Variations on a Theme by Eckart and Young

Jan de Leeuw Jan Graffelman

September 27, 2025

TBD

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/glspca

1 Introduction

X is a "tall" matrix of rank r with n rows and m columns, $r \leq m \leq n$. We want to approximate A in the least-squarese sense by the product of two matrices A and B, where A is $n \times p$ and B is $m \times p$. Thus we want to find A and B in such a way that the sum-of-squares

$$\sigma(A, B) = \text{SSQ}(X - AB')$$

is minimized.

$$X \Dashv [$$

$$\sigma(A) = \text{SSQ}(C - AA')$$

2 Weighting

2.1 Elementwise Weighting

2.2 Kronecker Weighting

2.3 General Weighting

3 Intercept

4 GLS Loss

$$\sigma(A, B, D) = \text{tr}\, U(X - D - AB')V(X - D - AB')'$$

- X is $n \times m$ with $m \leq n$, completely known,
- U is positive definite of order n, completely known,
- V is positive definite of order m, completely known,
- A is $n \times p$ with $p \leq m$, to be estimated,
- B is $m \times p$ with $p \leq m$, to be estimated,
- D is $n \times m$, constrained to be in $\mathcal{D} \subseteq \mathbb{R}^{n \times m}$.

Example for $\mathcal{D}$: $\mu + \alpha_i + \beta_j$ but can be nonlinear.

Note: Correspondence analysis is a special case with $\mathcal{D}$ equal to zero, but more generally D can be used in correspondence analysis to impute missing data. If there is no D then A and B can be found from a single SVD, no matter what U and V are.

Note: In ordinary multinormal analysis U is the identity. But in analyzing multivariate time series data (time points are rows of X) we need U. For spacetime data we need a third weighting matrix in a triple Kronecker product.

Minimize loss over A, B, and $D \in \mathcal{D}$. Let

$$H(D) := U^{\frac{1}{2}}(X - D)V^{\frac{1}{2}}$$

then

$$\sigma_\star(D) = \min_{A,B} \sigma(A, B, D) = \sum_{s=p+1}^{m} \lambda_s^2(H(D)).$$

with λ_s the $m - p$ smallest singular values.

Note: if there are constraints on A and/or B (as in canonical correspondence analysis) projection becomes more complicated. But we can always set $p = 0$ (there are no A and B) and put all unknowns into D.

Note: If there is no D then A and B can be found with a single SVD.

Note: Both U and V are supposed to be completely known. If they must be estimated we may run into Anderson-Rubin.

5 Symmetric case

$$\sigma(A, D) = \operatorname{tr} W(C - D - AA')W(C - D - AA')'$$

$$H(D) := W^{\frac{1}{2}}(C - D)W^{\frac{1}{2}}$$

$$\sigma_\star(D) = \min_{A} \sigma(A, D) = \sum_{s=p+1}^{m} \lambda_s^2(H(D)).$$

with λ_s the $m - p$ smallest eigenvalues.

Example for D: diagonal matrix of uniquenesses

6 Algorithm

Use majorization for the initial estimate – reduce to a sequence of ULS problems. Then use optim() or Newton to minimize $\sigma_\star$ over $D \in \mathcal{D}$, using the eigen/singular value derivatives of deleeuw(2025). Majorization is usually still feasible if there are linear constraints on A/B (Takane).

7 Majorization

We first transform using $x := \text{vec}(X)$ and $y := \text{vec}(Y)$.

$$\sigma(Y) = \text{tr}\, U(X-Y)V(X-Y)' = (x-y)'(V \otimes U)(x-y).$$

Now suppose $\tilde{y}$ is the previous solution. Then

$$\sigma(y) = ((x-\tilde{y}) - (y-\tilde{y}))'(V \otimes U)((x-\tilde{y}) - (y-\tilde{y}))$$

