Matrix Decomposition Factor Analysis

Jan de Leeuw

October 24, 2025

TBD

Table of contents

Note: This is a working manuscript which will be expanded/updated frequently. All suggestions for improvement are welcome. All qmd, tex, html, pdf, R, and C files are in the public domain. Attribution will be appreciated, but is not required. The files can be found at https://www.github.com/deleeuw/mdfa

1 Introduction

Suppose X is an $n \times m$ "tall" data matrix ($n \geq m$).

Definition 1.1. We say that $X = YT'$, with Y an $n \times p$ matrix and T an $m \times p$ matrix, is a *decomposition of order* p of X.

Decompositions of various kinds play a key role in multivariate data analysis, as well as in theoretical and numerical linear algebra (Stewart (1998)). Prime examples are the Singular Value Decomposition (SVD) $X = YT'$ with Y and T both orthogonal[1], the QR Decomposition $X = YT'$ with Y orthonormal and T lower-triangular, and the Polar Decomposition $X = YT'$ with Y orthonormal and T positive semi-definite.

Definition 1.2. A decomposition XT' of order p of the $n \times m$ matrix X is an *factor analytic decomposition of order* p, if $m < p < n$.

Note that Y is a "tall" matrix ($n \geq p$), while T is "wide" ($m \leq p$). The order p of the decomposition is consequently larger than the rank of the matrix X. In order to have a non-trivial data analysis technique we need constraints on T and Y, which we write in the general form $T \in \mathcal{T}$ and $Y \in \mathcal{Y}$, with $\mathcal{T}$ a subset of $\mathbb{R}^{m \times p}$, the space of all $m \times p$ matrices, and $\mathcal{Y}$ a subset of $\mathbb{R}^{n \times p}$, the space of all $n \times p$ matrices

Factor analysis techniques aim to find a solution of the system

$$X = YT', \tag{1a}$$

$$Y \in \mathcal{Y}, \tag{1b}$$

$$T \in \mathcal{T}. \tag{1c}$$

If no exact solution exists an approximate solution must be found.

Mathematically this introduces the problem to find the conditions under which equations (1a)-(1c) can be solved for Y and T, and to describe the set of solutions if the system is solvable. Computationally the problem is to define what is meant by "approximately" and to find a technique that produces such an approximate solution. Typically this is done by defining a non-negative loss function that measures departure from perfect fit and an algorithm for minimizing it.

In *Orthogonal Common Factor Analysis*, which is the most important special case of factor analysis, the set $\mathcal{T}$ is a set of partitioned matrices $\begin{bmatrix} A & \mid & D \end{bmatrix}$, and there are separate constraints

[1] A matrix Y is orthogonal if $Y'Y$ is diagonal, an orthogonal matrix is orthonormal if $Y'Y = I$.

$A \in \mathcal{A}$ and $D \in \mathcal{D}$. matrix of *common factor loadings* A is $m \times q$, with[2] $q := p - m$, and the matrix of *unique factor loadings* D is of order m. The set $\mathcal{Y}$ are the orthonormal $n \times p$ matrices Y with $Y'Y = I$. There is a corresponding partition of Y as $\begin{bmatrix} F & \mid & U \end{bmatrix}$, with F the $n \times q$ matrix of *common factor scores* and U the $n \times m$ matrix of *unique factor scores*. If A is unrestricted and D is required to be diagonal then the factor analysis is *exploratory*, otherwise it is *confirmatory*. If D is diagonal then the squares of its diagonal elements are called *unique variances* or *uniquenesses*.

For orthogonal common factor anaysis the system (1a)-(1c) thus becomes

$$\begin{aligned} X &= FA' + UD, && \text{(2a)}\\ F'F &= I, && \text{(2b)}\\ U'U &= I, && \text{(2c)}\\ F'U &= 0, && \text{(2d)}\\ D &\in \mathcal{D}, && \text{(2e)}\\ A &\in \mathcal{A}. && \text{(2f)} \end{aligned}$$

Although common factor analysis will always be in the back of our mind, we will develop equations and algorithms for the general orthognal case $X = YT$ with $Y'Y = I$ and $T \in \mathcal{T}$.

1.1 Some Tools

Definition 1.3. Suppose X is an $n \times m$ "tall" matrix of rank r. A "fat" SVD for X is any decomposition

$$X = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L' & L'_\perp \end{bmatrix}. \tag{3}$$

with

- Λ diagonal and positive definite of order r with elements decreasing[3] along the diagonal,
- K is $n \times r$ with $K'K = I$,
- L is $m \times r$ with $L'L = I$,
- $K_\perp$ is $n \times (n - r)$ with $K'_\perp K_\perp = I$,
- $L_\perp$ is $m \times (m - r)$ with $L'_\perp L_\perp = I$,
- $K'K_\perp = 0$,
- $L'L_\perp = 0$.

[2] The symbol $:=$ is used for definitions.

[3] Decreasing means $\lambda_1 \geq \cdots \geq \lambda_r$

Remark 1.1. In the "fat" SVD the diagonal matrix Λ of *singular values* is uniquely defined. The matrices $K_\perp$ and $L_\perp$ have orthonormal bases for the left and the right null-spaces of X, and are consequently only unique up to a rotation. The matrices of left and right *singular vectors* K and L have orthonormal bases for the column and row spaces of X, and are uniquely defined if and only if the singular values are all different.

Definition 1.4. The Moore-Penrose Inverse of an $n \times m$ matrix X is any $m \times n$ matrix X^+ satisfying the four Penrose conditions

1. XX^+ is symmetric,
2. X^+X is symmetric,
3. $X^+XX^+ = X^+$,
4. $XX^+X = X$.

Theorem 1.1. *If X has "fat" SVD given by* (3) *then $Y = X^+$ if and only if*

$$Y = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix}. \tag{4}$$

Proof. To prove sufficiency we merely have to verify that Y from (4) satisfies the four Penrose conditions. Necessity (and uniqueness) is a bit more involved. Any $m \times n$ matrix Y can be written as

$$Y = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix}. \tag{5}$$

Thus

$$XY = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda P & \Lambda Q \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix}. \tag{6}$$

If Y is a Moore-Penrose Inverse of X then XY must be symmetric, and thus $Q = 0$. Next

$$YX = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} P\Lambda & 0 \\ R\Lambda & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}. \tag{7}$$

If Y is a Moore-Penrose Inverse of X then YX must be symmetric, and thus $R = 0$. Next

$$XYX = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda P\Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}. \tag{8}$$

If Y is a Moore-Penrose Inverse of X then XYX must be equal to X, and thus $P = \Lambda^{-1}$. Finally

$$YXY = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} P\Lambda P & P\Lambda Q \\ R\Lambda P & R\Lambda Q \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}. \tag{9}$$

If Y is a Moore-Penrose Inverse of X then R and Q are zero, and we must have $YXY = Y$, which implies $S = 0$. □

Definition 1.5. Suppose X is an $n \times m$ "tall" matrix. Suppose $\mathcal{Y}$ is the set of all $n \times m$ matrices with $Y'Y = I$. Then the *Procrustus Transformation* of X is defined as

$$\Pi_{\mathcal{Y}}(X) := \underset{Y\in\mathcal{Y}}{\text{argmin}}\ \text{SSQ}(X - Y) = \underset{Y\in\mathcal{Y}}{\text{argmax}}\ \text{tr}\, Y'X.$$

Theorem 1.2. *Suppose X is an $n \times m$ "tall" matrix of rank r with "fat" SVD given by* (3)*. Then*

$$\Pi_{\mathcal{Y}}(X) := \{Y \mid Y = KL' + K_\perp SL'_\perp\}$$

with S any $(n - r) \times (m - r)$ matrix satisfying $S'S = I$.

Corollary 1.1. *If X is an $n \times m$ "tall" matrix of rank r with "fat" SVD given by* (3) *then*

$$\max_{Y\in\mathcal{Y}} \mathit{tr}\, Y'X = \mathit{tr}\, \Lambda,$$

and the maximum is attained for any $Y \in \Pi_Y(X)$.

Definition 1.6. The positive semidefinite square root of a positive semidefinite A is positive semidefinite
matrix B with $B^2 = A$.

Theorem 1.3. *A positive semi-definite matrix A has a unique positive semidefinite square root $A^{\frac{1}{2}}$.*

Proof. The "fat" SVD of A (which is identical here to its eigenvalue decomposition) is

$$A = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix}. \tag{10}$$

Any other symmetric matrix of the same size as A can be written as

$$B = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} P & Q \\ Q' & R \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix}, \tag{11}$$

with P and R symmetric. Thus

$$B^2 = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} P^2 + QQ' & PQ + QR \\ Q'P + RQ' & Q'Q + R^2 \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix} \tag{12}$$

Now $B^2 = A$ implies $Q'Q + R^2 = 0$, which implies $Q = 0$ and $R = 0$. Also $P^2 = \Lambda$, and thus

$$A^{\frac{1}{2}} = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda^{\frac{1}{2}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix} \tag{13}$$

□

Definition 1.7. $X = YS$ is a polar decomposition of the $n \times m$ matrix X if $Y'Y = I$ and S is positive semidefinite.

