $\mathbf{x}$  and  $\boldsymbol{\Sigma}_c := \mathbf{L}\mathbf{L}'$  is the covariance matrix corresponding to the common factors. Decomposition (1) suggests that  $\boldsymbol{\Sigma}$  can be written as the sum of a positive semidefinite (PSD) matrix  $\boldsymbol{\Sigma}_c$  of rank  $r_1$  and a nonnegative diagonal matrix ( $\boldsymbol{\Phi}$ ) corresponding to the errors. In particular, the variance of the  $i$ th coordinate of  $\mathbf{x} := (x_1, \dots, x_p)$ , i.e.,  $\text{var}(x_i) = \sum_k l_{ik}^2 + \Phi_i, i = 1, \dots, p$ , splits into two parts, where  $\mathbf{L} = ((l_{ik}))$ . The first part ( $\sum_k l_{ik}^2$ ) is known as the *communality estimate* (since this is the variance of the factors common to all the  $x_i$ 's) and the remaining part  $\Phi_i$  is the variance specific to the  $i$ th variable ( $\Phi_i$ 's are also referred to as the *unique variances* or simply *uniquenesses*).

**Formulation of the estimator:** In decomposition (1), the assumption that the rank ( $r_1$ ) of  $\boldsymbol{\Sigma}_c$  is small compared to  $p$  is fairly stringent—see Guttman (1958); Shapiro (1982); ten Berge (1998) for a historical overview of the concept. In a classical paper of Guttman (1958), the author argued based on psychometric evidence that  $\boldsymbol{\Sigma}_c$  is often found to have high algebraic rank. In psychometric case studies it is rather rare that the covariance structure can be *completely* explained by a *few* common factors corresponding to mental abilities—in fact there is evidence of at least hundreds of common factors being present with the number growing without an upper bound. Formally, this means that instead of assuming that  $\boldsymbol{\Sigma}_c$  has *exactly* low-rank it is practical to assume that it can be well-approximated by a low-rank matrix, namely,  $\mathbf{L}_1\mathbf{L}'_1$  with  $\mathbf{L}_1 \in \mathbb{R}^{p \times r}$ . More precisely,  $\mathbf{L}_1\mathbf{L}'_1$  is the best rank- $r$  approximation to  $\boldsymbol{\Sigma}_c$  in the matrix  $q$ -norm (also known as the Schatten norm), as defined in (5), and  $(\boldsymbol{\Sigma}_c - \mathbf{L}_1\mathbf{L}'_1)$  is the *residual* component. Following psychometric terminology,  $\mathbf{L}_1$  corresponds to the  $r$  *most significant* factors representative of mental abilities and the residual  $\boldsymbol{\Sigma}_c - \mathbf{L}_1\mathbf{L}'_1$  corresponds to the remaining psychometric factors unexplained by  $\mathbf{L}_1\mathbf{L}'_1$ . Thus we can rewrite decomposition (1) as follows:

$$\boldsymbol{\Sigma} = \underbrace{\mathbf{L}_1\mathbf{L}'_1}_{:=\boldsymbol{\Theta}} + \underbrace{(\boldsymbol{\Sigma}_c - \mathbf{L}_1\mathbf{L}'_1)}_{:=\boldsymbol{\mathcal{N}}} + \boldsymbol{\Phi}, \quad (2)$$

where we use the notation  $\boldsymbol{\Theta} = \mathbf{L}_1\mathbf{L}'_1$  and  $\boldsymbol{\mathcal{N}} = (\boldsymbol{\Sigma}_c - \mathbf{L}_1\mathbf{L}'_1)$  with  $\boldsymbol{\Theta} + \boldsymbol{\mathcal{N}} = \boldsymbol{\Sigma}_c = \boldsymbol{\Sigma} - \boldsymbol{\Phi}$ . Note that  $\boldsymbol{\Theta}$  denotes the best rank- $r$  approximation to  $(\boldsymbol{\Sigma} - \boldsymbol{\Phi})$ , with the residual component being  $\boldsymbol{\mathcal{N}} = \boldsymbol{\Sigma} - \boldsymbol{\Phi} - \boldsymbol{\Theta}$ . Note that the entries in  $\boldsymbol{\Phi}$  need to be non-negative<sup>1</sup> and  $\boldsymbol{\Sigma} - \boldsymbol{\Phi} \succeq \mathbf{0}$ . In fact, in the words of ten Berge (1998) (see p. 326)

1. Negative estimates of the diagonals of  $\boldsymbol{\Phi}$  are unwanted since they correspond to variances, but some FA estimation procedures often lead to negative estimates of  $\boldsymbol{\Phi}$ —these are popularly known in the literature as Heywood cases and have invited a significant amount of discussion in the community.

“...However, when  $\Sigma - \Phi$  the covariance matrix for the common parts of the variables, would appear to be indefinite, that would be no less embarrassing than having a negative unique variance in  $\Phi$ ...”

We further refer the reader to Mardia et al. (1979) discussing the importance of  $\Sigma - \Phi$  being PSD.<sup>2</sup> We thus have the following natural structural constraints on the parameters:

$$\Theta \succeq \mathbf{0}, \quad \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \quad \text{and} \quad \Sigma - \Phi \succeq \mathbf{0}. \quad (3)$$

Motivated by the above considerations, we present the following rank-constrained estimation problem for FA:

$$\begin{aligned} & \text{minimize} \quad \|\Sigma - (\Theta + \Phi)\|_q^q \\ & \text{s.t.} \quad \text{rank}(\Theta) \leq r \\ & \quad \Theta \succeq \mathbf{0} \\ & \quad \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\ & \quad \Sigma - \Phi \succeq \mathbf{0}, \end{aligned} \quad (4)$$

where  $\Theta \in \mathbb{R}^{p \times p}$ ,  $\Phi \in \mathbb{R}^{p \times p}$  are the optimization variables, and for a real symmetric matrix  $\mathbf{A}_{p \times p}$ , its matrix  $q$ -norm, also known as the Schatten norm (or Schatten-von-Neumann norm) is defined as

$$\|\mathbf{A}\|_q := \left( \sum_{i=1}^p |\lambda_i(\mathbf{A})|^q \right)^{\frac{1}{q}}, \quad (5)$$

where  $\lambda_i(\mathbf{A})$ ,  $i = 1, \dots, p$ , are the (real) eigenvalues of  $\mathbf{A}$ .

**Interpreting the estimator:** The estimation criterion (4) seeks to *jointly* obtain the (low-rank) common factors and *uniquenesses* that best explain  $\Sigma$  in terms of minimizing the matrix  $q$ -norm of the error  $\Sigma - (\Theta + \Phi)$  under the PSD constraints (3). Note that criterion (4) does not necessarily assume that  $\Sigma$  *exactly* decomposes into a low-rank PSD matrix and a non-negative diagonal matrix. Problem (4) enjoys curious similarities with Principal Component Analysis (PCA). In PCA, given a PSD matrix  $\Sigma$  the leading  $r$  principal component directions of  $\Sigma$  are obtained by minimizing  $\|\Sigma - \Theta\|_q$  subject to  $\Theta \succeq \mathbf{0}$  and  $\text{rank}(\Theta) \leq r$ . If the optimal solution  $\Phi$  to Problem (4) is *given*, Problem (4) is analogous to a rank- $r$  PCA on the residual matrix  $\Sigma - \Phi$ —thus it is naturally desirable to have  $\Sigma - \Phi \succeq \mathbf{0}$ . In PCA one is interested in understanding the proportion of variance explained by the top- $r$  principal component directions:  $\sum_{i=1}^r \lambda_i(\Sigma) / \sum_{i=1}^p \lambda_i(\Sigma)$ . The denominator  $\sum_{i=1}^p \lambda_i(\Sigma) = \text{Tr}(\Sigma)$  accounts for the total variance explained by the covariance matrix  $\Sigma$ . Analogously, the proportion of variance explained by  $\hat{\Theta}_r$  (which denotes the best rank  $r$  approximation to  $\Sigma - \Phi$ ) is given by  $\sum_{i=1}^r \lambda_i(\Sigma - \Phi) / \sum_{i=1}^p \lambda_i(\Sigma - \Phi)$ —for this quantity to be interpretable it is imperative that  $\Sigma - \Phi \succeq \mathbf{0}$ . In the above argument, of course, we assumed that  $\Phi$  is given. In general,  $\Phi$  needs to be estimated: Problem (4) achieves this goal by *jointly* learning  $\Phi$  and  $\Theta$ . We note that certain popular approaches of FA (see Sections 1.1 and 1.2) do not impose the PSD constraint  $\Sigma - \Phi$  as a part of the estimation

2. However, the estimation method described in Mardia et al. (1979) does not guarantee that  $\Sigma - \Phi \succeq \mathbf{0}$ .

scheme—leading to indefinite  $\Sigma - \Phi$ —thereby rendering statistical interpretations troublesome. Our numerical evidence suggests that the quality of estimates of  $\Theta$  and  $\Phi$  obtained from Problem (4) outperform those obtained by other competing procedures which do not incorporate the PSD constraints into their estimation criteria.

**Choice of  $r$ :** In exploratory FA, it is standard to consider several choices of  $r$  and study the manner in which the proportion of variance explained by the common factors *saturates* with increasing  $r$ . We refer the reader to popularly used methods described in Anderson (2003); Bartholomew et al. (2011); Mardia et al. (1979) and more modern techniques (Bai and Ng, 2008, see also references therein) for the choice of  $r$ .

**Estimate of Covariance Matrix:** In the finite sample setting, we set  $\Sigma$  to be the sample covariance matrix. A *consequence* of solving Problem (4) is that we get an *estimate* for the covariance matrix given by  $\hat{\Theta} + \hat{\Phi}$ —in this sense, criterion (4) can be viewed as a regularization scheme: the rank constraint on  $\Theta$  encourages parsimony and the PSD constraints encourage interpretability, as discussed above.

In this paper, we propose a general computational framework to solve Problem (4) for any  $q \geq 1$ . The well-known Schatten  $q$ -norm appearing in the loss function is chosen for flexibility—it underlines the fact that our approach can be applied for *any*  $q \geq 1$ . Note that the estimation criterion (4) (even for the case  $q = 1$ ) does not seem to appear in prior work on *approximate minimum rank Factor Analysis* (MRFA) (ten Berge and Kiers, 1991; Shapiro and ten Berge, 2002). However, we show in Proposition 1 that Problem (4) for the special case  $q = 1$  turns out to be equivalent to MRFA. For  $q = 2$ , the loss function is the familiar squared Frobenius norm also used in MINRES, though the latter formulation is not equivalent to Problem (4), as explained in Section 1.1. We place more emphasis on studying the computational properties for the more common norms  $q \in \{1, 2\}$ .

The presence of the rank constraint in Problem (4) makes the optimization problem non-convex. Globally optimizing Problem (4) or for that matter obtaining a good stationary point is quite challenging. We propose a new *equivalent* smooth formulation to Problem (4) which does not contain the combinatorial rank constraint. We employ simple and tractable sequential convex relaxation techniques with guaranteed convergence properties and excellent computational properties to obtain a stationary point for Problem (4). An important novelty of this paper is to present certifiable lower bounds on Problem (4) without resorting to structural assumptions, thus making it possible to solve Problem (4) to *provable* optimality. Towards this end we propose new methods and ideas that incorporate state-of-the-art developments in nonlinear and global optimization.<sup>3</sup>

### 1.1 A selective overview of related FA estimators

FA has a long and influential history which dates back more than a hundred years. The notion of FA possibly first appeared in Spearman (1904) for the one factor model, which was then generalized to the multiple factors model by various authors (see, for example,

---

3. The class of optimization problems studied in this paper involve global minimization of nonconvex, continuous semidefinite optimization problems. Computational methods for this class of problems are in a nascent stage; further, such methods are significantly less developed when compared to those for mixed integer linear optimization problems, thus posing a major challenge in this work.

Thurstone 1947). Significant contributions related to computational methods for FA have been nicely documented in Bai and Ng (2008); Harman and Jones (1966); Jöreskog (1967, 1978); Lawley and Maxwell (1962); Ledermann (1937); Rao (1973); Shapiro (1982); and ten Berge and Kiers (1991), among others.

We will briefly describe some widely used approaches for FA that are closely related to the approach pursued herein and also point out their connections.

**Constrained Minimum Trace Factor Analysis (MTFA):** This approach (ten Berge et al., 1981; Riccia and Shapiro, 1982) seeks to decompose  $\Sigma$  exactly into the sum of a diagonal matrix and a low-rank component, which are estimated via the following convex optimization problem:

$$\begin{aligned} & \text{minimize} && \text{Tr}(\Theta) \\ & \text{s.t.} && \Theta \succeq \mathbf{0} \\ & && \Sigma = \Theta + \Phi \\ & && \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0}, \end{aligned} \tag{6}$$

with variables  $\Theta, \Phi$ . Because  $\Theta$  is PSD,  $\text{Tr}(\Theta) = \sum_{i=1}^p \lambda_i(\Theta)$  is a convex surrogate (Fazel, 2002) for the rank of  $\Theta$ —Problem (6) may thus be viewed as a convexification of the rank minimization problem

$$\begin{aligned} & \text{minimize} && \text{rank}(\Theta) \\ & \text{s.t.} && \Theta \succeq \mathbf{0} \\ & && \Sigma = \Theta + \Phi \\ & && \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0}. \end{aligned} \tag{7}$$

In general, Problems (6) and (7) are *not* equivalent. See Riccia and Shapiro (1982); Saunderson et al. (2012); Shapiro (1982) (and references therein) for further connections between the minimizers of (6) and (7).

A main difference between formulations (4) and (7) is that the former allows an error in the residual  $(\Sigma - \Theta - \Phi)$  by constraining  $\Theta$  to have low-rank, unlike (7) which imposes a hard constraint  $\Sigma = \Theta + \Phi$ . As noted earlier, this can be quite restrictive in various applications. Even if one views Problem (6) as imposing a less stringent requirement than that of Problem (7), we see two distinct advantages of Problem (4) over Problem (6): it offers the modeling flexibility of controlling the complexity, viz. rank, of solutions via the choice of  $r$ ; and it provides smaller estimates of rank for a comparable amount of explained variance, as substantiated by experimental findings presented in Section 5.

**Approximate Minimum Rank Factor Analysis (MRFA):** This method (see for example, ten Berge and Kiers, 1991; Shapiro and ten Berge, 2002) considers the following optimization problem:

$$\begin{aligned} & \text{minimize} && \sum_{i=r+1}^p \lambda_i(\Sigma - \Phi) \\ & \text{s.t.} && \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\ & && \Sigma - \Phi \succeq \mathbf{0}, \end{aligned} \tag{8}$$

where the optimization variable is  $\Phi$ . Proposition 1 presented below establishes that Problem (8) is equivalent to the rank-constrained FA formulation (4) for the case  $q = 1$ . This connection does not appear to be formally established in ten Berge and Kiers (1991). We believe that criterion (4) for  $q = 1$  is easier to interpret as an estimation criterion for FA models over (8). ten Berge and Kiers (1981, 1991) describe a method<sup>4</sup> for numerically optimizing (8)—as documented in the code for MRFA (ten Berge and Kiers, 2003), their implementation can handle problems of size  $p \leq 20$ .

**Principal Component (PC) Factor Analysis:** Principal Component factor analysis or PC in short (Connor and Korajczyk, 1986; Bai and Ng, 2008) implicitly assumes that  $\Phi = \sigma^2 \mathbf{I}_{p \times p}$ , for some  $\sigma^2 > 0$  and performs a low-rank PCA on  $\Sigma$ . It is not clear how to estimate  $\Phi$  via this method such that  $\Phi_i \geq 0$  and  $\Sigma - \Phi \succeq \mathbf{0}$ . Following Mardia et al. (1979), the  $\Phi$ 's may be estimated after estimating  $\hat{\Theta}$  via the update rule  $\hat{\Phi} = \text{diag}(\Sigma - \hat{\Theta})$ —the estimates thus obtained, however, need not be non-negative. Furthermore, it is not guaranteed that the condition  $\Sigma - \Phi \succeq \mathbf{0}$  is met.

**Minimum Residual Factor Analysis (MINRES):** This approach (Harman and Jones, 1966; Shapiro and ten Berge, 2002) considers the following optimization problem (with respect to the variable  $\mathbf{L}_{p \times r}$ ):

$$\text{minimize} \quad \sum_{1 \leq i \neq j \leq p} (\sigma_{ij} - (\mathbf{L}\mathbf{L}')_{ij})^2, \quad (9)$$

where  $\Sigma := ((\sigma_{ij}))$ ,  $\Theta = \mathbf{L}\mathbf{L}'$  and the sum (in the objective) is taken over all the off-diagonal entries. Formulation (9) is equivalent to the nonconvex optimization problem

$$\begin{aligned} \text{minimize} \quad & \|\Sigma - (\Theta + \Phi)\|_2^2 \\ \text{s.t.} \quad & \text{rank}(\Theta) \leq r \\ & \Theta \succeq \mathbf{0} \\ & \Phi = \text{diag}(\Phi_1, \dots, \Phi_p), \end{aligned} \quad (10)$$

with variables  $\Theta$  and  $\Phi$ , where  $\Phi_i$ ,  $i \geq 1$ , are unconstrained. If  $\hat{\Theta}$  is a minimizer of Problem (10), then *any*  $\hat{\mathbf{L}}$  satisfying  $\hat{\mathbf{L}}\hat{\mathbf{L}}' = \hat{\Theta}$  minimizes (9) and vice-versa.

Various heuristic approaches are used to for Problem (9). The R package `psych`, for example, uses a black box gradient based tool `optim` to minimize the nonconvex Problem (9) with respect to  $\mathbf{L}$ . Once  $\hat{\mathbf{L}}$  is estimated, the diagonal entries of  $\Phi$  are estimated as  $\hat{\Phi}_i = \sigma_{ii} - (\hat{\mathbf{L}}\hat{\mathbf{L}}')_{ii}$ , for  $i \geq 1$ . Note that  $\hat{\Phi}_i$  obtained in this fashion may be negative<sup>5</sup> and the condition  $\Sigma - \Phi \succeq \mathbf{0}$  may be violated.

**Generalized Least Squares, Principal Axis and variants:** The Ordinary Least Squares (OLS) method for FA (Bartholomew et al., 2011) considers formulation (10) with the additional constraint that  $\Phi_i \geq 0 \forall i$ . The Weighted Least Squares (WLS) or the generalized least squares method (see for example, Bartholomew et al., 2011) considers a weighted least squares objective:

$$\|\mathbf{W}(\Sigma - (\Theta + \Phi))\|_2^2.$$

4. The method is similar to Algorithm 1 presented herein for the case of  $q = 1$ ; however, ten Berge and Kiers (1981, 1991) rely on a heuristic procedure, as described in Bentler and Woodward (1980), for the subproblem with respect to  $\Phi$ .

5. If  $\hat{\Phi}_i < 0$ , some ad-hoc procedure is used to threshold it to a nonnegative quantity.

As in the ordinary least squares case, here too we assume that  $\Phi_i \geq 0$ . Various choices of  $\mathbf{W}$  are possible depending upon the application, with  $\mathbf{W} \in \{\Sigma^{-1}, \Phi^{-1}\}$  being a couple of popular choices.

The Principal Axis (PA) FA method (Bartholomew et al., 2011; Revelle, 2015) is popularly used to estimate factor model parameters based on criterion (10) along with the constraints  $\Phi_i \geq 0 \forall i$ . This method starts with a nonnegative estimate  $\hat{\Phi}$  and performs a rank  $r$  eigendecomposition on  $\Sigma - \hat{\Phi}$  to obtain  $\hat{\Theta}$ . The matrix  $\hat{\Phi}$  is then updated to match the diagonal entries of  $\Sigma - \hat{\Theta}$ , and the above steps are repeated until the estimate  $\hat{\Phi}$  stabilizes.<sup>6</sup> Note that in this procedure the estimate  $\hat{\Theta}$  may fail to be PSD and the entries of  $\hat{\Phi}_i$  may be negative as well. Heuristic restarts and various initializations are often carried out to arrive at a reasonable solution (see for example discussions in Bartholomew et al. 2011).

In summary, the least squares stylized methods described above may lead to estimates that violate one or more of the constraints:  $\Sigma - \Phi \succeq \mathbf{0}$ ,  $\Theta \succeq \mathbf{0}$  and  $\Phi \succeq \mathbf{0}$ .

**Maximum Likelihood for Factor Analysis:** This approach (Bai and Li, 2012; Jöreskog, 1967; Mardia et al., 1979; Roweis and Ghahramani, 1999; Rubin and Thayer, 1982) is another widely used method in FA and typically assumes that the data follows a multivariate Gaussian distribution. This procedure maximizes a likelihood function and is quite different from the loss functions pursued in this paper and discussed above. The estimator need not exist for any  $\Sigma$ —see for example Robertson and Symons (2007).

Most of the methods described in Section 1.1 are widely used and their implementations are available in statistical packages `psych` (Revelle, 2015), `nFactors` (Raiche and Magis, 2011), `GPArotation` (Bernaards and Jennrich, 2005), and others and are publicly available from CRAN.<sup>7</sup>

## 1.2 Broad categories of factor analysis estimators

A careful investigation of the methods described above suggests that they can be divided into two broad categories. Some of the above estimators explicitly incorporate a PSD structural assumption on the residual covariance matrix  $\Sigma - \Phi$  in *addition* to requiring  $\Theta \succeq \mathbf{0}$  and  $\Phi \succeq \mathbf{0}$  while the others do not. As already pointed out, these constraints are important for statistical interpretability. We propose to distinguish between the following two broad categories of FA algorithms:

- (A) This category is comprised of FA estimation procedures cast as nonlinear Semidefinite Optimization (SDO) problems—estimation takes place in the presence of constraints of the form  $\Sigma - \Phi \succeq \mathbf{0}$ , along with  $\Theta \succeq \mathbf{0}$  and  $\Phi \succeq \mathbf{0}$ . Members of this category are MRFA, MTFA and more generally Problem (4).

