Derivatives of Low Rank PSD Approximation

Jan de Leeuw

Version 01, November 23, 2016

Abstract

In De Leeuw (2008) we studied the derivatives of the least squares rank p approximation in the case of general rectangular matrices. We modify these results for the symmetric positive semi-definite case, using basically the same derivation. We apply the formulas to compute an expression for the convergence rate of Thomson's iterative principal component algorithm for factor analysis.

Contents

Note: This is a working paper which will be expanded/updated frequently. All suggestions for improvement are welcome. The directory deleeuwpdx.net/pubfolders/rank has a pdf version, the complete Rmd file with all code chunks, the bib file, and the R source code.

1 Introduction

We have a symmetric matric B that we want to approximate, in the least squares sense, by a symmetric positive semi-definite matrix of rank less than or equal to p. This is a problem that arises in various forms of multivariate analysis, specifically in factor analysis and multidimensinal scaling. A brief history is in De Leeuw (1974).

Thus the problem is to minimize

$$\sigma(X) := \mathbf{SSQ}\ (B - XX') \tag{1}$$

over $X \in \mathbb{R}^{n\times p}$. The stationary equations are

$$BX = X(X'X). \tag{2}$$

Clearly the loss function is in variant under rotation, which means we can require without loss of generality that $X'X$ is diagonal. Thus each column of the optimal X is either zero or an eigenvector corresponding with a positive eigenvalue of B. At stationary points $\sigma(X) = \mathbf{SSQ}(B) - \mathbf{tr}\ X'BX$. If the signature of B is (n_+, n_-, n_0) then the optimal X has $\min(n_+, p)$ non-zero columns equal to $\sqrt{\lambda_s}k_s$, where the k_s are normalized eigenvectors corresponding to positive eigenvalues λ_s, and $\max(0, p - n_+)$ zero columns. Although the optimal X is never unique, the optimal XX' is unique if and only if $\lambda_p > \lambda_{p+1}$.

If $B = K\Lambda K'$ is a complete eigen-decomposition, with eigenvalues in decreasing order along the diagonal, then

$$\Gamma_p(B) = K_p\Lambda_p^{\frac{1}{2}} \tag{3}$$

where the k_s are the normalized eigenvectors corresponding with the p largest eigenvalues. To guarantee differentiability we assume throughout that $\lambda_p > \lambda_{p+1}$ as well as $\lambda_p > 0$.

2 Derivative

2.1 Formula

Theorem 1 gives an expression for the first order term in the Taylor expansion

$$\Gamma_p(B + E) = \Gamma_p(B) + \mathcal{D}\Gamma_p(B)(E) + o(\|E\|). \tag{4}$$

Note that Γ_p is defined on the space of real doubly-centered symmetric matrices $\mathcal{S}^{n\times n}$, with values in that same space. The derivative $\mathcal{D}\Gamma_p(B)$ at B is a linear operator from $\mathcal{S}^{n\times n}$ to $\mathcal{S}^{n\times n}$, which we apply to the real symmetric and doubly-centered perturbation E. The symmetry makes our results different from those of De Leeuw (2008). Define the matrix $\Xi := K'EK$ with elements ξ_{st}.

Theorem 1: [Projection Derivative]

$$\mathcal{D}\Gamma_p(B)(E) = \sum_{s=1}^{p} \xi_{ss}k_sk_s' - \sum_{s=1}^{p}\sum_{\substack{t=1\\t\neq s}}^{n} \frac{\lambda_s}{\lambda_t - \lambda_s}\xi_{st}(k_tk_s' + k_sk_t'). \tag{5}$$

Proof: From De Leeuw (2007) (and many other places) we know that

$$\begin{aligned}\lambda_s(B+E)&=\lambda_s+\xi_{ss}+o(\|E\|),\\ k_s(B+E)&=k_s-(B-\lambda_sI)^+Ek_s+o(\|E\|).\end{aligned}$$

Thus

$$\begin{aligned}\Gamma_p(B+E)&=\sum_{s=1}^p(\lambda_s+\xi_{ss})(k_s-(B-\lambda_sI)^+Ek_s)(k_s-(B-\lambda_sI)^+Ek_s)'+o(\|E\|)=\\ &=\Gamma_p(B)+\sum_{s=1}^p\left\{\xi_{ss}k_sk_s'-\lambda_sk_sk_s'E(B-\lambda_sI)^+-\lambda_s(B-\lambda_sI)^+Ek_sk_s'\right\}+o(\|E\|)=\\ &=\Gamma_p(B)+\sum_{s=1}^p\left\{\xi_{ss}k_sk_s'-\sum_{\substack{t=1\\t\neq s}}^n\frac{\lambda_s}{\lambda_t-\lambda_s}\xi_{st}(k_sk_t'+k_tk_s')\right\}+o(\|E\|).\end{aligned}$$

QED

Note that if B and E are SDC, then so is $\mathcal{D}\Gamma_p(B)(E)$.