Theorem 1.4. *$X = YS$ is a polar decomposition if and only if $S = (X'X)^{\frac{1}{2}}$ and $Y \in \Pi_y(X)$.*

Proof. $X = YS$ implies $X'X = S^2$, and thus S is the unique square root of $X'X$. Suppose X has "fat" SVD given by (3). Any $n \times m$ matrix Y can be written as

$$Y = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}. \tag{14}$$

Also

$$S = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix} \tag{15}$$

Thus

$$YS = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} P\Lambda & 0 \\ R\Lambda & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}. \tag{16}$$

This is equal to X if and only if $P = I$ and $R = 0$. Taking this into account

$$Y'Y = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} I & Q \\ Q' & Q'Q + S'S \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}. \tag{17}$$

which is equal to the identity if and only if $Q = 0$ and $S'S = I$. □

1.2 Fundamental Theorem

In this section we look at finding exact solutions of the "full" system of equations (1a)-(1c). As we will see, the basic solvability result, which we call the "Fundamental Theorem of Factor Analysis", following Kestelman (1952), also has computational consequences.

We start by considering the "reduced" OFA system, using $C := X'X$,

$$C = TT', \tag{18a}$$

$$T \in \mathcal{T}. \tag{18b}$$

Solvability of the reduced system is a necessary condition for solvability of the full system (1a)-(1c).

Most factor analysis procedures are two-step methods. In the first step they find an approximate solution T to the reduced system, and then use this T to find an approximate solution Y to the full system. The two steps may actually use two different loss functions, the first one to assess the fit of $C = TT'$ and the second one the fit of $X = YT'$. This practice is motivated, and to some extent justified, by the following theorem, first proved by Garnett (1919).

Theorem 1.5. *The* full *system* (1a)-(1c) *is solvable if and only if the* reduced *system* (18a)-(18b) *is solvable.*

Proof. Necessity is trivial. For sufficiency we have to prove that if T is any solution of the reduced system then there is a Y such that (T, Y) satisfies the full system. Our proof uses the "fat" singular value decomposition (SVD) of the $n \times m$ matrix X, assumed to be of rank r, which is

$$X_{n\times m} = \begin{bmatrix} K^1_{n\times r} & K^0_{n\times(n-r)} \end{bmatrix} \begin{bmatrix} \Lambda_{r\times r} & 0_{r\times(m-r)} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(m-r)} \end{bmatrix} \begin{bmatrix} L^1_{m\times r} & L^0_{m\times(m-r)} \end{bmatrix}'. \tag{19}$$

The singular values in $\Lambda_{r\times r}$ are positive and decrease along the diagonal. Subscripts are used to indicate the dimension of the matrices.

From $C = TT'$ we know that a "fat" SVD of T is

$$T_{m\times p} = \begin{bmatrix} L^1_{m\times r} & L^0_{m\times(m-r)} \end{bmatrix} \begin{bmatrix} \Lambda_{r\times r} & 0_{r\times(p-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(p-r)} \end{bmatrix} \begin{bmatrix} M^1_{p\times r} & M^0_{p\times(p-r)} \end{bmatrix}', \tag{20}$$

for some square orthonormal M.

Write Y as

$$Y_{n\times p} = \begin{bmatrix} K^1_{n\times r} & K^0_{n\times(n-r)} \end{bmatrix} \begin{bmatrix} V^{11}_{r\times r} & V^{10}_{r\times(p-r)} \\ V^{01}_{(n-r)\times r} & V^{11}_{(n-r)\times(p-r)} \end{bmatrix} \begin{bmatrix} M^1_{p\times r} & M^0_{p\times(p-r)} \end{bmatrix}', \tag{21}$$

where the partitioned matrix V in the middle must satisfy $V'V = I$.

Now $X = YT'$ becomes

$$\begin{bmatrix} \Lambda_{r\times r} & 0_{r\times(m-r)} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(m-r)} \end{bmatrix} = \begin{bmatrix} V^{11}_{r\times r} & V^{10}_{r\times(p-r)} \\ V^{01}_{(n-r)\times r} & V^{11}_{(n-r)\times(p-r)} \end{bmatrix} \begin{bmatrix} \Lambda_{r\times r} & 0_{r\times(m-r)} \\ 0_{(p-r)\times r} & 0_{(p-r)\times(m-r)} \end{bmatrix}, \tag{22}$$

which implies

$$V = \begin{bmatrix} I_{r\times r} & 0_{r\times(p-r)} \\ 0_{(n-r)\times r} & S_{(n-r)\times(p-r)} \end{bmatrix} \tag{23}$$

where S satisfies $S'S = I$, but is otherwise arbitrary.

From (21) and (23) we see that

$$Y = K_1 M_1' + K_0 S M_0' \tag{24}$$

provides us with the required solution of the full system. □

If the non-zero singular values of X, and consequently of T, are all different then K_1 and M_1 are uniquely determined. Matrices K_0 and M_0 consist of orthonormal bases for the null-spaces of X and T, which are only determined up to rotations. They do not have to come from an SVD, they can be computed more efficiently by QR decomposoition or by reducing the matrix to row/column echelon form. We can select any one of these bases and absorb the rotations in the arbitrary matrix S. The fundamental theorem implies that for any solution $T \in \mathcal{T}$ of $C = TT'$ there is a nonlinear manifold of solutions Y of $X = YT'$. Even if T is identified, Y is not. This is the *factor indeterminacy problem*, which has haunted the field for more than 100 years. (Steiger and Schönemann (1978)).

2 Least Squares Factor Analysis

2.1 ULS

There are several OFA techniques that use least squares loss functions[4]. The oldest and most obvious one is

$$\sigma(T) = \mathrm{SSQ}(C - TT'), \tag{25}$$

where SSQ() stands for the unweighted sum of squares. Minimizing (**??**) is the *Unweighted Least Squares (ULS)* method of orthogonal factor analysis.

For OCFA (25) becomes

$$\sigma(A, D) = \mathrm{SSQ}(C - AA' - D^2). \tag{26}$$

An alternating least squares technique to minimize (26) was first proposed by Thomson (1934). It alternates minimizing the loss function (26) over D for the current A and minimizing over A for the current D. Of course ULS only finds an approximate solution to the reduced system, and it leaves open the question on how to compute the factor scores Y. Alternative algorithms for minimizing (26) are MINRES of Harman and Jones (1966) and the Newton-Raphson technique of Derflinger (1969) and Jöreskog and Van Thillo (1971).

The file lsfa.R in the repository has a function lsfa() to do an ULS for OCFA. It needs an initial estimate of D, for which we can take $\mathrm{diag}^{-1}(C^{-1})$. Then it applies a small number of Thomson iterations before switching to Newton iterations. First and second derivatives are taken from De Leeuw (2025). Eigenvalue/eigenvector calculations use the RSpectra package (Qiu and Mei (2024)).

2.2 GLS

Joreskog and Goldberger (1972)

$$\sigma(T) := \mathrm{tr}\, C^{-1}(C - TT')C^{-1}(C - TT')$$

majorization, through ULS.

$$\mathbf{SSQ}(I - \tilde{A}\tilde{A}' - C^{-\frac{1}{2}}DC^{-\frac{1}{2}}) = \sum_{s=p+1}^{m} \lambda_s^2(I - C^{-\frac{1}{2}}DC^{-\frac{1}{2}})$$

[4] We use SSQ() for the unweighted sum of squares of a matrix or vector.

2.3 MLS

Also for OCFA Young (1940) and Whittle (1952) propose minimizing the weighted least squares loss function

$$\sigma(A, D, F) = \text{tr }(X - FA')D^{-2}(X - FA')', \tag{27}$$

which has the disadvantage that it assumes D is known and the advantage that the problem becomes a form of principal component analysis (PCA) in which the solution can be computed with a single singular value decomposition.

It is tempting to use a block method to minimize (27), alternating finding the optimum F and A for fixed D, and then repeating this iteratively with the new D equal to

$$D^2 = n^{-1}\text{diag }(X - FA')'(X - FA'). \tag{28}$$

This in fact corresponds to alternating minimization of the loss function

$$\sigma(A, D, F) = n \log \det(D^2) + \text{tr }(X - FA')D^{-2}(X - FA')'. \tag{29}$$

But (29) is the negative likelihood loss function for the fixed factor model proposed by Lawley (1941), and we know this loss function is unbounded below, does not have a minimum, and converges to a perfect but trivial solution which has p of the uniquenesses in D^2 equal to zero (Anderson and Rubin (1956)).

$$\min_{F,\tilde{A}} \text{SSQ}(XD^{-1} - F\tilde{A}) = \sum_{s=p+1}^{m} \lambda_s(D^{-1}CD^{-1})$$

2.4 MDFA

A more direct way of fitting the full orthogonal common factor equations was discussed (independently and around the same time) in the dissertation of Sočan (2003)[5] and in the conference chapter of De Leeuw (2004) (presented at the conference in 2002). Recently this technique has become known as *Matrix Decomposition Factor analysis (MDFA)*.

