Differentiating Generalized Eigenvalues and Eigenvectors

Jan de Leeuw

August 12, 2025

We give formulae for first and second derivatives of generalized eigenvalues/eigenvectors of symmetric matrices and generalized singular values/singular vectors of rectangular matrices when the matrices are linear or nonlinear functions of a vector of parameters. In addition we provide functions in R to compute these derivatives, both in the general case and in various special cases. Formulae are checked against Jacobians and Hessians computed by numerical differentiation. Some applications to multivariate data analysis are discussed.

Table of contents

Note: This is a working manuscript which will be expanded/updated frequently. All suggestions for improvement are welcome. All Rmd, 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://github.com/deleeuw/generalized-eigen-derivatives.

1 Introduction

The generalized eigenvalue decomposition (GEVD) problem for a pair (A, B) of square symmetric matrices of order n is to find the solution (X, Λ) of the system

$$AX = BX\Lambda, \tag{1a}$$
$$X'BX = I. \tag{1b}$$

We assume that B is positive definite. The matrix X of eigenvectors and the diagonal matrix Λ of eigenvalues are both square of order n. As an aside we mention that the generalized eigenvalues of (1a) and (1b) are also the eigenvalues of the asymmetric matrix $B^{-1}A$ and of the symmetric matrix $B^{-1/2}AB^{-1/2}$.

GEVD is at the basis of a great many computations in multivariate statistics. In multinormal small sample theory the derivatives are needed for the Jacobians in change-of-variable calculations to find the distribution of many interesting statistics (mathai). The applications we have in mind, however, are in large-sample statistics, where the derivatives are used in Delta Method computations of bias estimates, standard errors, and confidence intervals.

The equations (1a) and (1b) implicitly define $(X(), \Lambda())$ as a function of (A, B). It turns out that under suitable assumptions on the matrix arguments these implicit functions are actually differentiable, and this makes it interesting to compute their derivatives.

There is a humongous and scattered literature on formulae for and computations of derivatives of eigenvalues and eigenvectors in linear algebra, numerical mathematics, engineering, multivariate statistics, and even in physics. Reviewing and collecting all relevant literature is an impossible task. We only give the basic mathematical references, which provide the foundations upon which our results will be built (Kato (1984), Baumgärter (1985), Brustad (2019)).

We have written code in R (R Core Team (2025)) to implement our final formulae for the first and second derivatives. Code is available at https://github.com/deleeuw/generalized-eigen-derivatives. In addition we have also written R code, available in the same repository, to numerically verify the final formulae. This uses the functions jacobian() and hessian() from the excellent R package numDeriv (Gilbert and Varadhan (2019)). Of course numerical verification, especially in the nonlinear case, is limited to a small number of examples, but if we get the same results as from our formulae we do gain some confidence.

I should perhaps also mention a previous version of this paper (De Leeuw (2007)), which has some errors and is not as complete as the current version.

2 Basic Results

Suppose $A()$ and $B()$ are differentiable symmetric matrix valued functions of order n on an open subset Θ of $\mathbb{R}^p$, and suppose at $\theta_0 \in \Theta$ the matrix $B(\theta_0)$ is positive definite and the generalized eigenvalues of $(A(\theta_0), B(\theta_0))$ are all different. Then the ordered eigenvalues $\Lambda()$ and the eigenvectors $X()$ are analytic functions of the parameters in a neighborhood of θ_0.

We use subscripts $i, j = 1, \cdots, n$ for the rows and columns of A and B, subscripts $\eta, \nu = 1, \cdots, n$ for the eigenvalues and eigenvectors, and $s, t = 1, \cdots, p$ for the parameters in θ. Thus $x_{i\nu}$ is element i of eigenvector x_ν. The partial derivative of a function $A()$ on Θ with respect to θ_s, evaluated at θ_0, is written as $\mathcal{D}_s A(\theta_0)$. In order not to clutter our formulae the parameter vector where the derivative is evaluated is usually not explicitly specified. Note also that it is sometimes necessary to use parentheses to distinguish $(\mathcal{D}_s A)x$ from $\mathcal{D}_s(Ax)$.

2.1 First Partials

Differentiate (1a) with respect to θ_s. Then

$$(\mathcal{D}_s A)X + A(\mathcal{D}_s X) = BX(\mathcal{D}_s\Lambda) + B(\mathcal{D}_s X)\Lambda + (\mathcal{D}_s B)X\Lambda. \tag{2}$$

Premultiplying (2) by X' and rearranging gives

$$\mathcal{D}_s\Lambda = \{X'(\mathcal{D}_s A)X - X'(\mathcal{D}_s B)X\Lambda\} + \{\Lambda X'B(\mathcal{D}_s X) - X'B(\mathcal{D}_s X)\Lambda\}. \tag{3}$$

The matrix $\Lambda X'B(\mathcal{D}_s X) - X'B(\mathcal{D}_s X)\Lambda$ is anti-symmetric and consequently has a zero diagonal. Taking the diagonal[1] on both sides of (3) gives

$$\mathcal{D}_s\Lambda = \text{diag}\{X'(\mathcal{D}_s A)X - X'(\mathcal{D}_s B)X\Lambda\}, \tag{4}$$

or, for a single eigenvalue λ_ν with corresponding eigenvector x_ν,

$$\mathcal{D}_s\lambda_\nu = x_\nu'(\mathcal{D}_s A - \lambda_\nu\mathcal{D}_s B)x_\nu. \tag{5}$$

Taking the off-diagonal[2] on both sides of (3) gives

$$\text{off}\{\Lambda X'B(\mathcal{D}_s X) - X'B(\mathcal{D}_s X)\Lambda\} = -\text{off}\{X'(\mathcal{D}_s A)X - X'(\mathcal{D}_s B)X\Lambda\}. \tag{6}$$

[1]The diagonal $\text{diag}(X)$ of a square matrix X is a diagonal matrix with the same diagonal as X.
[2]The off-diagonal $\text{off}(X)$ of a square matrix X is X with its diagonal replaced by zeroes.

X is non-singular, and thus there is a unique square H_s such that $\mathcal{D}_s X = X H_s$. Using this substitution (6) becomes

$$\text{off}\{\Lambda H_s - H_s \Lambda\} = -\text{off}\{X'(\mathcal{D}_s A)X - X'(\mathcal{D}_s B)X\Lambda\}. \tag{7}$$

Switch to subscript notation and solve for H_s. For $s \neq t$

$$\{H_s\}_{\nu\eta} = -\frac{x_\nu'\{\mathcal{D}_s A - \lambda_s \mathcal{D}_s B\}x_\eta}{\lambda_\nu - \lambda_\eta}. \tag{8}$$

This does not give a value for the diagonal of H_s. Differentiating (1b) gives

$$X'B(\mathcal{D}_s X) + (\mathcal{D}_s X)'BX + X'\mathcal{D}_s BX = 0. \tag{9}$$

Using $\mathcal{D}_s X = X H_s$ and taking the diagonal of (9) gives

$$h_{\nu\nu} = -\tfrac{1}{2} x_\nu'(\mathcal{D}_s B)x_\nu. \tag{10}$$

Combining (8) and (10) shows that for the eigenvector corresponding with λ_ν we have

$$\mathcal{D}_s x_\nu = -\sum_{\substack{\eta=1 \\ \eta\neq\nu}}^{n} \frac{x_\eta'\{\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B\}x_\nu}{\lambda_\eta - \lambda_\nu} x_\eta - \tfrac{1}{2} x_\nu'(\mathcal{D}_s B)x_\nu \cdot x_\nu, \tag{11}$$

which can also be written as

$$\mathcal{D}_s x_\nu = -\sum_{\substack{\eta=1 \\ \eta\neq\nu}}^{n} \left\{ \frac{x_\eta x_\eta'}{\lambda_\eta - \lambda_\nu} \right\} (\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)x_\nu - \tfrac{1}{2}(x_\nu'(\mathcal{D}_s B)x_\nu)x_\nu. \tag{12}$$

