Some Majorization Theory for Weighted Least Squares

Jan de Leeuw

Version 13, May 26, 2017

Abstract

In many situations in numerical analysis least squares loss functions with diagonal weight matrices are much easier to minimize than least square loss functions with full positive semi-definite weight matrices. We use majorization to replace problems with a full weight matrix by a sequence of diagonal weight matrix problems. Diagonal weights which optimally approximate the full weights are computed using a simple semi-definite programming procedure.

Contents

1 Introduction **2**

2 Majorization **2**
2.1 Singularity 3

3 Rate of Convergence **3**
3.1 Linear Least Squares 3
3.2 Non-linear Least Squares 4

4 Minimum Trace Majorization Bound **5**
4.1 Extreme Correlation Matrices 6

5 Algorithm **6**
5.1 Examples 7

6 Majorization Examples **8**
6.1 Monotone Regression 8
6.2 Multidimensional Scaling 9
6.3 Structural Equation Models 10

7 Appendix: Code **10**
7.1 maxR 10
7.2 jbkWPava 11

References **13**

Note: This is a working paper which will be expanded/updated frequently. All suggestions for improvement are welcome. The directory gifi.stat.ucla.edu/wls has a pdf version, the complete Rmd file, the R code, and the bib file.

1 Introduction

In this paper we study the problem of minimizing a weighted least squares loss function $\sigma(x) = x'Wx$ over a set $\mathcal{X} \subseteq \mathbb{R}^n$. Here W is a positive semi-definite matrix of order n.

Examples are linear least squares, where $\mathcal{X}$ is of the form $y - A\beta$, or isotone regression, where $\mathcal{X} = y - \mathcal{K}$ with K the cone of isotone vectors, or nonlinear least squares, where the elements of $\mathcal{X}$ are of the form $y - F(\theta)$ for $F : \mathbb{R}^p \to \mathbb{R}^n$ and $\theta \in \mathbb{R}^p$.

The idea is that weighted least squares problems with general W are more difficult to solve than problems with a diagonal W. We use majorization theory to reduce problems with non-diagonal weights to sequences of problems with diagonal weights. General discussions of majorization (also known as MM) algorithms are in De Leeuw (2016) and Lange (2016). Majorization theory has been used earlier in the context of weighted least squares problems by Heiser (1995), Kiers (1997) and Groenen, Giaquinto, and Kiers (2003), although their applications are limited to linear regression and low-rank matrix approximation (also known as principal component analysis) with rather simple bounds.

2 Majorization

Write $x = z + (x - z)$ and thus

$$\sigma(x) = (z + (x - z))'W(z + (x - z)) = \sigma(z) + 2z'W(x - z) + (x - z)'W(x - z).$$

Now suppose we have a diagonal D satisfying $D \gtrsim W$ in the Loewner sense, i.e. $D - W$ is positive semi-definite. Then $\sigma(x) \leq \eta(x, z)$ for all $x, z \in \mathcal{X}$, where

$$\eta(x, z) := \sigma(z) + 2z'W(x - z) + (x - z)'D(x - z),$$

and, of course, $\sigma(x) = \eta(x, x)$ for all $x \in \mathcal{X}$. Thus $\eta : \mathcal{X} \otimes \mathcal{X} \to \mathbb{R}$ is a *majorization scheme* in the sense of De Leeuw (2016), chapter 7. The corresponding majorization algorithm is

$$x^{(k+1)} \in \underset{x \in \mathcal{X}}{\textbf{Argmin}} \ \eta(x, x^{(k)}),$$

and the *sandwich inequality* is

$$\sigma(x^{(k+1)}) \leq \eta(x^{(k+1)}, x^{(k)}) \leq \eta(x^{(k)}, x^{(k)}) = \sigma(x^{(k)}).$$

The first inequality is strict if the minimizer of $\eta(x, x^{(k)})$ is unique, the second is strict if $D - W$ is positive definite.

Define $r(z) := (I - D^{-1}W)z$. Then, completing the square, minimizing $\eta(x, z)$ over x is the same as minimizing

$$\rho(x, z) := (x - r(z))'D(x - r(z))$$

over x.