Existing approaches for these problems are typically not scalable: for example, we are not aware of any algorithm (prior to this paper) for Problem (8) (MRFA) that scales to covariance matrices with  $p$  larger than thirty. Indeed, while theoretical guarantees

6. We are not aware of a proof, however, showing the convergence of this procedure.

7. <http://cran.us.r-project.org>

of optimality exist in certain cases (Saunderson et al., 2012), the conditions required for such results to hold are generally difficult to verify in practice.

- (B) This category includes classical FA methods which are not based on nonlinear SDO based formulations (as in Category (A)). MINRES, OLS, WLS, GLS, PC and PA based FA estimation procedures (as described in Section 1.1) belong to this category. These methods are generally scalable to problem sizes where  $p$  is of the order of a few thousand—significantly larger than most procedures belonging to Category (A)—and are implemented in open-source R-packages.

**Contributions:** Our contributions in this paper may be summarized as follows:

1. We consider a flexible family of FA estimators which can be obtained as solutions to rank-constrained nonlinear SDO problems. In particular, our framework provides a unifying perspective on several existing FA estimation approaches.
2. We propose a novel *exact* reformulation of the rank-constrained FA problem (4) as a smooth optimization problem with convex compact constraints. We also develop a unified algorithmic framework utilizing modern optimization techniques to obtain high quality solutions to Problem (4). Our algorithms, at every iteration, simply require computing a low-rank eigendecomposition of a  $p \times p$  matrix and a structured scalable convex SDO. Our proposal is capable of solving FA problems involving covariance matrices having dimensions up to a few thousand, thereby making it on par with the most scalable FA methods used currently.<sup>8</sup>
3. Our SDO formulation enables us to estimate the underlying factors and unique variances under the restriction that the residual covariance matrix is PSD—a characteristic that is absent in several popularly used FA methods. This aids statistical interpretability, especially in drawing parallels with PCA and understanding the proportion of variance explained by a given number of factors. Methods proposed herein produce superior quality estimates, in terms of various performance metrics, when compared to existing FA approaches. To our knowledge, this is the first paper demonstrating that certifiably optimal solutions to a rank-constrained problem can be found for problems of realistic sizes, without making any assumptions on the underlying data.
4. Using techniques from discrete and global optimization, we develop a branch-and-bound algorithm which proves that the low-rank solutions found are often optimal in seconds for problems on the order of  $p = 10$  variables, in minutes for problems on the order of  $p = 100$ , and in days for some problems on the order of  $p = 4000$ . As the selected rank increases, so too does the computational burden of proving optimality. It is particularly crucial to note that the optimal solutions for all problems we consider are found very quickly, and that vast majority of computational time is then spent on proving optimality. Hence, for a practitioner who is not particularly concerned with certifying optimality, our techniques for finding feasible solutions provide high-quality estimates quickly.

---

8. An implementation of our approach is available at <https://github.com/copenhaver/factoranalysis>.



5. We provide computational evidence demonstrating the favorable performance of our proposed method. Finally, to the best of our knowledge this is the first paper that views various FA methods in a unified fashion via a modern optimization lens and attempts to compare a wide range of FA techniques in large scale.

**Structure of the paper:** The paper is organized as follows. In Section 1 we propose a flexible family of optimization Problems (4) for the task of statistical estimation in FA models. Section 2 presents an exact reformulation of Problem (4) as a nonlinear SDO without the rank constraint. Section 3 describes the use of nonlinear optimization techniques such as the Conditional Gradient (CG) method (Bertsekas, 1999) adapted to provide feasible solutions (upper bounds) to our formulation. First order methods employed to compute the convex SDO subproblems are also described in the same section. In Section 4, we describe our method for certifying optimality of the solutions from Section 3 in the case when  $q = 1$ . In Section 5, we present computational results demonstrating the effectiveness of our proposed method in terms of (a) modeling flexibility in the choice of the number of factors  $r$  and the parameter  $q$ , (b) scalability, and (c) the quality of solutions obtained in a wide array of real and synthetic datasets—comparisons with several existing methods for FA are considered. Section 6 contains our conclusions.

## 2. Reformulations of Problem (4)

Let  $\lambda(\mathbf{A})$  denote the vector of eigenvalues of  $\mathbf{A}$ , arranged in decreasing order, i.e.,

$$\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \dots \geq \lambda_p(\mathbf{A}). \quad (11)$$

The following proposition presents the first reformulation of Problem (4) as a continuous eigenvalue optimization problem with convex compact constraints. Proofs of all results can be found in Appendix A.

**Proposition 1 (a)** *For any  $q \geq 1$ , Problem (4) is equivalent to:*

$$\begin{aligned} \text{(CFA}_q\text{)} \quad & \text{minimize} && f_q(\Phi; \Sigma) := \sum_{i=r+1}^p \lambda_i^q(\Sigma - \Phi) \\ & \text{s.t.} && \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\ & && \Sigma - \Phi \succeq \mathbf{0}, \end{aligned} \quad (12)$$

where  $\Phi$  is the optimization variable.

(b) *Suppose  $\Phi^*$  is a minimizer of Problem (12), and let*

$$\Theta^* = \mathbf{U} \text{diag}(\lambda_1(\Sigma - \Phi^*), \dots, \lambda_r(\Sigma - \Phi^*), 0, \dots, 0) \mathbf{U}^T, \quad (13)$$

where  $\mathbf{U}_{p \times p}$  is the matrix of eigenvectors of  $\Sigma - \Phi^*$ . Then  $(\Theta^*, \Phi^*)$  is a solution to Problem (4).

Problem (12) is a nonlinear SDO in  $\Phi$ , unlike the original formulation (4) that estimates  $\Theta$  and  $\Phi$  jointly. Note that the rank constraint does not appear in Problem (12) and the constraint set of Problem (12) is convex and compact. However, Problem (12) is nonconvex due to the nonconvex objective function  $\sum_{i=r+1}^p \lambda_i^q(\Sigma - \Phi)$ . For  $q = 1$  the function appearing in the objective of (12) is concave and for  $q > 1$ , it is neither convex nor concave.

**Proposition 2** *The estimation Problem (4) is equivalent to*

$$\begin{aligned}
 & \text{minimize} && \|\boldsymbol{\Sigma} - (\boldsymbol{\Theta} + \boldsymbol{\Phi})\|_q^q \\
 & \text{s.t.} && \text{rank}(\boldsymbol{\Theta}) \leq r \\
 & && \boldsymbol{\Theta} \succeq \mathbf{0} \\
 & && \boldsymbol{\Phi} = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\
 & && \boldsymbol{\Sigma} - \boldsymbol{\Phi} \succeq \mathbf{0} \\
 & && \boldsymbol{\Sigma} - \boldsymbol{\Theta} \succeq \mathbf{0}.
 \end{aligned} \tag{14}$$

Note that Problem (14) has an additional PSD constraint  $\boldsymbol{\Sigma} - \boldsymbol{\Theta} \succeq \mathbf{0}$  which does not explicitly appear in Problem (4). It is interesting to note that the two problems are equivalent. By virtue of Proposition 2, Problem (14) can as well be used as the estimation criterion for rank constrained FA. However, we will work with formulation (4) because it is easier to interpret from a statistical perspective.

**Special instances of (CFA<sub>q</sub>):** We show that some well-known FA estimation problems can be viewed as special cases of our general framework.

For  $q = 1$ , Problem (12) reduces to MRFA, as described in (8). For  $q = 1$  and  $r = 0$ , Problem (12) reduces to MTFA (6). When  $q = 2$ , we get a variant of (10), i.e., a PSD *constrained* analogue of MINRES

$$\begin{aligned}
 & \text{minimize} && \sum_{i=r+1}^p \lambda_i^2(\boldsymbol{\Sigma} - \boldsymbol{\Phi}) \\
 & \text{s.t.} && \boldsymbol{\Phi} = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\
 & && \boldsymbol{\Sigma} - \boldsymbol{\Phi} \succeq \mathbf{0}.
 \end{aligned} \tag{15}$$

Note that unlike Problem (10), Problem (15) explicitly imposes PSD constraints on  $\boldsymbol{\Phi}$  and  $\boldsymbol{\Sigma} - \boldsymbol{\Phi}$ . The objective function in (12) is continuous but non-smooth. The function is differentiable at  $\boldsymbol{\Phi}$  if and only if the  $r$  and  $(r+1)$ th eigenvalues of  $\boldsymbol{\Sigma} - \boldsymbol{\Phi}$  are distinct (Lewis, 1996; Shapiro and ten Berge, 2002), i.e.,  $\lambda_{r+1}(\boldsymbol{\Sigma} - \boldsymbol{\Phi}) < \lambda_r(\boldsymbol{\Sigma} - \boldsymbol{\Phi})$ . The non-smoothness of the objective function in Problem (12) makes the use of standard gradient based methods problematic (Bertsekas, 1999). Theorem 1 presents a reformulation of Problem (12) in which the objective function is continuously differentiable.

**Theorem 1 (a)** *The estimation criterion given by Problem (4) is equivalent to*<sup>9</sup>

$$\begin{aligned}
 & \text{minimize} && g_q(\mathbf{W}, \boldsymbol{\Phi}) := \text{Tr}(\mathbf{W}(\boldsymbol{\Sigma} - \boldsymbol{\Phi})^q) \\
 & \text{s.t.} && \boldsymbol{\Phi} = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\
 & && \boldsymbol{\Sigma} - \boldsymbol{\Phi} \succeq \mathbf{0} \\
 & && \mathbf{I} \succeq \mathbf{W} \succeq \mathbf{0} \\
 & && \text{Tr}(\mathbf{W}) = p - r.
 \end{aligned} \tag{16}$$

**(b)** *The solution  $\widehat{\boldsymbol{\Theta}}$  of Problem (4) can be recovered from the solution  $\widehat{\mathbf{W}}, \widehat{\boldsymbol{\Phi}}$  of Problem (16) via:*

$$\widehat{\boldsymbol{\Theta}} := \widehat{\mathbf{U}} \text{diag}(\widehat{\lambda}_1, \dots, \widehat{\lambda}_r, 0, \dots, 0) \widehat{\mathbf{U}}', \tag{17}$$

---

9. For any PSD matrix  $\mathbf{A}$ , with eigendecomposition  $\mathbf{A} = \mathbf{U}_A \text{diag}(\lambda_1, \dots, \lambda_p) \mathbf{U}'_A$ , we define  $\mathbf{A}^q := \mathbf{U}_A \text{diag}(\lambda_1^q, \dots, \lambda_p^q) \mathbf{U}'_A$ , for any  $q \geq 1$ .

where  $\widehat{\mathbf{U}}$  is the matrix formed by the  $p$  eigenvectors corresponding to the eigenvalues  $\widehat{\lambda}_1, \dots, \widehat{\lambda}_p$  (arranged in decreasing order) of the matrix  $\mathbf{\Sigma} - \widehat{\mathbf{\Phi}}$ . Given  $\widehat{\mathbf{\Phi}}$ , any solution  $\widehat{\mathbf{\Theta}}$  (given by (17)) is independent of  $q$ .

In Problem (16), if we partially minimize the function  $g_q(\mathbf{W}, \mathbf{\Phi})$  over  $\mathbf{\Phi}$  (with fixed  $\mathbf{W}$ ), the resulting function is concave in  $\mathbf{W}$ . This observation leads to the following proposition.

**Proposition 3** *The function  $G_q(\mathbf{W})$  obtained upon (partially) minimizing  $g_q(\mathbf{W}, \mathbf{\Phi})$  over  $\mathbf{\Phi}$  (with  $\mathbf{W}$  fixed) in Problem (16), given by*

$$\begin{aligned} G_q(\mathbf{W}) := & \inf_{\mathbf{\Phi}} g_q(\mathbf{W}, \mathbf{\Phi}) \\ \text{s.t. } & \mathbf{\Phi} = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\ & \mathbf{\Sigma} - \mathbf{\Phi} \succeq \mathbf{0}, \end{aligned} \quad (18)$$

is concave in  $\mathbf{W}$ . The sub-gradients of the function  $G_q(\mathbf{W})$  exist and are given by:

$$\nabla G_q(\mathbf{W}) = (\mathbf{\Sigma} - \widehat{\mathbf{\Phi}}(\mathbf{W}))^q, \quad (19)$$

where  $\widehat{\mathbf{\Phi}}(\mathbf{W})$  is a minimizer of the convex optimization Problem (18).

In light of Proposition 3, we present another reformulation of Problem (4) as the following concave minimization problem:

$$\begin{aligned} \text{minimize } & G_q(\mathbf{W}) \\ \text{s.t. } & \mathbf{I} \succeq \mathbf{W} \succeq \mathbf{0} \\ & \text{Tr}(\mathbf{W}) = p - r, \end{aligned} \quad (20)$$

where the function  $G_q(\mathbf{W})$  is differentiable if and only if  $\widehat{\mathbf{\Phi}}(\mathbf{W})$  is unique.

Note that by virtue of Proposition 1, Problems (12) and (20) are equivalent. Therefore, it is natural to ask whether one formulation might be favored over the other from a computational perspective. Towards this end, note that both Problems (12) and (20) involve the minimization of a non-smooth objective function, over convex compact constraints. However, the objective function of Problem (12) is nonconvex (for  $q > 1$ ) whereas the one in Problem (20) is concave (for all  $q \geq 1$ ). We will see in Section 3 that CG-based algorithms can be applied to a concave minimization problem (even if the objective function is not differentiable); however, CG applied to general non-smooth objective functions has limited convergence guarantees. Thus, formulation (20) is readily amenable to CG-based optimization algorithms, unlike formulation (12). This seems to make Problem (20) computationally more appealing than Problem (12).

### 3. Finding Upper Bounds

This section presents a unified computational framework for the class of problems ( $\text{CFA}_q$ ). Problem (16) is a nonconvex smooth optimization problem and obtaining a stationary point is quite challenging. We propose iterative schemes based on the Conditional Gradient (CG) algorithm (Bertsekas, 1999)—a generalization of the Frank-Wolfe algorithm (Frank

and Wolfe, 1956)—to obtain a stationary point of the problem. The appealing aspect of our framework is that every iteration of the algorithm requires solving a convex SDO problem which is computationally tractable. While off-the-shelf interior point algorithms—for example SDPT3 (Toh et al., 1999), Yalmip (Löfberg, 2004), and MOSEK (Andersen and Andersen, 2000)—can be used to solve the convex SDO problems, they typically do not scale well for large problems due to intensive memory requirements. In this vein, first order algorithms have received a lot of attention (Nesterov, 2004, 2005, 2007) in convex optimization of late, due to their low cost per iteration, low-memory requirements, and ability to deliver solutions of moderate accuracy for large problems within a modest time limit. We use first order methods to solve the convex SDO problems. We present one primary scheme based on the CG algorithm:

**Algorithm 1:** This scheme, described in Section 3.1, applies CG on the optimization Problem (20), where the function  $G_q(\mathbf{W})$  defined in (18) is concave (and possibly non-smooth).

In addition, in Appendix B we present an alternative approach that applies CG to Problem (16), where the objective function  $g_q(\mathbf{W}, \Phi)$  is smooth.

To make notation simpler, we will use the following shorthand:

$$\Psi_{p,p-r} := \{\mathbf{W} \in \mathbb{R}^{p \times p} : \mathbf{I} \succeq \mathbf{W} \succeq \mathbf{0}, \text{Tr}(\mathbf{W}) = p - r\} \quad (21)$$

and  $\mathcal{F}_\Sigma = \{\Phi : \Sigma - \Phi \succcurlyeq \mathbf{0}, \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succcurlyeq \mathbf{0}\}$ .

### 3.1 A CG based algorithm for Problem (20)

The CG method for Problem (20) requires solving a linearization of the concave objective function. At iteration  $k$ , if  $\mathbf{W}^{(k)}$  is the current estimate of  $\mathbf{W}$ , the new estimate  $\mathbf{W}^{(k+1)}$  is obtained by

$$\mathbf{W}^{(k+1)} \in \arg \min_{\mathbf{W} \in \Psi_{p,p-r}} \langle \nabla G_q(\mathbf{W}^{(k)}), \mathbf{W} \rangle = \arg \min_{\mathbf{W} \in \Psi_{p,p-r}} \text{Tr}(\mathbf{W}(\Sigma - \Phi^{(k)})^q), \quad (22)$$

where by Proposition 3,  $(\Sigma - \Phi^{(k)})^q$  is a sub-gradient of  $G_q(\mathbf{W})$  at  $\mathbf{W}^{(k)}$  with  $\Phi^{(k)}$  given by

$$\Phi^{(k)} \in \arg \min_{\Phi \in \mathcal{F}_\Sigma} \text{Tr}(\mathbf{W}^{(k)}(\Sigma - \Phi)^q) \quad (23)$$

No explicit line search is necessary here because the minimum will always be at the new point, i.e.,  $\Phi^{(k)}$ , due to the concavity of the objective function. The sequence  $\mathbf{W}^{(k)}$  is recursively computed via (22) until the convergence criterion

$$G_q(\mathbf{W}^{(k)}) - G_q(\mathbf{W}^{(k+1)}) \leq \text{TOL} \cdot G_q(\mathbf{W}^{(k)}), \quad (24)$$

for some user-defined tolerance,  $\text{TOL} > 0$ , is met. A short description of the procedure appears in Algorithm 1.

Before we present the convergence rate of Algorithm 1, we will need to introduce some notation. For any point  $\overline{\mathbf{W}}$  belonging to the feasible set of Problem (20) let us define  $\Delta(\overline{\mathbf{W}})$  as follows:

$$\Delta(\overline{\mathbf{W}}) := \inf_{\mathbf{W} \in \Psi_{p,p-r}} \langle \nabla G_q(\overline{\mathbf{W}}), \mathbf{W} - \overline{\mathbf{W}} \rangle. \quad (25)$$

---

**Algorithm 1** A CG based algorithm for formulation (20)
 

---

1. Initialize with  $\mathbf{W}^{(1)}$  ( $k = 1$ ), feasible for Problem (20) and repeat, for  $k \geq 2$ , Steps 2-3 until convergence criterion (24) is satisfied.
  2. Update  $\Phi$  (with  $\mathbf{W}$  fixed) by solving (23).
  3. Update  $\mathbf{W}$  (with  $\Phi$  fixed) by solving (22), to get  $\mathbf{W}^{(k+1)}$ .
- 

Further,  $\mathbf{W}^*$  satisfies the first order stationary condition for Problem (20) if  $\mathbf{W}^*$  is feasible for the problem and  $\Delta(\mathbf{W}^*) \geq 0$ .

We now present Theorem 2 establishing the rate of convergence and associated convergence properties of Algorithm 1. The proof (along with all other omitted proofs) is contained in Appendix A.

**Theorem 2** *If  $\mathbf{W}^{(k)}$  is a sequence produced by Algorithm 1, then  $G_q(\mathbf{W}^{(k)})$  is a monotone decreasing sequence and every limit point  $\mathbf{W}^{(\infty)}$  of the sequence  $\mathbf{W}^{(k)}$  is a stationary point of Problem (20). Furthermore, Algorithm 1 has a convergence rate of  $O(1/K)$  (with  $K$  denoting the iteration index) to a first order stationary point of Problem (20), i.e.,*

$$\min_{i=1, \dots, K} \left\{ -\Delta(\mathbf{W}^{(i)}) \right\} \leq \frac{G_q(\mathbf{W}^{(1)}) - G_q(\mathbf{W}^{(\infty)})}{K}. \quad (26)$$

### 3.2 Solving the convex SDO problems

Algorithm 1 requires sequentially solving convex SDO problems in  $\mathbf{W}$  and  $\Phi$ . We describe herein how these subproblems can be solved efficiently.

#### 3.2.1 SOLVING THE SDO PROBLEM WITH RESPECT TO $\mathbf{W}$

A generic SDO problem associated with Problem (22) requires updating  $\mathbf{W}$  as

$$\widehat{\mathbf{W}} \in \arg \min_{\mathbf{W} \in \Psi_{p,p-r}} \langle \mathbf{W}, \widetilde{\mathbf{W}} \rangle, \quad (27)$$

for some fixed symmetric  $\widetilde{\mathbf{W}}_{p \times p}$ , depending upon the algorithm and the choice of  $q$ , described as follows. For Algorithm 1 the update in  $\mathbf{W}$  at iteration  $k$  for Problem (22), corresponds to  $\widetilde{\mathbf{W}} = (\Sigma - \Phi^{(k+1)})^q$ .

A solution to Problem (27) is given by  $\widehat{\mathbf{W}} = \sum_{i=r+1}^p \mathbf{u}_i \mathbf{u}_i'$ , where  $\mathbf{u}_1, \dots, \mathbf{u}_p$  are the eigenvectors of the matrix  $\widetilde{\mathbf{W}}$ , corresponding to the eigenvalues  $\lambda_1(\widetilde{\mathbf{W}}), \dots, \lambda_p(\widetilde{\mathbf{W}})$ .

#### 3.2.2 SOLVING THE SDO PROBLEM WITH RESPECT TO $\Phi$

The SDO problem arising from the update of  $\Phi$  is not as straightforward as the update with respect to  $\mathbf{W}$ . Before presenting the general case, it helps to consider a few special cases of (CFA $_q$ ). For  $q = 1$  the objective function of Problem (16)

$$g_1(\mathbf{W}, \Phi) = \langle \mathbf{W}, \Sigma \rangle - \sum_{i=1}^p w_{ii} \Phi_i \quad (28)$$

is linear in  $\Phi$  (for fixed  $\mathbf{W}$ ). For  $q = 2$ , the objective function of Problem (16)

$$g_2(\mathbf{W}, \Phi) = \text{Tr}(\mathbf{W}\Sigma^2) + \sum_{i=1}^p (w_{ii}\Phi_i^2 - 2\langle \mathbf{w}_i, \sigma_i \rangle \Phi_i) \quad (29)$$

is a convex quadratic in  $\Phi$  (for fixed  $\mathbf{W}$ ).

For Algorithm 1, the partial minimizations with respect to  $\Phi$ , for  $q = 1$  and  $q = 2$ , require minimizing Problems (28) and (29), respectively.