It is notationally convenient to have a matrix expression for this derivative. Define the matrices

$$(A - \lambda_\nu B)^- := \sum_{\substack{\eta=1 \\ \eta\neq\nu}}^{n} \left\{ \frac{x_\eta x_\eta'}{\lambda_\eta - \lambda_\nu} \right\} = X(\Lambda - \lambda_\nu I)^+ X', \tag{13}$$

where $(\Lambda - \lambda_\nu I)^+$ is the Moore-Penrose inverse of $\Lambda - \lambda_\nu I$. Matrix (13) is a reflexive g-inverse of $A - \lambda_\nu B$ (Rao and Mitra (1971), section 2.5). Of the four Penrose conditions only the first two are satisfied. To verify this we use definition (13) and $A - \lambda_\nu B = X^{-T}(\Lambda - \lambda_\nu I)X^{-1}$. We find

$$(A - \lambda_\nu B)(A - \lambda_\nu B)^-(A - \lambda_\nu B) = A - \lambda_\nu B, \tag{14a}$$

$$(A - \lambda_\nu B)^-(A - \lambda_\nu B)(A - \lambda_\nu B)^- = (A - \lambda_\nu B)^-, \tag{14b}$$

$$(A - \lambda_\nu B)(A - \lambda_\nu B)^- = X^{-T}(I - e_\nu e_\nu')X', \tag{14c}$$

$$(A - \lambda_\nu B)^-(A - \lambda_\nu B) = X(I - e_\nu e_\nu')X^{-1}. \tag{14d}$$

We see that $(A-\lambda_\nu B)^-$ is a Moore-Penrose inverse of $A-\lambda_\nu B$ if and only if B is identically equal to one (and thus $X'X=XX'=I$).

Using (13) gives

$$\mathcal{D}_s x_\nu = -(A-\lambda_\nu B)^-(\mathcal{D}_s A-\lambda_\nu \mathcal{D}_s B)x_\nu - \tfrac{1}{2}(x_\nu'(\mathcal{D}_s B)x_\nu)x_\nu. \tag{15}$$

The equations (5) and (15) will be used frequently throughout this paper.

For eigenvalues (5) shows that computing $\mathcal{D}_s\lambda_\nu$ only requires us to know λ_ν and x_ν, not the other eigenvalues and eigenvectors. Equation (15) suggests the same thing is true for $\mathcal{D}_s x_\nu$, but this is only apparent, because we need all generalized eigenvalues and eigenvectors to compute $(A-\lambda_\nu B)^-$. If we only need the derivatives of, say, the first few eigenvectors from a very large system this can be computationally quite expensive.

There is an alternative way of computing $\mathcal{D}_s x_\nu$ which does not require a full GEVD. We know that $\mathcal{D}_s x_\nu$ is a solution of the linear equations

$$(A-\lambda_\nu B)\mathcal{D}_s x_\nu = -(\mathcal{D}_s A-\lambda_\nu \mathcal{D}_s B)x_\nu + (\mathcal{D}_s\lambda_\nu)Bx_v. \tag{16}$$

Because λ_ν is a simple eigenvalue, the matrix $A-\lambda_\nu B$ is of rank $n-1$, with a null space consisting of all vectors proportional to x_ν. Write y_ν for the right hand side of (16). The system is solvable because $x_\nu' y_\nu = 0$ by (5), and its general solution is

$$\mathcal{D}_s x_\nu = (A-\lambda_\nu B)^- y_\nu + \theta x_\nu, \tag{17}$$

with $(A-\lambda_\nu B)^-$ any generalized inverse of $A-\lambda_\nu B$ and with θ arbitrary.

Since we must have

$$x_\nu' B(\mathcal{D}_s x_\nu) = -\tfrac{1}{2}x_\nu'(\mathcal{D}_s B)x_\nu \tag{18}$$

from (9), we see that

$$\theta = -\tfrac{1}{2}x_\nu'(\mathcal{D}_s B)x_\nu - x_\nu' B(A-\lambda_\nu B)^- y_\nu. \tag{19}$$

The value of θ depends on the choice of the generalized inverse. For our previous choice of the reflexive inverse from (13) we actually have $x_\nu' B(A-\lambda_\nu B)^- = x_\nu' X^{-T}X^{-1}X(\Lambda-\lambda_\nu I)^+X'y_\nu = e_\nu'(\Lambda-\lambda_\nu I)^+X'y = 0$, and thus $\theta = -\frac{1}{2}x_\nu'(\mathcal{D}_s B)x_\nu$, but for other generalized inverses this may not be true. One possible other choice is the Moore-Penrose inverse, which we can compute with

$$(A-\lambda_\nu B)^+ = (A-\lambda_\nu B+\tilde{x}_\nu\tilde{x}_\nu')^{-1} - \tilde{x}_\nu\tilde{x}_\nu', \tag{20}$$

with $\tilde{x}_\nu$ defined as x_ν normalized to length one. Using (20) means that to compute $\mathcal{D}_s x_\nu$ we do not need the complete GEVD of (A,B) but instead the inverse of the symmetric matrix $A-\lambda_\nu B+\tilde{x}_\nu\tilde{x}_\nu'$. Yet another generalized inverse uses one of the non-singular

principal submatrices $A - \lambda_\nu B$ of order $n-1$. If $x_{i\nu} \neq 0$ then the submatrix leaving out row and column i is non-singular, so it makes sense to choose i with the largest $|x_{i\nu}|$. The computation now requires us to invert a matrix of order $n-1$. Using this last choice of generalized inverse is known in the engineering literature as Nelson's method for computing derivatives of eigenvectors, after Nelson (1976).

2.2 Second Partials

To find second partials of the eigenvalues we differentiate (5) with respect to θ_t. This gives

$$\mathcal{D}_{st}\lambda_\nu = 2x_\nu'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)\mathcal{D}_t x_\nu + x_\nu'(D_{st}A - \lambda_\nu \mathcal{D}_{st}B)x_\nu - x_\nu'(\mathcal{D}_s B)x_\nu \cdot \mathcal{D}_t\lambda_\nu. \quad (21)$$

Substituting from (5) and (15) gives, using the reflexive inverse from (13),

$$\begin{aligned}\mathcal{D}_{st}\lambda_\nu = &-2x_\nu'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)(A-\lambda_\nu B)^-(\mathcal{D}_t A - \lambda_\nu \mathcal{D}_t B)x_\nu + x_\nu'(D_{st}A - \lambda_\nu \mathcal{D}_{st}B)x_\nu \\ &- x_\nu'(\mathcal{D}_t B)x_\nu \cdot x_\nu'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)x_\nu - x_\nu'(\mathcal{D}_\jmath B)x_\nu \cdot x_\nu'(\mathcal{D}_t A - \lambda_\nu \mathcal{D}_t B)x_\nu. \quad (22)\end{aligned}$$

Formula (22) shows that $\mathcal{D}_{st}\lambda_\nu = \mathcal{D}_{ts}\lambda_\nu$, just as it should be.