Consequently an iteration of the majorization algorithm finds $x^{(k+1)}$ by projecting $r(x^{(k)})$ on $\mathcal{X}$, using the metric D.

2.1 Singularity

It should be mentioned that our majorization method also applies if W is singular. In fact we can show that D will be positive definite in all situations of interest. Suppose D is singular, i.e. it has zero diagonal elements. Then $D \gtrsim W$ implies that W also has the same zero diagonal elements, and because W is positive semi-definite all off-diagonal elements of W corresponding with these diagonal elements are zero as well. If I is the index set for which the diagonal elements of W are zero, then we must minimize $z_I'W_I z_I$ and for majorization use a positive definite $D_I \gtrsim W_I$. Thus majorization can be also be used to regularize singular problems.

If W is indefinite it is still possible to choose a positive definite diagonal D such that $D - W \gtrsim 0$. We will still have a majorization algorithm that generates a decreasing sequence of loss function values, but the loss function may no longer be bounded below, and thus the sequence of loss function values may not converge.

3 Rate of Convergence

3.1 Linear Least Squares

For simplicity we start with the linear case. Thus $\mathcal{X} = \{x \mid x = y - A\beta\}$. Assume A to be of full column rank, and assume D to be positive definite. We can write the majorization scheme as

$$\eta(\beta, \gamma) = \sigma(\gamma) + 2(y - A\gamma)'WA(\gamma - \beta) + (\gamma - \beta)'A'DA(\gamma - \beta).$$

Thus

$$\mathcal{D}_1\eta(\beta, \gamma) = -2A'W(y - A\gamma) - 2A'DA(\gamma - \beta),$$

which is zero for

$$\beta = \gamma + (A'DA)^{-1}A'W(y - A\gamma) = (A'DA)^{-1}A'Wy + (I - (A'DA)^{-1}A'WA)\gamma.$$

The iterations take the form

$$\beta^{(k+1)} = (A'DA)^{-1}A'Wy + (I - (A'DA)^{-1}A'WA)\beta^{(k)},$$

and thus

$$\beta^{(k+1)} - \beta^{(k)} = (I - (A'DA)^{-1}A'WA)(\beta^{(k)} - \beta^{(k-1)}). \tag{1}$$

The speed of convergence of the iteration is determined by the eigenvalues of $(A'DA)^{-1}A'WA$, which are less than or equal to one. Thus the smaller D is, given that $D - W \gtrsim 0$, the faster the convergence will be. The linear convergence rate of our majorization algorithm is the largest eigenvalue of $I - (A'DA)^{-1}A'WA$.

3.2 Non-linear Least Squares

For non-linear least squares $\mathcal{X} = \{x \mid y - F(\theta)\}$. The majorization scheme is

$$\eta(\theta, \xi) = \sigma(\xi) - 2(y - F(\xi))'W(F(\theta) - F(\xi)) + (F(\theta) - F(\xi))'D(F(\theta) - F(\xi)).$$

Minimizing over θ for fixed ξ means solving

$$\mathcal{D}_1\eta(\theta, \xi) = -2G(\theta)'W(y - F(\xi)) + 2G(\theta)'D(F(\theta) - F(\xi)) = 0,$$

where $G\theta)$ is short for $\mathcal{D}F(\theta)$. This implicitly defines θ as a function of ξ, and we have, at a fixed point where $\theta(\xi) = \xi$,

$$\mathcal{D}\theta(\xi) = -\left[\mathcal{D}_{11}\eta(\xi, \xi)\right]^{-1}\mathcal{D}_{12}\eta(\xi, \xi).$$

What remains to be done is computing expressions for the derivatives. We have

$$\mathcal{D}_{12}\eta(\xi, \xi) = 2G(\xi)'WG(\xi) - 2G(\xi)'DG(\xi),$$

$$\mathcal{D}_{11}\eta(\theta, \xi) = 2G(\xi)'DG(\xi) - 2H(\xi),$$

where

$$H(\xi) = \sum_{i=1}^{n}\mathcal{D}^2 f_i(\xi)\sum_{j=1}^{n} w_{ij}(y_j - f_j(\xi)).$$