Thus

$$\sigma(y) = \sigma(\tilde{y}) - 2(y-\tilde{y})'(V \otimes U)(x-\tilde{y}) + (y-\tilde{y})'(V \otimes U)(y-\tilde{y}),$$

and, with λ_{max} the largest eigenvalue of $U \otimes V$,

$$\sigma(y) \leq \sigma(\tilde{y}) - 2(y-\tilde{y})'(V \otimes U)(x-\tilde{y}) + \lambda_{\text{max}}(y-\tilde{y})'(y-\tilde{y}),$$

with equality if $y = \tilde{y}$. Define

$$g := \tilde{y} + \lambda_{\text{max}}^{-1}(V \otimes U)(x-\tilde{y}).$$

. Then

$$\sigma(y) \leq \sigma(\tilde{y}) + \lambda_{\text{max}}(y-g)'(y-g) - \lambda_{\text{max}} g'g$$

Now

$$(V \otimes U)(x-\tilde{y}) = \text{vec}(U(X-\tilde{Y})V)$$

so that the majorization step can also be written as the minimization of SSQ$(Y-G)$ with

$$G = \tilde{Y} + \lambda_{\text{max}}^{-1}\, U(X-\tilde{Y})V$$

Remember that

$$\lambda_{\text{max}}(V \otimes U) = \lambda_{\text{max}}(V)\lambda_{\text{max}}(U)$$

8 Code

To check the majorization result we analyze a simple example. The R function is glsAdd(), which can fit one of three types of "models".

1. $\mu + \alpha_i + \beta_j$
2. $\sum_{s=1}^{p} a_{is} b_{js}$
3. $\mu + \alpha_i + \beta_j + \sum_{s=1}^{p} a_{is} b_{js}$

We first generate some random matrices for X, U, and V.

```
set.seed(12345)
x <- matrix(rnorm(40), 10, 4)
u <- crossprod(matrix(rnorm(100), 10, 10)) / 10
v <- crossprod(matrix(rnorm(16), 4, 4)) / 4
```

The R code is

```
library("RSpectra")

ulsPCA <- function(x, p) {
  s <- svd(x, nu = p, nv = p)
  a <- s$u
  b <- s$v %*% diag(s$d[1:p])
  return(list(a = a, b = b, ab = tcrossprod(a, b)))
}

ulsAdd <- function(x) {
  m <- mean(x)
  r <- apply(x, 1, mean) - m
  s <- apply(x, 2, mean) - m
  return(list(
    m = m,
    r = r,
    s = s,
    rs = outer(r, s, "+") + m
  ))
}

ulsBoth <- function(x, p) {
```

```r
  h1 <- ulsAdd(x)
  h2 <- ulsPCA(x - h1$rs, p)
  return(list(
    m = h1$m,
    r = h1$r,
    s = h1$s,
    a = h2$a,
    b = h2$b,
    y = h1$rs + h2$ab
  ))
}

glsLoss <- function(x, y, u, v) {
  d <- x - y
  return(sum(v * crossprod(d, (u %*% d))))
}

glsAdd <- function(x,
                   u,
                   v,
                   type = 3,
                   p = 2,
                   itmax = 10000,
                   eps = 1e-6,
                   verbose = FALSE) {
  yold <- switch(type,
                 ulsAdd(x)$rs,
                 ulsPCA(x, p)$ab,
                 ulsBoth(x, p)$y
  )
  sold <- glsLoss(x, yold, u, v)
  lbdm <- eigs_sym(u, 1)$values * eigs_sym(v, 1)$values
  itel <- 1
  repeat {
    d <- x - yold
    g <- yold + u %*% d %*% v / lbdm
    ynew <- switch(type,
          ulsAdd(g)$rs,
          ulsPCA(g, p)$ab,
          ulsBoth(g, p)$y
```

```
    )
    snew <- glsLoss(x, ynew, u, v)
    if (verbose) {
      cat(
        "itel ",
        formatC(itel, format = "d"),
        "sold ",
        formatC(sold, digits = 10, format = "f"),
        "snew ",
        formatC(snew, digits = 10, format = "f"),
        "\n"
      )
    }
    if ((itel == itmax) || ((sold - snew) < eps)) {
      break
    }
    itel <- itel + 1
    sold <- snew
    yold <- ynew
  }
  return(list(y = ynew, loss = snew, itel = itel))
}
```