The second partials of the generalized eigenvalues are, not surprisingly, more complicated. We start with (11) and differentiate with respect to θ_t. We give some intermediate calculations for this case, because they are also used in our software. First

$$\begin{aligned}\mathcal{D}_{st}x_\nu = &-\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n} \mathcal{D}_t\left\{\frac{x_\jmath'\{\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B\}x_\nu}{\lambda_\eta - \lambda_\nu}\right\} x_\eta \\ &- \sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n} \frac{x_\eta'\{\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B\}x_\nu}{\lambda_\eta - \lambda_\nu}\mathcal{D}_t x_\eta - \tfrac{1}{2}\mathcal{D}_t\{x_\nu'(\mathcal{D}_s B)x_\nu \cdot x_\nu\}. \quad (23)\end{aligned}$$

The terms in the first summation in (23) are

$$\begin{aligned}\mathcal{D}_t\left\{\frac{x_\eta'\{\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B\}x_\nu}{\lambda_\eta - \lambda_\nu}\right\} = &\\ \frac{\mathcal{D}_t\{x_\eta'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)x_\nu\}}{\lambda_\nu - \lambda_\eta} &- \frac{(\mathcal{D}_t\lambda_\eta - \mathcal{D}_t\lambda_\nu)x_\eta'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)x_\nu}{(\lambda_\eta - \lambda_\nu)^2}, \quad (24)\end{aligned}$$

and the numerator in the first term on the right-hand side of (24) is

$$\begin{aligned}\mathcal{D}_t\{x_\eta'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)x_\nu\} = &(\mathcal{D}_t x_\eta)'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)x_\nu + x_\eta'(\mathcal{D}_s A - \lambda_\nu \mathcal{D}_s B)\mathcal{D}_t x_\nu \\ &+ x_\eta'(\mathcal{D}_{st}A - \lambda_\nu \mathcal{D}_{st}B)x_\nu - \mathcal{D}_t\lambda_\nu \cdot x_\eta'(\mathcal{D}_s B)x_\nu. \quad (25)\end{aligned}$$

Finally

$$\mathcal{D}_t\{x_\nu'(\mathcal{D}_sB)x_\nu\cdot x_\nu\} = \\ 2(\mathcal{D}_tx_\nu)'(\mathcal{D}_sB)x_\nu\cdot x_\nu + x_\nu'(\mathcal{D}_{st}B)x_\nu\cdot x_\nu + x_\nu'(\mathcal{D}_tB)x_\nu\cdot\mathcal{D}_tx_\nu. \quad (26)$$

In computing the final result we first evaluate (25), substitute the result in (24), and then substitute that result, together with the result of (26), into (23). For this to work we need to have both $\mathcal{D}\Lambda$ and $\mathcal{D}X$ available. We could of course substitute the expressions of $\mathcal{D}\Lambda$ and $\mathcal{D}X$ into (23)-(26) but that would result in very long and opaque formulae. We will rely on our software to give us the numerical values for any specific value of θ.

2.3 Multiple Eigenvalues

We would be remiss if we did not say anything about matrices with multiple eigenvalues, where differentiability fails. The main problem is that there is no unique eigenvector corresponding with a multiple eigenvalue. If X are is an $n\times r$ matrix of eigenvectors corresponding with eigenvalue λ, i.e. if $AX=\lambda BX$, then XS are also eigenvectors corresponding with λ for any $r\times p$ matrix S.

A related problem has to do with the order of the eigenvalues. Let's illustrate both problems with a embarrassingly simple example. Suppose $B(\theta)=I$ and

$$A(\theta)=\begin{bmatrix}4&0&0&0\\0&4&0&0\\0&0&2&0\\0&0&0&2\end{bmatrix}+\theta\begin{bmatrix}4&0&0&0\\0&3&0&0\\0&0&2&0\\0&0&0&1\end{bmatrix}.$$

The eigenvectors are the four unit vectors e_1-e_4[3] The corresponding eigenvalues are $\lambda_1=4+4\theta$, $\lambda_2=4+3\theta$, $\lambda_3=2+2\theta$, and $\lambda_4=2+\theta$. In the figure below they are drawn, respectively, in RED, BLUE, GREEN, and PURPLE.

[3]A unit vector e_i has zeroes everywhere, except for element i, which is one.

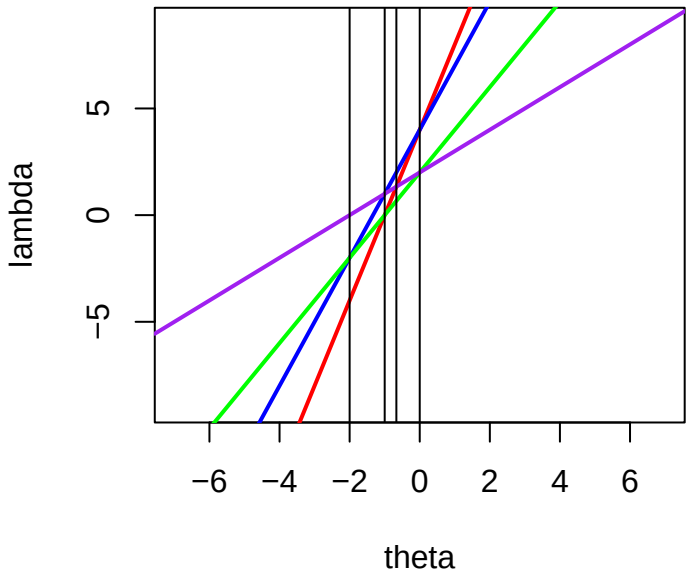

Figure 1: Eigenvalues

With this ordering the eigenvalues are clearly differentiable, even analytic. The corresponding eigenvectors are constant unit vectors, which also makes them differentiable. If $A()$ and $B()$ are analytic functions of a single parameter θ then there are n analytic functions $\lambda_\nu()$ such that $\lambda_\nu(\theta)$ is a generalized eigenvalue of the pair $A(\theta), B(\theta)$ for each θ. Moreover there are corresponding analytic B-orthogonal eigenvectors $x_\nu()$ (Sun (1990)).

But if we look at the ordered eigenvalues $\lambda_{(1)} \geq ... \geq \lambda_{(4)}$ the figure tells us something different. The largest eigenvalue $\lambda_{(1)}$ is RED for $\theta > 0$, BLUE for $-1 < \theta < 0$, and PURPLE for $\theta < -1$. It is convex, but not differentiable at 0 and -1, although it has left and right derivatives (i.e. directional derivatives) at all θ. The smallest eigenvalue $\lambda_{(4)}$ is concave, PURPLE for $\theta > 0$, GREEN for $-1 < \theta < 0$, and RED for $\theta < -1$. The intermediate eigenvalues $\lambda_{(2)}$ and $\lambda_{(3)}$ change color, and thus slope, even more frequently. At zero none of the four ordered eigenvalues is differentiable. The situation with eigenvectors is even worse. At the points where the ordered eigenvalues change color the eigenvector $\lambda_{(i)}$ changes discontinuously from one unit vector to another.

One way to deal with this problem is to give up the idea of looking at individual eigenvalues and eigenvectors. The averages of blocks of eigenvalues are differentiable, even if all eigenvalues in the block are the same (and the ones outside the block are different) (Chu (1990)). In our example we compute the averages of the first two and the last two eigenvalues and we see they are differentiable at zero. The figure below shows the two averages. If $\theta = -1$ the average of the first two largest eigenvalues is equal to the average of the last two, and there are problems again with differentiability.

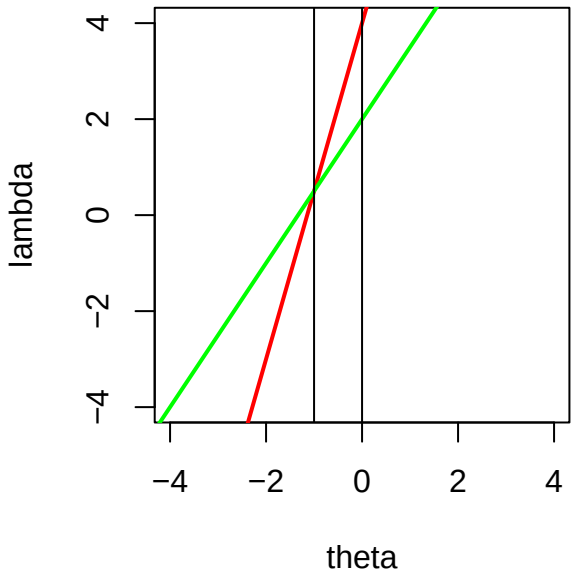

Figure 2: Eigenvalue Averages

Unfortunately the result on the analyticity of suitable ordered eigenvalues and eigenvectors is no longer true if there is more than one parameter. Consider

$$A(\theta) = \begin{bmatrix} 1+\theta_1 & \theta_2 \\ \theta_2 & 1-\theta_1 \end{bmatrix}$$

and $B(\theta) = I$. The generalized eigenvalues are $1 \pm \sqrt{\theta_1^2 + \theta_2^2}$. The functions $\lambda_1(\theta) := 1 + \|\theta\|_2$ and $\lambda_2(\theta) := 1 - \|\theta\|_2$ are two parabolic shells, the first one convex, the second one concave. The first one is larger than the second one for all θ, except at the origin where the two shells touch. Thus there is no problem with the "crossing" of the two functions. But neither of the two functions is differentiable at the origin, although by convexity/concavity they both are directionally differentiable there.