In MDFA the least squares loss function is

$$\sigma(Y, T) := \text{SSQ}(X - YT'), \tag{30}$$

which must be minimized over $Y'Y = I$ and $T \in \mathcal{T}$.

[5] Attributes to a 2001 personal communication and some unpublished notes of Henk Kiers.

MDFA was not immediately accepted as an alternative factor analysis technique. It made its first journal appearance in a series of papers by Unkel and Trendafilov, based largely on Unkel's dissertation (Unkel (2009)). Over the years they contributed a "robust" version of MDFA (Unkel and Trendafilov (2010)) and a version for a "wide" data matrix X (Trendafilov and Unkel (2011)). There is a nice review of their contributions in Unkel and Trendafilov (2010), with an update in Trendafilov, Unkel, and Krzanowski (2013).

There have been additional important contributions to MDFA in Adachi (2012), Adachi and Trendafilov (2018), Stegeman (2016), Terada (2025), and Yamashita (2025). We will discuss these recent contributions in various places in our present paper. When googling MDFA, keep in mind that Adachi initially used *Data-Fitting Factor Analysis*, while Stegeman used the *Data Factor Model*.

3 MDFA Algorithms

3.1 Algorithm A

The Alternating Least Squares (ALS) algorithm proposed for MDFA by both Kiers (in Sočan (2003)) and De Leeuw (2004) alternates finding the optimum Y for given T and the optimum T for given Y.

Finding the optimum T for given Y is straightforward. We complete the square, as in

$$\sigma(Y,T) = \text{tr}\, C + \text{tr}(T - X'Y)'(T - X'Y) - \text{tr}\, Y'CY. \tag{31}$$

Thus the optimum T for given Y is obtained by projecting the $m \times p$ matrix $X'Y$ on the set of matrices $\mathcal{T}$. We shall use $\Pi_{\mathcal{T}}()$ for the unweighted least squares projection on $\mathcal{T}$.

$$T \leftarrow \Pi_{\mathcal{T}}(X'Y)$$

For exploratory OCFA this gives

$$\left[A \quad | \quad D\right] \leftarrow \left[X'F \quad | \quad \text{diag}(X'U)\right]. \tag{32}$$

For confirmatory MDFA in which we require some loadings to be equal to each other and/or to to fixed constants finding $\Pi_{\mathcal{T}}(X'Y)$ is still a simple linear least squares problem. If there are inequality constraints we need some form of quadratic programming.

Expanding loss gives

$$\sigma(Y,T) = \text{tr}\, C + \text{tr}\, T'T - 2\, \text{tr}\, Y'XT. \tag{33}$$

To compute the optimal Y for given T we have to maximize tr $Y'XT$ over all $n \times p$ orthonormal Y. We start with some general lemmas. Remember that the *trace norm* (a.k.a. the *nuclear norm*) $\|Z\|_{\tau}$ of a matrix Z is the sum of its singular values. The trace norm is also equal to the sum of the square roots of the eigenvalues of $Z'Z$ and thus equal to the trace of $(Z'Z)^{1/2}$.

Lemma 3.1. *If Z and Y are $n \times p$ with $Y'Y = I$ then*

$$\textit{tr}\, Y'Z \leq \|Z\|_{\tau}. \tag{34}$$

Proof. The stationary equations are $Z = YM$ with $Y'Y = I$ and M a symmetric matrix of Lagrange multipliers of order p. It follows that $M^2 = Z'Z$ and thus M is the (unique) positive semi-definite symmetric square root $(Z'Z)^{1/2}$ At the maximum $Z = YM$ implies tr $Y'Z =$ tr $M = \|X\|_{\tau}$. □

Note that Lemma 3.1 does not say that the optimizing Y is unique. We get more insight from an alternative and more constructive proof (also taken from De Leeuw (2004), Appendix). It uses the "fat" singular value decomposition of the $n \times p$ matrix Z of rank r.

Lemma 3.2. *If the $n \times p$ matrix Z with rank r has the "fat" SVD*

$$Z = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}, \tag{35}$$

then the $n \times p$ orthonormal Y maximizes tr $Y'Z$ if and only if

$$Y = KL' + K_\perp S L'_\perp, \tag{36}$$

for some $(n-r) \times (p-r)$ matrix S with $S'S = I$.

Proof. Partition the columnwise orthonormal $n \times p$ matrix Y in the same way as Z.

$$Y = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}, \tag{37}$$

where the four-block partitioned $n \times p$ matrix in the middle of (37) must be columnwise orthonormal. Now tr $Y'X =$ tr ΛP and the constraint on the $r \times r$ matrix P is that $P'P \lesssim I$ in the Loewner[6] sense (because we must have $P'P + R'R = I$). It follows that the diagonal elements of P must be less than or equal to one, and thus the maximum of tr $Y'X$ is tr Λ, uniquely attained for $P = I$. But $P = I$ implies that at the maximum $R = 0$ and $Q = 0$, and S is any $(n-r) \times (p-r)$ matrix with $S'S = I$. Thus the optimum Y is as given by (36). □

We can now apply the lemmas to minimizing the loss in (33). This generalizes the fundamental theorem of factor analysis to the case of imperfect fit.

Theorem 3.1. *If the $n \times p$ matrix XT has rank r with $r \leq m < p < n$ and "fat" singular value decomposition*

$$XT = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}, \tag{38}$$

then the maximum of tr $Y'XT$ over $n \times p$ orthonormal Y is $\|XT\|_\tau$, attained for all Y with

$$Y = KL' + K_\perp S L'_\perp, \tag{39}$$

where S is any $(n-r) \times (p-r)$ matrix with $S'S = I$.

[6] $A \lesssim B$ means $A - B$ positive semi-definite.

Proof. Directly from Lemma 3.2. □

It is important to see that in MDFA we cannot use $Y = KL'$, because then $Y'Y = LL'$, which is of rank $r \leq p$ and thus not equal to the identity if $r < p$. Also note that formula (39) is identical to formula (24) in the fundamental theorem. This is not surprising, since both compute the optimal Y for given T. The fundamental theorem merely adds that the optimal Y makes loss equal to zero if $C = TT'$.

In factor analytic literature it is often said that the factor scores Y have the "determinate" part KL' and the "indeterminate" part $K_\perp SL'_\perp$. There are some simple rank conditions which imply that the indeterminate part does not play a role in Algorithm A, which uses $X'Y$ to update T, and in the fitted values, which use YT'. And consequently in the loss function values, which use $T'X'Y$.

Corollary 3.1. *If XT has rank m and $Y = KL' + K_\perp SL'_\perp$ then*

1. $X'Y = X'KL' = CT(T'CT)^{-1/2}$.
2. $YT' = KL'T' = XT(T'CT)^{-1/2}T'$.

Proof. If XT is of rank m then both X and T are of rank m. We know that $K'_\perp XT = 0$. Because $\text{rank}(T) = m$ this implies $X'K_\perp = 0$. This proves part one. Because $\text{rank}(X) = m$ and $XTL_\perp = 0$ we have $L'_\perp T = 0$. This proves part two. □

The CFA equations (2a)-(2f) imply that $0 \lesssim AA' \lesssim C$ and $0 \lesssim D^2 \lesssim C$. The second one in turn implies that $0 \lesssim D^2 \lesssim \text{diag}(C)$. In most CFA techniques it is possible that the computed uniquenesses do not satisfy these inequality constraints and are consequently "improper" solutions. This is true for both MINRES and ML. We show that it is "less true" for MDFA.

Theorem 3.2. *In the exploratory MDFA solution of CFA we have $0 \lesssim AA' \lesssim C$ and $0 \lesssim D^2 \lesssim \text{diag}(C)$.*

Proof. At the solution we have $A = X'F$ and thus $AA' = X'FF'X$. Because $FF' \lesssim I$ we have $AA' \lesssim C$. At the solution we also have $D = \text{diag}(X'U)$, or $d_j = x'_j u_j$. Since $u'_j u_j = 1$ we have from Cauchy-Schwartz $d_j^2 \leq x'_j x_j = c_{jj}$. □

Theorem 3.2 shows that worst-case improper solutions do not happen in MDFA. Uniquenesses are always non-negative and bounded above by the diagonal of C, i.e. for correlation matrices they are always between zero and one. Note however that we have not shown that $D^2 \lesssim C$, so not all improper solutions are excluded. If it so happens that $U'X$ is diagonal at the solution, which is certainly not guaranteed, then indeed $D^2 = X'UU'X \lesssim C$.

The repository has a file mdfaAlgorithmA.R, with the function mdfaAlgorithmA() implementing Algorithm A. There is an argument proj which is a function that takes care of the least squares projections on $\mathcal{T}$. By default proj is mdfaCFAProject(), which replaces the last m columns of T by its diagonal, but changing this default projection routine allows us to incorporate any set of constraints.