Various instances of optimization problems with respect to  $\Phi$  such as those appearing in Algorithm 1 can be viewed as special cases of the following family of SDO problems:

$$\underset{\Phi \in \mathcal{F}_\Sigma}{\text{minimize}} \quad \sum_{i=1}^p (c_i \Phi_i^2 + d_i \Phi_i) \quad (30)$$

where  $c_i \geq 0$  and  $d_i$  for  $i = 1, \dots, p$  are problem parameters that depend upon the choice of algorithm and  $q$ . We now present a first order convex optimization scheme for solving (30).

#### A FIRST ORDER SCHEME FOR PROBLEM (30)

With the intention of providing a simple and scalable algorithm for the convex SDO problem, we use the Alternating Direction Method of Multipliers (Bertsekas, 1999; Boyd et al., 2011) (ADMM). We introduce a splitting variable  $\Lambda = \Sigma - \Phi$  and rewrite Problem (30) in the following equivalent form:

$$\begin{aligned} \underset{\Phi, \Lambda}{\text{minimize}} \quad & \sum_{i=1}^p (c_i \Phi_i^2 + d_i \Phi_i) \\ \text{s.t.} \quad & \Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0} \\ & \Lambda \succeq \mathbf{0} \\ & \Lambda = \Sigma - \Phi. \end{aligned} \quad (31)$$

The Augmented Lagrangian for the above problem is:

$$\mathcal{L}_\rho(\Phi, \Lambda, \nu) := \sum_{i=1}^p (c_i \Phi_i^2 + d_i \Phi_i) + \langle \nu, \Lambda - (\Sigma - \Phi) \rangle + \frac{\rho}{2} \|\Lambda - (\Sigma - \Phi)\|_2^2, \quad (32)$$

where  $\rho > 0$  is a scalar and  $\langle \cdot, \cdot \rangle$  denotes the standard trace inner product. ADMM involves the following three updates:

$$\Phi^{(k+1)} \in \underset{\Phi = \text{diag}(\Phi_1, \dots, \Phi_p) \succeq \mathbf{0}}{\text{arg min}} \quad \mathcal{L}_\rho(\Phi, \Lambda^{(k)}, \nu^{(k)}), \quad (33)$$

$$\Lambda^{(k+1)} \in \underset{\Lambda \succeq \mathbf{0}}{\text{arg min}} \quad \mathcal{L}_\rho(\Phi^{(k+1)}, \Lambda, \nu^{(k)}), \quad (34)$$

$$\nu^{(k+1)} = \nu^{(k)} + \rho(\Lambda^{(k+1)} - (\Sigma - \Phi^{(k+1)})), \quad (35)$$

and produces a sequence  $\{(\Phi^{(k)}, \Lambda^{(k)}, \nu^{(k)})\}$ ,  $k \geq 1$  — the convergence properties of the algorithm are quite well known (Boyd et al., 2011).

Problem (33) can be solved in closed form as

$$\Phi_i^{(k+1)} = \frac{\rho}{\rho + 2c_i} \max \left\{ (\sigma_{ii} - \lambda_{ii}^{(k)}) - \frac{(d_i + \nu_{ii}^{(k)})}{\rho}, 0 \right\}, \quad i = 1, \dots, p. \quad (36)$$

The update with respect to  $\mathbf{\Lambda}$  in (34) requires an eigendecomposition:

$$\begin{aligned} \mathbf{\Lambda}^{(k+1)} &= \arg \min_{\mathbf{\Lambda} \succeq \mathbf{0}} \left\| \mathbf{\Lambda} - \left( \mathbf{\Sigma} - \mathbf{\Phi}^{(k+1)} - \frac{1}{\rho} \boldsymbol{\nu}^{(k)} \right) \right\|_2^2 \\ &= \mathcal{P}_{S_p^+} \left( \mathbf{\Sigma} - \mathbf{\Phi}^{(k+1)} - \frac{1}{\rho} \boldsymbol{\nu}^{(k)} \right), \end{aligned} \quad (37)$$

where, the operator  $\mathcal{P}_{S_p^+}(\mathbf{A})$  denotes the projection of a symmetric matrix  $\mathbf{A}$  onto the cone of PSD matrices of dimension  $p \times p$ :  $\mathcal{P}_{S_p^+}(\mathbf{A}) = \mathbf{U}_A \text{diag} \left( \max \{ \lambda_1, 0 \}, \dots, \max \{ \lambda_p, 0 \} \right) \mathbf{U}'_A$ , where,  $\mathbf{A} = \mathbf{U}_A \text{diag}(\lambda_1, \dots, \lambda_p) \mathbf{U}'_A$  is the eigendecomposition of  $\mathbf{A}$ .

**Stopping criterion:** The ADMM iterations (33)–(35) are continued till the values of  $\| \mathbf{\Lambda}^{(k+1)} - (\mathbf{\Sigma} - \mathbf{\Phi}^{(k+1)}) \|_2$  and the relative change in the objective values of Problem (30) become smaller than a certain threshold, say,  $\text{TOL} \times \alpha$ , where  $\alpha \in \{10^{-1}, \dots, 10^{-3}\}$ —this is typically taken to be smaller than the convergence threshold for the CG iterations (TOL).

**Computational cost of Problem (30):** The most intensive computational stage in the ADMM procedure is in performing the projection operation (37); this requires  $O(p^3)$  operations due to the associated eigendecomposition. This needs to be done for as many ADMM steps, until convergence.

Since Problem (30) is embedded inside iterative procedures like Algorithm 1, the estimates of  $(\mathbf{\Phi}, \mathbf{\Lambda}, \boldsymbol{\nu})$  obtained by solving Problem (30) for a iteration index (of the CG algorithm) provides a good warm-start for the Problem (30) in the subsequent CG iteration. This is often found to decrease the number of iterations required by the ADMM algorithm to converge to a prescribed level of accuracy.<sup>10</sup>

### 3.3 Computational cost of Algorithm 1

For Algorithm 1 and other CG-based algorithms for factor analysis (see Appendix B), the computational bottleneck is in performing the eigendecomposition of a  $p \times p$  matrix: the  $\mathbf{W}$  update requires performing a low-rank eigendecomposition of a  $p \times p$  matrix and the  $\mathbf{\Phi}$  update requires solving a problem of the form (30), which also costs  $O(p^3)$ . Since eigendecompositions can easily be done for  $p$  of the order of a few thousands, the proposed algorithms can be applied to that scale.

Note that most existing popular algorithms for FA belonging to Category (B) (see Section 1.2) also perform an eigendecomposition with cost  $O(p^3)$ . Thus it appears that

10. The utility of warm starts is another compelling reason to apply a first-order-based approach instead of interior point methods. Indeed, warm starts are well-known to perform quite poorly when incorporated into interior point methods—often they can perform worse than cold starts (Yildirim and Wright, 2002; John and Yildirim, 2008). Given the need to repeatedly solve similarly structured SDOs for both upper bounds (as presented in this section) as well as lower bounds (as presented in Section 4), the ability to effectively incorporate warm start information is crucial.

Category (B) and the algorithms proposed herein have the same computational complexity and hence these two estimation approaches are equally scalable.

#### 4. Certificates of Optimality via Lower Bounds

In this section, we outline our approach to computing lower bounds to  $(\text{CFA}_q)$  via techniques from global optimization and matrix analysis. In particular, we focus on the case when  $q = 1$ .<sup>11</sup> We begin with an overview of the method. We then discuss initialization parameters for the method as well as branching rules and other refinements employed in our approach.

##### 4.1 Overview of Method

Our primary problem of interest is to provide lower bounds to  $(\text{CFA}_1)$ , i.e.,

$$\underset{\Phi \in \mathcal{F}_\Sigma}{\text{minimize}} \sum_{i=r+1}^p \sigma_i(\Sigma - \Phi), \quad (38)$$

or equivalently vis-à-vis Theorem 1,

$$\underset{\substack{\mathbf{W} \in \Psi_{p,p-r} \\ \Phi \in \mathcal{F}_\Sigma}}{\text{minimize}} \langle \mathbf{W}, \Sigma - \Phi \rangle. \quad (39)$$

One possible approach is to consider *convex* lower bounds to (39).

**Definition 3** For a function  $f : \Gamma \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  we define its convex envelope on  $\Gamma$ , denoted  $\text{conv}_{\Gamma}(f)$ , to be the largest convex function  $g$  with  $g \leq f$ . In symbols,

$$g = \sup\{h : \Gamma \rightarrow \mathbb{R} \mid h \text{ convex on } \Gamma \text{ and } h \leq f\}.$$

The convex envelope acts as the best possible convex lower bound for a given function. Further, its precise form is well-known for certain classes of functions; one such function of principle interest here is described in the following theorem. This is in some contexts referred to as a McCormick hull and is widely used throughout the nonconvex optimization literature (Floudas, 1999; Tawarmalani and Sahinidis, 2002a).

**Theorem 4 (Al-Khayyal and Falk, 1983)** If  $f : \Gamma = [0, 1] \times [\ell, u] \rightarrow \mathbb{R}$  is defined by  $f(x, y) = -xy$ , then the convex envelope of  $f$  on  $\Gamma$  is precisely

$$\text{conv}_{\Gamma}(f)(x, y) = \max\{-ux, \ell - \ell x - y\}.$$

Further,  $|f(x, y) - \text{conv}_{\Gamma}(f)(x, y)| \leq (u - \ell)/4$ . In particular, if  $|u - \ell| \rightarrow 0$ , then  $\text{conv}_{\Gamma}(f) \rightarrow f$ .

---

11. The general case for  $q > 1$  can be addressed using similar (although more complicated) techniques as applied to Problem (16), again applying principles developed in global optimization.



Using this result we proceed to describe our approach for computing lower bounds to (CFA<sub>1</sub>) as in (39). First observe that if  $\Phi_i \in [\ell_i, u_i]$  and  $W_{ii} \in [0, 1]$ , then

$$\begin{aligned} \text{convex}(-W_{ii}\Phi_i)(W_{ii}, \Phi_i) &= \max\{-u_i W_{ii}, \ell_i - \ell_i W_{ii} - \Phi_i\} \\ &= -\min\{u_i W_{ii}, \Phi_i + \ell_i W_{ii} - \ell_i\}. \end{aligned}$$

Hence, the best possible convex lower bound to the objective in Problem (39), namely  $\langle \mathbf{W}, \boldsymbol{\Sigma} - \boldsymbol{\Phi} \rangle = \langle \mathbf{W}, \boldsymbol{\Sigma} \rangle - \sum_i W_{ii} \Phi_i$ , for  $\boldsymbol{\ell} \leq \text{diag}(\boldsymbol{\Phi}) \leq \mathbf{u}$  and  $\mathbf{I} \succeq \mathbf{W} \succeq \mathbf{0}$  is

$$\langle \mathbf{W}, \boldsymbol{\Sigma} \rangle - \sum_{i=1}^p \min\{u_i W_{ii}, \Phi_i + \ell_i W_{ii} - \ell_i\}.$$

By introducing auxiliary variables  $\mathbf{e} \in \mathbb{R}^p$  to represent these convex envelopes, we obtain the following linear SDO that is a lower bound to (39):

$$\begin{aligned} & \underset{\mathbf{W}, \boldsymbol{\Phi}, \mathbf{e}}{\text{minimize}} && \langle \mathbf{W}, \boldsymbol{\Sigma} \rangle - \sum_{i=1}^p e_i \\ & \text{s. t.} && \boldsymbol{\Phi} \in \mathcal{F}_{\boldsymbol{\Sigma}} \\ & (\text{LS}_{\boldsymbol{\ell}, \mathbf{u}}) && \left. \begin{aligned} & \mathbf{W} \in \boldsymbol{\Psi}_{p, p-r} \\ & \ell_i \leq \Phi_i \leq u_i \\ & e_i \leq \Phi_i + \ell_i W_{ii} - \ell_i \\ & e_i \leq u_i W_{ii} \end{aligned} \right\} \forall i \end{aligned}$$

The approach now lies in iteratively refining the lower and upper bounds on the diagonal entries of  $\boldsymbol{\Phi}$ , denoted  $\boldsymbol{\ell}$  and  $\mathbf{u}$ , respectively, in order to improve the quality of the approximations obtained via convex envelopes (*cf.* Theorem 4). This classical scheme is known as *spatial branch and bound* and is shown in pseudocode in Algorithm 2 as it applies to solving (39) by way of using (LS <sub>$\boldsymbol{\ell}, \mathbf{u}$</sub> ).

In words, Algorithm 2 involves treating a given “node”  $\mathbf{n} = [\boldsymbol{\ell}, \mathbf{u}]$ , which represents bounds on  $\boldsymbol{\Phi}$ , namely,  $\boldsymbol{\ell} \leq \text{diag}(\boldsymbol{\Phi}) \leq \mathbf{u}$ . Here we solve (LS <sub>$\boldsymbol{\ell}, \mathbf{u}$</sub> ) with the lower and upper bounds  $\boldsymbol{\ell}$  and  $\mathbf{u}$ , respectively, and see whether the resulting new feasible solution is better (lower in objective value) than the best known incumbent solution encountered thus far. We then see if the the bound for this node as obtained via (LS <sub>$\boldsymbol{\ell}, \mathbf{u}$</sub> ) is better than the currently known best feasible solution; if it is not at least the current best feasible solution’s objective value (up to some numerical tolerance), then we must further branch on this node, generating two new nodes  $\mathbf{n}_1$  and  $\mathbf{n}_2$  which partition the existing node  $\mathbf{n}$ . Throughout, we keep track of the worst lower bound encountered, which allows for us to terminate the algorithm early while still having a provable suboptimality guarantee on the best feasible solution  $\boldsymbol{\Phi}_f \in \mathcal{F}_{\boldsymbol{\Sigma}}$  found thus far.

In light of Theorem 4, we have as a corollary the following theorem.

**Theorem 5** *Given numerical tolerance  $TOL > 0$ , Algorithm 2 (with an appropriate branching rule)<sup>12</sup> terminates in finitely many iterations and solves (CFA<sub>1</sub>) to within an additive optimality gap of at most  $TOL$ . Further, if Algorithm 2 is terminated early (*i.e.*, before  $\text{Nodes} = \emptyset$ ), then the best feasible solution  $\boldsymbol{\Phi}_f$  at termination is guaranteed to be within an additive optimality gap of  $z_f - z_{lb}$ .*

12. A branching rule that is sufficient for convergence is selecting  $i \in \arg \max_i (u_i^c - \ell_i^c)$  and  $\alpha = (u_i^c + \ell_i^c)/2$ .

The algorithm we have considered here omits some important details. After discussing properties of  $(\text{LS}_{\ell, \mathbf{u}})$  in Section 4.2, we will discuss various aspects of Algorithm 2. In Section 4.3, we detail how to choose input  $\mathbf{u}^0$ . We then turn our attention to branching (line 5) in Section 4.4. In Section 4.5, we use results from matrix analysis coupled with ideas from the modern practice of discrete optimization to make tailored refinements to Algorithm 2. Finally, in Section 4.6 we include a discussion of node selection strategies.

---

**Algorithm 2** Spatial branch and bound scheme to solve (39). The inputs are as follows: (a) upper bounds  $\mathbf{u}^0$  such that for any  $i$  and any  $\Phi \in \mathcal{F}_{\Sigma}$ , we have  $\Phi_i \leq u_i^0$  (see Section 4.3); (b) optimality tolerance TOL; and (c) initial feasible solution  $\Phi_f \in \mathcal{F}_{\Sigma}$ .

---

1. Initialize  $z_f \leftarrow \sum_{i>r} \sigma_i(\Sigma - \Phi_f)$ ;  $\text{Nodes} \leftarrow \{([\mathbf{0}, \mathbf{u}^0], -\infty)\}$ ; and  $z_{\text{lb}} \leftarrow -\infty$ .
2. While  $\text{Nodes} \neq \emptyset$ , remove some node  $([\ell^c, \mathbf{u}^c], z^c) \in \text{Nodes}$ .
3. Solve  $(\text{LS}_{\ell^c, \mathbf{u}^c})$ . Let  $\Phi$  be an optimal solution with  $z$  the optimal objective value; set  $z_u \leftarrow \sum_{i>r} \sigma_i(\Sigma - \Phi)$ .
4. If  $z_u < z_f$  (i.e., a better feasible solution is found), update the best feasible solution found thus far ( $\Phi_f$ ) to be  $\Phi$  and update the corresponding value ( $z_f$ ) to  $z_u$ .
5. If  $z < z_f - \text{TOL}$  (i.e., a TOL-optimal solution has not yet been found), then pick some  $i \in \{1, \dots, p\}$  and some  $\alpha \in (\ell_i^c, u_i^c)$ . Then add two new nodes to  $\text{Nodes}$ :

$$\left( \prod_{j<i} [\ell_j^c, u_j^c] \times [\ell_i^c, \alpha] \times \prod_{j>i} [\ell_j^c, u_j^c], z \right) \quad \text{and} \quad \left( \prod_{j<i} [\ell_j^c, u_j^c] \times [\alpha, u_i^c] \times \prod_{j>i} [\ell_j^c, u_j^c], z \right).$$

6. Update the best lower bound  $z_{\text{lb}} \leftarrow \min_{([\ell, \mathbf{u}], z) \in \text{Nodes}} z$  and return to Step 2.
- 

**Global Optimization—State of the Art:** We close this section by discussing similarities between the branch-and-bound approach we develop here and existing methods in nonconvex optimization. Our approach is very similar in spirit to approaches to global optimization (Floudas, 1999), and in particular for (nonconvex) quadratic optimization problems, quadratically-constrained convex optimization problems, and bilinear optimization problems (Hansen et al., 1993; Bao et al., 2009; Tawarmalani and Sahinidis, 2002b; Tawarmalani et al., 2013; Costa and Liberti, 2012; Anstreicher and Burer, 2010; Misener and Floudas, 2012). The primary similarity is that we work within a branch and bound framework using successively better convex lower bounds. However, while global optimization software for a variety of nonconvex problems with underlying vector variables is generally well-developed (as evidenced by solvers like BARON, see Sahinidis 2014), this is not the case for problems with underlying matrix variables and semidefinite constraints.

The presence of semidefinite structure presents several substantial computational challenges. First and foremost, algorithmic implementations for solving linear SDOs are not nearly as advanced as those which exist for linear optimization problems. Therefore, each

subproblem, which is itself a linear SDO, carries a larger computational cost than the usual corresponding linear program which typically arises in other global optimization problems with vector variables. Secondly, a critical component of the success of global optimization software is the ability to quickly resolve multiple instances of subproblems which have similar structure. Corresponding methods for SDOs, as solved via interior point methods, are generally not well-developed. Finally, semidefinite structure complicates the traditional process of computing convex envelopes. Such computations are critical to the success of modern global optimization solvers like BARON.

There are a variety of other possible approaches to computing lower bounds to  $(\text{CFA}_q)$ . One such approach is the method of moments (Lasserre, 2009). However, for problems of the size we are considering, such an approach is likely not computationally feasible, so we do not make a direct comparison here. There is also recent work in complementarity constraints literature (Bai et al., 2016) which connects rank-constrained optimization problems to copositive optimization (Burer, 2012). In short, such an approach turns (38) into an equivalent convex problem; despite the transformation, the new problem is not particularly amenable to computation at this time. For this reason, we do not consider the copositive optimization approach.

## 4.2 Properties of $(\text{LS}_{\ell, \mathbf{u}})$

We now examine properties of  $(\text{LS}_{\ell, \mathbf{u}})$ , the main subproblem of interest. Observe that it is a linear SDO problem, and therefore we can consider its dual, namely

$$\begin{aligned}
 & \underset{\substack{q, \mathbf{f}_u, \mathbf{f}_\ell, \boldsymbol{\mu}, \\ \boldsymbol{\sigma}, \mathbf{M}, \mathbf{N}, \mathbf{P}}}{\text{maximize}} && (p-r)q - \mathbf{u}' \mathbf{f}_u - \text{Tr}(\mathbf{N}) - \langle \mathbf{P}, \boldsymbol{\Sigma} - \text{diag}(\boldsymbol{\ell}) \rangle \\
 & \text{s. t.} && \boldsymbol{\mu} + \boldsymbol{\sigma} = \mathbf{1} \\
 & && \text{diag}(\mathbf{P}) + \mathbf{f}_u - \mathbf{f}_\ell - \boldsymbol{\mu} = \mathbf{0} \\
 & && \boldsymbol{\Sigma} - \text{diag}(\mathbf{u}) + \mathbf{M} + \mathbf{N} + \text{diag}(\text{diag}(\mathbf{u} - \boldsymbol{\ell})\boldsymbol{\mu}) - q\mathbf{I} = \mathbf{0} \\
 & && \mathbf{f}_u, \mathbf{f}_\ell, \boldsymbol{\mu}, \boldsymbol{\sigma} \geq \mathbf{0} \\
 & && \mathbf{M}, \mathbf{N}, \mathbf{P} \succcurlyeq \mathbf{0}.
 \end{aligned}
 \tag{DS}_{\ell, \mathbf{u}}$$

**Observation 1** *We now include some remarks about structural properties of  $(\text{LS}_{\ell, \mathbf{u}})$  and its dual  $(\text{DS}_{\ell, \mathbf{u}})$ .*

1. *If  $\text{rank}(\boldsymbol{\Sigma}) = p$  then the Slater condition (Boyd and Vandenberghe, 2004) holds and hence there is strong duality, so we can work with  $(\text{DS}_{\ell, \mathbf{u}})$  instead of  $(\text{LS}_{\ell, \mathbf{u}})$  as an exact reformulation.*
2. *There exists an optimal solution to the dual with  $\mathbf{f}_u = \mathbf{0}$ . This is a variable reduction which is not immediately obvious. Note that  $\mathbf{f}_u$  appears as the multiplier for the constraints in the primal of the form  $\text{diag}(\boldsymbol{\Phi}) \leq \mathbf{u}$ . To claim that we can set  $\mathbf{f}_u = \mathbf{0}$  it suffices to show that the corresponding constraints in the primal can be ignored. Namely, if  $(\mathbf{W}^*, \boldsymbol{\Phi}^*)$  solves  $(\text{LS}_{\ell, \mathbf{u}})$  with the constraints  $\Phi_i \leq u_i \forall i$  omitted, then the pair  $(\mathbf{W}^*, \tilde{\boldsymbol{\Phi}})$  is feasible and optimal to  $(\text{LS}_{\ell, \mathbf{u}})$  with all the constraints included, where  $\tilde{\boldsymbol{\Phi}}$  is defined by*

$$\tilde{\Phi}_i = \min\{\Phi_i^*, u_i\}.$$

Hereinafter we set  $\mathbf{f}_u = \mathbf{0}$  and omit the constraint  $\text{diag}(\Phi) \leq \mathbf{u}$  (with the caveat that, upon solving a problem and identifying some  $\Phi^*$ , we must instead work with  $\min\{\Phi^*, \text{diag}(\mathbf{u})\}$ , taken entrywise).