This points to another solution of part of the problem with multiple eigenvalues. We can use more general derivatives, such as one-sided directional derivatives, subdifferentials, epi-derivatives, or lexicographic derivatives. This is the approach taken in convex and nonsmooth analysis (Hiriart-Urruty and Ye (1995), Lewis and Overton (1996)). It has been applied mainly to eigenvalues in EVD problems with $B = I$, and not to the corresponding eigenvectors. Eigenvalues have a variational characterization and eigenvectors do not. The basic result from this approach is that eigenvalues, either simple or multiple, are directionally differentiable functions of the matrix. There are different definitions around of second-order differentiability, but for the most common ones eigenvalues are twice directionally differentiable (Torki (1999), Torki (2001)).

3 Software

In the github repository there are R programs evaluating $A, B, \mathcal{D}\Lambda, \mathcal{D}X, \mathcal{D}^2\Lambda$, and $\mathcal{D}^2X$ at a given θ. The function gevdNonlinear() can be used for any non-linear GEVD. Its only argument is the vector θ. It computes the first partials and then calls the subroutines gevdHessianValues() and gevdHessianVectors() to compute the second partials. The gevdHessianVectors() routine uses the stepwise method suggested at the end of the previous section. First compute all first derivatives, and then use their numerical values in equations (23)-(26).

In order to attain the necessary level of generality gevNonlinear() needs to be run from or after a template which defines global variables, specifically the R functions theA(), theB(), dA(), dB(), ddA(), and ddB(), together with the parameters and additional values that these auxiliary functions need. The template also has values for

- p, the number of parameters;
- n, the order of the matrices;
- hessianl, do we compute second derivatives of eigenvalues;
- hessianx, do we compute second derivatives of eigenvectors.

If only first order information is needed we can set hessianl and hessianx to FALSE.

The result of gevdNonlinear() is a list with

- $A(\theta)$, an $n \times n$ matrix, typical element $a_{ij}(\theta)$,
- $B(\theta)$, an $n \times n$ matrix, typical element $b_{ij}(\theta)$,
- $\lambda(\theta)$, an n-element vector, typical element $\lambda_\nu(\theta)$,
- $X(\theta)$, an $n \times n$ matrix, typical element $x_{i\nu}(\theta)$,
- $\mathcal{D}\Lambda(\theta)$, an $n \times p$ matrix, element (ν, s) is $\mathcal{D}_s\lambda_\nu(\theta)$,
- $\mathcal{D}X(\theta)$, an $n \times p \times n$ array, element (i, s, ν) is $\mathcal{D}_s x_{i\nu}(\theta)$,
- $\mathcal{D}^2\Lambda(\theta)$, a $p \times p \times n$ array,element (s, t, ν) is $\mathcal{D}_{st}\lambda_\nu(\theta)$,
- $\mathcal{D}^2X(\theta)$, a $p \times p \times n \times n$ array, element (s, t, i, ν) is $\mathcal{D}_{st}x_{i\nu}(\theta)$.

The names in the code and in the results for the eight matrices in this list are a, b, l, x, dl, dx, ddl, and ddx. The R print function prints $\mathcal{D}\Lambda$ as an $n \times p$ matrix and $\mathcal{D}X$ as a sequence of n matrices of dimensions $n \times p$, one for each eigenvalue. $\mathcal{D}^2\Lambda$ is printed as n symmetric matrices of order p, and $\mathcal{D}^2X$ as n^2 symmetric matrices of order p, one for each element of X. Printing $\mathcal{D}^2X$ produces a lot of output if n and p are at all large. If hessianx and hessianl are set to FALSE then ddl and ddx are set to NULL.

There is a corresponding numDeriv based check function gevdNonlinearNum(), which has the same single argument θ and gives the same output as gevdNonlinear(). The same procedure is followed for evdNonlinear() and gsvdNonlinear(), which are the functions

for the (non-generalized) eigenvalue decomposition and for the generalized singular value decomposition we shall discuss in later sections.

4 Template Examples

The function gevNonlinear() can handle any parametric model, as long as the appropriate template is provided with routines for $A, B, \mathcal{D}A, \mathcal{D}B, \mathcal{D}^2A$, and $\mathcal{D}^2B$. In special cases more compact and elegant formulae and more efficient computations will exist. The question then becomes if it is worthwhile to derive and retype the formulae and to reprogram the R routines for these special cases. In this section we discuss some important special cases and indicate what should be included in the templates.

We do not give the numerical results of the various derivative calculations, which for these examples are just a large amount of meaningless numbers. The software in the repository, together with the templates, can be used to calculate Jacobians and Hessians for various values of the parameter vector. I should mention that in all examples the results for the derivatives based on the formulas are the same as the results based on numerical differentiation.

4.1 Power Series

In our first example we define

$$A(\theta) = A_0 + \theta A_1 + \frac{1}{2}\theta^2 A_2 + ... + \frac{1}{p}\theta^p A_p \tag{27}$$

$$B(\theta) = B_0 + \theta B_1 + \frac{1}{2}\theta^2 B_2 + ... + \frac{1}{p}\theta^p B_p \tag{28}$$

By a suitable choice of the A_s and B_s we can approximate functions for the cells of $A()$ and $B()$ with convergent power series expansions (possibly different functions for different cells).

The template for a case with $n = 4$ and $p = 3$ is gevdTemplatePowerSeries.R in the github repository. U=It uses random symmetric matrices for A_s and B_s.

4.2 Linear Combinations

Suppose $A()$ is a linear combination of p known symmetric matrices A_s and $B()$ is a linear combination of p known symmetric matrices B_s. There may also be "intercepts"

A_0 and B_0. Thus

$$A(\theta) = A_0 + \sum_{s=1}^{p} \theta_s A_s, \tag{29a}$$

$$B(\theta) = B_0 + \sum_{s=1}^{p} \theta_s B_s. \tag{29b}$$

It may seem somewhat limiting that the same parameter vector θ is used for both $A()$ and $B()$. But we can decide to make the last q matrices A_s and the first $p - q$ matrices B_s equal to zero. This means, in effect, that $A()$ and $B()$ depend on different parameter vectors.

The template for a special case with $n = 4$ and $p = 6$ is gevdTemplateLinear.R in the github repository. In this example the last three A_s and the first three B_s are indeed zero. The basic simplication is of course that $\mathcal{D}_s A = A_s$ and $\mathcal{D}_s B = B_s$, and that consequently the second partials $\mathcal{D}_{st} A$ and $\mathcal{D}_{st} B$ are all zero.

4.3 The Elements

As a special case of thus linear combinations special case we have the partial derivatives with respect to the elements of A and B. Differentiation with respect to the matrix elements can be useful in combination with the chain rule, which computes, for example, the derivative of the eigenvalue with respect to a parameter as the derivative of the eigenvalue with respect to the matrix evaluated at the derivative of the matrix with respect to the parameter.