Thus

$$\mathcal{D}\theta(\xi) = [G(\xi)'DG(\xi) - H(\xi)]^{-1}(G(\xi)'DG(\xi) - G(\xi)'WG(\xi)). \tag{2}$$

In the linear case $H(\xi) = 0$ and $G(\xi) = A$, and we find the result in equation (1). Again we see that, in general terms, it will be good to have D as small as possible. Note that if we have perfect fit, i.e. if $y = F(\xi)$, then again $H(\xi) = 0$ and

$$\mathcal{D}\theta(\xi) = (I - [G(\xi)'DG(\xi)]^{-1}G(\xi)'WG(\xi)).$$

Both in the linear and nonlinear case the derivatives $G(\xi)$ and A complicate the comparison of different diagonal matrices of weights, but the general idea that we want both $D - W \gtrsim 0$ and D as small as possible seems to be a useful summary of the results so far.

4 Minimum Trace Majorization Bound

Let us look at various ways in which we can have a small D, while satisfying $D-W \gtrsim 0$.There are many ways, of course, to measure how large D is. In the context of linear iterations, we have seen that what matters is the size of $A'DA$ relative to $A'WA$, and something very similar is true in the non-linear case. But in this section we will look for diagonal matrices D that are small in a more general sense, independent of the value of any derivatives or coefficient matrices.

The set of all scalar matrices $D = \lambda I$ with $D \gtrsim W$ is easy to describe. We have $D \gtrsim W$ if and only if $\lambda \geq \lambda_{\max}(W)$, the largest eigenvalue of W. Thus we also have $D \gtrsim W$ if $D = \|W\|I$, with W any matrix norm. Perhaps the easiest bound is $D = \mathbf{tr}(W)I$, although that will tend to be really unsharp.

Another easy bound uses $V = \mathbf{diag}(W)$. Now $V^{-\frac{1}{2}}WV^{-\frac{1}{2}}$ is a correlation matrix, which means its largest eigenvalue is less than or equal to n. Thus $D = nV$ satisfies $D \gtrsim W$. In general, however, it seems that requiring D to be scalar, and working with the largest eigenvalue, or even bound on the largest eigenvalue, will lead to large D, and consequently slow convergence of the majorization algorithm. So we will attempt to do better. For now we adopt the trace as a simple measure of the size of D, which does not take specific aspects of the least squares problem into account.

What we try to solve in this section is to compute $\min_D \mathbf{tr}\ D$ over $D - W \gtrsim 0$. Let's call this the *minimum trace majorization bound* or MTMB problem. Before we start, let us note the similarity of the MTMB problem to *minimum trace factor analysis* (MTFA), in which we compute $\max_D \mathbf{tr}\ D$ over $W - D \gtrsim 0$ (and $D \gtrsim 0$ in the case of constrained MTFA). See Watson (1992) and Jamshidian and Bentler (1998) for MTFA algorithms. Both MTFA and MTMB are examples of *semidefinite programming* or SDP (Vandenberghe and Boyd (1996)).

We first analyse the MTMB problem a bit more in detail. The Lagrangian is

$$\mathcal{L}(D, R) = \mathbf{tr}\ D - \mathbf{tr}\ R(D - W),$$

where $R \gtrsim 0$ is a symmetric matrix of Lagrange multipliers (or dual variables). The nececessary and sufficient conditions for a minimum are

$$\begin{aligned} D - W &\gtrsim 0, \\ R &\gtrsim 0, \\ \mathbf{diag}(R) &= I, \\ R(D - W) &= 0. \end{aligned}$$

Because

$$\max_{R \gtrsim 0} \mathcal{L}(D, R) = \begin{cases} \mathbf{tr}\ D & \text{if } D - W \gtrsim 0, \\ +\infty & \text{otherwise,} \end{cases}$$

the primal problem is $\min_D \max_{R \gtrsim 0} \mathcal{L}(D, R)$.

The dual function is

$$\min_D \mathcal{L}(D, R) = \begin{cases} \mathbf{tr}\ RW & \text{if } \mathbf{diag}(R) = I, \\ -\infty & \text{otherwise,} \end{cases}$$

and thus dual probem is to maximize $\mathbf{tr}\ RW$ over $R \gtrsim 0$ with $\mathbf{diag}(R) = I$, i.e. over all correlation matrices. Because both the primal and the dual problem are structly feasible, the optimal value of both problems are equal.