**Solving Subproblems:** We briefly detail how to solve  $(\text{LS}_{\ell, \mathbf{u}})$ . In light of the discussion in Section 3.2, we choose to apply a first-order method. Observe that we cannot solve the primal form  $(\text{LS}_{\ell, \mathbf{u}})$  within the branch-and-bound framework unless we solve it to optimality. Therefore, we instead choose to work with its dual  $(\text{DS}_{\ell, \mathbf{u}})$ . We apply an off-the-shelf solver SCS (O’Donoghue et al., 2016) to solve  $(\text{DS}_{\ell, \mathbf{u}})$  and find reasonably accurate, *feasible* solutions for this dual problem, which guarantees that we have a lower bound to  $(\text{LS}_{\ell, \mathbf{u}})$ .<sup>13</sup> In this way, we maintain the provable optimality properties of Algorithm 2 without needing to solve nodes in the branch-and-bound tree to full optimality.

### 4.3 Input Parameters

In solving the root node of Algorithm 2, we must begin with some choice of  $\mathbf{u}^0 = \mathbf{u}$ . An obvious first choice for  $u_i$  is  $u_i = \Sigma_{ii}$ , but one can do better. Let us optimally set  $u_i$ , defining it as

$$u_i := \begin{array}{ll} \text{maximize} & x \\ x \in \mathbb{R} & \\ \text{s.t.} & \Sigma - x\mathbf{E}^{(i)} \succcurlyeq \mathbf{0}, \end{array} \quad (40)$$

where  $\mathbf{E}^{(i)} \in \mathbb{R}^{p \times p}$  is a matrix with all zeros except  $\mathbf{E}_{ii}^{(i)} = 1$ . These bounds are useful because if  $\Phi \in \mathcal{F}_\Sigma$ , then  $\Phi_i \leq u_i$ . Note that problem (40) is a linear SDO for which strong duality holds. Its dual is precisely

$$u_i = \begin{array}{ll} \text{minimize} & \langle \mathbf{M}, \Sigma \rangle \\ \mathbf{M} \in \mathbb{R}^{p \times p} & \\ \text{s.t.} & M_{ii} = 1 \\ & \mathbf{M} \succcurlyeq \mathbf{0}, \end{array} \quad (41)$$

a linear SDO in standard form with a single equality constraint. By a result of (Barvinok, 1995; Pataki, 1998), there exists a rank one solution to (41). This implies that (41) can actually be solved as a convex quadratic program:

$$u_i = \begin{array}{ll} \text{minimize} & \mathbf{m}'\Sigma\mathbf{m} \\ \mathbf{m} \in \mathbb{R}^p & \\ \text{s.t.} & m_i^2 = 1. \end{array} = \begin{array}{ll} \text{minimize} & \mathbf{m}'\Sigma\mathbf{m} \\ \mathbf{m} & \\ \text{s.t.} & m_i = 1. \end{array} \quad (42)$$

This formulation given in (42) is computationally inexpensive to solve (given a large number of specialized convex quadratic problem solvers), in contrast to both formulations (40) and (41).<sup>14</sup>

13. One notable feature of the ADMM-based approach is that we can extract an approximately feasible primal solution  $\Phi$ , which is useful for branching. Note that in Algorithm 2, we can replace the best incumbent solution if we find a new  $\Phi$  which has better objective value  $\sum_{i>r} \sigma_i(\Sigma - \Phi)$ . Because  $\Phi$  may not be feasible (i.e.,  $\Phi \notin \mathcal{F}_\Sigma$ ), we take care here. Namely, compute  $t = \sum_{i>r} \lambda_i(\Sigma - \Phi)$ , where  $\lambda_1(\Sigma - \Phi) \geq \dots \geq \lambda_p(\Sigma - \Phi)$  are the sorted eigenvalues of  $\Sigma - \Phi$ . If  $t < z_f$ , then we perform an iteration of CG scheme for finding feasible solutions (outlined in Section 3) to find a feasible  $\bar{\Phi} \in \mathcal{F}_\Sigma$ . We then use this as a possible candidate for replacing the incumbent.

14. In the case when  $\Sigma \succ \mathbf{0}$ , it is not necessary to use quadratic optimization problems to compute  $\mathbf{u}$ . In this case one can apply a straightforward Schur complement argument (Boyd and Vandenberghe, 2004) to show that  $\mathbf{u}$  can be computed by solving for inverse of  $p$  different  $(p-1) \times (p-1)$  matrices (or equivalently, by finding the diagonal of the precision matrix  $\Sigma^{-1}$ ).

#### 4.4 Branching

Here we detail two methods for branching (line 5 in Algorithm 2). The problem of branching is as follows: having solved  $(\text{LS}_{\ell, \mathbf{u}})$  for some particular  $\mathbf{n} = [\ell, \mathbf{u}]$ , we must choose some  $i \in \{1, \dots, p\}$  and split the interval  $[\ell_i, u_i]$  to create two new subproblems. We begin with a simple branching rule. Given a solution  $(\mathbf{W}^*, \Phi^*, \mathbf{e}^*)$  to  $(\text{LS}_{\ell, \mathbf{u}})$ , compute  $i \in \operatorname{argmax}_i |e_i^* - W_{ii}^* \Phi_i^*|$  and branch on variable  $\Phi_i$ , generating two new subproblems with the intervals

$$\prod_{j<i} [\ell_j, u_j] \times [\ell_i, \Phi_i^*] \times \prod_{j>i} [\ell_j, u_j] \quad \text{and} \quad \prod_{j<i} [\ell_j, u_j] \times [\Phi_i^*, u_i] \times \prod_{j>i} [\ell_j, u_j].$$

Observe that, so long as  $\max_i |e_i^* - W_{ii}^* \Phi_i^*| > 0$ , the solution  $(\mathbf{W}^*, \Phi^*, \mathbf{e}^*)$  is not optimal for either of the subproblems created.

We now briefly describe an alternative rule which we employ instead. We again pick the branching index  $i$  as before, but now the two new nodes we generate are

$$\prod_{j<i} [\ell_j, u_j] \times [\ell_i, (1-\epsilon)\Phi_i^* + \epsilon\ell_i] \times \prod_{j>i} [\ell_j, u_j] \quad \text{and} \quad \prod_{j<i} [\ell_j, u_j] \times [(1-\epsilon)\Phi_i^* + \epsilon u_i, u_i] \times \prod_{j>i} [\ell_j, u_j],$$

where  $\epsilon \in [0, 1)$  is some parameter. For the computational experiments, we set  $\epsilon = 0.4$ .

Such an approach, which lowers the location of the branch in the  $i$ th interval  $[\ell_i, u_i]$  from  $\Phi_i^*$ , serves to improve the objective value from the first node, while hurting the objective value from the second node (here by objective value, we mean the objective value of the optimal solution to the two new subproblems). In this way, it spreads out the distance between the two, and so it is more likely that the first node may have an objective value that is higher than  $z_f - \text{TOL}$  than before, and hence, this would mean there are fewer nodes necessary to consider to solve for an additive gap of  $\text{TOL}$ . While this heuristic explanation is only partially satisfying, we have observed throughout a variety of numerical experiments that this rule, even though simple, performs better across a variety of example classes than the basic branching rule outlined. At the same time, recent work on the theory of branching rules supports such a heuristic rule (le Bodic and Nemhauser, 2015). In Section 5.4, we give evidence on the impact of the use of the modified branching rule.

#### 4.5 Weyl's Method—Pruning and Bound Tightening

In this subsection, we develop another method for lower bounds for the factor analysis problem. While we use it to supplement our approach detailed throughout Section 4, it is of interest as a standalone method, particularly for its computational speed and simplicity. In Section 5.3, we discuss the performance of this approach in both contexts.

The central problem of interest in factor analysis involves the spectrum of a symmetric matrix  $(\Sigma - \Phi)$  which is the difference of two other symmetric matrices ( $\Sigma$  and  $\Phi$ ). From a linear algebraic perspective, the spectrum of the sum of real symmetric matrices is an extensively studied problem (Horn and Johnson, 2013; Bhatia, 2007). Therefore, it is natural to inquire how such results carry over to our setting. We discuss the implications of a well-known result from this literature, namely *Weyl's inequality* (see, e.g., Bhatia 2007):

**Theorem 6 (Weyl’s Inequality)** For symmetric matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{p \times p}$  with sorted eigenvalues

$$\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \dots \geq \lambda_p(\mathbf{A}) \quad \text{and} \quad \lambda_1(\mathbf{B}) \geq \lambda_2(\mathbf{B}) \geq \dots \geq \lambda_p(\mathbf{B})$$

one has for any  $k \in \{1, \dots, p\}$  that

$$\lambda_k(\mathbf{A} + \mathbf{B}) \geq \lambda_{k+j}(\mathbf{A}) + \lambda_{p-j}(\mathbf{B}) \quad \forall j = 0, \dots, p - k.$$

For any vector  $\mathbf{x} \in \mathbb{R}^p$  we let  $\{x_{(i)}\}_{i=1}^p$  denote sorted  $\{x_i\}_{i=1}^p$  with  $x_{(1)} \geq x_{(2)} \geq \dots \geq x_{(p)}$ . Using this notation, we arrive at the following theorem.

**Theorem 7** For any diagonal matrix  $\bar{\Phi}$  one has that

$$\underset{\Phi \in \mathcal{F}_\Sigma}{\text{minimize}} \sum_{i>r} \sigma_i(\Sigma - \Phi) \geq \underset{\Phi \in \mathcal{F}_\Sigma}{\text{minimize}} \left( \sum_{i>r} \max_{j=0, \dots, p-i} \left\{ \lambda_{i+j}(\Sigma - \bar{\Phi}) + \text{diag}(\bar{\Phi} - \Phi)_{(p-j)}, 0 \right\} \right).$$

**Proof** Apply Weyl’s inequality with  $\mathbf{A} = \Sigma - \text{diag}(\bar{\Phi})$  and  $\mathbf{B} = \bar{\Phi} - \Phi$ , and use the fact that  $\Phi \in \mathcal{F}_\Sigma$  so  $\Sigma - \Phi \succcurlyeq \mathbf{0}$ . ■

**Weyl’s Method:** The new lower bound introduced in Theorem 7 is a nonconvex problem in general. We begin by discussing one situation in which Theorem 7 provides computationally tractable (and fast) lower bounds; we deem this *Weyl’s method*, detailed as follows:

1. Compute bounds  $u_i$  as in (40), so that for all  $\Phi \in \mathcal{F}_\Sigma$ , one has  $\Phi_i \leq u_i \forall i$ .
2. For each  $r \in \{1, \dots, p\}$ , one can compute a lower bound to (CFA<sub>1</sub>) (for a given  $r$ ) of

$$\sum_{i>r} \max\{\lambda_i(\Sigma - \text{diag}(\mathbf{u})), 0\},$$

by taking Theorem 7 with  $\bar{\Phi} = \text{diag}(\mathbf{u})$ .

As per the above remarks, computing  $\mathbf{u}$  in Step 1 of Weyl’s method can be carried out efficiently. Step 2 only relies on computing the eigenvalues of  $\Sigma - \text{diag}(\mathbf{u})$ . Therefore, this lower bounding procedure is quite simple to carry out. What is perhaps surprising is that this simple lower bounding procedure is effective as a standalone method. We describe such results in Section 5.3. We now turn our attention to how Weyl’s method can be utilized within the branch and bound tree as described in Algorithm 2.

**Pruning:** We begin by considering how Weyl’s method can be used for *pruning*. The notion of pruning for branch and bound trees is grounded in the theory and practice of discrete optimization. In short, pruning is the elimination of nodes from the tree without actually solving them. We make this precise in our context.

Consider some point in the branch and bound process in Algorithm 2, where we have some collection of nodes,  $([\ell^c, \mathbf{u}^c], z^c) \in \text{Nodes}$ . Recall that  $z^c$  is the optimal objective

value of the parent node of  $\mathbf{n}$ . Vis-à-vis Weyl’s method, we know *a priori*, without solving  $(\text{LS}_{\ell^c, \mathbf{u}^c})$ , that

$$\underset{\substack{\Phi \in \mathcal{F}_\Sigma \\ \ell^c \leq \text{diag}(\Phi) \leq \mathbf{u}^c}}{\text{minimize}} \sum_{i>r} \sigma_i(\Sigma - \Phi) \geq \sum_{i>r} \max\{\lambda_i(\Sigma - \text{diag}(\mathbf{u}^c)), 0\}.$$

Hence, if  $z_f - \text{TOL} < \sum_{i>r} \max\{\lambda_i(\Sigma - \text{diag}(\mathbf{u}^c)), 0\}$ , where  $z_f$  is as in Algorithm 2, then node  $\mathbf{n}$  can be discarded, i.e., there is no need to actually compute  $(\text{LS}_{\ell^c, \mathbf{u}^c})$  or further consider this branch. This is because if we were to solve  $(\text{LS}_{\ell^c, \mathbf{u}^c})$ , and then branch again, solving further down this branch to optimality, then the final lower bound obtained would necessarily be at least as large as the best feasible objective already found (within tolerance TOL).

In this way, because Weyl’s method is relatively fast, this provides a simple method for pruning. In the computational results detailed in Section 5.3, we always use Weyl’s method to discard nodes which are not fruitful to consider.

**Bound Tightening:** We now turn our attention to another way in which Weyl’s method can be used to improve the performance of Algorithm 2—*bound tightening*. In short, bound tightening is the use of implicit constraints to strengthen bounds obtained for a given node. We detail this with the same node notation as above. Namely, consider a given node  $\mathbf{n} = [\ell^c, \mathbf{u}^c]$ . Fix some  $j \in \{1, \dots, p\}$  and let  $\alpha \in (\ell_j^c, u_j^c)$ . If we have that

$$z_f - \text{TOL} < \sum_{i>r} \max\{\lambda_i(\Sigma - \text{diag}(\tilde{\mathbf{u}})), 0\},$$

where  $\tilde{\mathbf{u}}$  is  $\mathbf{u}^c$  with the  $j$ th entry replaced by  $\alpha$ , then we can replace the node  $\mathbf{n}$  with the “tightened” node  $\tilde{\mathbf{n}} = [\tilde{\ell}, \mathbf{u}^c]$ , where  $\tilde{\ell}$  is  $\ell^c$  with the  $j$ th entry replaced by  $\alpha$ .

We consider why this is valid. Suppose that one were to solve  $(\text{LS}_{\ell^c, \mathbf{u}^c})$  and choose to branch on index  $j$  at  $\alpha$ . Then one would create two new nodes:  $[\ell^c, \tilde{\mathbf{u}}]$  and  $[\tilde{\ell}, \mathbf{u}^c]$ . We would necessarily then prune away the node  $[\ell^c, \tilde{\mathbf{u}}]$  as just described; hence, we can replace  $[\ell^c, \mathbf{u}^c]$  without loss of generality with  $[\tilde{\ell}, \mathbf{u}^c]$ . Note that here  $\alpha \in (\ell_j^c, u_j^c)$  and  $j \in \{1, \dots, p\}$  were arbitrary. Hence, for each  $j$ , one can choose the largest such  $\alpha_j \in (\ell_j^c, u_j^c)$  (if one exists) so that  $z_f - \text{TOL} < \sum_{i>k} \max\{\lambda_i(\Sigma - \text{diag}(\tilde{\mathbf{u}})), 0\}$ , and then replace  $\ell^c$  by  $\tilde{\ell}$ .<sup>15</sup>

Such a procedure is somewhat expensive (because of its use of repeated eigenvalue calculations), but can be thought of as “optimal” pruning via Weyl’s method. In our experience the benefit of bound tightening does not warrant its computational cost when used at every node in the branch-and-bound tree except in a small number of problems. For this reason, in the computational results in Section 5.3 we only employ bound tightening at the root node  $\mathbf{n} = [\mathbf{0}, \mathbf{u}^0]$ .

## 4.6 Node Selection

In this section, we briefly describe our method of node selection. The problem of node selection has been considered extensively in discrete optimization and is still an active area of research. Here we describe a simple node selection strategy.

15. An obvious choice to find such an  $\alpha$  is a grid-search-based bisection method. For simplicity we use a linear search on a grid instead of resorting to the bisection method.

To be precise, consider some point in Algorithm 2 where we have a certain collection of nodes,  $([\ell^c, \mathbf{u}^c], z^c) \in \mathbf{Nodes}$ . The most obvious node selection strategy is to pick the node  $\mathbf{n}$  for which  $z^c$  is smallest among all nodes in  $\mathbf{Nodes}$ . In this way, the algorithm is likely to improve the gap  $z_f - z_{\text{lb}}$  at every iteration. Such greedy selection strategies tend to not perform particularly well in general global optimization problems (see, e.g., Tawarmalani and Sahinidis, 2002a).

For these reasons, we employ a slightly modified greedy selection strategy which utilizes Weyl’s method. For a given node  $\mathbf{n}$ , we also consider its corresponding lower bound  $w^c$  obtained from Weyl’s method, namely,  $w^c := \sum_{i>r} \max\{\lambda_i(\Sigma - \text{diag}(\mathbf{u}^c)), 0\}$ . For each node, we now consider  $\max\{z^c, w^c\}$ . There are two cases to consider:

1. With probability  $\beta$ , we select the node with smallest value of  $\max\{z^c, w^c\}$ .
2. In the remaining scenarios (occurring with probability  $1 - \beta$ ), we choose randomly between selecting the node with smallest value of  $z^c$  and the node with smallest value of  $w^c$ . To be precise, let  $Z$  be the minimum of  $z^c$  over all nodes and likewise let  $W$  be the minimum of  $w^c$  over all nodes. Then with (independent) probability  $\beta$ , we choose the node with worst  $z^c$  or  $w^c$  (i.e., with  $\min\{z^c, w^c\} = \min\{Z, W\}$ ); with probability  $1 - \beta$ , if  $Z < W$  we choose a node with  $w^c = W$ , and if  $Z > W$  we choose a node with  $z^c = Z$ .

In this way, we allow for the algorithm to switch between trying to make progress towards improving the convex envelope bounds and making progress towards improving the best of the two bounds (the convex envelope bounds along with the Weyl bounds). We set  $\beta = 0.9$  for all computational experiments. It is possible that a more dynamic branching strategy could perform substantially better; however, the method here has a desirable level of simplicity.

We close by noting that while this node selection strategy appears naïve, it is not necessarily so simple. Improved node selection strategies from discrete optimization often take into account some sort of duality theory. Weyl’s inequality is at its core a result from duality theory (principally Wielandt’s minimax principle; see Bhatia, 2007), and therefore our strategy is not as unsophisticated as it might appear on first inspection.

## 5. Computational experiments

In this section, we perform various computational experiments to study the properties of our different algorithmic proposals for  $(\text{CFA}_q)$ , for  $q \in \{1, 2\}$ . Using a variety of statistical measures, we compare our methods with existing popular approaches for FA, as implemented in standard R statistical packages `psych` (Revelle, 2015), `nFactors` (Raiche and Magis, 2011), and `GPArotation` (Bernaards and Jennrich, 2005). We then turn our attention to certificates of optimality as described in Section 4 for  $(\text{CFA}_1)$ .

### 5.1 Synthetic examples

For our synthetic experiments, we considered distinctly different groups of examples. Classes  $A_1$  and  $A_2$  have subspaces of the low-rank common factors which are random and the values



of  $\Phi_i$  are taken to be equally spaced. The underlying matrix corresponding to the common factors in type  $A_1$  is exactly low-rank, while this is not the case in type  $A_2$ .

**Class  $A_1(R/p)$ .** We generated a matrix  $\mathbf{L}_{p \times R} := ((L_{ij}))$  (with  $R < p$ ) with  $L_{ij} \stackrel{\text{iid}}{\sim} N(0, 1)$ . The *unique variances*  $\Phi_1, \dots, \Phi_p$ , are taken to be proportional to  $p$  equi-spaced values on the interval  $[\lambda_R(\mathbf{L}'\mathbf{L}), \lambda_1(\mathbf{L}'\mathbf{L})]$  such that  $\Phi_i = \bar{\phi} \cdot \left( \lambda_1(\mathbf{L}'\mathbf{L}) + (\lambda_R(\mathbf{L}'\mathbf{L}) - \lambda_1(\mathbf{L}'\mathbf{L})) \frac{i-1}{p} \right)$ , for  $1 \leq i \leq p$ . Here  $\bar{\phi}$ , which controls the ratio of the variances between the *uniquenesses* and the common latent factors is chosen such that  $\sum_{i=1}^p \Phi_i = \text{Tr}(\mathbf{L}\mathbf{L}')$ , i.e., the contribution to the total variance from the common factors matches that from the uniqueness factors. The covariance matrix is thus given by:  $\Sigma = \mathbf{L}\mathbf{L}' + \Phi$ .