We emphasize that mdfaAlgorithmA() updates Y using the determinate part KL' only, which means that in our iterations (and at convergence) we do not have $Y'Y = I$. Corollary 3.1 shows that if we were to complete Y by adding $K_\perp SL_\perp'$ in each iteration we would get the same sequence of loss function values and the same sequence of T values.

It does mean however that if we want a proper Y then we have to do some extra work after convergence. Adachi (2012), section 2.5, makes a case for using only the determinate part, but it remains true that this does not solve the original MDFA problem, which requires $Y'Y = I$. Also computing a proper Y corresponding with the optimal T forces us to make a choice of S in (39). In mdfaAlgorithmA() after convergence we optionally use the QR decompositon to compute an orthonormal bases $K_\perp$ of the left null space and an orthonormal basis $L_\perp$ for the right null space of XT. We use a function identical to Null() from the package MASS (Venables and Ripley (2002)). We choose the arbitrary $(n-m) \times (p-m)$ matrix S by setting its first $p - m$ rows to the identity matrix, with the remaining rows zero.

3.2 Algorithm B

Write $\Pi_Y()$ for the (generally set-valued) least squares projection of a matrix on the columnwise orthonormal matrices of the same dimension. Also write $\Pi_\mathcal{T}()$ for least squares projection on $\mathcal{T}$. We also use the $\leftarrow$ symbol for updates. Thus $x \leftarrow f(x, y)$ means the update of x (the "new" X) in an iteration is a function f of the current x and the current y. On the left hand side is what we are computing now and on the right hand side is what we have computed before.

Then ALS iteration of Algorithm A is

$$Y \leftarrow \Pi_\mathcal{Y}(XT), \tag{40a}$$

$$T \leftarrow \Pi_\mathcal{T}(X'Y). \tag{40b}$$

If we assume that $\text{rank}(X) = \text{rank}(T) = m$ then it does not matter how we choose S in (36), because by Corollary 3.1 for all choices of S we have tr $Y'XT = \|XT\|_{\mathcal{T}}$, and all choices of S give the same update and the same loss function value.

In most factor analytic applications we have $n >> m$. The computations in (40a) and (40b) use the matrices X and Y, which each have n rows. Every iteration involves multiplications with these potentially very large matrices. This will tend to be expensive computationally. It has been pointed out by Adachi (2012) that we can rewrite the ALS algorithm in such a way that it only involves matrices of order m. This alternative derivation of the algorithm has the additional property that it can be applied to C without having to know X, which has obvious advantages in secondary analysis. And computational factor analysis publications typically have secondary analyses examples, using the mountain of correlation matrices that have been published since 1900.

Theorem 3.3. *Suppose T has rank m. Then the update is*

$$T \leftarrow \Pi_{\mathcal{T}}\{CT(T'CT)^{-1/2}\}. \tag{41}$$

Proof. We rewrite (36) as

$$Y = XT(T'CT)^{-1/2} + K_{\perp}SL'_{\perp} \tag{42}$$

where $(T'CT)^{-1/2}$ is the square root of the Moore-Penrose inverse of $T'CT$, $K_{\perp}$ and $L_{\perp}$ are orthonormal bases for the left and right null spaces of XT, and S is arbitrary.

$$T \leftarrow \Pi_{\mathcal{T}}(X'Y) = \Pi_{\mathcal{T}}\left\{CT(T'CT)^{-\frac{1}{2}} + X'K_{\perp}SL'_{\perp}\right\} \tag{43}$$

We have $K'_{\perp}XT = 0$. If T has rank m then Corollary 3.1 implies $X'K_{\perp} = 0$ which leads to the result in the theorem. □

Algorithm (41) requires us to compute $(T'CT)^{-\frac{1}{2}}$ in each iteration, which means finding the m non-zero eigenvalues and their corresponding eigenvectors. The update is then projected on the set $\mathcal{T}$.

Alternative: polar decomposition iterative

The ALS interpretation of the algorithm shows directly that a decreasing sequence of loss function values is produced, which proves convergence of T under the usual identification and rank conditions.

Since $\phi(T) := \|XT\|_{\mathcal{T}}$ is convex and homogeneous in T the algorithm (49) is also a majorization (or MM) algorithm (De Leeuw (1994), Lange (2016)). To show this, it follows from convexity that for all T and $\overline{T}$ we have $\phi(T)\geq\text{tr }T'\overline{G}$, where $\overline{G}$ is any element of the subdifferential $\partial\phi(\overline{T})$. Thus

$$\sigma_\star(T)\leq\text{tr }C+\text{tr }T'T-2\text{ tr }T'\overline{G}=\text{tr }C+\text{SSQ}(T-\overline{G})+\text{SSQ}(\overline{G}),\tag{44}$$

with equality if $T=\overline{T}$. Since $\|XT\|_\tau=\max_{Y'Y=I}\text{tr }Y'XT$ it follows from the theorem about subgradients of linear transformations (Rockafellar (1970), theorem 23.9) and the theorem about subgradients of supremum functions (Mordukhovich and Nam (2022), section 4.4) that

$$\partial\phi(T)=X'(KL'+K_\perp\overline{\text{co}}(\mathcal{S})L_\perp')\tag{45}$$

where K and L refer to the SVD of XT, where $\mathcal{S}$ is the Stiefel manifold of all S with $S'S=I$, and where $\overline{\text{co}}()$ is the closed convex hull. Finally (Watson (1992), Gallivan and Absil (2010)) $\overline{\text{co}}(\mathcal{S})$ is the compact convex set $\mathcal{H}$ of all matrices with largest singular value less than or equal to one. Thus

$$T\leftarrow\Pi_\mathcal{T}(\partial\phi(T))=\Pi_\mathcal{T}(X'(KL'+K_\perp HL_\perp')),\tag{46}$$

with $H\in\mathcal{H}$ is a convergent MM algorithm. Since $\mathcal{H}$ is larger than $\mathcal{S}$ this extends Algorithm B. On the other hand as long as T has rank m $K_\perp'XT=0$ implies $X'K_\perp=0$, as in Corollary 3.1, and thus in that case the MM algorithm iterates the determinate part KL' and is identical to Algorithm B.

If XT has rank m then $\mathcal{D}\phi(T)=X'KL'$

In Algorithm A we compute the determinate part of Y, and optionally after convergence a proper orthonormal Y. In Algorithm B there is no Y at all, and if there is no X we cannot compute factor scores. In both cases finding appropiate (indeterminate) factor scores Y requires additional work, not unlike what must be done in the MINRES or ML techniques for CFA. One option is to simply compute the optimal T and then use one of the various classical methods to compute factor scores (McDonald and Burr (1967)).

3.3 Algorithm C

The minimum of $\sigma(Y,T)$ over $Y'Y=I$ is

$$\begin{aligned}\sigma_\star(T):=&\text{tr }X'X+\text{tr }T'T-2\,\|XT\|_\tau\\=&\text{tr }X'X+\text{tr }T'T-2\sum_{\nu=1}^m\lambda_\nu^{\frac12}(T'CT).\end{aligned}\tag{47}$$

First and second derivatives of the trace norm with respect to T were computed in De Leeuw (2025). The gradient is

$$\mathcal{D}\sigma_\star(T)=2\{T-CT(T'CT)^{-1/2}\}.\tag{48}$$

Thus the exploratory iterative algorithm

$$T \leftarrow CT(T'CT)^{-1/2} \tag{49}$$

is a fixed-point algorithm, and $T \in \mathcal{T}$ is a fixed point of (49) if and only if $\mathcal{D}\sigma_\star(T) = 0$.

If the problem is defined in such a way that some elements of T are fixed, usually at zero, then the gradient is still the same, but with the understanding that the fixed elements get derivatives equal to zero. In other words there is a binary $m \times p$ matrix B such that $T \in \mathcal{T}$ if and only if $B * T = T$, using symbol $*$ for the elementwise (or Hadamard) product. For constraints of this type the gradient is

$$\mathcal{D}\sigma_\star(T) = B * \{T - CT(T'CT)^{-1/2}\}. \tag{50}$$

With this formula for the gradient many general purpose optimization methods for minimizing the projected loss function (47) become available. The R function mdfaAlgorithmG() in the repository uses the BFGS method of the optim() function from the stats package (R Core Team (2025)).