We generate some random data to illustrate theorem 1.

```
set.seed(12345)
eps <- .001
b <- crossprod (matrix (rnorm(400), 100, 4)) / 100
e <- eps * crossprod (matrix (rnorm(400), 100, 4)) / 100
```

Then compute $\Gamma_p(B+E)$.

```
##            [,1]          [,2]          [,3]          [,4]
## [1,]  1.2781679170  0.1413226188  0.0155429929  0.2557498035
## [2,]  0.1413226188  0.3754844674 -0.3835591107  0.4570154983
## [3,]  0.0155429929 -0.3835591107  0.4126808466 -0.4559121477
## [4,]  0.2557498035  0.4570154983 -0.4559121477  0.5619744567
```

Then compute the zero-order approximation $\Gamma_p(B)$.

```
##            [,1]          [,2]          [,3]          [,4]
## [1,]  1.2773620868  0.1408232165  0.0156044480  0.2562189505
## [2,]  0.1408232165  0.3753719549 -0.3833537174  0.4567557210
## [3,]  0.0156044480 -0.3833537174  0.4122604048 -0.4554195169
## [4,]  0.2562189505  0.4567557210 -0.4554195169  0.5616655283
```

And then compute the first-order approximation $\Gamma_p(B)+\mathcal{D}\Gamma_p(B)(E)$, using the program `perturb()` from the appendix.

```
##              [,1]          [,2]          [,3]          [,4]
## [1,]  1.2781680631  0.1413211691  0.0155430195  0.2557508933
## [2,]  0.1413211691  0.3754850527 -0.3835595774  0.4570158930
## [3,]  0.0155430195 -0.3835595774  0.4126805687 -0.4559116839
## [4,]  0.2557508933  0.4570158930 -0.4559116839  0.5619738470
```

Clearly the first order approximates is better than the zero order one . For zero order the sum of the absolute values of the approximation errors is 0.0056233246, and for first order it is 0.0000094019.

2.2 Structure

To get more insight into the structure of the derivative we look at its eigenvalues and eigenvectors, i.e. those SDC perturbations E for which $\mathcal{D}\Gamma_p(B)(E) = \omega E$ for some ω. We investigate, again following De Leeuw (2008), perturbation matrices E of the form $E = k_\alpha k_\beta' + k_\beta k_\alpha'$, with the k_α the eigenvectors of B. It suffices to look at those $\frac{1}{2}n(n-1)$ perturbations for which $\alpha < \beta$. Note they are orthogonal and consequenty are a basis for the space of SDC matrices. Also

$$\xi_{st} = k_s' E k_t = k_s' k_\alpha k_\beta' k_t + k_s' k_\beta k_\alpha' k_t = \delta^{s\alpha}\delta^{t\beta} + \delta^{s\beta}\delta^{t\alpha}. \tag{6}$$

Theorem 2: [Projection Eigen Structure] For $\alpha < \beta$

$$\mathcal{D}\Gamma_p(B)(k_\alpha k_\beta' + k_\beta k_\alpha') = \begin{cases} k_\alpha k_\beta' + k_\beta k_\alpha' & \text{if } 1 \leq \alpha < \beta \leq p, \\ \frac{\lambda_\alpha}{\lambda_\alpha - \lambda_\beta}(k_\alpha k_\beta' + k_\beta k_\alpha') & \text{if } 1 \leq \alpha \leq p < p+1 \leq \beta \leq n, \\ 0 & \text{if } p+1 \leq \alpha < \beta \leq n. \end{cases} \tag{7}$$

Proof: From (6) we have $\xi_{ss} = 0$ if $\alpha < \beta$. Thus the first term on the rhs of (5) is zero. It remains to evaluate

$$\sum_{s=1}^{p}\sum_{\substack{t=1\\t\neq s}}^{n} \frac{\lambda_s}{\lambda_t - \lambda_s}(\delta^{s\alpha}\delta^{t\beta} + \delta^{s\beta}\delta^{t\alpha})(k_s k_t' + k_t k_s'). \tag{8}$$