**Class  $A_2(p)$ .** Here  $\mathbf{L}_{p \times p}$  is generated as  $L_{ij} \stackrel{\text{iid}}{\sim} N(0, 1)$ . We did a full singular value decomposition on  $\mathbf{L}$ —let  $\mathbf{U}_L$  denote the set of  $p$  (left) singular vectors. We created a positive definite matrix with exponentially decaying eigenvalues as follows:  $\tilde{\mathbf{L}}\tilde{\mathbf{L}}' = \mathbf{U}_L \text{diag}(\lambda_1, \dots, \lambda_p)\mathbf{U}_L'$ , where the eigenvalues were chosen as  $\lambda_i = 0.8^{i/2}$ ,  $i = 1, \dots, p$ . We chose the diagonal entries of  $\Phi$  (like data Type- $A_1$ ), as a scalar multiple ( $\bar{\phi}$ ) of a uniformly spaced grid in  $[\lambda_p, \lambda_1]$  and  $\bar{\phi}$  was chosen such that  $\sum_i \Phi_i = \text{Tr}(\tilde{\mathbf{L}}\tilde{\mathbf{L}}')$ .

In contrast, classes  $B_1$ ,  $B_2$ , and  $B_3$  are qualitatively different from the aforementioned ones—the subspaces corresponding to the common factors are more structured, and hence different from the coherence-like assumptions on the eigenvectors which are necessary for nuclear norm based methods (Saunderson et al., 2012) to work well.

**Class  $B_1(R/p)$ .** We set  $\Theta = \mathbf{L}\mathbf{L}'$ , where  $\mathbf{L}_{p \times R}$  is given by

$$L_{ij} = \begin{cases} 1, & i \leq j \\ 0, & i > j. \end{cases}$$

**Class  $B_2(r/R/p)$ .** Here we set  $\Theta = \mathbf{L}\mathbf{L}'$ , where  $\mathbf{L}_{p \times R}$  is such that

$$L_{ij} = \begin{cases} 1, & i, j = 1, \dots, r \\ \stackrel{\text{iid}}{\sim} N(0, 1), & i > r, j = 1, \dots, R \\ 0, & i = 1, \dots, r, j > r. \end{cases}$$

**Class  $B_3(r/R/p)$ .** Here we define  $\Theta = \mathbf{L}\mathbf{L}'$ , where  $\mathbf{L}_{p \times R}$  is such that

$$L_{ij} = \begin{cases} 1, & j = 1, \dots, r, i \leq j \\ \stackrel{\text{iid}}{\sim} N(0, 1), & j > r, i = 1, \dots, R \\ 0, & i > j, j = 1, \dots, r. \end{cases}$$

In all the  $B$  classes, we generated  $\Phi_i \stackrel{\text{iid}}{\sim} \text{abs}(N(0, 1))$  and the covariance matrix  $\Sigma$  was taken to be  $\Sigma = \Theta + \alpha\Phi$ , where  $\alpha$  is so that  $\text{Tr}(\Theta) = \alpha \text{Tr}(\Phi)$ .

## COMPARISONS WITH OTHER FA METHODS

We performed a suite of experiments using Algorithm 1 for the cases  $q \in \{1, 2\}$ . We compared our proposed algorithm with the following popular FA estimation procedures as described in Section 1.1:

1. MINRES: minimum residual factor analysis
2. WLS: weighted least squares method with weights being the *uniquenesses*
3. PA: this is the principal axis factor analysis method
4. MTFA: constrained minimum trace factor analysis—formulation (6)
5. PC: The method of principal component factor analysis
6. MLE: this is the maximum likelihood estimator (MLE)
7. GLS: the generalized least squares method

For MINRES, WLS, GLS, and PA, we used the implementations available in the R package `psych` (Revelle, 2015) available from CRAN. For MLE we investigated the methods `factanal` from R package `stats` and the `fa` function from R package `psych`. The estimates obtained by the different MLE implementations were similar in our experiments; therefore, we report the results obtained from `factanal`.

For MTFA, we used our own implementation by adapting the ADMM algorithm (Section 3.2.2) to solve Problem (6). For the experiments in Section 5.1, we took the convergence thresholds for Algorithm 1 as  $\text{TOL} = 10^{-5}$  and ADMM as  $\text{TOL} \times \alpha = 10^{-9}$ . For the PC method we followed the description in Bai and Ng (2008) (as described in Section 1.1)—the  $\Phi$  estimates were thresholded at zero if they became negative.

Note that all the methods considered in the experiments, apart from MTFA, allow the user to specify the desired number of factors in the problem. Since standard implementations of MINRES, WLS and PA require  $\Sigma$  to be a correlation matrix, we standardized all covariance matrices to correlation matrices at the outset.

### 5.1.1 PERFORMANCE MEASURES

We consider the following measures of “goodness of fit” (see Bartholomew et al. 2011 and references therein) to assess the performances of the different FA estimation procedures.

**Estimation error in  $\Phi$ :** We use the following measure to assess the quality of an estimator for  $\Phi$ :

$$\text{Error}(\Phi) := \sum_{i=1}^p (\hat{\Phi}_i - \Phi_i)^2. \quad (43)$$

The estimation of  $\Phi$  plays an important role in FA—given a good estimate  $\hat{\Phi}$ , the  $r$ -common factors can be obtained by a rank- $r$  eigendecomposition on the residual covariance matrix  $\Sigma - \hat{\Phi}$ .

**Proportion of variance explained and semi-definiteness of  $(\Sigma - \Phi)$ :** A fundamental objective in FA lies in understanding how well the  $r$ -common factors explain the residual covariance, i.e.,  $(\Sigma - \hat{\Phi})$ —a direct analogue of what is done in PCA, as explained in Section 1. For a given  $r$ , the proportion of variance explained by the  $r$  common factors is given by

$$\text{Explained Variance} = \sum_{i=1}^r \lambda_i(\hat{\Theta}) / \sum_{i=1}^p \lambda_i(\Sigma - \hat{\Phi}). \quad (44)$$

As  $r$  increases, the explained variance increases to one. This trade-off between  $r$  and “Explained Variance” plays an important role in exploratory FA and in particular the choice

of  $r$ . For the expression (44) to be meaningful, it is desirable to have  $\Sigma - \widehat{\Phi} \succeq \mathbf{0}$ . Note that our framework (CFA $_q$ ), and in particular MTFA, estimates  $\Phi$  under a PSD constraint on  $\Sigma - \Phi$ . However, as seen in our experiments ( $\widehat{\Phi}, \widehat{\Theta}$ ) estimated by the remaining methods MINRES, PA, WLS, GLS, MLE and others often violate the PSD condition on  $\Sigma - \widehat{\Phi}$  for some choices of  $r$ , thereby rendering the interpretation of ‘‘Explained Variance’’ troublesome.

For the MTFA method with estimator  $\widehat{\Theta}$ , the measure (44) applies only for the value of  $r = \text{rank}(\widehat{\Theta})$  and the explained variance is one.

For the methods we have included in our comparisons, we report the values of ‘‘Explained Variance’’ as delivered by the R-package implementations.

**Proximity between  $\widehat{\Theta}$  and  $\Theta$ :** A relevant measure of the proximity between  $\Theta$  and its estimate ( $\widehat{\Theta}$ ) is given by

$$\text{Error}(\Theta) := \|\widehat{\Theta} - \Theta_r\|_2^2 / \|\Theta_r\|_2^2, \quad (45)$$

where  $\Theta_r$  is the best rank- $r$  approximation to  $\Theta$  and can be viewed as the natural ‘‘oracle’’ counterpart of  $\widehat{\Theta}$ . Note that MTFA does not incorporate any constraint on  $\text{rank}(\widehat{\Theta})$  in its formulation. Since the estimates obtained by this procedure satisfy  $\widehat{\Theta} = \Sigma - \widehat{\Phi}$ ,  $\text{rank}(\widehat{\Theta})$  may be quite different from  $r$ .

**Discussion of experimental results.** We next discuss our findings from the numerical experiments for the synthetic datasets.

Table 1 shows the performances of the various methods for different choices of  $p$  and  $R$  for class  $A_1$ . For the problems (CFA $_q$ ),  $q \in \{1, 2\}$ , we present the results of Algorithm 1. (Results obtained by the approach in Appendix B were similar.) In all the examples, with the exception of MTFA, we set the number of factors to be  $r = (R - 1)$ , one less than the rank of the covariance matrix for the common underlying factors; in other words, the ‘‘remaining’’ rank one component can be considered as noise. For MTFA, the rank of  $\widehat{\Theta}$  was computed as the number of eigenvalues of  $\widehat{\Theta}$  larger than  $10^{-5}$ . MTFA and (CFA $_q$ ),  $q \in \{1, 2\}$ , estimate  $\Phi$  with zero error—significantly better than competing methods. While MTFA and (CFA $_q$ ),  $q \in \{1, 2\}$ , result in estimates such that  $\Sigma - \widehat{\Phi}$  is PSD, other methods, however, fail to do so; indeed, the discrepancy can often be quite large. MTFA performs poorly in terms of estimating  $\Theta$  since the estimated  $\Theta$  has rank different than  $r$ . In terms of the proportion of variance explained (CFA $_q$ ) performs significantly better than all other methods. The notion of ‘‘Explained Variance’’ by MTFA for  $r = (R - 1)$  is not applicable since the rank of the estimated  $\Theta$  is larger than  $r$ .

Figure 1 displays results for type  $A_2$ . Here we present the results for (CFA $_q$ ),  $q \in \{1, 2\}$ , using Algorithm 1. For all the methods (with the exception of MTFA) we computed estimates of  $\Theta$  and  $\Phi$  for a range of values of  $r$ . MTFA and (CFA $_1$ ) do a perfect job in estimating  $\Phi$  and both deliver PSD matrices  $\Sigma - \widehat{\Phi}$ . MTFA computes solutions ( $\widehat{\Theta}$ ) with a higher numerical rank and with large errors in estimating  $\Theta$  (for smaller values of  $r$ ). Among the four performance measures corresponding to MTFA,  $\text{Error}(\Theta)$  is the only one that varies with different  $r$  values. Each of the other three measures deliver a single value corresponding to  $r = \text{rank}(\widehat{\Theta})$ . Overall, it appears that (CFA $_q$ ) is significantly better than all other methods.

Figure 2 shows the results for classes  $B_1, B_2$ , and  $B_3$ . We present the results for (CFA $_q$ ) for  $q \in \{1, 2\}$  using Algorithm 1 as before. Figure 2 shows the performance of the different

Performance measure	Method Used								Problem Size ( $R/p$ )
	(CFA <sub>1</sub> )	(CFA <sub>2</sub> )	MTFA	MINRES	WLS	PA	PC	MLE	
Error( $\Phi$ )	0.0	0.0	0.0	699.5	700.1	700.0	639.9	699.6	3/200
Explained Var.	0.6889	0.6889	-	0.2898	0.2898	0.2898	0.2956	0.2898	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.6086	-0.6194	-0.6195	-0.6140	-0.6034	
Error( $\Theta$ )	0.0	0.0	689.68	0.1023	0.1079	0.1146	0.7071	0.1111	
Error( $\Phi$ )	0.0	0.0	0.0	197.18	197.01	196.98	139.60	197.18	5/200
Explained Var.	0.8473	0.8473	-	0.3813	0.3813	0.3814	0.3925	0.3813	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.4920	-0.5071	-0.5081	-0.4983	-0.4919	
Error( $\Theta$ )	0.0	0.0	190.26	0.0726	0.0802	0.0817	1.1638	0.0725	
Error( $\Phi$ )	0.0	0.0	0.0	39.85	39.18	39.28	2.47	40.04	10/200
Explained Var.	0.9371	0.9371	-	0.4449	0.4452	0.4452	0.4686	0.4449	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.2040	-0.2289	-0.2329	-0.2060	-0.2037	
Error( $\Theta$ )	0.0	0.0	35.94	0.0531	0.0399	0.0419	2.47	0.0560	
Error( $\Phi$ )	0.0	0.0	0.0	3838	3859	3858	3715	3870	2/500
Explained Var.	0.7103	0.7103	-	0.3036	0.3033	0.3033	0.3056	0.3031	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.7014	-0.7126	-0.7125	-0.7121	-0.7168	
Error( $\Theta$ )	0.0	0.0	3835.6	0.0726	0.0770	0.0766	0.5897	0.1134	
Error( $\Phi$ )	0.0	0.0	0.0	1482.9	1479.1	1478.9	1312.9	1485.1	5/500
Explained Var.	0.8311	0.8311	-	0.3753	0.3754	0.3754	0.3798	0.3752	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.5332	-0.5433	-0.5445	-0.5423	-0.5379	
Error( $\Theta$ )	0.0	0.0	1459.5	0.1176	0.0752	0.0746	1.1358	0.1301	
Error( $\Phi$ )	0.0	0.0	0.0	301.94	301.08	301.04	160.29	302.1	10/500
Explained Var.	0.9287	0.9287	-	0.4443	0.4444	0.4444	0.4538	0.4443	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.3290	-0.3280	-0.3281	-0.3210	-0.3302	
Error( $\Theta$ )	0.0	0.0	291.44	0.0508	0.0415	0.0412	2.3945	0.0516	
Error( $\Phi$ )	0.0	0.0	0.0	19123	19088	19090	18770	19120	2/1000
Explained Var.	0.6767	0.6767	-	0.2885	0.2886	0.2886	0.2898	0.2885	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.6925	-0.7335	-0.7418	-0.7422	-0.6780	
Error( $\Theta$ )	0.0	0.0	19037	50.76	0.3571	0.0984	0.8092	18.5930	
Error( $\Phi$ )	0.0	0.0	0.0	6872	6862	6861	6497	6876	5/1000
Explained Var.	0.8183	0.8183	-	0.3716	0.3717	0.3717	0.3739	0.3716	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.6202	-0.6785	-0.6797	-0.6788	-0.6209	
Error( $\Theta$ )	0.0	0.0	6818.1	0.1706	0.0776	0.0780	1.1377	0.1696	
Error( $\Phi$ )	0.0	0.0	0.0	1682	1681	1681	1311	1682	10/1000
Explained Var.	0.9147	0.9147	-	0.4360	0.4360	0.4360	0.4408	0.4360	
$\lambda_{\min}(\Sigma - \hat{\Phi})$	0.0	0.0	0.0	-0.2643	-0.2675	-0.2676	-0.2631	-0.2643	
Error( $\Theta$ )	0.0	0.0	1654.3	0.0665	0.0568	0.0570	2.4202	0.0665	

Table 1: Comparative performances of the various FA methods for data of type  $A_1$ , for different choices of  $R$  and  $p$ . In all the above methods (apart from MTFA)  $r$  was taken to be  $(R - 1)$ . In all the cases  $\text{rank}(\hat{\Theta})$  obtained by MTFA is seen to be  $R$ . The “-” symbol implies that the notion of explained variance is not meaningful for MTFA for  $r = R - 1$ . No method in Category (B) satisfies  $\Sigma - \Phi \succeq \mathbf{0}$ . Methods proposed in this paper seem to significantly outperform their competitors, across the different performance measures.

methods in terms of four different metrics: error in  $\Phi$  estimation, proportion of variance explained, violation of the PSD constraint on  $\Sigma - \Phi$ , and error in  $\Theta$  estimation. For the case of  $B_1$ , we see that the proportion of explained variance for  $(CFA_q)$  reaches one at a rank smaller than that of MTFA—this shows that the nonconvex criterion  $(CFA_q)$  provides smaller estimates of the rank than its convex relaxation MTFA when one seeks a model that explains the full proportion of residual variance. This result is qualitatively different from the behavior seen for  $A_1$  and  $A_2$  where the benefit of  $(CFA_q)$  over MTFA was mainly due to its flexibility to control the rank of  $\Theta$ . Algorithms in Category (A) do an excellent job in estimating  $\Phi$ . All other competing methods perform poorly in estimating  $\Phi$  for small/moderate values of  $r$ . We observe that none of the methods apart from  $(CFA_q)$  and MTFA lead to PSD estimates of  $\Sigma - \hat{\Phi}$  (unless  $r$  becomes sufficiently large which corresponds to a model with a saturated fit). In terms of the proportion of variance explained, our proposal performs much better than the competing methods. We see that the error in  $\Theta$  estimation incurred by  $(CFA_q)$ , increases marginally as soon as the rank  $r$  becomes larger than a certain value for  $B_1$ . Note that around the same values of  $r$ , the proportion of explained variance reaches one in both these cases, thereby suggesting that this is possibly not a region of statistical interest. In summary, Figure 2 suggests that  $(CFA_q)$  performs very well compared to all its competitors.

**Summary:** All methods of Category (B) (see Section 1.2) used in the experimental comparisons perform worse than Category (A) in terms of measures  $\text{Error}(\Phi)$ ,  $\text{Error}(\Theta)$  and Explained Variance for small/moderate values of  $r$ . They also lead to indefinite estimates of  $\Sigma - \hat{\Phi}$ . MTFA performs well in estimating  $\Phi$  but fails in estimating  $\Theta$  mainly due to the lack in flexibility of imposing a rank constraint; in some cases the trace heuristic falls short of doing a good job in approximating the rank function when compared to its nonconvex counterpart  $(CFA_q)$ . The estimation methods proposed herein have a significant edge over existing methods in producing high quality solutions across various performance metrics.

## 5.2 Real data examples

This section describes the performance of different FA methods on some real-world benchmark datasets popularly used in the context of FA. These datasets can be found in the R libraries `datasets` (R Core Team, 2014), `psych` (Revelle, 2015), and `FactoMineR` (Husson et al., 2015) and are as follows:

- The `bfi` data set has 2800 observations on 28 variables (25 personality self-reported items and 3 demographic variables).
- The `neo` data set has 1000 measurements for  $p = 30$  dimensions.
- The `Harman` data set is a correlation matrix of 24 psychological tests given to 145 seventh- and eighth-grade children.
- The `geomorphology` data set is a collection of geomorphological data collected across  $p = 10$  variables and 75 observations. (The `geomorphology` data set originally has  $p = 11$ , but we remove the one categorical feature, leaving  $p = 10$ .)

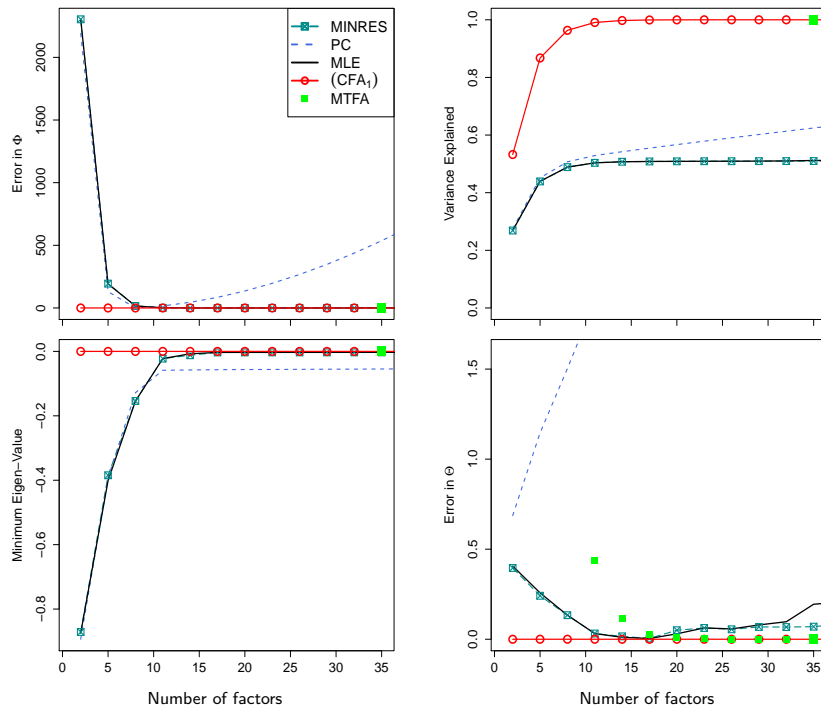


Figure 1: Performance of various FA methods for examples from class  $A_2(p = 200)$  as a function of the number of factors. The  $y$ -axis label “Minimum Eigenvalue” refers to  $\lambda_{\min}(\Sigma - \hat{\Phi})$ . We present the results of  $(CFA_1)$ , as obtained via Algorithm 1—the results of  $(CFA_2)$  were similar, and hence omitted from the plot. Our methods seems to perform better than all the other competitors. For large values of  $r$ , as the fits saturate, the methods become similar. The methods (as available from the R package implementations) that experienced convergence difficulties do not appear in the plot.

- The JO data set records athletic performance in Olympic events, and involves 24 observations with  $p = 58$ . This example is distinctive because the corresponding correlation matrix  $\Sigma$  is not full rank (having more variables than observations).

We present the results in Figure 3. We also experimented with other methods—such as WLS and GLS—but the results were similar to MINRES and hence have not been shown in the figure. For the real examples, most of the performance measures described in Section 5.1.1 do not apply<sup>16</sup>; however, the notion of explained variance (44) does apply. We used this metric to compare the performance of different competing estimation procedures. We observe that solutions delivered by Category (B) explain the maximum amount of residual variance for a given rank  $r$ , which is indeed desirable, especially in the light of its analogy with PCA on the residual covariance matrix  $\Sigma - \Phi$ .

16. Understanding performance of FA methods on real data is difficult because it is an unsupervised problem. However, we can understand the performances of different methods by drawing parallels with PCA in terms of the proportion of variance explained of the matrix  $\Sigma - \Phi$ —see our discussion in Section 1.