Compared with Algorithm B we can perhaps expect some gain in efficiency, because of the superlinear convergence of the BFGS method, but we lose some flexibility by not having the general projections $\Pi_{\mathcal{T}}()$ available.

box constraints

3.4 Algorithm D

$\phi(T) = \|XT\|_\tau$

$\mathcal{D}\phi(T) = X'KL'$

Algorithm B: $T - \Pi_T(X'KL') = T - \Pi_T(CT(T'CT)^{-\frac{1}{2}}) = 0$

In Algorithms A and B the projection $\Pi_{\mathcal{T}}()$ can be quite general. Algorithm F uses the binary template B to encode fixed and variable elements, and projection is just elementwise multiplication with the template. We can get more generality by defining $\mathcal{T}$ as the set of matrices of the form

$$T = T_0 + \sum_{s=1}^{h} \theta_s T_s, \tag{51}$$

which allows for patterns with some elements fixed to known values but also allows some elements required to be equal to other elements. Fixed elements go into T_0 and the T_s with $s > 0$ have zeroes where the fixed elements are. If the fixed elements are all fixed to zero, as they are in exploratory CFA, there is no need for a T_0. If (51) is combined with non-negativity restrictions on the θ_s then we can also incorporate inequality between the elements of T, i.e. $\mathcal{T}$ can be a polyhedral convex cone.

For the parametrization (51)

$$\mathcal{D}_s\sigma_\star(T) = \text{tr } T'T_s - \text{tr } (T'CT)^{-1/2}T'CT_s. \tag{52}$$

The derivatives of the loss function (52) with respect to the parameters θ_s are most easily expressed by first computing the derivatives of the eigenvalues of $T'CT$.

$$\mathcal{D}_s\lambda_\nu = x_\nu'Q_sx_\nu, \tag{53}$$

where $Q_s := T_s'CT + T'CT_s$. It follows that

$$\mathcal{D}_{st}\lambda_\nu = -2x_\nu'Q_s(A - \lambda_\nu I)^+Q_tx_\nu + 2x_\nu'T_s'CT_tx_\nu. \tag{54}$$

3.5 Algorithm E

An MDFA procedure $X \rightarrow (Y, T)$ maps data matrices X to a scores Y and loadings T. The procedure is scale-free if for all diagonal Φ we have $X\Phi \rightarrow (Y, T\Phi)$. Because of the uncertainly of appropriate units for the variables it is generally thought that scale-freeness is a desirable property of multivariate technique.

The MDFA least squares techniques we have discussed so far are not scale-free, but they can be made scale-free in various ways. The first way is a traditional one. We normalize the data in X to unit length, so that $C = X'X$ becomes a correlation matrix. This can be formalized as minimizing

$$\sigma_W(T, Y) := \text{tr } (X - YT')W(X' - TY')$$

with $W := \text{diag}^{-1}(C)$. Adachi (2012) proposes instead to use $W = C^{-1}$. But no matter how we choose the positive definite W the computational consequences are easy to handle.

We have

$$\sigma_W(T, Y) = \text{tr } XWX' + \text{tr } (T - X'Y)'W(T - X'Y) - \text{tr } Y'XWX'Y, \tag{55}$$

and consequently the optimal T for given Y is the projection in the metric W of $X'Y$ on $\mathcal{T}$. This may or may not be more complicated than unweighted projection, depending on the definition of $\mathcal{T}$.

Also

$$\sigma_W(T, Y) = \text{tr } XWX' + \text{tr } T'WT - 2 \text{ tr } Y'XWT, \tag{56}$$

which means the optimal Y for given T is the (unweighted) projection of XWT on the orthonormal $n \times p$ matrices.

As in the unweighted case we can combine the two ALS steps into a one-step upgrade for T.

$$T \leftarrow \Pi_{\mathcal{T}}(CWT(T'WCWT)^{-1/2}). \tag{57}$$

Algorithm W is the version of Algorithm B that uses the update in (57). Of course it does need a weighted projection routine implementing $\Pi_{\mathcal{T}}()$. For exploratory

4 Examples

4.1 MacDonell

Our first example illustrates and compares several techniques for exploratory CFA. The example is of some historical interest. The data are one of the first published correlation matrices, from the very first volume of Biometrika. MacDonell (1902), a retired businessman volunteering in Karl Pearson's lab, analyzed the politically incorrect correlations between seven physical characteristics of 3000 criminals.

```
             [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]
Head Length  1.00000 0.40163 0.39454 0.30071 0.30539 0.33886 0.33993
Head Breadth 0.40163 1.00000 0.61779 0.15040 0.13515 0.20614 0.18308
Face Breadth 0.39454 0.61779 1.00000 0.32097 0.28869 0.36322 0.34527
Finger       0.30071 0.15040 0.32097 1.00000 0.84638 0.75871 0.66084
Cubit        0.30539 0.13515 0.28869 0.84638 1.00000 0.79699 0.79986
Foot         0.33886 0.20614 0.36322 0.75871 0.79699 1.00000 0.73636
Height       0.33993 0.18308 0.34527 0.66084 0.79986 0.73636 1.00000
```

MacDonell mentions that Pearson advised him to use principal component analysis to find numerical indices that could be used to rank and classify criminals. He promises the PCA results in a follow-up paper, which never appeared (I think). It was probably too early for PCA, which was brand new at the time (Pearson (1901)).

We will compute two-factor exploratory MDFA, ULS, and ML solutions. To compare them we rotate the loadings of all three so that $A'D^{-2}A$ is diagonal.

We cannot apply algorithm A here, because we do not have the data matrix X, only the correlation matrix C. Algorithm B converges in 86 iterations. The loadings are

```
             [,1]      [,2]
Head Length  +0.371534 +0.395183
Head Breadth +0.212495 +0.798347
Face Breadth +0.386271 +0.668991
Finger       +0.865813 -0.038468
Cubit        +0.958579 -0.102605
Foot         +0.859463 +0.040946
Height       +0.825508 +0.027657
```

and the uniquenesses are

```
[1] +0.705545 +0.317129 +0.402941 +0.245376 +0.067481 +0.258232 +0.314327
```

The ULS algorithm uses 5 Thomson and 4 Newton iterations to minimize loss. Because of the Newton iterations the loss function values converged to 15 decimals precision. The loadings are

```
             [,1]      [,2]
Head Length  +0.376812 +0.388622
Head Breadth +0.218102 +0.813147
Face Breadth +0.394111 +0.651819
Finger       +0.856975 -0.042366
Cubit        +0.955459 -0.112785
Foot         +0.869081 +0.027989
Height       +0.824056 +0.022833
```

and the uniquenesses are

```
[1] +0.706986 +0.291224 +0.419809 +0.263799 +0.074378 +0.243915 +0.320411
```

Next we apply the maximum likelihood method, using factanal() from the stats package (R Core Team (2025)). The loadings are

```
             [,1]      [,2]
Head Length  +0.365333 +0.401091
Head Breadth +0.214124 +0.764567
Face Breadth +0.380562 +0.697782
Finger       +0.871261 -0.036722
Cubit        +0.960740 -0.102181
Foot         +0.847288 +0.051981
Height       +0.828229 +0.030106
```

and the uniquenesses are

```
[1] +0.705658 +0.369588 +0.368273 +0.239556 +0.066538 +0.279402 +0.313131
```

Because the data are already a correlation matrix Algorithm W with weights diag(C) gives the same result as the unweighted Algorithm B. But choosing weights C^{-1}

Algorithm B converges in 96 iterations. The loadings are

```
             [,1]      [,2]
Head Length  +0.364870 +0.401951
Head Breadth +0.212980 +0.770783
Face Breadth +0.379746 +0.697102
Finger       +0.874182 -0.037232
Cubit        +0.961129 -0.101275
Foot         +0.848774 +0.053606
Height       +0.830887 +0.032425
```

and the uniquenesses are

```
[1] +0.699296 +0.355826 +0.366204 +0.213144 +0.064469 +0.265352 +0.277413
```

4.2 Emmett

Our next example is the correlation matrix of order nine from Emmett (1949), also used in Lawley and Maxwell (1971), pp. 42-44. We use three common factors, and compare our results with those obtained using multinormal maximum likelihood (Lawley and Maxwell (1971), table 4.2).