Apply our code.

```
h1 <- glsAdd(x, u, v, type = 1)
h2 <- glsAdd(x, u, v, type = 2)
h3 <- glsAdd(x, u, v, type = 3)
```

- For type 1 we have convergence in 291 iterations to loss 31.1720174.
- For type 2 we have convergence in 1275 iterations to loss 0.7924819.
- For type 3 we have convergence in 2427 iterations to loss 0.1039566.

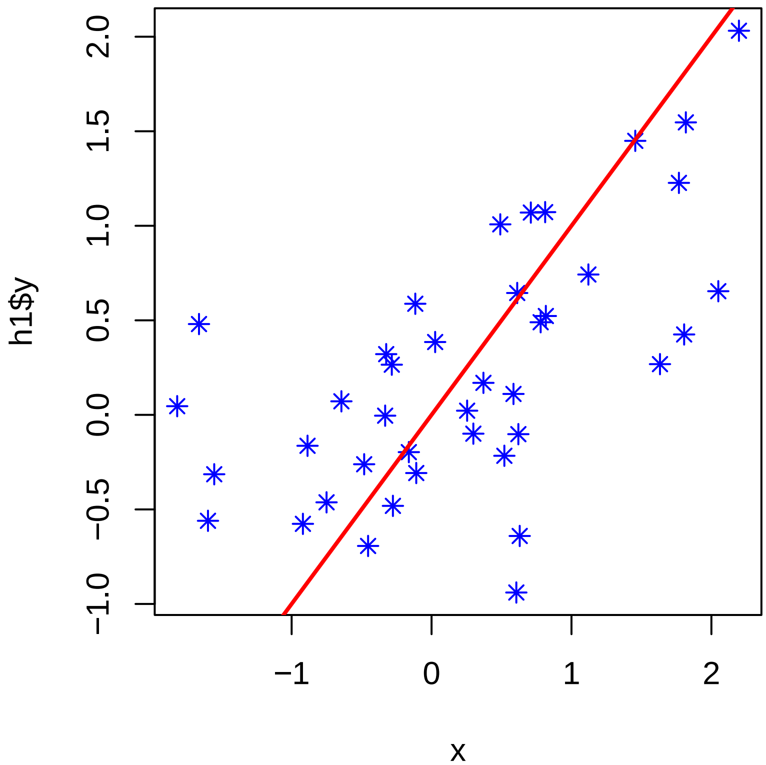

Figure 1: Type 1

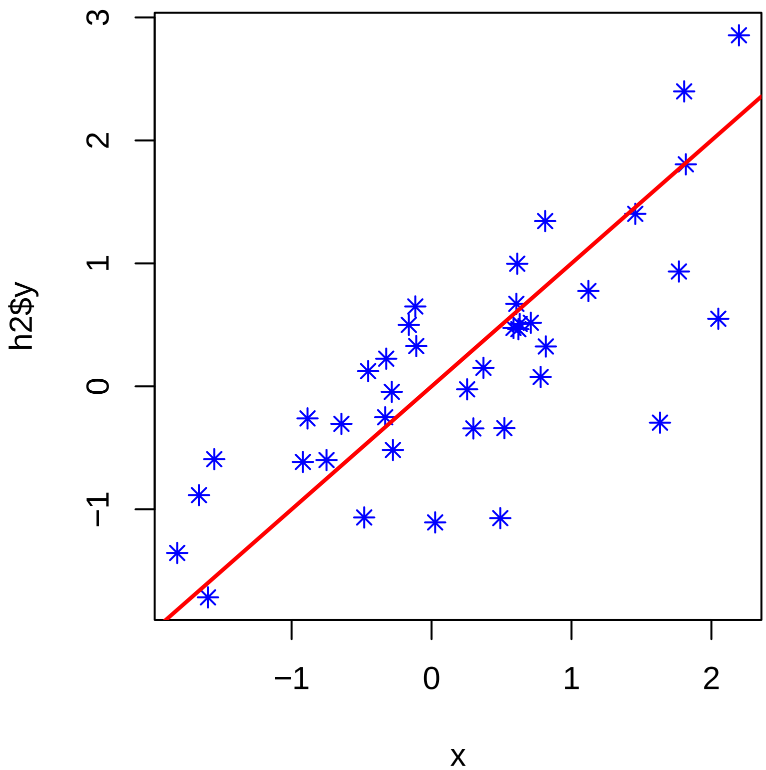

Figure 2: Type 2

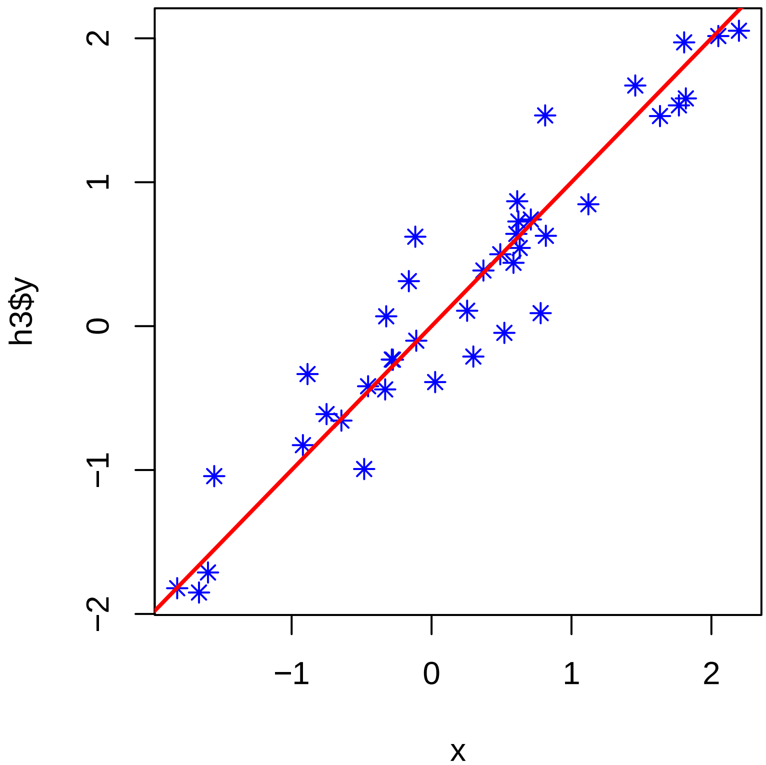

Figure 3: Type 3

9 Generalization

A far-reaching generalization of type 3 defines $D = EMF' + AB'$, with E and Q known matrices. In the unweighted case (and thus in the update step of the majorization algorithm) this amounts to an SVD of the residuals $P_E X P_F$, where P_E and P_F are projectors on the null spaces of E and F.

10 Correspondence Analysis

$$\sigma(A, B) = \text{tr } E^{-1}(F - AB')D^{-1}(F - AB')'$$

CCA

$$\sigma(A, B) = \text{tr } (Z'EZ)^{-1}(Z'F - AB')D^{-1}(Z'F - AB')'$$

Canonical Analysis

$$\sigma(A,B) = \text{tr}\ (X'X)^{-1}(X'Y - AB')(Y'Y)^{-1}(X'Y - AB')'$$

11 Redundancy Analysis

$$\sigma(A,B) = \text{tr}\ (X'X)^{-1}(X'Y - AB')(X'Y - AB')'$$

12 Aside

$$\sigma(A,B) = \mathbf{SSQ}(X - GAB'H')$$

with G and H known.