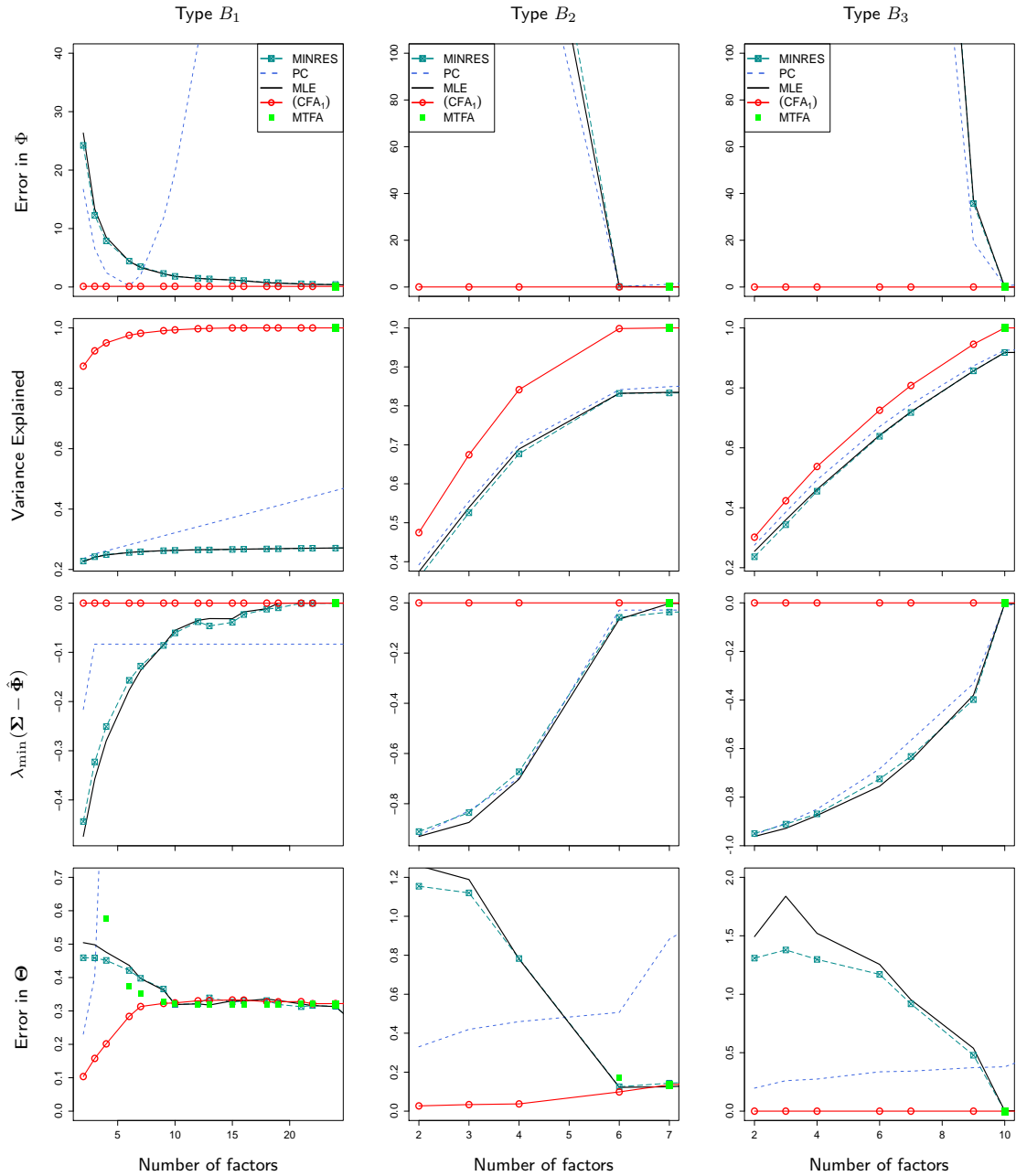


Figure 2: Performance of different methods for instances of  $B_1(30/100)$ ,  $B_2(5/10/100)$ , and  $B_3(5/10/100)$ . We see that (CFA)<sub>1</sub> exhibits very good performance across all instances, significantly outperforming the competing methods (the results of (CFA)<sub>2</sub> were similar).

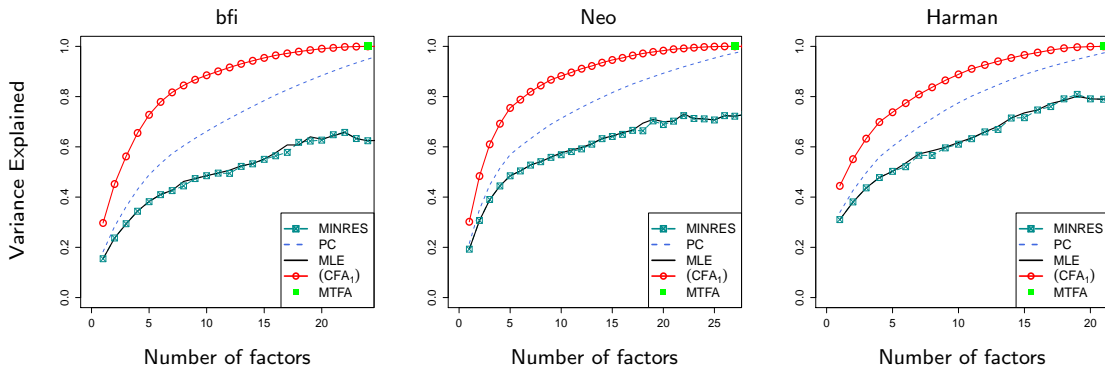


Figure 3: Figure showing proportion of variance explained by different methods for real-data examples. We see that in terms of the proportion of explained variance,  $(CFA_1)$  delivers the largest values for different values of  $r$ , which is indeed desirable.  $(CFA_1)$  also shows nice flexibility in delivering different models with varying  $r$ , in contrast to MTFA which delivers one model with proportion of variance explained equal to one. The results of  $(CFA_2)$  were found to be similar to  $(CFA_1)$ .

### 5.3 Certificates of Optimality via Algorithm 2

We now turn our attention to certificates of optimality using Algorithm 2. Computational results of Algorithm 2 for a variety of problem sizes across all six classes can be found in Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, and 11. In general, we provide results for  $p$  ranging between 10 and 4000. Parametric choices are outlined in depth in Table 2.<sup>17</sup> We always initialize Algorithm 2 with an initial feasible solution as found via Algorithm 1 so that we can understand if the estimators found via the CG-based approach are indeed optimal. In particular, if the best feasible objective value does not change throughout the branch-and-bound algorithm, then the initial estimator was indeed optimal (within the numerical tolerance).

**Root Node Gap:** Let us first consider the gap at the root node. In classes  $A_1$ ,  $B_1$ ,  $B_2$ , and  $B_3$ , we see that the Weyl bound at the root node often provides a better bound than the one given by using convex envelopes. Indeed, the bound provided by Weyl’s method can in many instances certify optimality (up to numerical tolerance) at the *root node*. For example, this is the case in many instances of classes  $A_1$  and half of the instances in  $B_3$ . Given that Weyl’s method is computationally inexpensive (only requiring the computation of  $p$  convex quadratic optimization problems and an eigenvalue decomposition), this suggests that Weyl’s inequality as used within the context of factor analysis is particularly fruitful.

In contrast, in class  $A_2$  and the real examples, we see that the convex envelope bound tends to perform better. Because of the structure of Weyl’s inequality, Weyl’s method is well-suited for correlation matrices  $\Sigma$  with very quickly decaying eigenvalues. Examples in

17. All computational experiments are performed in a shared cluster computing environment with highly variable demand, and therefore runtimes are not necessarily a reliable measure of problem complexity; hence, the number of nodes considered is always displayed. Further, Algorithm 2 is highly parallelizable, like many branch-and-bound algorithms; however, our implementation is serial. Therefore, with improvements in code design, it is very likely that runtimes can be substantially improved beyond those shown here.



		Root node			Terminal node			
Problem size ( $R/p$ )	Instance	Upper bound	CE LB	Weyl LB	Upper bound	Lower bound	Nodes	Time (s)
2/10	1	1.54	1.44	1.43	1.54	1.44	1	0.14
	2	1.50	1.40	1.40	1.50	1.40	1	0.38
	3	1.44	1.25	1.32	1.44	1.35	5	0.20
3/10	1	0.88	0.49	0.70	0.88	0.78	78	4.08
	2	0.49	0.25	0.39	0.49	0.39	1	1.27
	3	0.52	0.17	0.42	0.52	0.42	14	0.34
5/10	1	0.43	-0.90	0.10	0.43	0.33	28163	19432.70
	2	0.06	-0.75	0.00	0.06	0.00	1	0.20
	3	0.17	-0.98	0.00	0.17	0.07	3213	380.66
2/20	1	3.99	3.91	3.94	3.99	3.94	1	0.19
	2	4.64	4.58	4.60	4.64	4.60	1	2.07
	3	3.34	3.26	3.28	3.34	3.28	1	0.66
3/20	1	2.33	2.06	2.24	2.33	2.33	1	0.21
	2	2.55	2.38	2.49	2.55	2.49	1	6.62
	3	2.04	1.86	1.97	2.04	1.97	1	3.51
5/20	1	0.61	-0.07	0.49	0.61	0.51	626	75.98
	2	0.50	-0.01	0.40	0.50	0.40	41	2.88
	3	0.92	0.18	0.81	0.92	0.82	267	24.81

Table 2: Computational results for Algorithm 2 for class  $C = A_1(R/p)$ . All computations are performed in `julia` using SDO solvers `MOSEK` (for primal feasible solutions) and `SCS` (for dual feasible solutions within tolerance  $10^{-3}$ ). Computation time does not include preprocessing (such as computation of  $\mathbf{u}$  as in (40) and finding an initial incumbent feasible solution  $\Phi_f$  as computed via the conditional gradient algorithm in Section 3). We always use default tolerance `TOL = 0.1` for algorithm termination. Parameters for branching, pruning, node selection, etc., are detailed throughout Section 4. For each problem size, we run three instances (the instance number is the random seed provided to `julia`). Upper bounds denote  $z_f$ , which is the best feasible solution found thus far (either at the root node or at algorithm termination). At the root node, we display two lower bounds: the lower bound arising from convex envelopes (denoted “CE LB”) and the one arising from the Weyl bound (denoted “Weyl LB”). Note that for lower bound at the termination node, we mean the worst bound  $\max\{z^c, w^c\}$  (see Section 4.6;  $z^c$  is from the convex envelope approach, while  $w^c$  comes from Weyl’s method). “Nodes” indicates the number of nodes considered in the course of execution, while “Time (s)” denotes the runtime (in seconds). We set  $r^*$ , the rank used within Algorithm 2, to  $r^* = R - 1$ , where  $R$  is the generative rank displayed. All results displayed to two decimals. Computations run on high-demand, shared cluster computing environment with variable architectures. Runtime is capped at 400000s (approximately 5 days), and any instance which is still running at that time is marked with an asterisk next to its runtime.

		Root node			Terminal node			
Problem size ( $R/p$ )	Instance	Upper bound	CE LB	Weyl LB	Upper bound	Lower bound	Nodes	Time (s)
3/50	1	7.16	7.09	7.14	7.16	7.14	1	18.66
	2	6.74	6.66	6.71	6.74	6.71	1	13.19
	3	6.78	6.74	6.77	6.78	6.77	1	33.28
5/50	1	3.04	2.86	3.01	3.04	3.01	1	19.13
	2	3.32	3.12	3.29	3.32	3.29	1	7.33
	3	3.70	3.56	3.67	3.70	3.67	1	52.53
10/50	1	0.88	-0.15	0.81	0.88	0.81	1	6.22
	2	1.20	0.08	1.12	1.20	1.12	1	17.41
	3	1.14	-0.02	1.07	1.14	1.07	1	8.88
3/100	1	13.72	13.69	13.72	13.72	13.72	1	125.17
	2	14.07	14.03	14.06	14.07	14.06	1	63.25
	3	14.67	14.64	14.66	14.67	14.66	1	28.36
5/100	1	8.54	8.46	8.53	8.54	8.53	1	117.71
	2	7.16	7.04	7.14	7.16	7.14	1	70.09
	3	7.07	6.94	7.06	7.07	7.06	1	78.60
10/100	1	2.62	2.10	2.58	2.62	2.58	1	36.91
	2	2.81	2.38	2.78	2.81	2.78	1	9.25
	3	3.16	2.64	3.12	3.16	3.12	1	49.04

Table 3: Computational results for larger examples from class  $C = A_1(R/p)$ . Same parameters as in Table 2 but with larger problem instances. Again we set  $r^* = R - 1$ .

these two classes do not have such a spectrum, and indeed Weyl’s method does not provide the best root node bound.<sup>18</sup> Because neither Weyl’s method nor the convex envelope bound strictly dominate one another at the root node across all examples, our approach incorporating both can leverage the advantages of each.

Observe that the root node gap (either in terms of the absolute difference between the initial feasible solution found and the better of the convex envelope bound and the Weyl bound) tends to be smaller when  $r^*$  is much smaller than  $p$ . This suggests that the approach we take is well-suited to certify optimality of particularly low-rank decompositions in noisy factor analysis settings. We see that this phenomenon is true across all classes. The numerical results suggest that corresponding theoretical guarantees for Weyl’s method are potentially of interest in their own right and warrant further consideration.

Finally, we remark that if the true convex envelope of the objective over the set of semidefinite constraints was taken, then the convex envelope objective would always be non-negative. However, because we have taken the convex envelope of the objective over the polyhedral constraints only, this is not the case.

**Performance at Termination:** It is particularly encouraging that the initial feasible solutions provided via Algorithm 1 remain the best feasible solution throughout the execution of Algorithm 2 for all but three problem instances. (Of course, this need not be

18. However, it is worth remarking that Weyl’s method still provides lower bounds on the rank of solutions to the noiseless factor analysis problem. Hence, even in settings where Weyl’s method is not necessarily well-suited for proving optimality for the noisy factor analysis problem, it can still be applied successfully to lower bound rank for noiseless factor analysis.

Problem size ( $p$ )	Instance	$r^*$ used	Root node			Terminal node		Nodes	Time (s)
			Upper bound	CE LB	Weyl LB	Upper bound	Lower bound		
10	1	2	0.98	-0.29	0.26	0.96	0.86	1955	91.99
	2		0.85	-0.28	0.29	0.85	0.75	736	37.70
	3		0.89	-0.29	0.23	0.89	0.79	1749	53.30
10	1	3	0.53	-0.43	0.13	0.53	0.43	3822	409.68
	2		0.58	-0.45	0.15	0.58	0.48	8813	1623.38
	3		0.64	-0.44	0.10	0.59	0.49	13061	1671.61
20	1	2	5.13	4.14	2.13	5.13	5.03	29724	36857.46
	2		4.68	3.44	2.00	4.68	4.58	28707	35441.37
	3		4.83	3.67	2.00	4.83	4.73	17992	11059.55
20	1	3	4.26	2.68	1.55	4.26	4.05	86687	400002.3*
	2		3.88	2.00	1.46	3.88	3.71	132721	400003.7*
	3		4.03	2.25	1.48	4.03	3.84	100042	400002.1*
50	1	3	17.83	16.54	11.49	17.83	17.44	35228	400002.5*
	2		18.57	17.34	12.19	18.57	18.20	34006	400007.1*
	3		18.21	17.06	11.96	18.21	17.86	35874	400004.9*
100	1	3	38.65	37.50	33.08	38.65	38.18	20282	400015.2*
	2		38.84	37.66	33.25	38.84	38.37	17306	400013.7*
	3		39.02	37.86	33.67	39.02	38.58	19889	400004.9*
100	1	5	30.98	29.05	25.69	30.98	29.91	16653	400005.6*
	2		31.37	29.39	26.07	31.37	30.32	26442	400021.4*
	3		31.20	29.24	26.13	31.20	30.15	19445	400043.4*

Table 4: Computational results for class  $C = A_2(p)$ . All parameters as per Table 2. Here we show the behavior across a variety of choices of the parameter  $r^*$ . Recall that the class  $A_2$  is generated by a common variance component with high rank and exponentially decaying eigenvalues.

universally true.) This is an important observation to make because without a provable optimality scheme such as the one we consider here, it is difficult to quantify the performance of heuristic upper bound methods. As we demonstrate here, despite the only local guarantees of solutions obtained via a conditional gradient scheme, they tend to perform quite well in the setting of factor analysis. Indeed, even in the three examples where the best feasible solution is improved, the improved solution is found very early in the branching process.

Across the different example classes, we see that in general the gap tends to decrease more when  $r^*$  is small relative to  $p$  and  $p$  is smaller. To appropriately contextualize and appreciate the number of nodes solved for a problem with  $p = 100$  on the timescale of 100s, with state-of-the-art implementations of interior point methods, solving a single node in the branch and bound tree can take on the order of 40s (for the specifically structured problems of interest and on the same machine). In other words, if one were to naively use interior point methods, it would only be possible to solve approximately three nodes during a 100s time limit. In contrast, by using a first-order method approach which facilitates warm starts, we are able to solve hundreds of nodes in the same amount of time.

We see that Algorithm 2 performs particularly well for classes  $A_1$  (Tables 2 and 3) and  $B_1$  (Table 5), for problems of reasonable size with relatively small underlying ranks. This

			Root node			Terminal node			
Problem size ( $r/p$ )	Instance	$r^*$ chosen	Upper bound	CE LB	Weyl LB	Upper bound	Lower bound	Nodes	Time (s)
4/10	1	1	0.25	0.06	0.11	0.25	0.20	24	0.66
	2	1	0.17	-0.04	0.04	0.17	0.12	17	0.50
	3	1	0.16	-0.08	0.00	0.16	0.11	26	0.16
6/10	1	2	0.09	-0.27	0.00	0.09	0.04	77	0.62
	2	2	0.07	-0.36	0.00	0.07	0.02	118	0.75
	3	2	0.07	-0.36	0.00	0.07	0.02	125	3.79
10/10	1	3	0.10	-0.46	0.00	0.10	0.05	2162	166.44
	2	3	0.10	-0.62	0.00	0.10	0.05	3282	406.82
	3	4	0.06	-0.49	0.00	0.04	0.00	122	11.69
4/20	1	1	0.26	0.10	0.13	0.26	0.21	21	0.17
	2	1	0.19	0.01	0.07	0.19	0.14	15	0.40
	3	1	0.19	-0.04	0.04	0.19	0.14	26	0.59
6/20	1	2	0.10	-0.24	0.01	0.10	0.05	73	1.47
	2	2	0.08	-0.32	0.00	0.08	0.03	103	2.82
	3	2	0.08	-0.30	0.00	0.08	0.03	96	1.64
10/20	1	3	0.13	-0.44	0.00	0.13	0.08	2602	113.82
	2	3	0.12	-0.59	0.00	0.12	0.07	3971	188.93
	3	3	0.15	-0.45	0.01	0.15	0.10	3910	531.04
6/50	1	2	0.11	-0.15	0.04	0.11	0.06	50	0.66
	2	2	0.10	-0.18	0.01	0.10	0.05	61	1.95
	3	2	0.10	-0.20	0.01	0.10	0.05	64	0.58
10/50	1	3	0.17	-0.35	0.03	0.17	0.12	2283	333.62
	2	3	0.17	-0.46	0.02	0.17	0.12	4104	544.71
	3	3	0.19	-0.36	0.03	0.19	0.14	4362	620.61
6/100	1	2	0.12	-0.07	0.05	0.12	0.05	1	0.71
	2	2	0.11	-0.09	0.04	0.11	0.07	12	0.82
	3	2	0.11	-0.12	0.03	0.11	0.06	30	0.46
10/100	1	3	0.19	-0.22	0.06	0.19	0.14	1535	237.19
	2	3	0.19	-0.28	0.05	0.19	0.14	2625	383.27
	3	3	0.20	-0.26	0.05	0.20	0.15	2829	355.08

Table 5: Computational results for class  $C = B_1(R/p)$ . We choose  $r^*$  during computation as the largest  $r$  such that the CG method of Section 3 is strictly positive (up to additive tolerance 0.05; we use a smaller value here because the objective values are smaller across this class). For this class, examples can be preprocessed because  $\Sigma \sim B_1(R/p)$  has a block of size  $R \times R$  in the upper left, with all other entries set to zero except the diagonal. Hence, it suffices to perform factor analysis with the truncated matrix  $\Sigma_{1:R,1:R}$ .

is highly encouraging. Class  $A_1$  forms a set of prototypical examples for which theoretical recovery guarantees perform well; in stark contrast, problems such as those in  $B_1$  which have highly structured underlying factors tend to not satisfy the requirements of such recovery guarantees. Indeed, if  $\Sigma \sim B_1(R/p)$ , then there generally appears to be a rank  $R/2$  matrix  $\Theta \succcurlyeq \mathbf{0}$  so that  $\Sigma - \Theta$  is positive semidefinite and diagonal. In such a problem, for  $r^*$  on the order of  $R/2 - 1$ , we provide nearly optimal solutions within a reasonable time frame.

Further, we note that similar results to those obtained for classes  $A_1$  and  $B_1$  are obtained for the other classes and are detailed in Tables 4 (class  $A_2$ ), 6 (class  $B_2$ ), 7 (class  $B_3$ ), and

Problem size ( $r/R/p$ )	Instance	$r^*$ chosen	Root node			Terminal node		Nodes	Time (s)
			Upper bound	CE LB	Weyl LB	Upper bound	Lower bound		
2/5/20	1	4	1.15	0.4	1.03	1.15	1.05	517	146.72
	2	4	1.08	0.50	0.97	1.08	0.98	366	60.56
	3	4	0.89	0.29	0.79	0.89	0.79	1	0.56
3/5/20	1	4	1.24	0.46	1.11	1.24	1.14	933	232.74
	2	4	0.99	0.55	0.92	0.99	0.92	1	0.87
	3	4	0.72	0.08	0.62	0.72	0.62	123	20.16
2/5/100	1	4	7.54	7.46	7.53	7.54	7.53	1	84.20
	2	4	6.45	6.34	6.44	6.45	6.44	1	119.62
	3	4	7.20	7.09	7.18	7.20	7.18	1	97.17
3/5/100	1	4	8.40	8.32	8.39	8.40	8.39	1	135.73
	2	4	5.79	5.70	5.78	5.79	5.78	1	38.19
	3	4	7.58	7.47	7.57	7.58	7.57	1	73.51
2/10/100	1	9	3.56	3.11	3.53	3.56	3.53	1	71.10
	2	9	3.00	2.61	2.97	3.00	2.97	1	85.47
	3	9	2.58	2.14	2.55	2.58	2.55	1	22.99
3/10/100	1	9	3.23	2.80	3.20	3.23	3.20	1	50.63
	2	9	2.94	2.56	2.91	2.94	2.91	1	51.99
	3	9	2.57	2.12	2.54	2.57	2.54	1	72.45

Table 6: Computational results for class  $B_2(r/R/p)$ . Parameters as set in Table 2. Here  $r^*$  is chosen during computation as largest  $k$  such that the CG method of Section 3 is strictly positive (up to 0.1). Here  $r^* = R - 1$  ends up being an appropriate choice.