The linear combinations we use in this case are

$$A = \sum_{1 \leq i \leq j} \sum a_{ij} E_{ij}, \tag{30a}$$

$$B = \sum_{1 \leq i \leq j} \sum b_{ij} E_{ij}. \tag{30b}$$

Here $E_{ij} := (e_i e_j' + e_j e_i')$ for $i \neq j$ and $E_{ii} = e_i e_i'$ with e_i and e_j unit vectors. Thus

$$\mathcal{D}_{(i,j)} A = \mathcal{D}_{(i,j)} B = E_{ij}. \tag{31}$$

In gevdTemplateLinear.R from the previous secton the matrices A_s and B_s were arbitrary and were explicitly given as input. In gevdTemplateElement.R there are auxiliary functions that generate the matrices E_{ij} when needed. This requires some manipulationof indices to go from the vector of parameters θ_s to the matrices of parameters a_{ij} or b_{ij}. The included numerical example is of order four, and consequently has 20 parameters.

4.4 Eigenvalue Decomposition

An Eigenvalue Decomposition (EVD) problem, without the "generalized", is of the form

$$AX = X\Lambda, \tag{32a}$$
$$X'X = I. \tag{32b}$$

This is the special case of GEVD with $B = I$ and $\mathcal{D}B = 0$. In EVD the matrices $(A - \lambda_\nu B)^-$ from (13) are actually the Moore-Penrose inverses of $A - \lambda_\nu I$. There are substantial, and rather obvious, simplifications of the first and second derivatives.

From (5)

$$\mathcal{D}_s\lambda_\nu = x_\nu'(\mathcal{D}_s A)x_\nu, \tag{33}$$

and from (15)

$$\mathcal{D}_s x_\nu = -(A - \lambda_\nu I)^+(\mathcal{D}_s A)x_\nu. \tag{34}$$

From (22)

$$\mathcal{D}_{st}\lambda_\nu = -2x_\nu'(\mathcal{D}_s A)(A - \lambda_\nu I)^+(\mathcal{D}_t A)x_\nu + x_\nu'(D_{st}A)x_\nu. \tag{35}$$

An alternative, which may be more efficient computationally, is

$$\mathcal{D}_{st}\lambda_\nu = -2\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n}\frac{1}{\lambda_\eta - \lambda_\nu}x_\nu'(\mathcal{D}_s A)x_\eta \cdot x_\nu'(\mathcal{D}_t A)x_\eta + x_\nu'(\mathcal{D}_{st}A)x_\nu.$$

This is also the formula given in Overton and Womersley (1995).

For the second derivatives of the eigenvectors we again use the stepwise computation of (23). We do not give a single formula, but a recipe instead.

$$\mathcal{D}_{st}x_\nu = -\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n}\mathcal{D}_t\left\{\frac{x_\eta'\{\mathcal{D}_s A\}x_\nu}{\lambda_\eta - \lambda_\nu}\right\}x_\eta - \sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n}\frac{x_\eta'\{\mathcal{D}_s A\}x_\nu}{\lambda_\eta - \lambda_\nu}\mathcal{D}_t x_\eta. \tag{36}$$

The terms in the first summation in (36) are

$$\mathcal{D}_t\left\{\frac{x_\eta'\{\mathcal{D}_s A\}x_\nu}{\lambda_\eta - \lambda_\nu}\right\} = \frac{\mathcal{D}_t\{x_\eta'(\mathcal{D}_s A)x_\nu\}}{\lambda_\nu - \lambda_\eta} - \frac{(\mathcal{D}_t\lambda_\eta - \mathcal{D}_t\lambda_\nu)x_\eta'(\mathcal{D}_s A)x_\nu}{(\lambda_\eta - \lambda_\nu)^2}, \tag{37}$$

and the numerator in the first term of (37) is

$$\mathcal{D}_t\{x_\eta'(\mathcal{D}_s A)x_\nu\} = (\mathcal{D}_t x_\eta)'(\mathcal{D}_s A)x_\nu + x_\eta'(\mathcal{D}_s A)\mathcal{D}_t x_\nu + x_\eta'(\mathcal{D}_{st}A)x_\nu. \tag{38}$$

As before, we substitute (38) into (37) and then substitute the result in (36).

It is easy and not very wasteful to plug in $B = I$ and $\mathcal{D}_s B = \mathcal{D}_{st}B = 0$ in the formulas and the template of the generalized nonlinear case. Nevertheless, because of the importance of EVD, we do include both evdTemplate.R and evdNonlinear.R in the repository. The template has a one-parameter example $A(\theta) = A_0 + \theta A_1$ of order four.

4.5 Generalized Singular Value Decomposition

Suppose F is a rectangular $n \times m$ matrix, and G and H are two positive definite matrices of orders n and m. Without loss of generality we assume $m \leq n$. The Generalized Singular Value Decomposition (GSVD) is finding a solution to the system

$$FY = GX\Lambda, \tag{39a}$$
$$F'X = HY\Lambda, \tag{39b}$$
$$X'GX = I, \tag{39c}$$
$$Y'HY = I, \tag{39d}$$

Here Y and Λ are square of order m, with Λ diagonal and non-negative. The left singular vectors are $n \times m$, and there is an $n \times (n-m)$ matrix $X_\perp$ that satisfies $F'X_\perp = 0$, with $X_\perp'GX = 0$ and $X_\perp'GX_\perp = I$.

The GSVD system (39a)-(39d) is (very) closely related to the GEVD system

$$\begin{bmatrix} 0 & F \\ F' & 0 \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} G & 0 \\ 0 & H \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} \Psi, \tag{40}$$

with normalization $U'GU + V'HV = I$.

System (40) has $n + m$ solutions that can be described using the GSVD solutions (X, Y, Λ). There are m solutions with $U = \frac{1}{2}\sqrt{2}\ X, V = \frac{1}{2}\sqrt{2}\ Y$ and $\Psi = \Lambda$. There are another m solutions with $U = \frac{1}{2}\sqrt{2}\ X, V = -\frac{1}{2}\sqrt{2}\ Y$ and $\Psi = -\Lambda$. And finally there are $n - m$ solutions with $U = X_\perp, V = 0$, and $\Lambda = 0$. For the non-zero eigenvalues we have $u_\nu'Gu_\nu = v_\nu'Hv_\nu = \frac{1}{2}$, which shows that the GSVD solutions are normalized slightly different from the GEVD ones. The m solutions with non-negative eigenvalues are the interesting ones, and they provide us with the m solutions of the GSVD.

Thus, in stead of tackling the GSVD from (39a)-(39d) directly, we apply our results for the derivatives of the GEVD to the system (40). All three matrices F, G, and H are assumed to be functions of the parameters θ and we will only consider the first m generalized eigenvalues (assuming that F has rank $r = m$, otherwise only the first r).

To bring the notation in line with our previous results we rewrite the equations in the GEVD system (40) as

$$A \begin{bmatrix} x_\nu \\ y_\nu \end{bmatrix} = \lambda_\nu B \begin{bmatrix} x_\nu \\ y_\nu \end{bmatrix}, \tag{41}$$

with

$$A = \begin{bmatrix} 0 & F \\ F' & 0 \end{bmatrix} \tag{42}$$

and

$$B = \begin{bmatrix} G & 0 \\ 0 & H \end{bmatrix} \tag{43}$$

The first derivatives are

$$\mathcal{D}_s\lambda_\nu = 2\ x'_\nu(\mathcal{D}_sF)y_\nu - \lambda_\nu(x'_\nu(\mathcal{D}_sG)x_\nu + y'_\nu(\mathcal{D}_sH)y_\nu), \tag{44}$$

and

$$\mathcal{D}_s \begin{bmatrix} x_\nu \\ y_\nu \end{bmatrix} = -(A - \lambda_\nu B)^- \begin{bmatrix} (\mathcal{D}_sF)y_\nu - \lambda_\nu(\mathcal{D}_sG)x_\nu \\ (\mathcal{D}_sF')x_\nu - \lambda_\nu(\mathcal{D}_sH)y_\nu \end{bmatrix} - \tfrac{1}{2}(x'_\nu(\mathcal{D}_sG)x_\nu + y'_\nu(\mathcal{D}_sH)y_\nu) \cdot \begin{bmatrix} x_\nu \\ y_\nu \end{bmatrix}. \tag{45}$$

Note that

$$(A - \lambda_\nu B)^- = \sum_{\substack{\eta=1 \\ \eta\neq\nu}}^{n+m} \frac{1}{\lambda_\eta - \lambda_\nu} \begin{bmatrix} x_\eta x'_\eta & x_\eta y'_\eta \\ y_\eta x'_\eta & y_\eta y'_\eta \end{bmatrix} \tag{46}$$

with the summation over all $n+m$ eigenvalues, including m negative ones and the $n-m$ that are equal to zero.