4.1 Extreme Correlation Matrices

Correlation matrices, i.e. positive semi-definite matrices with a unit diagonal, form a compact convex $\mathcal{E}_{n\times n}$ set in the space of all symmetric matrices. Laurent and Poljak (1996) call this set the *elliptope*. A linear function on a compact convex set attains its maximum at one of the extreme poonts of the set, i.e. one of the points that cannot be represented as a convex combination of two other points in the set. Thus the solution of the dual problem of maximizing $\mathbf{tr}\ RW$ is an extreme point of the elliptope. This makes it interesting for us to look at the structure of such extreme points

Correlation matrices of rank one are of the form xx', where x is a vector with element ± 1. Such matrices are called *cut matrices* in Laurent and Poljak (1996), and their convex hull is the *cut polytope*. The max-cut problem is maximizing a linear function over the cut polytope, and clearly maximizing that linear function over the elliptope is a convex relaxation of the *max cut* problem. Cut matrices are extreme points of the elliptope, but having rank one is only sufficient, not necessary, for extreme points. A comprehensive recent paper on the rank and structure of the extreme points of the elliptope, which also reviews previous results on the topic, is Li and Tam (1994).

It was first shown by Grone, Pierce, and Watkins (1990) that the elliptope $\mathcal{E}_{n\times n}$ contains an extreme point of rank r if and only if $r(r+1) \leq 2n$. The face structure of the elliptope was analyzed in Laurent and Poljak (1996), and necessary and sufficient conditions for a correlation matrix of rank r to be an extreme point are given in Ycart (1985), Li and Tam (1994), Parthasarathy (2002), and Hürlimann (2015).

5 Algorithm

Although we have a choice of many different SDP algorithms, we use a special purpose method to solve MTMB. It seems to work well. We maximize $\mathbf{tr}\ U'UW$ by cycling over the columns of U, changing one of them at a time, requiring throughout that $u_i'u_i = 1$. Clearly the solution for column u_i, keeping all other column at their current value, is given by $v = \sum_{k\neq i} w_{ij}u_k$ and then $u_i = v/\|v\|$. The appendix has a simple R function `maxR()` to do exactly this

If we have found the solution, we use $R(D - W) = 0$ to find the solution of the primal problem.

5.1 Examples

We give some example of application of our dual algorithm. Our first matrix W is

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] +1.0 -1.0 +0.0 +0.0 +0.0 +0.0
## [2,] -1.0 +2.0 -1.0 +0.0 +0.0 +0.0
## [3,] +0.0 -1.0 +2.0 -1.0 +0.0 +0.0
## [4,] +0.0 +0.0 -1.0 +2.0 -1.0 +0.0
## [5,] +0.0 +0.0 +0.0 -1.0 +2.0 -1.0
## [6,] +0.0 +0.0 +0.0 +0.0 -1.0 +1.0
```

This has largest eigenvalue 3.7320508076, and thus the best scalar D has trace 22.3923048454. Using the trace of W for a scalar D gives a trace of D of 60. The iterations of our algorithm, which always start with $R = I$, are

```
## itel       1  fold   10.000000  fnew   17.656854
## itel       2  fold   17.656854  fnew   19.431440
## itel       3  fold   19.431440  fnew   19.804726
## itel       4  fold   19.804726  fnew   19.914796
## itel       5  fold   19.914796  fnew   19.962473
## itel       6  fold   19.962473  fnew   19.983844
## itel       7  fold   19.983844  fnew   19.993073
## itel       8  fold   19.993073  fnew   19.997032
## itel       9  fold   19.997032  fnew   19.998729
## itel      10  fold   19.998729  fnew   19.999455
## itel      11  fold   19.999455  fnew   19.999767
## itel      12  fold   19.999767  fnew   19.999900
## itel      13  fold   19.999900  fnew   19.999957
## itel      14  fold   19.999957  fnew   19.999982
## itel      15  fold   19.999982  fnew   19.999992
## itel      16  fold   19.999992  fnew   19.999997
## itel      17  fold   19.999997  fnew   19.999999
## itel      18  fold   19.999999  fnew   19.999999
```

The optimum R is a cut matrix, given as

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] +1.0 -1.0 +1.0 -1.0 +1.0 -1.0
## [2,] -1.0 +1.0 -1.0 +1.0 -1.0 +1.0
## [3,] +1.0 -1.0 +1.0 -1.0 +1.0 -1.0
## [4,] -1.0 +1.0 -1.0 +1.0 -1.0 +1.0
## [5,] +1.0 -1.0 +1.0 -1.0 +1.0 -1.0
```

```
## [6,] -1.0 +1.0 -1.0 +1.0 -1.0 +1.0
```