8 and 9. In a class such as  $A_2$ , which is generated as a high rank matrix (with decaying spectrum) with added individual variances, theoretical recovery guarantees do not generally apply, so again it is encouraging to see that our approach still makes significant progress towards proving optimality. Further, as shown in Table 11, for a variety of problems with  $p$  on the order of 1000 or 4000, solutions can be found in seconds and optimality can be certified within minutes via Weyl bounds, with no need for convex envelope bounds as computed via Algorithm 2, so long as the rank  $r^*$  is sufficiently small (for classes  $A_1$ ,  $B_2$ , and  $B_3$  on the order of hundreds, and for  $A_2$  on the order of tens). This strongly supports the value of such an eigenvalue-based approach. When computing lower bounds solely via Weyl’s method, the only necessary computations are solving quadratic programs (to find  $\mathbf{u}$  as in Section 4.3) and an eigenvalue decomposition. As Table 11 suggests, for sufficiently small rank, one can still quickly find certifiably optimal solutions even for very large-scale factor analysis problems.

A particularly interesting example that stands apart from the rest is JO from the set of real examples. In particular, the correlation matrix for this example is not full rank (as there are fewer observations than variables). As displayed in Table 10, solutions found via conditional gradient methods are certified to be optimal at the root node. Indeed, the initial  $\mathbf{u}_0$  as in Section 4.3 can be quickly verified to be  $\mathbf{u}_0 = \mathbf{0}$  for this example, i.e., factor analysis is only possible with no individual variances. Hence, Algorithm 2 terminates at the root node without needing any branching.

			Root node			Terminal node			
Problem size ( $r/R/p$ )	Instance	$r^*$ chosen	Upper bound	CE LB	Weyl LB	Upper bound	Lower bound	Nodes	Time (s)
2/5/20	1	3	0.30	-0.30	0.13	0.30	0.20	777	219.75
	2	3	0.22	-0.11	0.12	0.22	0.12	1	2.61
	3	3	0.19	-0.17	0.10	0.19	0.10	1	1.57
3/5/20	1	2	1.00	0.62	0.82	1.00	0.90	217	48.43
	2	2	0.95	0.47	0.74	0.95	0.85	305	26.14
	3	2	0.45	0.06	0.30	0.45	0.36	98	12.31
2/5/100	1	3	0.55	0.24	0.40	0.55	0.45	7099	47651.57
	2	3	0.28	-0.07	0.17	0.28	0.18	266	1431.88
	3	3	0.19	-0.07	0.11	0.19	0.11	1	574.61
3/5/100	1	2	1.03	0.70	0.81	1.03	0.93	4906	34514.69
	2	2	0.90	0.66	0.78	0.90	0.80	95	596.96
	3	2	0.55	0.39	0.47	0.55	0.47	1	1787.14
2/10/100	1	8	0.12	-0.58	0.05	0.12	0.05	1	77.38
	2	8	0.11	-0.53	0.05	0.11	0.05	1	41.24
	3	8	0.10	-0.43	0.05	0.10	0.05	1	110.17
3/10/100	1	6	0.33	-0.73	0.20	0.33	0.22	27770	400000.0*
	2	7	0.25	-0.43	0.17	0.25	0.17	1	128.63
	3	7	0.31	-0.28	0.24	0.31	0.24	1	213.72

Table 7: Computational results for class  $B_3(r/R/p)$ . Parameters as set in Table 2. As in Tables 5 and 6, here  $r^*$  is chosen during computation as largest  $r$  such that the CG method of Section 3 produces a feasible solution with strictly positive objective (up to 0.1).

	Root node			Terminal node			
$r^*$ used	Upper bound	CE LB	Weyl LB	Upper bound	Lower bound	Nodes	Time (s)
1	4.06	3.78	2.53	4.06	3.96	44	0.64
2	2.64	2.04	1.42	2.64	2.54	1885	142.06
3	1.56	0.62	0.61	1.56	1.46	11056	2458.23
4	0.88	-0.33	0.28	0.88	0.78	66877	60612.61
5	0.36	-0.88	0.0	0.36	0.25	155759	400012.4*

Table 8: Computational results for the **geomorphology** example ( $p = 10$ ). Parameters as set in Table 2. Here we display results for the choices of  $r^* \in \{1, 2, 3, 4, 5\}$  (for  $r^* > 5$ , the upper bound is below the numerical tolerance, so we do not include those).

	Root node			Terminal node			
$r^*$ used	Upper bound	CE LB	Weyl LB	Upper bound	Lower bound	Nodes	Time (s)
1	9.88	9.64	5.89	9.88	9.78	158	30.13
2	7.98	7.54	4.22	7.98	7.88	31710	49837.17
3	6.53	5.85	3.01	6.53	6.35	81935	400003.8*

Table 9: Computational results for the **Harman** example ( $p = 24$ ). Parameters as set in Table 2. Here we display results for the choices of  $r^* \in \{1, 2, 3\}$ .

$r^*$	Optimal value	Time (s)	$r^*$	Optimal value	Time (s)
1	51.85	5.59	12	12.52	3.37
2	46.30	6.86	13	10.81	2.97
3	41.29	6.08	14	9.25	1.07
4	36.54	6.08	15	7.78	6.21
5	32.36	1.21	16	6.44	5.40
6	28.72	7.83	17	5.15	0.91
7	25.39	1.13	18	3.98	8.50
8	22.10	3.87	19	2.84	5.33
9	19.20	4.97	20	1.87	2.15
10	16.63	4.17	21	1.07	2.13
11	14.30	7.87	22	0.48	1.65

Table 10: Computational results for the example JO ( $p = 58$ ). Parameters as set in Table 2. Here we display results for the choices of  $r^* \in \{1, 2, \dots, 22\}$  (for  $r^* > 22$ , the upper bound is below the numerical tolerance, so we do not include those). For all choices of  $r^*$ , the optimal solution is found and certified to be optimal at the root node, and therefore we do not display any information other than the optimal values.

Finally, we note that all synthetic examples we have considered have equal proportions of common and individual variances (although, of course, this is not exploited by our approach as this information is not *a priori* possible to specify without additional contextual information). If one modifies the classes so that the proportion of the common variances is higher than the proportion of individual variances (in the generative example), then Algorithm 2 is able to deliver better bounds on a smaller time scale. (Results are not included here.) This is not particularly surprising because the branch-and-bound approach we take essentially hinges on how well the products  $W_{ii}\Phi_i$  can be approximated. When there is underlying factor structure with a lower proportion of individual variances, the scale of  $W_{ii}\Phi_i$  is smaller and hence these products are easier to approximate well.

#### 5.4 Additional Considerations

We now turn our attention to assessing the benefits of various algorithmic modifications as presented in Section 4. We illustrate the impact of these by focusing on four representative examples from across the classes:  $A_1(3/10)$ ,  $B_1(6/50)$ ,  $B_3(3/5/20)$ , and  $G := \text{geomorphology}$ .

**Performance of Branching Strategy:** We begin by considering the impact of our branching strategy as developed in Section 4.4. The results across the four examples are shown in Table 12. Recall that  $\epsilon \in [0, 1)$  controls the extent of deviation from the canonical branching location, with  $\epsilon = 0$  corresponding to no deviation. Across all examples, we see that the number of nodes considered to prove optimality is approximately convex in  $\epsilon \in [0, 0.5]$ . In particular, for all examples, the “optimal” choice of  $\epsilon$  is not the canonical choice of  $\epsilon = 0$ . This contrast is stark for the examples  $B_3(3/5/20)$  and  $G$ . Indeed, for these two examples, the number of nodes considered when  $\epsilon = 0$  is over five times larger than for any  $\epsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ .

Example	$r^*$	Upper bound	Lower bound
$A_1(100/1000)$	10	460.35	457.46
	30	322.35	313.62
	50	198.70	197.13
	70	103.72	102.78
	90	28.65	28.34
$A_2(1000)$	10	184.22	183.18
	20	61.24	60.34
	30	20.25	19.50
	40	6.63	6.02
	50	2.17	1.71
$B_2(10/90/1000)$	20	379.02	377.23
	40	236.10	234.83
	60	122.08	121.33
	80	34.49	34.24
$B_2(20/90/1000)$	20	378.37	376.64
	40	235.16	233.94
	60	121.69	120.96
	80	34.33	34.09
$B_3(10/50/1000)$	10	384.40	384.04
	20	233.25	232.97
	30	105.68	105.49
	40	0.69	0.60
$B_3(20/150/1000)$	20	426.31	422.13
	45	285.56	282.33
	70	174.00	171.70
	95	86.54	85.15
	120	20.30	19.80

Example	$r^*$	Upper bound	Lower bound
$A_1(360/4000)$	100	1334.40	1327.60
	150	991.64	986.15
	200	693.02	688.87
	250	434.64	431.80
	300	213.29	211.75
$A_2(4000)$	350	30.75	30.50
	10	733.41	733.09
	20	244.13	243.85
	30	79.94	79.68
	40	26.17	25.96
$B_2(80/360/4000)$	50	8.57	8.39
	60	2.80	2.66
	70	0.90	0.79
	100	1360.1	1354.04
	150	1008.90	1004.04
$B_3(120/360/4000)$	200	703.78	700.21
	250	440.56	438.16
	300	215.87	214.62
	350	31.02	30.91
	100	1081.20	1078.74
$B_3(120/360/4000)$	150	630.25	628.54
	200	253.50	252.50
	240	7.89	7.46

Table 11: Computational results across several classes for larger scale instances with  $p \in \{1000, 4000\}$ . Here for a given example, the random seed is set as 1. Results are displayed across a variety of choices of rank  $r^*$ . Class  $B_1$  is not shown because of the reduction noted in Table 5 (which reduces this class to a much smaller-dimensional class, and hence it is not truly large scale). The “Upper bound” denotes the upper bound found by the conditional gradient method, while “Lower bound” denotes the Weyl bound at the root node (no convex envelope bounds via Algorithm 2 are shown here because of the large nature of the SDO-based convex envelope lower bounds which are prohibitive to solve for problems of this size). These instances are completed in `MATLAB` because of its superior support (over `julia`) for solving large-scale quadratic optimization problems, which is the only computation necessary for computing Weyl bounds.



Example	$r^*$	Nodes considered for $\epsilon =$					
		0.0	0.1	0.2	0.3	0.4	0.5
$A_1(3/10)$	2	103	72	59	62	78	101
$B_1(6/50)$	2	39	33	29	40	50	53
$B_3(3/5/20)$	2	8245	762	278	189	217	203
geomorphology	1	937	162	59	46	44	52

Table 12: Computational results for effect of branching strategy across different examples. In particular, we consider how the number of nodes needed to prove optimality (up to numerical tolerance) changes across different choices of  $\epsilon$ , where  $\epsilon$  is as in Section 4.4. Recall that  $\epsilon = 0$  corresponds to a naïve choice of branching, while  $\epsilon > 0$  corresponds to a modified branching. All other parameters as in Table 2, including the branching index. For synthetic examples, the instances are initialized with random seed 1.

Example	$r^*$	Nodes considered for	
		Naïve strategy	Modified strategy
$A_1(3/10)$	2	99	78
$B_1(6/50)$	2	135	50
$B_3(3/5/20)$	2	375	217
geomorphology	1	43	44

Table 13: Computational results for effect of node selection strategy across different examples. In particular, we consider how the number of nodes needed to prove optimality (up to numerical tolerance) changes across the naïve and modified branching strategies as described in Section 4.4. All other parameters as in Table 2. For synthetic examples, the instances are initialized with random seed 1.

Example	$r^*$	Upper Bound	CE LB	CE LB
			with tightening	without tightening
$A_1(3/10)$	2	0.88	0.70	0.39
$B_1(6/50)$	2	0.11	-0.15	-0.17
$B_3(3/5/20)$	2	1.00	0.62	0.51
geomorphology	1	4.06	3.78	3.78

Table 14: Computational results for effect of root node bound tightening across different examples. In particular, we consider how the convex envelope lower bound (denoted “CE LB”) compares with and without bound tightening at the root node (see Section 4.5). All other parameters as in Table 2. “Upper bound” displayed is as in Table 2 and is included for scale (i.e., to compare the relative impact of tightening). For synthetic examples, the instances are initialized with random seed 1.

Example	$r^*$	Nodes considered for TOL =		
		0.10	0.05	0.025
$A_1(3/10)$	2	78	287	726
$B_1(6/50)$	2	1	50	262
$B_3(3/5/20)$	2	217	1396	5132
geomorphology	1	44	217	978

Table 15: Computational results for numerical tolerance TOL across different examples. In particular, we consider how the number of nodes needed to prove optimality changes as a function of the additive numerical tolerance TOL chosen for algorithm termination. All other parameters as in Table 2. For synthetic examples, the instances are initialized with random seed 1.

In other words, the alternative branching strategy can have a substantial impact on the number of nodes considered in the branch-and-bound tree. As a direct consequence, this strategy can drastically reduce the computation time needed to prove optimality. As the examples suggest, it is likely that  $\epsilon$  should be chosen dynamically during algorithm execution, as the particular choice depends on a given problem’s structure. However, we set  $\epsilon = 0.4$  for all other computational experiments because this appears to offer a distinct benefit over the naïve strategy of setting  $\epsilon = 0$ .

**Performance of Node Selection Strategy:** We now turn our attention to the node selection strategy as detailed in Section 4.6. Recall that node selection considers how to pick which node to consider next in the current branch-and-bound tree at any iteration of Algorithm 2. We compare two strategies: the naïve strategy which selects the node with worst convex envelope bound (as explicitly written in Algorithm 2) and the modified strategy which employs randomness and Weyl bounds to consider nodes which might not be advantageous to fathom when only convex envelope bounds are considered.

The comparison is shown in Table 13. We see that this strategy is advantageous overall (with only the example  $G$  giving a negligible decrease in performance). The benefit is particularly strong for examples from the  $B$  classes which have highly structured underlying factors. For such examples, there is a large difference between the convex envelope bounds and the Weyl bounds at the root node (see e.g. Table 5). Hence, an alternative branching strategy which incorporates the eigenvalue information provided by Weyl bounds has potential to improve beyond the naïve strategy. Indeed, this appears to be the case across all examples where such behavior occurs.

**Performance of Bound Tightening** All computational results employ bound tightening as developed in Section 4.5, but only at the root node. Bound tightening, which requires repeated eigenvalue computations, is a computationally expensive process. For this reason, we have chosen not to employ bound tightening at every node in the branch-and-bound tree. From a variety of computational experiments, we observed that the most important node for bound tightening is the root node, and therefore it is a reasonable choice to only employ bound tightening there. Consequently, we employ pruning via Weyl’s method (detailed in Section 4.5 as well) at all nodes in the branch-and-bound tree. (Recall that bound tightening can be thought of as optimal pruning via Weyl’s method.)

In Table 14 we show the impact of bound tightening at the root node in terms of the improvement in the lower bounds provided by convex envelopes. The results for the class  $A_1$  are particularly distinctive. Indeed, for this class bound tightening has a substantial impact on the quality of the convex envelope bound (for the example  $A_1(3/10)$  given, the improvement is from a relative gap at the root node of 56% to a gap of 20%). For the examples shown, bound tightening offers the least improvement in the real example geomorphology. In light of Table 8 this is not too surprising, as Weyl’s method (at the root node) is not particularly effective for this example. As Weyl’s inequality is central to bound tightening, problems for which Weyl’s inequality is not particularly effective tend to experience less benefit from bound tightening at the root node.

**Influence of Optimality Tolerance** We close this section by assessing the influence of the optimality tolerance for termination, TOL. In particular, we study how the number of

nodes to prove optimality changes as a function of additive gap necessary for termination. The corresponding results across the four examples are shown in Table 15. Not surprisingly, as the gap necessary for termination is progressively halved, the corresponding number of nodes considered increases substantially. However, it is important to note that even though the gap at termination is smaller as this tolerance decreases (by design), for these examples the best feasible solution remains unchanged. In other words, the increase in the number of nodes is the price for more accurately proving optimality and not for finding better feasible solutions. Indeed, as noted earlier, the solutions found via conditional gradient methods at the outset are of remarkably high quality.

## 6. Conclusions

We analyze the classical rank-constrained FA problem from a computational perspective. We proposed a general, flexible family of rank-constrained, nonlinear SDO-based formulations for the task of approximating an observed covariance matrix  $\Sigma$  as the sum of a PSD low-rank component  $\Theta$  and a diagonal matrix  $\Phi$  (with nonnegative entries) subject to  $\Sigma - \Phi$  being PSD. Our framework enables us to estimate the underlying factors and unique variances under the restriction that the residual covariance matrix is semidefinite—this is important for statistical interpretability and understanding the proportion of variance explained by a given number of factors. This constraint, however, seems to be ignored by most other widely used methods in FA.

We introduce a novel *exact* reformulation of the rank-constrained FA problem as a smooth optimization problem with convex, compact constraints. We present a unified algorithmic framework, utilizing modern techniques in nonlinear optimization and first order methods in convex optimization to obtain high-quality solutions for the FA problem. At the same time, we use techniques from discrete and global optimization to demonstrate that these solutions are often provably optimal. We provide computational evidence demonstrating that the methods proposed herein provide high quality estimates with improved accuracy when compared to existing, popularly-used methods in FA.

In this work we have demonstrated that a previously intractable rank optimization problem can be solved to provable optimality. We envision that ideas similar to those used here can be used to address an even larger class of estimation problems with underlying matrix structure. In this way, we anticipate significant progress on such problems in the next decade, particularly in light of myriad advances throughout distinct areas of modern optimization.

## Acknowledgments

Copenhaver is supported by the U.S. Department of Defense, Office of Naval Research, through the National Defense Science and Engineering Graduate (NDSEG) Fellowship. Mazumder is partially supported by ONR Grant N000141512342. We gratefully acknowledge the thoughtful and detailed feedback of two anonymous referees which has led to a substantial improvement in the quality, organization, and completeness of the paper. Copen-

haver would also like to thank the JuliaOpt team, especially Joey Huchette, for assistance with experimental features in the `julia` language.

## Appendix A

This appendix contains all proofs for results presented in the main text.

### Proof of Proposition 1:

- (a) We start by observing that for any two real symmetric matrices  $\mathbf{A}, \mathbf{B}$  (with dimensions  $p \times p$ ) and the matrix  $q$ -norm, a result due to Mirsky (this is also known as the  $q$ -Wielandt-Hoffman inequality; see e.g. Stewart and Sun 1990, p. 205) states:

$$\|\mathbf{A} - \mathbf{B}\|_q \geq \|\boldsymbol{\lambda}(\mathbf{A}) - \boldsymbol{\lambda}(\mathbf{B})\|_q, \quad (46)$$

where  $\boldsymbol{\lambda}(\mathbf{A})$  and  $\boldsymbol{\lambda}(\mathbf{B})$  denote the vector of eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$ , respectively, arranged in decreasing order, i.e.,  $\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \dots \geq \lambda_p(\mathbf{A})$  and  $\lambda_1(\mathbf{B}) \geq \lambda_2(\mathbf{B}) \geq \dots \geq \lambda_p(\mathbf{B})$ . Using this result for Problem (4), it is easy to see that for fixed  $\Phi$

$$\{\Theta : \Theta \succeq \mathbf{0}, \text{rank}(\Theta) \leq r\} = \{\Theta : \boldsymbol{\lambda}(\Theta) \geq \mathbf{0}, \|\boldsymbol{\lambda}(\Theta)\|_0 \leq r\}, \quad (47)$$

where  $\|\boldsymbol{\lambda}(\Theta)\|_0$  counts the number of non-zero elements of  $\boldsymbol{\lambda}(\Theta)$ . If we partially minimize the objective function in Problem (4), with respect to  $\Theta$  (with  $\Phi$  fixed), and use (46) along with (47), we have the following inequality:

$$\begin{aligned} & \inf_{\Theta} \|\Sigma - \Phi - \Theta\|_q^q \quad \text{s.t.} \quad \Theta \succeq \mathbf{0}, \text{rank}(\Theta) \leq r \\ & \geq \inf_{\boldsymbol{\lambda}(\Theta)} \|\boldsymbol{\lambda}(\Sigma - \Phi) - \boldsymbol{\lambda}(\Theta)\|_q^q \quad \text{s.t.} \quad \boldsymbol{\lambda}(\Theta) \geq \mathbf{0}, \|\boldsymbol{\lambda}(\Theta)\|_0 \leq r. \end{aligned} \quad (48)$$

Since  $\Sigma - \Phi \succeq \mathbf{0}$ , it follows that the minimum objective value of the r.h.s. optimization Problem (48) is given by  $\sum_{i=r+1}^p \lambda_i^q(\Sigma - \Phi)$  and is achieved for  $\lambda_i(\Theta) = \lambda_i(\Sigma - \Phi)$  for  $i = 1, \dots, r$ . This leads to the following inequality:

$$\inf_{\Theta: \Theta \succeq \mathbf{0}, \text{rank}(\Theta) \leq r} \|\Sigma - \Phi - \Theta\|_q^q \geq \sum_{i=r+1}^p \lambda_i^q(\Sigma - \Phi). \quad (49)$$

Furthermore, if  $\mathbf{U}_{p \times p}$  denotes the matrix of  $p$  eigenvectors of  $\Sigma - \Phi$ , then the following choice of  $\Theta^*$ :

$$\Theta^* := \mathbf{U} \text{diag}(\lambda_1(\Sigma - \Phi), \dots, \lambda_r(\Sigma - \Phi), 0, \dots, 0) \mathbf{U}', \quad (50)$$

gives equality in (49). This leads to the following result:

$$\inf_{\Theta: \Theta \succeq \mathbf{0}, \text{rank}(\Theta) \leq r} \|\Sigma - (\Theta + \Phi)\|_q^q = \|\Sigma - (\Theta^* + \Phi)\|_q^q = \sum_{i=r+1}^p \lambda_i^q(\Sigma - \Phi). \quad (51)$$

- (b) The minimizer  $\Theta^*$  of Problem (51) is given by (50). In particular, if  $\Phi^*$  solves Problem (12) and we compute  $\Theta^*$  via (50) (with  $\Phi = \Phi^*$ ), then the tuple  $(\Phi^*, \Theta^*)$  solves Problem (4). This completes the proof of the proposition.