$\sigma(A,B) = \text{tr}\,X'X - 2\text{tr}\,X'GAB'H' + \text{tr}\,HBA'G'GAB'H' = \text{tr}\,X'X - 2\text{tr}\,H'X'GAB' + \text{tr}\,(H'H)BA'(G$

Let $\tilde{B} = (H'H)^{\frac{1}{2}}B$ and $\tilde{A} = (G'G)^{\frac{1}{2}}A$.

Then

$$\text{tr}\ H'X'GAB' = \text{tr}\ (H'H)^{-\frac{1}{2}}H'X'G(G'G)^{-\frac{1}{2}}\tilde{A}\tilde{B}'$$

$\text{tr}\,(H'H)BA'(G'G)AB' = \text{tr}\,(H'H)(H'H)^{-\frac{1}{2}}\tilde{B}\tilde{A}'(G'G)^{-\frac{1}{2}}(G'G)(G'G)^{-\frac{1}{2}}\tilde{A}\tilde{B}'(H'H)^{-\frac{1}{2}} = \text{SSQ}(\tilde{A}\tilde{B}'$

Thus

$$\min_{A,B}\sigma(A,B) = \min_{A,B}\text{tr}(H'H)^{-1}(H'XG - AB')(G'G)^{-1}(H'XG - AB')'$$

which can be solved by an SVD of $(H'H)^{-\frac{1}{2}}H'X'G(G'G)^{-\frac{1}{2}}$

13 Aside 2

This generalizes type 3.

$$\sigma(A,B,S) = \text{tr}\,U(X - GSH' - AB')V(X - GSH' - AB')'$$

Again it can be solved with a single SVD.

13.1 Aside 3

Suppose U and V are singular. Let

$$U = \begin{bmatrix} K & K_\perp \end{bmatrix} \begin{bmatrix} \Phi^2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K' \\ K'_\perp \end{bmatrix}$$

$$V = \begin{bmatrix} L & L_\perp \end{bmatrix} \begin{bmatrix} \Psi^2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L' \\ L'_\perp \end{bmatrix}$$

Let $A = KP + K_\perp Q$ and $B = LR + L_\perp S$.

$$\text{SSQ}(U^{\frac{1}{2}}(X - AB')V^{\frac{1}{2}}) = \text{SSQ}(\Phi K'XL\Psi - \Phi PR'\Psi)$$

which is minimizwed by the SVD of $\Phi K'XL\Psi$.

14 Elementwise weights

Use vec()

$$\sigma(Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}(x_{ij} - y_{ij})^2$$

Now any $V \geq W$ (elementwise) can be used to majorize.

$$\sigma(Y) \leq \sigma(\tilde{Y}) - 2\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}(y_{ij} - \tilde{y}_{ij})(x_{ij} - \tilde{y}_{ij}) + \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}(y_{ij} - \tilde{y}_{ij})^2$$

which leads to the majorization step of minimizing

$$\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}(z_{ij} - y_{ij})^2$$

with

$$z_{ij} = \tilde{y}_{ij} + \frac{w_{ij}}{v_{ij}}(x_{ij} - \tilde{y}_{ij})$$

In Groenen, Giaquinto, and Kiers (2003) $v_{ij} = \max_{j=1}^{m} w_{ij}$. In (**?**)?? $v_{ij} = \theta_i\xi_j$, where θ and xi are chosen such that $\log\theta_i + \log\xi_j \geq \log w_{ij}$ and $\sum_{i=1}^{n}\sum_{j=1}^{m}(\log\theta_i + \log\xi_j)$ is minimized (a linear programming problem)

Groenen, P. J. F., P. Giaquinto, and H. A. L Kiers. 2003. "Weighted Majorization Algorithms for Weighted Least Squares Decomposition Models." Econometric Institute Report EI 2003-09. Econometric Institute, Erasmus University Rotterdam. https://repub.eur.nl/pub/1700.