It is clearly helpful to define

$$T_s := X'(\mathcal{D}_sF)Y, \tag{47a}$$
$$U_s := X'(\mathcal{D}_sG)X, \tag{47b}$$
$$V_s := Y'(\mathcal{D}_sH)Y. \tag{47c}$$

With definitions (47a)-(47c) the eigenvalue derivatives (44) become

$$\mathcal{D}_s\lambda_\nu = 2\ t^s_{\nu\nu} - \lambda_\nu(u^s_{\nu\nu} + v^s_{\nu\nu}). \tag{48}$$

Substitute (46) into (45), and use the definitions (47a)-(47c) for the eigenvector derivatives.

$$\mathcal{D}_sx_\nu = -\sum_{\substack{\eta=1 \\ \eta\neq\nu}}^{n+m} \frac{1}{\lambda_\eta - \lambda_\nu}(t^s_{\eta\nu} + t^s_{\nu\eta} - \lambda_\nu(u^s_{\eta\nu} + v^s_{\nu\eta}))x_\eta - \tfrac{1}{2}(u^s_{\nu\nu} + v^s_{\nu\nu})x_\nu \tag{49}$$

$$\mathcal{D}_sy_\nu = -\sum_{\substack{\eta=1 \\ \eta\neq\nu}}^{n+m} \frac{1}{\lambda_\eta - \lambda_\nu}(t^s_{\eta\nu} + t^s_{\nu\eta} - \lambda_\nu(u^s_{\eta\nu} + v^s_{\nu\eta}))y_\eta - \tfrac{1}{2}(u^s_{\nu\nu} + v^s_{\nu\nu})y_\nu \tag{50}$$

In a Singular Value Decomposition (SVD) we have both $G = I$ and $H = I$, and the corresponding derivatives are consequently zero, as are the matrices U_s and V_s. This simplifies matters considerably.

The repository has the template gsvdTemplate.R, which can be used together with gevdNonlinear() to compute first and second derivatives of singular values and vectors. The example in the template is a simple one with six parameters. $F(\theta)$ is 4×3, with the columns $(\theta, \theta^2, \theta^3)$. That requires four parameters. The remaining two parameters are used for G and H, which are both of the form $I + \theta(I - ee')$. Thus they have one on the diagonal and θ on the off-diagonal. The additional file gsvdNonlinear.R has the function gsvdNonlinear(), with its numDeriv checker, which computes first derivatives of singular values and vectors. It uses the formulas in this section and is less computationally wasteful as gevdNonlinear(). For now, it does not compute second derivatives,

4.6 Correspondence Analysis

The next two examples (correspondence analysis and multiple correpondence analysis) can be used in combination with the Delta Method. Again, the Delta Method literature is huge and diverse. Special cases have been around before there was a statistics discipline, ever since the beginning of error analysis in geodesy, physics, and astronomy (Gorroochurn (2020)). For the types of applications we have in mind all the relevant Delta Method details are given in Hsu (1949) and Hurt (1976).

Suppose $A()$ and $B()$ are functions of proportions based on n iid observations. We use the Dutch Convention of underlining (sequences of) random variables (Hemelrijk (1966)). The sample proportions are $\underline{p}_n$, and their expected values are π. We have asymptotic normality

$$n^{\frac{1}{2}}(\underline{p}_n - \pi) \xrightarrow{L} \mathcal{N}(0,\ \Pi - \pi\pi'), \tag{51}$$

where Π is a diagonal matrix with π on the diagonal. If f is a three times continuously differentiable function with values in $\mathbb{R}^m$ then

$$n^{\frac{1}{2}}(\underline{p}_n - \pi) \xrightarrow{L} \mathcal{N}(0,\ \mathcal{D}f(\pi)'(\Pi - \pi\pi')\mathcal{D}f(\pi)). \tag{52}$$

Moreover for the expected value we have the convergence in probability

$$n\mathbf{E}(f_\nu(\underline{p}_n) - f_\nu(\pi)) \xrightarrow{P} \tfrac{1}{2}\text{tr}\ \mathcal{D}^2_\nu f(\pi)(\Pi - \pi\pi'). \tag{53}$$

Plugin versions of formulae (52) and (53), with the π on the right hand side replace by $\underline{p}_n$, allow us to use the first and second derivatives of the eigenvalues and eigenvectors to compute asymptotic estimates of biases, standard errors, and confidence intervals.

Correspondence analysis is a GSVD, with F the contingency table, G a diagonal matrix with row sums, and H a diagonal matrix with column sums. All three matrices are linear functions of the proportions. Thus $\mathcal{D}F$ is a unit matrix[4], and $\mathcal{D}G$ and $\mathcal{D}H$ are diagonal unit matrices with row and columns sums of $\mathcal{D}F$. All second derivatives are zero.

The file gevdTemplateCA.R uses the GEVD results for the linear case and applies them to a social mobility contingency table from Glass (1954), also used in De Leeuw and Mair (2009). In this 7×7 table the occupational status of fathers (rows) and sons (columns) of 3497 British families were cross-classified. The categories are professional and high administrative, managerial and executive, higher supervisory, lower supervisory, skilled manual and routine non-manual, semi-skilled manual, and unskilled manual. Running the template followed by either gevdNonlinear() or gsvdNonlinear() will give the derivatives needed for Delta Method standard errors and confidence intervals.

4.7 Multiple Correspondence Analysis

In Multiple Correspondence Analysis (MCA) we have N observations on m categorical variables (also known as "factors"). Variable j has k_j possible values (also known as "levels"), and our observations are coded as unit vectors of length k_j, concatenated to binary vectors (also known as "profiles") of length $K := \sum k_j$. There are $M := \prod k_j$ possible observations, and each of them occurs in the data with relative frequency p_s (in most MCA applications $N << M$ and thus many π_s will be zero). Note that (ordinary) Correspondence Analysis (CA) is the special case with $m = 2$.

The matrices A and B are of the form

$$A(p) = \sum_{s=1}^{M} p_s g_s g_s', \tag{54a}$$

$$B(p) = m \sum_{s=1}^{M} p_s \ \mathrm{diag}(g_s g_s'). \tag{54b}$$

The template gevdTemplateMCA.R use the three-dimensional contingency table HairEyeColor from the R datasets package, which is a part of base R (R Core Team (2025)). The first variable is hair color (4 levels), the second is 2 is eye color (4 levels), and the third variable is sex (2 levels). There are 592 cross-classified individuals, students in an introductory statistics course at the University of Delaware. Thus $N = 592$, $M = 4 \times 4 \times 2 = 32$, $K = 4 + 4 + 2 = 10$.

It is abundantly clear that this example should really be analyzed with some special purpose software which does not require computing and storing the M arrays A_s and

[4] A unit matrix has one element equal to one, the others are zero.

the M arrays B_s, which are symmetric, integer, and very sparse. In this case we want to show, however, that gevNonlinear() can do it all. The template gevdExampleMCA.R can actually handle any three-dimensional table. After running the template, we apply gevNonlinear().

