A DECOMPOSITION METHOD FOR WEIGHTED LEAST SQUARES LOW-RANK APPROXIMATION OF SYMMETRIC MATRICES

JAN DE LEEUW

Abstract. We discuss an alternating least squares algorithm that uses both decomposition and block relaxation to find the optimal positive semidefinite approxation of given rank p to a known symmetric matrix of order n. Each iteration of the algorithm involves minimizing n quartics and solving n secular equations of order p.

1. Introduction

Least squares approximation of a symmetric matrix C by a symmetric positive definite matrix $\hat{C}$ of rank at most p is a classical problem. It is typically solved by computing an eigen-decomposition of C and by truncating the eigen-decomposition by only using the eigenvectors and associated with the $\min(p, q)$ largest positive eigenvalues of C. Here q is the number of positive eigenvalues. Thus if $q \leq p$ we have $\textbf{rank}(\hat{C}) = q$ and if $q \geq p$ we have $\textbf{rank}(\hat{C}) = p$.

Eigen-decomposition cannot be used, however, if we use weighted least squares, in which the squared deviation for each matrix element (i, j) is weighted by a non-negative number w_{ij}. This has, as a special case, incorporation of missing data, where we set $w_{ij} = 0$. And an even more special case has

$$w_{ij} = \begin{cases} 1 \text{ if } i \neq j, \\ 0 \text{ if } i = j \end{cases}.$$

Date: April 16, 2006.

2000 *Mathematics Subject Classification.* 62H25.

Key words and phrases. Multivariate Analysis.

This occurs in least squares factor analysis, and special case algorithms have been developed a long time ago by Thomson (Iterative Refactoring) and Harman (MINRES).

As we shall see below, the case in which the diagonal elements are missing is indeed quite special and leads to considerable simplification. If all non-zero weights are equal, then we can we can iteratively alternate imputing the missing elements and using eigen-decomposition on the completed matrix. But this does not cover all cases. It seems useful to give an algorithm for the case of general matrices of weights, and general patterns of missing data, where the only restriction we impose is that the weights are non-negative.

2. Problem

We want to solve

$$\min_{X\in\mathbb{R}^{n\times p}} \sigma(X) = \sum_{i=1}^n\sum_{j=1}^n w_{ij}(c_{ij}-x_i'x_j)^2, \tag{1}$$

where both C and W are symmetric of order n, and the elements of W are non-negative.

We use *decomposition* to construct a simple, and hopefully efficient, algorithm. Write X as $X = \Lambda Z$, where Λ is diagonal and Z satisfies $\mathbf{diag}(ZZ') = I$. Thus Λ contains the *lengths* and Z the *directions* of the n points in X. The problem becomes

$$\min_{\lambda\geq 0, diag(ZZ')=I} \sigma(\lambda, Z) = \sum_{i=1}^n\sum_{j=1}^n w_{ij}(c_{ij}-\lambda_i\lambda_j z_i'z_j)^2. \tag{2}$$

After decomposition, the next step is *block relaxation*. We will solve the minimization problem $\mathcal{P}$, by cycling through a sequence of problems $\mathcal{P}_i$. Problem $\mathcal{P}_i$ minimizes $\sigma(\lambda, Z)$ over λ_i and z_i, while keeping the $x_j = \lambda_j z_j$ with $j \neq i$ fixed at the current best values. Let us separate out the part of the loss function that depends on point i. Clearly

$$\sigma_i(\lambda, Z) = w_{ii}(c_{ii}-\lambda_i^2)^2 + 2\sum_{j\neq i}^n w_{ij}(c_{ij}-\lambda_i\lambda_j z_i'z_j)^2.$$

By collecting terms, and ignoring parts that do not depend on the unknowns, we find that problem $\mathcal{P}_i$ is minimization of

$$f(\lambda_i, z_i) = \lambda_i^4 w_{ii} + 2\lambda_i^2(z_i' A_i z_i - w_{ii} c_{ii}) - 4\lambda_i z_i' b_i, \tag{3}$$

where

$$A_i = \sum_{j \neq i}^{n} w_{ij} x_j x_j',$$

$$b_i = \sum_{j \neq i}^{n} w_{ij} c_{ij} x_j.$$

To minimize $f(\lambda_i, z_i)$ over λ_i and z_i satisfying $z_i' z_i = 1$ we again apply block relaxation. Minimization over λ_i and z_i are alternated.

So the structure of the algorithm is clear. One iteration to update X involves solving n problems $\mathcal{P}_i$. One problem P_i involves a number of *inner iterations*, in which we alternate updating λ_i for given z_i and z_i for given λ. Depending on the iterative nature of the algorithms chosen to solve the two substeps of a single inner iteration, we may also have to use *innermost iterations*.

3. Inner Iterations

Solving for the optimal λ_i for given z_i means finding the minimum of the quartic (3). Let's first deal with the case $w_{ii} = 0$ which makes the quartic a quadratic. Decomposition is not really necessary, and we can just set

$$x_i = A_i^{-1} b.$$

If less than p of the weights in row i are positive, then A_i will be singular, and we have to use a generalized inverse. This result makes the case with a hollow weight matrix so much simpler. No inner iterations are needed.