and the diagonal of the corresponding D is

```
## [1] 2.000000 4.000000 4.000000 4.000000 4.000000 2.000000
```

This is close to the best scalar D, because in this first example all eigenvalues of W are close.

In the second example we take W of order 6, but of rank one.

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] +0.0 +0.1 +0.1 +0.2 +0.2 +0.2
## [2,] +0.1 +0.2 +0.2 +0.3 +0.4 +0.5
## [3,] +0.1 +0.2 +0.4 +0.5 +0.6 +0.7
## [4,] +0.2 +0.3 +0.5 +0.6 +0.8 +1.0
## [5,] +0.2 +0.4 +0.6 +0.8 +1.0 +1.2
## [6,] +0.2 +0.5 +0.7 +1.0 +1.2 +1.4
```

Its largest eigenvalue (and its trace) is 3.64. The iterations are

```
## itel       1  fold    3.640000  fnew   17.132389
## itel       2  fold   17.132389  fnew   17.633437
## itel       3  fold   17.633437  fnew   17.639916
## itel       4  fold   17.639916  fnew   17.639999
## itel       5  fold   17.639999  fnew   17.640000
```

The cut matrix we find is

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] +1.0 +1.0 +1.0 +1.0 +1.0 +1.0
## [2,] +1.0 +1.0 +1.0 +1.0 +1.0 +1.0
## [3,] +1.0 +1.0 +1.0 +1.0 +1.0 +1.0
## [4,] +1.0 +1.0 +1.0 +1.0 +1.0 +1.0
## [5,] +1.0 +1.0 +1.0 +1.0 +1.0 +1.0
## [6,] +1.0 +1.0 +1.0 +1.0 +1.0 +1.0
```

which corresponds with D equal to

```
## [1] 0.840000 1.680000 2.520000 3.360000 4.200000 5.040000
```

6 Majorization Examples

6.1 Monotone Regression

Consider the problem of minimizing $(y-x)'W(y-x)$ over all x satisfying $x_1 \leq \cdots \leq x_n$. If W is diagonal, this problem can be solved quickly (in linear time) with the usual monotone regression algorithms, and thus we can use majorization to reduce the problem to a sequence of such monotone regressions.

In our example we use W equal to

```
##        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]  1    1    1    1    1    1    1    1    1    1
##  [2,]  1    2    2    2    2    2    2    2    2    2
##  [3,]  1    2    3    3    3    3    3    3    3    3
##  [4,]  1    2    3    4    4    4    4    4    4    4
##  [5,]  1    2    3    4    5    5    5    5    5    5
##  [6,]  1    2    3    4    5    6    6    6    6    6
##  [7,]  1    2    3    4    5    6    7    7    7    7
##  [8,]  1    2    3    4    5    6    7    8    8    8
##  [9,]  1    2    3    4    5    6    7    8    9    9
## [10,]  1    2    3    4    5    6    7    8    9   10
```

and y equal to

```
##  [1]  1  3  2  3  3  1  1  4  4  1
```

Matrix W has a largest eigenvalue equal to 44.7660686527 and a trace equal to 55. The trace-optimal D is

```
##  [1] 10 19 27 34 40 45 49 52 54 55
```

We make three runs of the majorization version `jbkWPava()`, which uses `jbkPava()` from De Leeuw (2017) for the diagonal monotone regression. Using a scalar matrix with the trace requires 355 iterations, using the largest eigenvalue uses 296 iterations, and using the trace-optimal D uses 113 iterations. Of course this does not address the question if the gain in the number of iterations offsets the extra computation needed for either the largest eigenvalue or the trace-optimal D.