■

**Proof of Proposition 2:** We build upon the proof of Proposition 1. Note that any  $\Phi$  that is feasible for Problems (14) and (4) is PSD. Observe that  $\Theta^*$  (appearing in the proof of Proposition 1) as given by (50) satisfies:

$$\Sigma - \Phi - \Theta^* \succeq \mathbf{0} \implies \Sigma - \Theta^* \succeq \mathbf{0}, \quad (52)$$

where, the right hand side of (52) follows since  $\Phi \succeq \mathbf{0}$ . We have thus established that the solution  $\Theta^*$  to the following problem

$$\begin{aligned} & \underset{\Theta}{\text{minimize}} && \|(\Sigma - \Phi) - \Theta\|_q^q \\ & \text{s.t.} && \Theta \succeq \mathbf{0} \\ & && \text{rank}(\Theta) \leq r, \end{aligned} \quad (53)$$

is feasible for the following optimization problem:

$$\begin{aligned} & \underset{\Theta}{\text{minimize}} && \|(\Sigma - \Phi) - \Theta\|_q^q \\ & \text{s.t.} && \Theta \succeq \mathbf{0} \\ & && \text{rank}(\Theta) \leq r \\ & && \Sigma - \Theta \succeq \mathbf{0}. \end{aligned} \quad (54)$$

Since Problem (54) involves minimization over a subset of the feasible set of Problem (53), it follows that  $\Theta^*$  is also a minimizer for Problem (54). This completes the proof of the equivalence. ■

### Proof of Theorem 1:

- (a) The proof is based on ideas appearing in Overton and Womersley (1992), where it was shown that the sum of the top  $r$  eigenvalues of a real symmetric matrix can be written as the solution to a linear SDO problem.

By an elegant classical result due to Fan (Stewart and Sun, 1990), the smallest  $(p - r)$  eigenvalues of a real symmetric matrix  $\mathbf{A}$  can be written as

$$\begin{aligned} \sum_{i=r+1}^p \lambda_i(\mathbf{A}) = & \inf_{\mathbf{V}} \text{Tr}(\mathbf{V}'\mathbf{A}\mathbf{V}) \\ \text{s.t.} & \mathbf{V}'\mathbf{V} = \mathbf{I}, \end{aligned} \quad (55)$$

where the optimization variable is  $\mathbf{V} \in \mathbb{R}^{p \times (p-r)}$ . We will show that the solution to the above nonconvex problem can be obtained via the following linear (convex) SDO problem:

$$\begin{aligned} & \text{minimize} && \text{Tr}(\mathbf{W}\mathbf{A}) \\ & \text{s.t.} && \mathbf{I} \succeq \mathbf{W} \succeq \mathbf{0} \\ & && \text{Tr}(\mathbf{W}) = p - r. \end{aligned} \quad (56)$$

Clearly, Problem (56) is a convex relaxation of Problem (55)—hence its minimum value is at least smaller than  $\sum_{i=r+1}^p \lambda_i(\mathbf{A})$ . By standard results in convex analysis (Rockafellar, 1996), it follows that the minimum of the above linear SDO problem (56) is attained at the extreme points of the feasible set of (56). The extreme points (see for example, Overton and Womersley, 1992; Vandenberghe and Boyd, 1996) of this set are given by the set of orthonormal matrices of rank  $p - r$ :

$$\{\mathbf{V}\mathbf{V}' : \mathbf{V} \in \mathbb{R}^{p \times (p-r)} : \mathbf{V}'\mathbf{V} = \mathbf{I}\}.$$

It thus follows that the (global) minima of Problems (56) and (55) are the same. Applying this result to the PSD matrix  $\mathbf{A} = (\boldsymbol{\Sigma} - \boldsymbol{\Phi})^q$  appearing in the objective of (12), we arrive at (16). This completes the proof of part (a).

(b) The statement follows from (50). ■

**Proof of Proposition 3:** Observe that, for every fixed  $\boldsymbol{\Phi}$ , the function  $g_q(\mathbf{W}, \boldsymbol{\Phi})$  is concave (in fact, it is linear). Since  $G_q(\mathbf{W})$  is obtained by taking a point-wise infimum (with respect to  $\boldsymbol{\Phi}$ ) of the concave function  $g_q(\mathbf{W}, \boldsymbol{\Phi})$ , the resulting function  $G_q(\mathbf{W})$  is concave (Boyd and Vandenberghe, 2004).

The expression of the sub-gradient (19) is an immediate consequence of Danskin's Theorem (Bertsekas, 1999; Rockafellar, 1996). ■

**Proof of Theorem 2:** Using the concavity of the function  $G_q(\mathbf{W})$  we have:

$$\begin{aligned} G_q(\mathbf{W}^{(i+1)}) &\leq G_q(\mathbf{W}^{(i)}) + \langle \nabla G_q(\mathbf{W}^{(i)}), \mathbf{W}^{(i+1)} - \mathbf{W}^{(i)} \rangle \\ &= G_q(\mathbf{W}^{(i)}) + \Delta(\mathbf{W}^{(i)}). \end{aligned} \tag{57}$$

Note that  $\Delta(\mathbf{W}^{(i)}) \leq 0$  and  $G_q(\mathbf{W}^{(i+1)}) \leq G_q(\mathbf{W}^{(i)})$  for all  $i \leq k$ . Hence the (decreasing) sequence of objective values converge to  $G_q(\mathbf{W}^{(\infty)})$  (say) and  $\Delta(\mathbf{W}^{(\infty)}) \geq 0$ . Adding up the terms in (57) from  $i = 1, \dots, k$  we have:

$$G_q(\mathbf{W}^{(\infty)}) - G_q(\mathbf{W}^{(1)}) \leq G_q(\mathbf{W}^{(k+1)}) - G_q(\mathbf{W}^{(1)}) \leq -k \min_{i=1, \dots, k} \{-\Delta(\mathbf{W}^{(i)})\} \tag{58}$$

from which (26) follows. ■

**Proof of Theorem 5:** By construction,

$$\max_{[\boldsymbol{\ell}, \mathbf{u}] \in \mathbf{Nodes}} \|\mathbf{u} - \boldsymbol{\ell}\|_1$$

decreases monotonically to zero as the iterative scheme proceeds. Therefore, to show convergence it suffices to show that the additive error at a node  $\mathbf{n} = [\boldsymbol{\ell}, \mathbf{u}] \in \mathbf{Nodes}$  is  $O(\|\mathbf{u} - \boldsymbol{\ell}\|_1)$ . Fix an arbitrary node  $\mathbf{n} = [\boldsymbol{\ell}, \mathbf{u}] \in \mathbf{Nodes}$ . Without loss of generality, we may assume that  $(\text{CFA}_1)$  as restricted to  $\text{diag}(\boldsymbol{\Phi}) \in [\boldsymbol{\ell}, \mathbf{u}]$  has a feasible solution; as

such, let  $\Phi^*$  be an optimal solution. Likewise, let  $\Phi^\dagger$  be an optimal solution  $(\text{LS}_{\ell, \mathbf{u}})$ . First note that  $\sum_{i=r+1}^p \sigma_i(\Sigma - \Phi^*) \leq \sum_{i=r+1}^p \sigma_i(\Sigma - \Phi^\dagger)$ . Now, by two applications of the bound in Theorem 4 and the fact that  $\Phi^\dagger$  is optimal to  $(\text{LS}_{\ell, \mathbf{u}})$ , we have that  $\sum_{i=r+1}^p \sigma_i(\Sigma - \Phi^\dagger) \leq \sum_{i=r+1}^p \sigma_i(\Sigma - \Phi^*) + \|\mathbf{u} - \ell\|_1/2$ . Hence, the additive error satisfies

$$\left| \sum_{i=r+1}^p \sigma_i(\Sigma - \Phi^\dagger) - \sum_{i=r+1}^p \sigma_i(\Sigma - \Phi^*) \right| \leq \|\mathbf{u} - \ell\|_1/2,$$

as was to be shown. ■

## Appendix B

This appendix contains an alternative conditional gradient method for finding feasible solutions. Since Problem (16) involves the minimization of a smooth function over a compact convex set, the CG method requires iteratively solving the following convex optimization problem:

$$\begin{aligned} & \text{minimize} \quad \left\langle \nabla g_p(\mathbf{W}^{(k)}, \Phi^{(k)}), \begin{pmatrix} \mathbf{W} \\ \Phi \end{pmatrix} \right\rangle \\ & \text{s.t.} \quad \mathbf{W} \in \Psi_{p, p-r} \\ & \quad \quad \Phi \in \mathcal{F}_\Sigma, \end{aligned} \tag{59}$$

where  $\nabla g_p(\mathbf{W}^{(k)}, \Phi^{(k)})$  is the derivative of  $g_p(\mathbf{W}^{(k)}, \Phi^{(k)})$  at the current iterate  $(\mathbf{W}^{(k)}, \Phi^{(k)})$ . Note that due to the separability of the constraints in  $\mathbf{W}$  and  $\Phi$ , Problem (59) splits into two independent optimization problems with respect to  $\mathbf{W}$  and  $\Phi$ . The overall procedure is outlined in Algorithm 3.

Since  $\nabla_{\mathbf{W}} g_p(\mathbf{W}^{(k)}, \Phi^{(k)}) = (\Sigma - \Phi^{(k)})^q$ , the update for  $\mathbf{W}$  appearing in (60) requires solving:

$$\text{minimize}_{\mathbf{W} \in \Psi_{p, p-r}} \langle \mathbf{W}, (\Sigma - \Phi^{(k)})^q \rangle. \tag{63}$$

Similarly, the update for  $\Phi$  appearing in (61) requires solving:

$$\text{minimize}_{\Phi \in \mathcal{F}_\Sigma} \sum_{i=1}^p \Phi_i \ell_i, \tag{64}$$

where the vector  $(\ell_1, \dots, \ell_p)$  is given by  $\text{diag}(\ell_1, \dots, \ell_p) = -q \text{diag}(\mathbf{W}^{(k)}(\Sigma - \Phi^{(k)})^{q-1})$ , where  $\text{diag}(\mathbf{A})$  is a diagonal matrix having the same diagonal entries as  $\mathbf{A}$ . The sequence  $(\mathbf{W}^{(k)}, \Phi^{(k)})$  is recursively computed via Algorithm 3 until a convergence criterion is met:

$$g_p(\mathbf{W}^{(k)}, \Phi^{(k)}) - g_p(\mathbf{W}^{(k+1)}, \Phi^{(k+1)}) \leq \text{TOL} \cdot g_p(\mathbf{W}^{(k)}, \Phi^{(k)}), \tag{65}$$

for some user-defined tolerance  $\text{TOL} > 0$ .

---

**Algorithm 3** A CG based algorithm for Problem (16)
 

---

- 1 Initialize with  $(\mathbf{W}^{(1)}, \Phi^{(1)})$ , feasible for Problem (16) and repeat the following Steps 2-3 until the convergence criterion described in (65) is met.
- 2 Solve the linearized Problem (59), which requires solving two separate (convex) SDO problems over  $\mathbf{W}$  and  $\Phi$ :

$$\overline{\mathbf{W}}^{(k+1)} \in \arg \min_{\mathbf{W} \in \Psi_{p,p-r}} \langle \mathbf{W}, \nabla_{\mathbf{W}} g_p(\mathbf{W}^{(k)}, \Phi^{(k)}) \rangle \quad (60)$$

$$\overline{\Phi}^{(k+1)} \in \arg \min_{\Phi \in \mathcal{F}_{\Sigma}} \langle \Phi, \nabla_{\Phi} g_p(\mathbf{W}^{(k)}, \Phi^{(k)}) \rangle \quad (61)$$

where  $\nabla_{\mathbf{W}} g_p(\mathbf{W}, \Phi)$  (and  $\nabla_{\Phi} g_p(\mathbf{W}, \Phi)$ ) is the partial derivative with respect to  $\mathbf{W}$  (respectively,  $\Phi$ ).

- 3 Obtain the new iterates:

$$\begin{aligned} \mathbf{W}^{(k+1)} &= \mathbf{W}^{(k)} + \eta_k (\overline{\mathbf{W}}^{(k+1)} - \mathbf{W}^{(k)}), \\ \Phi^{(k+1)} &= \Phi^{(k)} + \eta_k (\overline{\Phi}^{(k+1)} - \Phi^{(k)}). \end{aligned} \quad (62)$$

with  $\eta_k \in [0, 1]$  chosen via an Armijo type line-search rule (Bertsekas, 1999).

---

A tuple  $(\mathbf{W}^*, \Phi^*)$  satisfies the first order stationarity conditions (Bertsekas, 1999) for Problem (16), if the following condition is satisfied:

$$\begin{aligned} \inf \left\langle \nabla g_p(\mathbf{W}^*, \Phi^*), \begin{pmatrix} \mathbf{W} - \mathbf{W}^* \\ \Phi - \Phi^* \end{pmatrix} \right\rangle &\geq 0 \\ \text{s.t. } \mathbf{W} &\in \Psi_{p,p-r} \\ \Phi &\in \mathcal{F}_{\Sigma}. \end{aligned} \quad (66)$$

Note that  $\Phi^*$  defined above also satisfies the first order stationarity conditions for Problem (12).

The following theorem presents a global convergence guarantee for Algorithm 3:

**Theorem 8** (Bertsekas, 1999) *Every limit point  $(\mathbf{W}^{(\infty)}, \Phi^{(\infty)})$  of a sequence  $(\mathbf{W}^{(k)}, \Phi^{(k)})$  produced by Algorithm 3 is a first order stationary point of the optimization Problem (16).*

Numerical experiments (in line with those from Section 5) suggest that Algorithm 3 performs similarly to Algorithm 1, and therefore we only present the results for Algorithm 1 in the main text. Algorithm 1 has the advantage that it does not require a line search, unlike Algorithm 3. Finally, we note that for Algorithm 3 the update for  $\mathbf{W}$  at iteration  $k$  for solving Problem (63) corresponds to  $\widetilde{\mathbf{W}} = (\Sigma - \Phi^{(k)})^q$ .



## References

- F.A. Al-Khayyal and J.E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8:273–286, 1983.
- E.D. Andersen and K.D. Andersen. *High Performance Optimization*, chapter The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm. Springer, 2000.
- T. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New York, 3rd edition, 2003.
- K.M. Anstreicher and S. Burer. Computable representations for convex hulls of low-dimensional quadratic forms. *Mathematical Programming, Series B*, 124:33–43, 2010.
- J. Bai and K. Li. Statistical analysis of factor models of high dimension. *The Annals of Statistics*, 40(1):436–465, 2012.
- J. Bai and S. Ng. Large dimensional factor analysis. *Foundations and Trends in Econometrics*, 3(2):89–163, 2008.
- L. Bai, J.E. Mitchell, and J.-S. Pang. On conic QPCCs, conic QCQPs and completely positive programs. *Mathematical Programming*, 159(1-2):109–136, 2016.
- X. Bao, N.V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically constrained quadratic programs. *Optimization Methods and Software*, 24:485–504, 2009.
- D. Bartholomew, M. Knott, and I. Moustaki. *Latent Variable Models and Factor Analysis: A Unified Approach*. Wiley, 2011.
- A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete and Computational Geometry*, 12:189–202, 1995.
- P.M. Bentler and J.A. Woodward. Inequalities among lower-bounds to reliability: With applications to test construction and factor analysis. *Psychometrika*, 45:249–267, 1980.
- C. Bernaards and R. Jennrich. Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, 65:676–696, 2005.
- D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.
- R. Bhatia. *Perturbation bounds for matrix eigenvalues*. SIAM, 2007.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1042–1068, 2011.

- S. Burer. *Handbook on Semidefinite, Conic and Polynomial Optimization*, chapter Copositive programming, pages 201–218. Springer, 2012.
- G. Connor and R. Korajczyk. Performance measurement with the arbitrage pricing theory: A new framework for analysis. *Journal of Financial Economics*, 15(3):373–394, 1986.
- A. Costa and L. Liberti. Relaxations of multilinear convex envelopes: dual is better than primal. In *International Symposium on Experimental Algorithms*, pages 87–98. Springer, 2012.
- M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.
- C.A. Floudas. *Deterministic global optimization: theory, algorithms, and applications*. Kluwer, 1999.
- M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.
- L. Guttman. To what extent can communalities reduce rank? *Psychometrika*, 23(4):297–308, 1958.
- P. Hansen, B. Jaumard, M. Ruiz, and J. Xiong. Global minimization of indefinite quadratic functions subject to box constraints. *Naval Research Logistics*, 40:373–392, 1993.
- H. Harman and W. Jones. Factor analysis by minimizing residuals (MINRES). *Psychometrika*, 31:351–368, 1966.
- R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge University Press, 2nd edition, 2013.
- F. Husson, J. Josse, S. Le, and J. Mazet. *FactoMineR: Multivariate Exploratory Data Analysis and Data Mining*, 2015. URL <http://CRAN.R-project.org/package=FactoMineR>. R package version 1.31.3.
- E. John and E.A. Yildirim. Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, 41:151–183, 2008.
- K. Jöreskog. Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32(4):443–482, 1967.
- K. Jöreskog. Structural analysis of covariance and correlation matrices. *Psychometrika*, 43(4):443–477, 1978.
- J.B. Lasserre. *Moments, positive polynomials and their applications*. Imperial College Press, 2009.
- D. Lawley and A. Maxwell. Factor analysis as a statistical method. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 12(3):209–229, 1962.

- P. le Bodic and G.L. Nemhauser. An abstract model for branching and its application to mixed integer programming. *arXiv preprint arXiv:1511.01818*, 2015.
- W. Ledermann. On the rank of the reduced correlational matrix in multiple-factor analysis. *Psychometrika*, 2(2):85–93, 1937.
- A. Lewis. Derivatives of spectral functions. *Mathematics of Operations Research*, 21(3): 576–588, 1996.
- J. Löfberg. YALMIP: A toolbox for modeling and optimization in Matlab. In *International Symposium on Computer Aided Control Systems Design*, pages 284–289. IEEE, 2004.
- K. Mardia, J. Kent, and J. Bibby. *Multivariate analysis*. Academic Press, 1979.
- R. Misener and C.A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. *Mathematical Programming, Series B*, 136:155–182, 2012.
- Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, 2004.
- Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming, Series A*, 103:127–152, 2005.
- Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007. Technical Report number 76.
- B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.
- M. Overton and R. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal of Matrix Analysis and Applications*, 13:41–45, 1992.
- G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23:339–358, 1998.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2014. URL <http://www.R-project.org/>.
- G. Raiche and D. Magis. nFactors: Parallel analysis and non graphical solutions to the Cattell Scree test. URL <http://cran.r-project.org/web/packages/nFactors/index.html>, 2011.
- C. R. Rao. *Linear Statistical Inference and Its Applications*. Wiley, New York, 1973.
- W. Revelle. *psych: Procedures for Psychological, Psychometric, and Personality Research*, 2015. URL <http://CRAN.R-project.org/package=psych>. R package version 1.5.6.

- G. Riccia and A. Shapiro. Minimum rank and minimum trace of covariance matrices. *Psychometrika*, 47:443–448, 1982.
- D. Robertson and J. Symons. Maximum likelihood factor analysis with rank-deficient sample covariance matrices. *Journal of Multivariate Analysis*, 98(4):813–828, 2007.
- R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1996.
- S. Roweis and Z. Ghahramani. A unifying review of linear gaussian models. *Neural Computation*, 11(2):305–345, 1999.
- D. Rubin and D. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.
- N. V. Sahinidis. *BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs*, User’s Manual, 2014.
- J. Saunderson, V. Chandrasekaran, P. Parrilo, and A. Willsky. Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1395–1416, 2012.
- A. Shapiro. Rank-reducibility of a symmetric matrix and sampling theory of minimum trace factor analysis. *Psychometrika*, 47:187–199, 1982.
- A. Shapiro and J. ten Berge. Statistical inference of minimum rank factor analysis. *Psychometrika*, 67:79–94, 2002.
- C. Spearman. “General Intelligence,” Objectively Determined and Measured. *American Journal of Psychology*, 15:201–293, 1904.
- G. Stewart and J.-G. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.
- M. Tawarmalani and N.V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*, volume 65 of *Nonconvex Optimization and its Applications*. Kluwer, 2002a.
- M. Tawarmalani and N.V. Sahinidis. Convex extensions and envelopes of lower semi-continuous functions. *Mathematical Programming, Series A*, 93:247–263, 2002b.
- M. Tawarmalani, J-P.P. Richard, and C. Xiong. Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming, Series A*, 138:531–577, 2013.
- J. ten Berge. Some recent developments in factor analysis and the search for proper communalities. In *Advances in data science and classification*, pages 325–334. Springer, 1998.
- J. ten Berge and H. Kiers. Computational aspects of the greatest lower bound to the reliability and constrained minimum trace factor analysis. *Psychometrika*, 46:201–213, 1981.

- J. ten Berge and H. Kiers. A numerical approach to the approximate and the exact minimum rank of a covariance matrix. *Psychometrika*, 56:309–315, 1991.
- J. ten Berge and H. Kiers. The minimum rank factor analysis program MRFA. URL <http://www.ppsw.rug.nl/~kiers/>, 2003.
- J. ten Berge, T. Snijders, and F.E. Zegers. Computational aspects of the greatest lower bound to the reliability and constrained minimum trace factor analysis. *Psychometrika*, 46:201–213, 1981.
- L.L. Thurstone. *Multiple Factor Analysis: a Development and Expansion of the Vectors of the Mind*. University of Chicago Press, 1947.
- K.C. Toh, M.J. Todd, and R.H. Tutuncu. SDPT3 – a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.
- L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.
- E.A. Yildirim and S. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12:782–810, 2002.