The results for MCA can also be applied directly to the situation in which the possible data are a finite number of numerical vectors which occurs with different relative frequencies. Because of the limits of measurement precision one could argue that all multivariate data are actually of this type. But of course for so-called "continuous" data the number of possible values, although finite, can be enormously large. One solution for the discrete-continuous dilemma is to move from expectations that are weighted by relative frequencies to expectations that are weighted by the empirical distribution function. Our formulae for the derivatives can also be applied to get standard errors, confidence intervals, and bias corrections for the EVD of covariances and correlations using the infinitesimal versions of the bootstrap or the jackknife.

4.8 Unweighted Least Squares CCFA

The final three examples are more complicated. They are all versions of exploratory common factor analysis (ECFA). In Covariance Exploratory Common Factor Analysis (CECFA) we approximate a covariance or correlation matrix by the sum of a diagonal matrix and a matrix of small rank. There are many different ways in which the degree of approximation can be measured, and for a given loss function there are many different proposed algorithms to minimize it. In the next two examples we will compute the first and second derivatives to provide a basis for minimization by Newton's method of some of these loss functions.

In Least Squares Factor Analysis (LSFA) of a covariance matrix C of order n we minimize

$$\sigma(D) = \sum_{i=p+1}^{n} \lambda_\nu^2(C - D) \tag{55}$$

over diagonal matrices D. In a proper LSFA we require $D \geq 0$ and $C - D$ positive semi-definite, but these constraints need not bother us here because we are only interested in the first and second derivatives. The formulas we derive have been given earlier by Derflinger (1969) and Joreskog and Van Thillo (1971).

Because $A = C - D$ we have $\mathcal{D}_i A = -e_i e_i'$. It follows directly from (5) that

$$\mathcal{D}_i \sigma = -2 \sum_{\nu=p+1}^{n} \lambda_\nu x_{i\nu}^2. \tag{56}$$

From (22), taking into account that $B = I$, and thus $D_i B = D_{ij} B = 0$,

$$\mathcal{D}_{ij}\sigma_\star = 2\sum_{\nu=p+1}^{n} x_{i\nu}^2 x_{j\nu}^2 - 4\sum_{\nu=p+1}^{n}\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n} \frac{\lambda_\nu}{\lambda_\eta - \lambda_\nu} x_{i\nu}x_{j\nu}x_{i\eta}x_{j\eta}. \tag{57}$$

The file lsfa.R in the github repository has functions to compute the first and second derivatives, with the usual numDeriv checker.

4.9 The Swain CCFA Family

In an important paper Swain (1975) derived a class of factor analytic procedures which produce estimates that are asymptotically equivalent to the classical Lawley-Joreskog maximum likelihood estimates. Swain also wrote down the necessary first and second order derivatives that can be used in implementing Newton's method, thus generalizing the work of Jennrich and Robinson (1969), Derflinger (1969), and Clarke (1970) for the maximum likelihood method.

Suppose $\mathcal{F}$ is the set of twice-differentiable univariate functions defined on the positive real axis with the properties

$$f(1) = \mathcal{D}f(1) = 0, \tag{58a}$$
$$\mathcal{D}^2 f''(1) = 1, \tag{58b}$$
$$\mathcal{D}f(\theta) < 0 \text{ for } 0 < \theta < 1, \tag{58c}$$
$$\mathcal{D}f(\theta) > 0 \text{ for } \theta > 1. \tag{58d}$$

Note that this implies that each $f \in \mathcal{F}$ is unimodal and has a single minimum equal to zero at one. Also, by l'Hôpital,

$$\lim_{x\to 1} \frac{f(x)}{\frac{1}{2}(x-1)^2} = 1. \tag{59}$$

Define the signature of a real-valued function f at x to be -1 if $f(x) < 0$, +1 if $f(x) > 0$, and 0 if $f(x) = 0$. It follows from (58a)-(58d) that if $f, g \in \mathcal{F}$ then the signatures of $\mathcal{D}f$ and $\mathcal{D}g$ are the same.

Swain proposes to minimize

$$\sigma(D) := \sum_{\nu=p+1}^{n} f(\lambda_\nu(A)), \tag{60}$$

with λ the eigenvalues of $A := S^{-\frac{1}{2}}DS^{-\frac{1}{2}}$. These are also the eigenvalues of the asymmetric matrix $S^{-1}D$, and the generalized eigenvalues of the pair (D, S).

Maximum likelihood estimation, generalized least squares estimation (Joreskog and Goldberger (1972)), and the "geodesic distance" between covariance matrices (James (1973)) are special cases. For maximum likelihood

$$f_1(\theta) := 1/\theta + \log\theta - 1, \tag{61a}$$

for GLS with loss $\frac{1}{2}\text{tr } S^{-1}(\Sigma - S)S^{-1}(\Sigma - S)$ we have

$$f_2(\theta) := \frac{1}{2}(\theta - 1)^2, \tag{61b}$$

and for the "geodetic distance"

$$f_3(\theta) := \frac{1}{2}(\log\theta)^2. \tag{61c}$$

But (58a)-(58d) also covers other cases such as the GLS variant with loss $\frac{1}{2}\text{tr } \Sigma^{-1}(\Sigma - S)\Sigma^{-1}(\Sigma - S)$ which has

$$f_4(\theta) := \frac{1}{2}\frac{(\theta - 1)^2}{\theta^2}. \tag{61d}$$

In addition if $f(\theta)$ satisfies (58a)-(58d) for all θ then so does $f(\theta^{-1})$. In fact $f_4(\theta) = f_2(\theta^{-1})$. Also $f_3(theta) = f_3(\theta^{-1})$. Note that the unweighted least squares method from the previous section is not in the Swain family, but the GLS methods are, and they have the advantage of being scale-free.

From $A = S^{-\frac{1}{2}}DS^{-\frac{1}{2}}$ we have $\mathcal{D}_i A = S^{-\frac{1}{2}}e_i e_i' S^{-\frac{1}{2}}$. To apply the EVD derivative formulas it is convenient to define the matrix $Y := S^{-\frac{1}{2}}X$ with elements $y_{i\nu}$. It follows that

$$\mathcal{D}_i\lambda_\nu = (x_\nu' S^{-\frac{1}{2}}e_i)^2 = y_{i\nu}^2. \tag{62a}$$

and

$$\mathcal{D}_i x_\nu = -\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n} \frac{1}{\lambda_\eta - \lambda_\nu} x_\eta x_\eta'(S^{-\frac{1}{2}}e_i e_i' S^{-\frac{1}{2}})x_\nu = -\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n} \frac{1}{\lambda_\eta - \lambda_\nu} y_{i\nu}y_{i\eta}\cdot x_\eta \tag{62b}$$

Consequently

$$\mathcal{D}_i\sigma = \sum_{\nu=p+1}^{n} \mathcal{D}f(\lambda_\nu)\mathcal{D}_i\lambda_\nu = \sum_{\nu=p+1}^{n} \mathcal{D}f(\lambda_\nu)y_{i\nu}^2. \tag{63}$$

Differentiate once again

$$\mathcal{D}_{ij}\sigma = \sum_{\nu=p+1}^{n} \mathcal{D}^2 f(\lambda_\nu)y_{j\nu}^2 y_{i\nu}^2 - 2\sum_{\nu=p+1}^{n}\sum_{\substack{\eta=1\\ \eta\neq\nu}}^{n} \mathcal{D}f(\lambda_\nu)\{\frac{1}{\lambda_\eta - \lambda_\nu} y_{i\nu}y_{i\eta}y_{j\nu}y_{j\eta}\} \tag{64}$$

The github repository has a file swain.R which has functions to compute the first and second derivatives for the more common choices of $f \in \mathcal{F}$. There is also a numDeriv checker. The functions are programmed in such a way that they can easily be extended to other loss functions with f satisfying (58a)-(58d).

4.10 Matrix Decomposition Factor Analysis

In Matrix Decomposition Factor Analysis (MDFA) we have to maximize the trace or nuclear norm $\|.\|_\tau$, i.e. the sum of the singular values, of a matrix XT over T. This is equivalent to maximizing the sum of the square roots of the eigenvalues of $T'X'XT$.