6.2 Multidimensional Scaling

Suppose Δ_r with $r = 1, ..., R$ are independent and identically distributed symmetric and hollow dissimilarity matrices of order n. Define for $i < j$

$$\overline{\delta}_{ij} = \frac{1}{R}\sum_{r=1}^{R}\delta_{ijr},$$

and for $i < j$ and $k < l$

$$w_{ijk\ell} = \frac{1}{R}\sum_{r=1}^{R}(\delta_{ijr} - \overline{\delta}_{ij})(\delta_{k\ell r} - \overline{\delta}_{k\ell})$$

Also define the inverse V of $W = \{w_{ijk\ell}\}$, and now minimize the stress loss function

$$\sigma(X) = \sum_{i<j}\sum\sum_{k<\ell} v_{ijk\ell}(\overline{\delta}_{ij} - d_{ij}(X))(\overline{\delta}_{k\ell} - d_{k\ell}(X)).$$

The minimum of stress is distributed as chi-square with $\frac{1}{2}n(n-1) - (pn - \frac{1}{2}p(p+1))$ degrees of freedom. Bounding the matrix V by a diagonal matrix with elements e_{ij} allows us to use the usual smacof algorithm (De Leeuw and Mair (2009)).

More precisely minimizing the majorization function amounts to minimizing over configurations X the loss

$$\sigma(X, X^{(k)}) = \sum_{i<j}\sum e_{ij}(d_{ij}(X) - (d_{ij}(X^{(k)}) + r_{ij}(X^{(k)})))^2,$$

where

$$r_{ij}(X^{(k)}) = \frac{1}{e_{ij}}\sum_{k<\ell}\sum w_{ijk\ell}(\delta_{ij} - d_{ij}(X^{(k)})).$$

In an actual program we will perform a number of smacof *inner iterations* to update $X^{(k)}$ without insisting on convergence to define $X^{(k+1)}$. In fact, a single inner iteration may suffice.

This approach can be extended to group difference models, in which each group has a different set of distances, and all groups are large. Actual examples of this approach are a bit hard to come by, because usually existing data sets are already averaged over individuals, or they are too small to allow for an invertible covariance matrix of the dissimilarities.

6.3 Structural Equation Models

The generalized least squares loss function in multinormal multivariate analysis is

$$\sigma(\theta) = \mathbf{tr}\ S^{-1}(S - \Sigma(\theta))S^{-1}(S - \Sigma(\theta)).$$

If we define $r(\theta) := \mathbf{vec}(S - \Sigma(\theta))$ then $\sigma(\theta) = r(\theta)'(S^{-1} \otimes S^{-1})r(\theta)$. If $D \gtrsim S^{-1}$ then $D \otimes D \gtrsim S^{-1} \otimes S^{-1}$, and of course $D \otimes D$ is diagonal. And if D is scalar, so is $D \otimes D$. Thus we can majorize with a weighted sum or, less precisely, with an unweighted sum of squares.

7 Appendix: Code

7.1 maxR

```
maxR <- function (w,
                  eps = 1e-6,
                  itmax = 100,
                  verbose = FALSE) {
  n <- nrow (w)
  x <- diag (n)
  r <- diag (n)
  itel <- 1
  fold <- sum (diag (w))
  repeat {
    for (i in 1:n) {
      y <- x %*% w[, i] - w[i, i] * x[, i]
```

```
      x [, i] <- y / sqrt (sum (y ^ 2))
      r [, i] <- r[i, ] <- x[, i] %*% x
    }
    fnew <- sum (r * w)
    if (verbose)
      cat (
        "itel ",
        formatC(
          itel,
          digits = 4,
          width = 6,
          format = "d"
        ),
        " fold ",
        formatC(
          fold,
          digits = 6,
          width = 10,
          format = "f"
        ),
        " fnew ",
        formatC(
          fnew,
          digits = 6,
          width = 10,
          format = "f"
        ),
        "\n"
      )
    if (((fnew - fold) < eps) || (itel == itmax))
      break
    itel <- itel + 1
    fold <- fnew
  }
  d <- (r %*% w) / r
  return (list (r = r, d = d[1, ], itel = itel))
}
```