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
 [1,] 1.000 0.523 0.395 0.471 0.346 0.426 0.576 0.434 0.639
 [2,] 0.523 1.000 0.479 0.506 0.418 0.462 0.547 0.283 0.645
 [3,] 0.395 0.479 1.000 0.355 0.270 0.254 0.452 0.219 0.504
 [4,] 0.471 0.506 0.355 1.000 0.691 0.791 0.443 0.285 0.505
 [5,] 0.346 0.418 0.270 0.691 1.000 0.679 0.383 0.149 0.409
 [6,] 0.426 0.462 0.254 0.791 0.679 1.000 0.372 0.314 0.472
 [7,] 0.576 0.547 0.452 0.443 0.383 0.372 1.000 0.385 0.680
 [8,] 0.434 0.283 0.219 0.285 0.149 0.314 0.385 1.000 0.470
 [9,] 0.639 0.645 0.504 0.505 0.409 0.472 0.680 0.470 1.000
```

As the initial estimate for A we use the first three principal components, scaled to the length of the corresponding eigenvalues. The initial estimate of D is the square root of the diagonal of $I - AA'$. Algorithm B converges in 396 iterations to a loss of 0.0059884, where convergence is defined by a loss-decrease in successive iterations of less than 10^{-10}. The MDFA uniquenesses are

```
[1] 0.449 0.422 0.617 0.210 0.381 0.174 0.403 0.465 0.230
```

We use factanal() from the stats package to compute the ML solution.

The minimum function value and iteration information is

```
      objective counts.function counts.gradient
     0.03501729     19.00000000     19.00000000
```

The uniquenesses from the ML solution are

```
[1] 0.450 0.427 0.617 0.212 0.381 0.177 0.400 0.462 0.231
```

Comparing loadings is slightly more involved. The ML solution is rotated so that $A'D^{-2}A$ is diagonal, so we rotate the MDFA solution in the same way. The MDFA loadings are

```
      [,1]   [,2]   [,3]
 [1,] +0.663 +0.324 +0.078
 [2,] +0.687 +0.253 -0.202
 [3,] +0.492 +0.305 -0.217
 [4,] +0.839 -0.288 -0.037
 [5,] +0.705 -0.313 -0.152
 [6,] +0.820 -0.377 +0.104
 [7,] +0.661 +0.393 -0.069
 [8,] +0.456 +0.296 +0.489
 [9,] +0.764 +0.431 -0.012
```

The ML loadings are

```
      [,1]   [,2]   [,3]
 [1,] -0.664 +0.321 -0.074
 [2,] -0.689 +0.247 +0.193
 [3,] -0.493 +0.302 +0.222
 [4,] -0.837 -0.292 +0.035
 [5,] -0.705 -0.315 +0.153
 [6,] -0.819 -0.377 -0.105
 [7,] -0.661 +0.396 +0.078
 [8,] -0.458 +0.296 -0.491
 [9,] -0.766 +0.427 +0.012
```

The Emmett example is somewhat atypical, because the CFA model with three factors has a near perfect fit. This explains why the MDFA and ML solutions are nearly identical.

The CFA solution computed by Algorithm G is identical to the Algorithm B solution. Loss, with 10 digits, returned by Algorithm G is 0.0059884299, while Algorithm B gives 0.0059884321. Algorithm G requires 29 function evaluations and 27 gradient evaluations. Comparing running times with microbenchmark is somewhat problematic, because the default option in Algorithm B is to iterate until the function value decrease is less than 10^{-10}. Algorithm G uses the default stopping criteria of the optim() function. Comparing the default options gives

```
Unit: milliseconds
                                                         expr      min
 mdfaAlgorithmB(emmett, emtold, itmax = 1000, verbose = FALSE) 8.776009
                       mdfaAlgorithmG(emmett, emtold, emtemp) 3.177090
       lq     mean   median       uq       max neval
 8.989967 9.773209 9.091238 9.226619 14.357544   100
 3.351770 3.608415 3.413066 3.479896  7.792173   100
```

4.3 Maxwell

Lawley and Maxwell give a second example, using data taken from Maxwell (1961) (ten variables, 810 observations). This example illustrates what happens when ML converges to an improper solution (a.k.a. Heywood case). The correlation matrix is

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]
 [1,] 1.000 0.345 0.594 0.404 0.579 0.280 0.449 0.188 0.303 0.200
 [2,] 0.345 1.000 0.477 0.338 0.230 0.159 0.205 0.120 0.168 0.145
 [3,] 0.594 0.477 1.000 0.498 0.505 0.251 0.377 0.186 0.273 0.154
 [4,] 0.404 0.338 0.498 1.000 0.389 0.168 0.249 0.173 0.195 0.055
 [5,] 0.579 0.230 0.505 0.389 1.000 0.151 0.285 0.129 0.159 0.079
 [6,] 0.280 0.159 0.251 0.168 0.151 1.000 0.363 0.359 0.227 0.260
 [7,] 0.449 0.205 0.377 0.249 0.285 0.363 1.000 0.448 0.439 0.511
 [8,] 0.188 0.120 0.186 0.173 0.129 0.359 0.448 1.000 0.429 0.316
 [9,] 0.303 0.168 0.273 0.195 0.159 0.227 0.439 0.429 1.000 0.301
[10,] 0.200 0.145 0.154 0.055 0.079 0.260 0.511 0.316 0.301 1.000
```

Algorithm B converges in 2255 iterations to a loss of 0.0058263. The MDFA uniquenesses are

```
[1] 0.373 0.606 0.308 0.634 0.381 0.780 0.293 0.000 0.694 0.587
```

The uniquenesses from the ML solution in Lawley and Maxwell's Table 4.6 are

```
[1] 0.385 0.623 0.301 0.638 0.347 0.778 0.286 0.000 0.690 0.600
```

Although the uniquenesses are close, there is an important difference between MDFA and ML. Obtaining the ML solution required manual intervention, because the four-factor solution was "improper". The uniqueness of variable 8 was judged to be a Heywood case, and the analysis with repeated without variable 8. The final results identify the first factor with variable 8, so the loadings on the first factor are just the corresponding column of the correlation matrix. MDFA gives the same results without any manual intervention. The only consequence of the Heywood case is that more iterations are required to drive the uniqueness to zero.

The MDFA loadings are, after suitable rotation,

```
       [,1]   [,2]   [,3]   [,4]
 [1,] -0.190 -0.756 +0.038 -0.131
 [2,] -0.118 -0.476 +0.113 +0.375
 [3,] -0.187 -0.766 +0.181 +0.192
 [4,] -0.175 -0.525 +0.218 +0.109
 [5,] -0.126 -0.659 +0.233 -0.336
 [6,] -0.354 -0.263 -0.153 +0.043
 [7,] -0.449 -0.505 -0.498 -0.043
 [8,] -1.000 +0.000 +0.000 -0.000
 [9,] -0.426 -0.285 -0.202 +0.047
[10,] -0.315 -0.238 -0.502 +0.062
```

The ML loadings are

```
       [,1]   [,2]   [,3]   [,4]
 [1,] +0.188 +0.753 -0.035 -0.108
 [2,] +0.120 +0.468 -0.103 +0.365
 [3,] +0.186 +0.767 -0.167 +0.217
 [4,] +0.173 +0.526 -0.200 +0.124
 [5,] +0.129 +0.672 -0.251 -0.349
 [6,] +0.359 +0.259 +0.154 +0.048
 [7,] +0.448 +0.504 +0.507 -0.052
 [8,] +1.000 +0.000 +0.000 +0.000
 [9,] +0.429 +0.282 +0.209 +0.053
[10,] +0.316 +0.232 +0.496 +0.029
```

4.4 Tucker

We continue with a simple confirmatory CFA example, with data from Tucker (1958). This example was also analyzed with ML by Jöreskog (1969). There are nine variables, the first four and the last five in two different test batteries.

```
    t42   t54   t45   t46   t23   t24   t27   t10   t51
t42 1.000 0.554 0.227 0.189 0.461 0.506 0.408 0.280 0.241
t54 0.554 1.000 0.296 0.219 0.479 0.530 0.425 0.311 0.311
t45 0.227 0.296 1.000 0.769 0.237 0.243 0.304 0.718 0.730
t46 0.189 0.219 0.769 1.000 0.212 0.226 0.291 0.681 0.661
t23 0.461 0.479 0.237 0.212 1.000 0.520 0.514 0.313 0.245
t24 0.506 0.530 0.243 0.226 0.520 1.000 0.473 0.348 0.290
t27 0.408 0.425 0.304 0.291 0.514 0.473 1.000 0.374 0.306
t10 0.280 0.311 0.718 0.681 0.313 0.348 0.374 1.000 0.672
t51 0.241 0.311 0.730 0.661 0.245 0.290 0.306 0.672 1.000
```

Following earlier analysis we fit two general factors and two group factors, where the first group factor has only four non-zero loadings and the second one has five non-zero loadings.