In MDFA the matrix X is $n \times m$ and T is $m \times p$, with $m < p$. Thus $A = T'X'XT$ is of order p and of rank $m \leq p$. We suppose it is locally of constant rank n and we maximize the sum of the square roots of its m non-zero eigenvalues (which is equal to the trace norm).

The matrix T can be patterned, which means that some of its elements are fixed at known values, usually zero. In exploratory MDFA we typically have $T = \begin{bmatrix} U & D \end{bmatrix}$, with an $m \times r$ with $r = p - m$, matrix U of common factor loadings and a diagonal matrix D of uniquenesses. Thus the pattern in this case is that the off-diagonal elements of D are zero. In confirmatory MDFA some elements of U are fixed at known values as well.

So $A = T'CT$, with $C = X'X$ a constant matrix, and $T(\theta) = T_0 + \sum_{s=1}^{p} \theta_s T_s$. The fixed elements of the pattern are in T_0, which has zeroes for the free elements, while the T_s with $s > 0$ all have zeroes for the fixed elements. In this linear/quadratic case we have

$$\mathcal{D}_s A = T_s'CT + T'CT_s, \tag{65a}$$
$$\mathcal{D}_{st} A = T_s'CT_t + T_t'CT_s, \tag{65b}$$

which are symmetric matrices if order p. Thus

$$\mathcal{D}_s \lambda_\nu = x_\nu' Q_s x_\nu, \tag{66}$$

where $Q_s := \mathcal{D}_s A = T_s'CT + T'CT_s$. It follows that

$$\mathcal{D}_{st} \lambda_\nu = 2x_\nu' Q_s (\mathcal{D}_s x_t) + x_\nu' (\mathcal{D}_t Q_s) x_\nu. \tag{67}$$

From (15)

$$\mathcal{D}_t x_\nu = -(A - \lambda_\nu I)^+ Q_t x_\nu, \tag{68}$$

and thus, or directly from (35),

$$\mathcal{D}_{st} \lambda_\nu = -2x_\nu' Q_s (A - \lambda_\nu I)^+ Q_t x_\nu + 2x_\nu' T_s' C T_t x_\nu. \tag{69}$$

Formulae (66) and (69) apply to all EVD problems where $A = T'CT$ and T is a weighted linear combination of known matrices, with or without intercept.

We now specialize to the problem of differentiating the trace norm

$$\sigma(T) := \|XT\|_\tau = \sum_{\nu=1}^{m} \lambda_\nu^{\frac{1}{2}}(T'CT), \tag{70}$$

for which we have

$$\mathcal{D}_s\sigma = \sum_{\nu=1}^{m} \lambda_\nu^{-\frac{1}{2}} x_\nu' T'CT_s x_\nu = \text{tr } (T'CT)^{-\frac{1}{2}} T'CT_s, \tag{71}$$

where $(T'CT)^{-\frac{1}{2}}$ is the square root of the Moore-Penrose inverse of $T'CT$. Next differentiate (71) to find

$$\mathcal{D}_{st}\sigma = -\frac{1}{4}\sum_{\nu=1}^{m} \lambda_\nu^{-\frac{3}{2}}(\mathcal{D}_s\lambda_\nu)(\mathcal{D}_t\lambda_\nu) + \frac{1}{2}\sum_{\nu=1}^{m} \lambda_\nu^{-\frac{1}{2}}(\mathcal{D}_{st}\lambda_\nu), \tag{72}$$

and we can substitute from formulae (66) and (69) for a final result.

If the T_s are unit matrices we find

$$\mathcal{D}\sigma = CT(T'CT)^{-\frac{1}{2}}, \tag{73}$$

where the fixed elements correspond with zeroes in $\mathcal{D}\sigma$.

The repository has the file mdfa.R, which has functions (and numDeriv checkers) to compute the first and second derivatives of the eigenvalues of the general linear/quadratic problem and the first and second derivatives of the trace norm for the MDFA problem.

References

Baumgärter, H. 1985. *Analytic Perturbation Theory for Matrices and Operators*. Birkhäuser.

Brustad, K. K. 2019. "Total Derivatives of Eigenvalues and Eigenprojections of Symmetric Matrices." https://arxiv.org/abs/1905.06045.

Chu, K.-W. Eric. 1990. "On Multiple Eigenvalues of Matrices Depending on Several Parameters." *SIAM Journal on Numerical Analysis* 27 (5): 1368–85.

Clarke, M. R. B. 1970. "A Rapidly Convergent Method for Maximum-Likelihood Factor Analysis." *British Journal of Mathematical and Statistical Psychology* 23: 43–52.

De Leeuw, J. 2007. "Derivatives of Generalized Eigen Systems with Applications." Preprint Series 528. Los Angeles, CA: UCLA Department of Statistics. https://jansweb.netlify.app/publication/deleeuw-r-07-c/deleeuw-r-07-c.pdf.

De Leeuw, J., and P. Mair. 2009. "Simple and Canonical Correspondence Analysis Using the R Package Anacor." *Journal of Statistical Software* 31 (5): 1–18. https://www.jstatsoft.org/v31/i05/.

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

Gilbert, P., and R. Varadhan. 2019. *numDeriv: Accurate Numerical Derivatives*. https://CRAN.R-project.org/package=numDeriv.

Glass, D. V. 1954. *Social Mobility in Britain*. Free Press.

Gorroochurn, P. 2020. "Who Invented the Delta Method, Really ?" *The Mathematical INtelligencer* 42: 46–49.

Hemelrijk, J. 1966. "Underlining Random Variables." *Statistica Neerlandica* 20: 1–7.

Hiriart-Urruty, J.-B., and D. Ye. 1995. "Sensitivity Analysis of All Eigenvalues of a Symmetric Matrix." *Numerische Mathematik* 70: 45–72.

Hsu, P. L. 1949. "The Limiting Distribution of Functions of Sample Means and Application to Testing Hypotheses." In *Proceedings of the First Berkeley Symposium on Mathematical Statistics and Probability*, edited by Neyman J, 359–401. University of California Press.

Hurt, J. 1976. "Asymptotic Expansions of Functions of Statistics." *Aplikace Matematiky* 21: 444–56.

James, A. T. 1973. "The Variance Information Manifold and the Functions on It." In *Multivariate Analysis*, edited by P. R. Krishnaiah, III:157–69. North Holland Publishing Company.

Jennrich, R. I., and S. M. Robinson. 1969. "A Newton-Raphson Algorithm for Maximum Likelihood Factor Analysis." *Psychometrika* 34: 111–23.

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

Joreskog, 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.

Kato, T. 1984. *A Short Introduction to Perturbation Theory for Linear Operators*. Springer.

Lewis, A. S., and M. L. Overton. 1996. "Eigenvalue Optimization." *Acta Numerica* 33: 149–90.

Nelson, R. B. 1976. "Simplified Calculation of Eigenvector Derivatives." *AIAA Journal* 14 (9): 1201–5.

Overton, M. L., and R. S. Womersley. 1995. "Second Derivatives for Optimizing Eigenvalues of Symmetric Matrices." *SIAM Journal on Matrix Analysis and Applications* 16 (3): 697–718.

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

Rao, C. R., and MS K. Mitra. 1971. *Generalized Inverse of Matrices and Its Applications*. Wiley.

Sun, J.-G. 1990. "Multiple Eigenvalue Sensitivity Analysis." *Linear Algebra and Its Applications* 137/138: 183–211.

Swain, A. J. 1975. "A Class of Factor Analysis Estimation Procedures with Common Asymptotic Sampling Properties." *Psychometrika* 40 (3): 315–36.

Torki, M. 1999. "First- and Second-Order Epi-Differentiability in Eigenvalue Optimization." *Journal of Mathematical Analysis and Applications* 234: 391–416.

———. 2001. "Second-order Directional Derivatives of All Eigenvalues of a Symmetric Matrix." *Nonlinear Analysis* 46: 1133–50.