7.2 jbkWPava

```
dyn.load("jbkPava.so")
```

```
jbkPava <- function (x, w = rep(1, length(x))) {
  h <-
    .C("jbkPava",
       x = as.double(x),
       w = as.double (w),
       n = as.integer(length(x)))
  return (h$x)
}

jbkWPava <-
  function (y,
            w,
            d,
            eps = 1e-6,
            itmax = 100,
            verbose = FALSE) {
    n <- length (d)
    h <- (w %*% y) / d
    xold <- 1:n
    itel <- 1
    fold <- sum (y * (w %*% y))
    repeat {
      g <- h + xold - (w %*% xold) / d
      xnew <- jbkPava (g, d)
      fnew <- sum ((xnew - y) * (w %*% (xnew - y)))
      if (verbose) {
        cat (
          "itel ",
          formatC(itel, width = 3, format = "d"),
          " fold ",
          formatC(
            fold,
            width = 8,
            digits = 4,
            format = "f"
          ),
          " fnew ",
          formatC(
            fnew,
            width = 8,
            digits = 4,
            format = "f"
          ),
          "\n"
```

```
      )
    }
    if (((fold - fnew) < eps) || (itel == itmax))
      break
    fold <- fnew
    xold <- xnew
    itel <- itel + 1
  }
  return (list (x = xnew, itel = itel, f = fold))
}
```

References

De Leeuw, J. 2016. *Block Relaxation Methods in Statistics*. Bookdown. https://bookdown.org/jandeleeuw6/bras/.

———. 2017. "Exceedingly Simple Monotone Regression." doi:10.13140/RG.2.2.13636.63369.

De Leeuw, J., and P. Mair. 2009. "Multidimensional Scaling Using Majorization: SMACOF in R." *Journal of Statistical Software* 31 (3): 1–30. http://www.stat.ucla.edu/~deleeuw/janspubs/2009/articles/deleeuw_mair_A_09c.pdf.

Groenen, P.J.F., P. Giaquinto, and H.A.L. Kiers. 2003. "Weighted Majorization Algorithms for Weighted Least Squares Decomposition Models." EI 2003-09. Rotterdam, Netherlands: Econometric Institute, Erasmus University.

Grone, R., S. Pierce, and W. Watkins. 1990. "Extremal Correlation Matrices." *Linear Algebra and Its Applications* 134: 63–70.

Heiser, W.J. 1995. "Convergent Computing by Iterative Majorization: Theory and Applications in Multidimensional Data Analysis." In *Recent Advances in Descriptive Multivariate Analysis*, edited by W.J. Krzasnowski. Oxford, England: Clenrendon Press.

Hürlimann, W. 2015. "Extreme Points of the N-Dimensional Elliptope: Application to Universal Copulas." *Theoretical Mathematics and Applications* 5 (3): 51–62.

Jamshidian, M., and P.M. Bentler. 1998. "A Quasi-Newton Method for Minimum Trace Factor Analysis." *Journal of Statistical Computation and Simulation* 62 (1-2): 73–89.

Kiers, H.A.L. 1997. "Weighted Least Squares Fitting Using Iterative Ordinary Least Squares Algorithms." *Psychometrika* 62: 251–66.

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

Laurent, M., and S. Poljak. 1996. "On the Facial Structure of the Set of Correlation Matrices." *SIAM Journal Matrix Analysis and Applications* 17 (3): 530–47.

Li, C.-K., and B.-S. Tam. 1994. "A Note on Extreme Correlation Matrices." *SIAM Journal*

Matrix Analysis and Applications 15 (3): 903–8.

Parthasarathy, K.R. 2002. "On Extremal Correlations." *Journal of Statistical Planning and Inference* 103: 173–80.

Vandenberghe, L., and S. Boyd. 1996. "Semidefinite Programming." *SIAM Review* 38 (1): 49–95.

Watson, G.A. 1992. "Algorithms for Minimum Trace Factor Analysis." *SIAM Journal Matrix Analysis and Applications* 13 (4): 1039–53.

Ycart, B. 1985. "Extreme Points in Convex Sets of Symmetric Matrices." *Proceedings of the American Mathematical Society* 95 (4): 607–12.