Algorithm B takes 2141 iterations to converge to loss 0.0016132. Since the ML solution in Jöreskog (1969) is reported with two decimals precision we will do the same for the MDFA solution (although we have computed the solution to more than six decimals precision). The MDFA loadings are

```
     [,1]     [,2]     [,3]     [,4]
t42    -0.55    +0.45    -0.15    +0.00
t54    -0.60    +0.43    -0.21    +0.00
t45    -0.74    -0.51    -0.33    +0.00
t46    -0.69    -0.47    -0.08    +0.00
t23    -0.55    +0.35    +0.00    +0.37
t24    -0.60    +0.39    +0.00    +0.15
t27    -0.56    +0.21    +0.00    +0.35
t10    -0.76    -0.32    +0.00    +0.02
t51    -0.74    -0.34    +0.00    -0.12
```

and the uniquenesses are

```
[1] 0.47 0.41 0.09 0.31 0.44 0.46 0.51 0.32 0.32
```

The ML solution, taken from Jöreskog (1969), has uniquenesses

```
[1] 0.48 0.41 0.09 0.31 0.44 0.46 0.52 0.32 0.32
```

and loadings

```
      [,1]    [,2]    [,3]    [,4]
 [1,]   +0.70   -0.12   +0.15   +0.00
 [2,]   +0.74   -0.08   +0.22   +0.00
 [3,]   +0.39   +0.81   +0.33   +0.00
 [4,]   +0.37   +0.75   +0.08   +0.00
 [5,]   +0.65   -0.03   +0.00   +0.37
 [6,]   +0.72   -0.05   +0.00   +0.15
 [7,]   +0.60   +0.09   +0.00   +0.35
 [8,]   +0.51   +0.65   +0.00   +0.02
 [9,]   +0.48   +0.67   +0.00   -0.12
```

We see that the uniquenesses and the loadings on the group factors are almost identical, but the loadings on the two general factors look very different. But the loadings on the general factors are only identified up to a rotation. If we rotate the MDFA solution to a maximum match with the ML solution we find again an almost perfect correspondence. The rotated MDFA solution is

```
    [,1]    [,2]
t42   +0.70   -0.12
t54   +0.73   -0.08
t45   +0.39   +0.80
t46   +0.37   +0.74
t23   +0.65   -0.03
t24   +0.72   -0.05
t27   +0.60   +0.09
t10   +0.51   +0.65
t51   +0.48   +0.67
```

4.5 Cattell

This example uses the correlation matrix between 12 cognitive variables from Cattell (1963), included in the psych package (Revelle (2025)). It illustrates fitting general and group factors with correlated (oblique) factors.

Variables 1 and 2 measures verbal ability, 3 and 4 are spatial ability, 5 and 6 are reasoning, 7 and 8 are numerical ability. The first eight are from the Thurstone Primary Abilities test, and the last four are from the (IPAT) Culture Fair Intelligence Test.

We apply Algorithm B with 6 factors: one general factor and five group factors. The projection $\Pi_{\mathcal{A}}()$ copies all elements from the first column of $X'F$, elements one and two from column two, three and four from column three, five and six from column four, seven and eight from column five, and nine to twelve from column six. All other elements of $\Pi_{\mathcal{A}}(X'F)$ are zero, and $\Pi_{\mathcal{D}}(X'F) = \text{diag}(X'F)$ as usual.

Algorithm B uses 312 iterations to converge to loss 0.067063. The loadings are

```
         [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
Verbal   -0.638 +0.670 +0.000 +0.000 +0.000 +0.000
Verbal2  -0.642 +0.673 +0.000 +0.000 +0.000 +0.000
Space1   -0.464 +0.000 -0.754 +0.000 +0.000 +0.000
Space2   -0.438 +0.000 -0.772 +0.000 +0.000 +0.000
Reason1  -0.652 +0.000 +0.000 +0.589 +0.000 +0.000
Reason2  -0.642 +0.000 +0.000 +0.591 +0.000 +0.000
Number1  -0.543 +0.000 +0.000 +0.000 +0.728 +0.000
Number2  -0.493 +0.000 +0.000 +0.000 +0.704 +0.000
IPATSer  -0.444 +0.000 +0.000 +0.000 +0.000 +0.391
IPATCLAS -0.399 +0.000 +0.000 +0.000 +0.000 +0.417
IPATMatr -0.538 +0.000 +0.000 +0.000 +0.000 +0.530
IPATTOP  -0.389 +0.000 +0.000 +0.000 +0.000 +0.258
```

and the uniquenesses are

```
 [1] 0.142 0.134 0.208 0.202 0.220 0.235 0.173 0.250 0.647 0.658 0.428 0.777
```

For this example we can also compare Algorithm B with Algorithm G. The two algorithms give the same solution, with Algorithm G needing 67 function and 53 gradient evaluations. To compare speeds we use the microbenchmark package (Mersmann (2024)).

```
Unit: milliseconds
                                                     expr       min        lq     mean
 mdfaAlgorithmB(cattell, ctmat, cattellProjection)  9.874768 10.38850 11.78513
            mdfaAlgorithmG(cattell, ctmat, catemp) 10.264227 10.52169 11.18316
   median       uq      max neval
 10.57804 12.04453 69.93407   100
 10.65477 10.93714 14.43446   100
```

For this example there does not seem to be much difference, but there are several caveats. Both algorithms use their default values. For Algorithm B this means a maximum of 1000 iterations and a stop when the loss function changes less than 10^{-10} from one iteration to the next. For Algorithm G the defaults are whatever the defaults of optim() are.

Rindskopf and Rose (1988) also compute a group factor analysis solution, but they allow for correlations between the five group factors. It seems as if allowing for oblique factors means that we replace the constraint $Y'Y = I$ by something more general. But instead we will keep Y orthonormal and emulate oblique factors by suitable (non-linear) constraints on the loadings. Allowing for otherwise arbitrary correlations between the five group factors amounts to imposing the loadings structure

$$A = \begin{bmatrix} a_{11} & a_2 & 0 & 0 & 0 & 0 \\ a_{21} & 0 & a_3 & 0 & 0 & 0 \\ a_{31} & 0 & 0 & a_4 & 0 & 0 \\ a_{41} & 0 & 0 & 0 & a_5 & 0 \\ a_{51} & 0 & 0 & 0 & 0 & a_6 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & r_2' \\ 0 & r_3' \\ 0 & r_4' \\ 0 & r_5' \\ 0 & r_6' \end{bmatrix} = \begin{bmatrix} a_{11} & a_2 r_2' \\ a_{21} & a_3 r_3' \\ a_{31} & a_4 r_4' \\ a_{41} & a_5 r_5' \\ a_{51} & a_6 r_6' \end{bmatrix}$$

The vectors r_j are all of length five, a_2, a_3, a_4, a_5 are of length two, and a_6 is of length four. This shows that $\Pi_{\mathcal{A}}()$ copies the first column of $X'F$ and computes five rank one approximations to the remaining blocks. This is not a linear projection, but one that is computationally still easy if we use the SVD of each block.

Algorithm B with the nonlinear projections converges in 8225 iterations to loss 0.015236. The loadings are

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] -0.398 -0.838 +0.000 +0.000 +0.000 +0.000
[2,] -0.399 -0.838 +0.000 +0.000 +0.000 +0.000
[3,] -0.381 +0.000 -0.788 +0.000 +0.000 +0.000
[4,] -0.347 +0.000 -0.837 +0.000 +0.000 +0.000
[5,] -0.406 +0.000 +0.000 -0.813 +0.000 +0.000
[6,] -0.418 +0.000 +0.000 -0.740 +0.000 +0.000
```

```
 [7,] -0.353 +0.000 +0.000 +0.000 -0.752 +0.000
 [8,] -0.233 +0.000 +0.000 +0.000 -0.928 +0.000
 [9,] -0.526 +0.000 +0.000 +0.000 +0.000 +0.243
[10,] -0.506 +0.000 +0.000 +0.000 +0.000 +0.242
[11,] -0.972 +0.000 +0.000 +0.000 +0.000 -0.235
[12,] -0.413 +0.000 +0.000 +0.000 +0.000 +0.304
```

and the uniquenesses are

```
 [1] 0.138 0.138 0.232 0.177 0.173 0.277 0.308 0.083 0.663 0.684 0.000 0.737
```

The correlations between the factors are

```
     [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] +1.000 +0.000 +0.000 +0.000 +0.000 +0.000
[2,] +0.000 +1.000 +0.224 +0.385 +0.313 -0.285
[3,] +0.000 +0.224 +1.000 +0.133 +0.105 -0.490
[4,] +0.000 +0.385 +0.133 +1.000 +0.428 -0.231
[5,] +0.000 +0.313 +0.105 +0.428 +1.000 -0.126
[6,] +0.000 -0.285 -0.490 -0.231 -0.126 +1.000
```

Our MDFA solution is very different from the ML solution reported by Rindskopf and Rose (1988). They found a Heywood case for variable four, we found a Heywood case for variable eleven. This discrepancy requires further analysis, but this paper is not the place for that.

Note that the correlation matrix recovered by MDFA is always positive semi-definite, because it is of the form RR'. Also, even in these nonlinear cases, MDFA will never find negative uniquenesses, because the MDFA uniqueness are squares.

4.6 BFI

In this example we use the BFI data from the psych package (Revelle (2025)). There are 2800 subjects and 25 personality self report items. We use the impute() function from the e1071 package, with the mean option, to fill in the missing data (Meyer et al. (2024)). In this example we can use mdfaAlgorithmA(), because the complete data matrix is available.

Algorithms A and B both require 61 iterations to arrive at the same solution with loss 0.1830771.