If $\alpha > p$ then $\beta > p$ and thus (8) is zero. If $\alpha <= p$ and $\beta > p$ then $\delta^{s\beta} = 0$ and

$$\sum_{s=1}^{p}\sum_{\substack{t=1\\t\neq s}}^{n} \frac{\lambda_s}{\lambda_t - \lambda_s}(\delta^{s\alpha}\delta^{t\beta} + \delta^{s\beta}\delta^{t\alpha})(k_s k_t' + k_t k_s') = \frac{\lambda_\alpha}{\lambda_\beta - \lambda_\alpha}(k_\alpha k_\beta' + k_\beta k_\alpha').$$

If $\alpha < \beta \leq p$ then

$$\sum_{s=1}^{p}\sum_{\substack{t=1\\t\neq s}}^{n} \frac{\lambda_s}{\lambda_t - \lambda_s}(\delta^{s\alpha}\delta^{t\beta} + \delta^{s\beta}\delta^{t\alpha})(k_s k_t' + k_t k_s') = \frac{\lambda_\alpha}{\lambda_\beta - \lambda_\alpha}(k_\alpha k_\beta' + k_\beta k_\alpha') + \frac{\lambda_\beta}{\lambda_\alpha - \lambda_\beta}(k_\alpha k_\beta' + k_\beta k_\alpha') = k_\alpha k_\beta' + k_\beta k_\alpha'.$$

QED

Corollary 1: [**Projection Eigen Values**] The derivative $\mathcal{D}\Gamma_p(B)$ has $\frac12 p(p-1)$ eigenvalues equal to $+1$, $\frac12(n-p)(n-p-1)$ eigenvalues equal to zero, and $p(n-p)$ eigenvalues equal to $\frac{\lambda_\alpha}{\lambda_\alpha-\lambda_\beta}$, where $1\leq\alpha<\beta\leq n$ and the λ_α are the ordered eigenvalues of B.

Note that if B is positive definite then the $p(n-p)$ eigenvalues for $\alpha\leq p$ and $\beta>p$ are larger than one, if B is positive semi-definite of rank p they are equal to one.

3 Thomson's Factor Analysis Algorithm

Thomson (1934) proposed an alternating least squares algorithm to find factor analysis solutions. We minimize the sum of squares

$$\sigma(X,\Delta)=\mathbf{SSQ}(B-\Delta-XX')$$

over loadings X and the diagonal matrix of uniquenesses Δ. The algorithm starts with $\Delta^{(0)}$ and alternates

$$\begin{aligned}C^{(k)}&=\Gamma_p(B-\Delta^{(k)}),\\ \Delta^{(k+1)}&=\mathbf{diag}(B-C^{(k)}).\end{aligned}$$

By substituting the first step into the second step we can think of the algorithm as a map τ that updates uniquenesses in a vector $\delta^{(k)}$ to a vector $\delta^{(k+1)}$. A straightforward application of theorem 1 shows

$$\mathcal{D}_j\tau_i(\delta)=\sum_{s=1}^p\left\{k_{is}^2k_{js}^2-2\sum_{\substack{t=1\\t\neq s}}^n\frac{\lambda_s}{\lambda_t-\lambda_s}k_{is}k_{it}k_{js}k_{jt}\right\}.$$

using the eigenvalues and eigenvectors of $B-\Delta$.

As an example we use the Harman.Burt data from the `psych` package ((**revelle_16?**)).

```r
data(Harman, package = "psych")
h3 <- thomson (Harman.Burt, p = 3, verbose = FALSE, eps = 1e-15, itmax = 1000)
h2 <- thomson (Harman.Burt, p = 2, verbose = FALSE, eps = 1e-15, itmax = 1000)
h1 <- thomson (Harman.Burt, p = 1, verbose = FALSE, eps = 1e-15, itmax = 1000)
```

In one dimension we need 21 iterations to get to loss function value 0.5514339263, with an estimated convergence rate of 0.4818706226. In two dimensions 107 iterations get us to loss 0.1486134802 with estimated rate 0.8746643617. And in three dimensions 435 iterations get us to 0.0312374332 and rate 0.9702874395. The estimated convergence rates correspond closely to the largest eigenvalues of the Jacobian, computed by the function `tderivative`.

```r
mprint (eigen (tderivative (Harman.Burt, h1$u, p=1))$values)
```

```
## [1]    0.4819142774    0.3684002138    0.2782070726    0.2446414810
## [5]    0.1830500694    0.1244703151    0.1022921260    0.0597542139
```

```r
mprint (eigen (tderivative (Harman.Burt, h2$u, p=2))$values)
```

```
## [1]    0.8746643733    0.6956043771    0.4674777799    0.4166336265
## [5]    0.3547709022    0.2311944490    0.1756869942    0.0967558960
```

```r
mprint (eigen (tderivative (Harman.Burt, h3$u, p=3))$values)
```

```
## [1]    0.9702874840    0.8848642535    0.8207644608    0.6522238451
## [5]    0.4197176966    0.3589311780    0.2335541555    0.1622049624
```

4 Appendix: Code

4.1 auxilary.R

```r
ei <- function (i, n) {
  return (ifelse(i == (1:n), 1, 0))
}

aij <- function (i, j, n) {
  u <- ei(i, n) - ei (j, n)
  return (outer (u, u))
}

kdelta <- function (i, j) {
  ifelse (i == j, 1 , 0)
}

mprint <- function (x, d = 10, w = 15) {
  print (noquote (formatC (
    x, di = d, wi = w, fo = "f"
  )))
}

mnorm <- function (x, norm = 2) {
  return (sum (abs (x) ^ norm) ^ (1 / norm))
}

basis <- function (i, j, n) {
  s <- sqrt (2) / 2
  a <- matrix (0, n, n)
  if (i == j)
    a[i, i] <- 1
  else {
    a[i, j] <- a[j, i] <- s
  }
  return (a)
}

doubleCenter <- function (x) {
  n <- nrow (x)
  j <- diag(n) - (1 / n)
  return (j %*% x %*% j)
}

center <- function (x) {
  return (apply (x, 2, function (z) z - mean (z)))
}

squareDist <- function (x) {
  d <- diag (x)
  return (outer (d, d, "+") - 2 * x)
}

lrank <- function (x, p = 2) {
  e <- eigen (x)
  v <- e$vectors[, 1:p]
  return (tcrossprod (x %*% v, v))
}

mpower <- function (x, p, eps = 1e-16) {
  z <- eigen (x, symmetric = TRUE)
  zval <- z$values[1:(nrow(x) - 1)] ^ p
  zvec <- z$vectors[, 1:(nrow(x) - 1)]
  return (zvec %*% diag (zval) %*% t (zvec))
}
```

4.2 thomson.R

```r
thomson <-
  function (b,
            uold = rep (0, nrow (b)),
            p = 2,
            itmax = 1000,
            eps = 1e-10,
            verbose = TRUE) {
    bold <- b - diag (uold)
    cold <- lrank (bold, p)
    fold <- sum ((bold - cold) ^ 2)
    eold <- Inf
    itel <- 1
    repeat {
      unew <- diag (b - cold)
      enew <- mnorm (uold - unew)
      rnew <- enew / eold
      bnew <- b - diag (unew)
      cnew <- lrank (bnew, p)
      fnew <- sum ((bnew - cnew) ^ 2)
      if (verbose) {
        cat (
          formatC (itel, width = 4, format = "d"),
          formatC (
            fold,
            digits = 10,
            width = 15,
            format = "f"
          ),
          formatC (
            fnew,
            digits = 10,
            width = 15,
            format = "f"
          ),
          formatC (
            enew,
            digits = 10,
            width = 15,
            format = "f"
          ),
          formatC (
            rnew,
            digits = 10,
            width = 15,
            format = "f"
          ),
          "\n"
        )
      }
      if ((itel == itmax) || (fold - fnew) < eps)
        break
      itel <- itel + 1
      uold <- unew
      fold <- fnew
      cold <- cnew
      bold <- bnew
      eold <- enew
    }
    return (list (
      u = unew,
      f = fnew,
      e = enew,
      r = rnew,
      itel = itel
    ))
  }

tmap <- function (b, u, p = 2) {
  c <- lrank (b - diag (u))
  return (diag (b - c))
}

tderivative <- function (b, u, p = 2) {
  n <- length (u)
  e <- eigen (b - diag (u))
  k <- e$vectors
  l <- e$values
  m <- matrix (0, n, n)
  for (i in 1:n)
    for (j in 1:n)
      for (s in 1:p) {
          m[i,j] <- m[i,j] + (k[i,s] ^ 2) * (k[j, s] ^ 2)
        for (t in 1:n) {
          if (s == t)
            next
          xi <- l[s] / (l[t] - l[s])
          m[i, j] <-
            m[i, j] - 2 * xi * k[i, s] * k[i, t] * k[j, s] * k[j, t]
        }
      }
  return (m)
}
```

References

De Leeuw, J. 1974. "Approximation of a Real Symmetric Matrix by a Positive Semidefinite Matrix of Rank r." Technical Memorandum TM-74-1229-10. Murray Hill, N.J.: Bell Telephone Laboratories.

———. 2007. "Derivatives of Generalized Eigen Systems with Applications." Preprint Series 528. Los Angeles, CA: UCLA Department of Statistics.

———. 2008. "Derivatives of Fixed-Rank Approximations." Preprint Series 547. Los Angeles, CA: UCLA Department of Statistics.

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