```
Unit: milliseconds
                                          expr       min         lq       mean
 mdfaAlgorithmA(bfi, bftold, verbose = FALSE) 423.74054 428.080200 447.205102
 mdfaAlgorithmB(cfi, bftold, verbose = FALSE)   4.96756   5.213293   5.966972
     median         uq       max neval
 431.836682 476.891008 496.43534   100
   5.277397   5.406567  54.13574   100
```

The median execution time, as measured by the microbenchmark package (Mersmann (2024)), is 435.936128 milliseconds for Algorithm A and 5.39906 milliseconds for Algorithm B, which is consequently about 80 times faster.

5 Discussion

5.1 Statistical Considerations

$$\underline{x}_{ij} = \mathcal{N}(\mu_j + \sum f_{is}a_{js} + u_{ij}d_j, s^2)$$

The consistency and asymptotic normality of the MDFA estimates has recently been studied by Terada (2025).

Now define $C := X'X$ and for any square symmetric T use the notation $\lambda_s(T)$ for the ordered eigenvalues of T. Define

$$\sigma(A, C) := \text{tr}\, C + \text{tr}\, A'A - 2\sum_{s=1}^{m} \sqrt{\lambda_s(A'CA)}$$

Differentiate this with repect to A, assuming the eigenvalues are all different.

$$\mathcal{D}_1\sigma(A, C) = 2\{A - CA(A'CA)^{-\frac{1}{2}}\}$$

Note that $\mathcal{D}_1\sigma(A, C) = 0$ if $C = AA'$.

1. The minimizer $A(C)$ is a continuous function of C.
2. The minimizer $A(C)$ is a differentiable function of C.
3. $\underline{C}_n$ converges weakly to $\Gamma = AA'$.
4. $n^{-\frac{1}{2}}(\underline{C}_n - \Gamma)$ is asymptotically normal.

References

Adachi, K. 2012. "Some Contributions to Data Fitting Factor Analysis with Empirical Comparisons to Covariance-Fitting Factor Analysis." *Journal of the Japanese Society for Computational Statistics* 25: 25–38.

Adachi, K., and N. T. Trendafilov. 2018. "Some Mathematical Properties of the Matrix Decomposition Solution in Factor Analysis." *Psychometrika* 83 (2): 407–24.

Anderson, T. W., and H. Rubin. 1956. "Statistical Inference in Factor Analysis." In *Proeedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, edited by J. Neyman, V:111–50. Berkeley; Los Angeles: University of California Press.

Cattell, R. B. 1963. "Theory of Fluid and Crystallized Intelligence: A Critical Experiment." *Journal of Educational Psychology* 54 (1–22).

De Leeuw, J. 1994. "Block Relaxation Algorithms in Statistics." In *Information Systems and Data Analysis*, edited by H. H. Bock, W. Lenski, and M. M. Richter, 308–24. Berlin: Springer Verlag. https://jansweb.netlify.app/publication/deleeuw-c-94-c/deleeuw-c-94-c.pdf.

———. 2004. "Least Squares Optimal Scaling of Partially Observed Linear Systems." In *Recent Developments in Structural Equation Models*, edited by K. van Montfort, J. Oud, and A. Satorra. Dordrecht, Netherlands: Kluwer Academic Publishers.

———. 2025. "Differentiating Generalized Eigenvalues and Eigenvectors." 2025. https://doi.org/10.48550/arXiv.2508.09355.

Derflinger, G. 1969. "Efficient Methods for Obtaining the Minres and Maximum-Likelihood Solutions in Factor Analysis." *Metrika* 14: 214–31.

Emmett, W. G. 1949. "Factor Analysis by Lawley's Method of Maximum Likelihood." *British Journal of Psychology, Statistical Section* 2: 90–97.

Gallivan, K. A., and P.-A. Absil. 2010. "Note on the Convex Hull of the Stiefel Manifold." https://studyres.com/doc/19971030/note-on-the-convex-hull-of-the-stiefel-manifold.

Garnett, J. C. Maxwell. 1919. "On Certain Independent Factors in Mental Measurements." *Proceedings of the Royal Society of London, Series A* 96: 91–111.

Harman, H. H., and W. H. Jones. 1966. "Factor Analysis by Minimizing Residuals ." *Psychometrika* 31: 351–68.

Joreskog, K. G., and A. S. Goldberger. 1972. "Factor Analysis by Generalized Least Squares." *Psychometrika* 37 (3): 243–60.

Jöreskog, K. G. 1969. "A General Approach to Confirmatory Maximum Likelihood Factor Analysis." *Psychometrika* 34 (2): 183–202.

Jöreskog, K. G., and M. Van Thillo. 1971. "New Rapid Algorithms for Factor Analysis by Unweighted Least Squares, Generalized Least Squares and Maximum Likelihood." Research Memorandum RM-71-5. Educational Testing Service.

Kestelman, H. 1952. "The Fundamental Equation of Factor Analysis." *British Journal of Psychology, Statistical Section* 5: 1–6.

Lange, K. 2016. *MM Optimization Algorithms*. SIAM.

Lawley, D. N. 1941. "Further Investigations in Factor Estimation." *Proceedings of the Royal Society of Edinburgh* A 61: 176–85.

Lawley, D. N., and A. E. Maxwell. 1971. *Factor Analysis as a Statistical Method*. New York, N.Y.: American Elsevier.

MacDonell, W. R. 1902. "On Criminal Anthropometry and the Identification of Criminals." *Biometrika* 1 (2): 177–227.

Maxwell, A. E. 1961. "Recent Trends in Factor Analysis." *Journal of the Royal Statistical Society, Series A* 124: 49–59.

McDonald, R. P., and E .J. Burr. 1967. "A Comparison of Four Methods of Constructing Factor Scores." *Psychometrika* 32 (4): 381–401.

Mersmann, O. 2024. *microbenchmark: Accurate Timing Functions*. https://CRAN.R-project.org/package=microbenchmark.

Meyer, David, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. 2024. *E1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. https://doi.org/10.32614/CRAN.package.e1071.

Mordukhovich, B. S., and N. M. Nam. 2022. *Convex Analysis and Beyond, Volume i: Basic Theory*. Springer.

Pearson, K. 1901. "On Lines and Planes of Closest Fit to Systems of Points in Space." *Philosophical Magazine* 2 (11): 559–72.

Qiu, Y., and J. Mei. 2024. *RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems*. https://CRAN.R-project.org/package=RSpectra.

R Core Team. 2025. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Revelle, W. 2025. *Psych: Procedures for Psychological, Psychometric, and Personality Research*. Evanston, Illinois: Northwestern University. https://CRAN.R-project.org/package=psych.

Rindskopf, D., and T. Rose. 1988. "Some Theory and Applications of Confirmatory Second-Order Factor Analysis." *Multivariate Behavioral Research* 23: 51–67.

Rockafellar, R. T. 1970. *Convex Analysis*. Princeton University Press.

Sočan, G. 2003. "The Incremental Value of Minimum Rank Factor Analysis." PhD thesis, Groningen University. https://research.rug.nl/en/publications/the-incremental-value-of-minimum-rank-factor-analysis.

Stegeman, A. 2016. "A New Method for Simultaneous Estimation of the Factor Model Parameters, Factor Scores, and Unique Parts." *Computational Statistics and Data Analysis* 99: 189–203.

Steiger, J. H., and P. H. Schönemann. 1978. "A History of Factor Indeterminacy." In *Theory Construction and Data Analysis in the Behavioral Sciences*, edited by S. Shye. San Francisco: Jossey-Bass.

Stewart, G. W. 1998. *Matrix Algorithms. Volume i: Basic Decompositions*. SIAM.

Terada, Y. 2025. "Statistical Properties of Matrix Decomposition Factor Analysis." https:

//arxiv.org/abs/2403.06968.

Thomson, G. H. 1934. "Hotelling's Method Modified to Give Spearman's *g*." *Journal of Educational Psychology* 25: 366–74.

Trendafilov, N. T., and S. Unkel. 2011. "Exploratory Factor Analysis of Data Matrices with More Variables Than Observations." *Journal of Computational and Graphical Statistics* 20 (4): 874–91.

Trendafilov, N. T., S. Unkel, and W. Krzanowski. 2013. "Exploratory Factor and Principal Component Analyses: Some New Aspects." *Statistics and Computing* 23: 209–20.

Tucker, L. R. 1958. "An Interbattery Method of Factor Analysis." *Psychometrika* 23: 111–36.

Unkel, S. 2009. "Factor Analysis of Data Matrices: New Theoretical and Computational Aspects with Applications." PhD thesis, The Open University. https://oro.open.ac.uk/62007/1/13837664.pdf.

Unkel, S., and N. T. Trendafilov. 2010. "Simultaneous Parameter Estimation in Exploratory Factor Analysis: An Expository Review." *International Statistical Review* 78 (3): 363–82.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Fourth Edition. New York: Springer.

Watson, G. A. 1992. "Characterization of the Subdifferential of Some Matrix Norms." *Linear Algebra and Its Applications* 170: 33–45.

Whittle, P. 1952. "On Principal Components and Least Square Methods of Factor Analysis." *Scandinavian Actuarial Journal* 3-4: 223–39.

Yamashita, N. 2025. "Identification of Factor Scores by Regression with External Variables in Exploratory Factor Analysis." *Psychometrika* (in press).

Young, G. 1940. "Maximum Likelihood and Factor Analysis." *Psychometrika* 6 (1): 49–53.