In the general case the quartic has no cubic term, which makes the explicit formula for the location of the minimum of Jeffrey [1997, Theorem 3] especially attractive. So we use this in our algorithm.

The optimum z_i for given λ_i must be computed by minimizing

$$g(z) = z'\overline{A}z - 2z'\overline{b},$$

where $\overline{A} = \lambda_i^2 A_i$ and $\overline{b} = \lambda_i b_i$ and where we require $z'z = 1$. Using a Lagrange multiplier μ shows

$$z = (\overline{A} - \mu I)^{-1}\overline{b} \tag{4a}$$

where μ must satisfy

$$b'(A - \mu I)^{-2}b = 1 \tag{4b}$$

For the minimum we must have, in addition, $\overline{A} - \mu I$ positive definite, i.e. μ must be stricty smaller than the smallest eigenvalue ω_p of $\overline{A}$. On (∞, ω_p) the function $b'(A - \mu I)^{-2}b$ is convex and increases from zero to $+\infty$, and thus crosses the line $\mu = 1$ exactly once. The bounds on the solution can be improved [Hager, 2001, page 190]. If $\overline{A} = K\Omega K'$ is the eigen-decomposition of $\overline{A}$, and $\beta = K'\overline{b}$, then

$$\omega_p - \sqrt{\sum_{s=1}^{p}\beta_s^2} \leq \mu \leq \omega_p - \sqrt{\sum_{\omega_s=\omega_p}\beta_s^2}.$$

Our algorithm uses a standard root finding method to locate μ in this interval.

Equation (4b) is one of the *secular equations*, which have been studied in great detail. There is a huge literature, which is excellently reviewed in Conn et al. [2000, Chapter 7].

4. Example

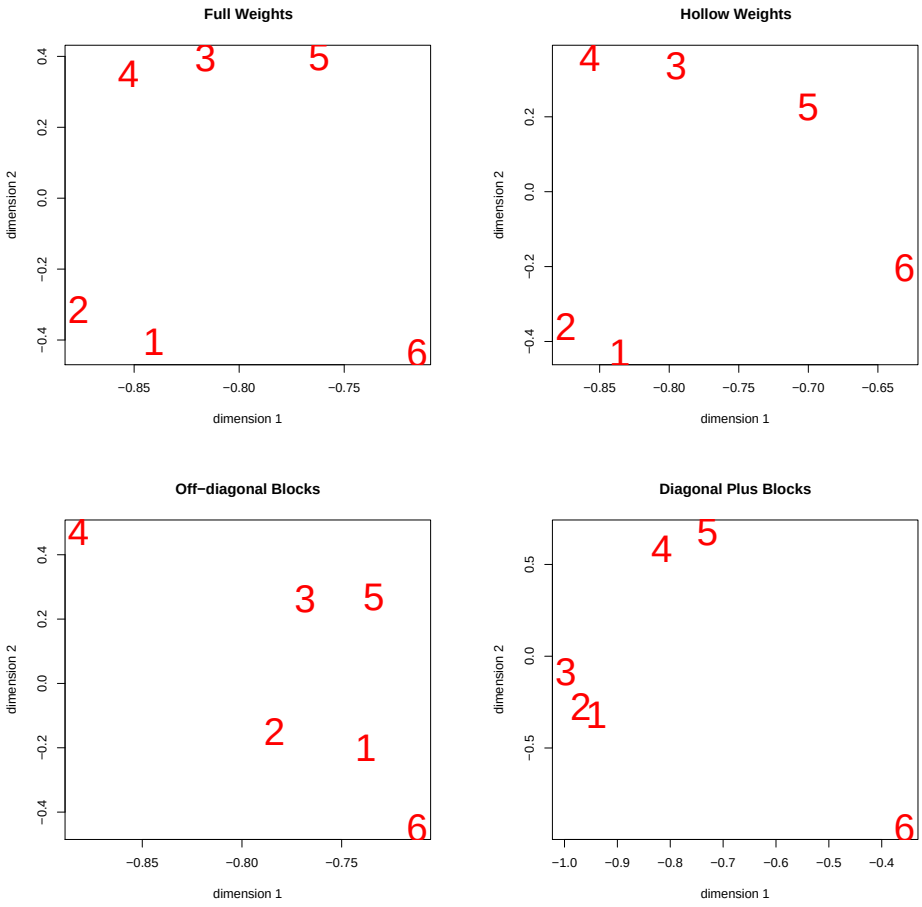
We have taken the correlation matrix in Table 1 from Spearman [1927, page 144]. Anthropometric measurements were made by Doll on 477 boys and girls.

Four different solutions were computed. In all cases we iterate until the loss function decreases less than $1e-6$. Plots of the four solutions are given in Table 2. The first solution has a full weight matrix, with all $w_{ij} = 1$. Since the program starts with the eigenvalue solution it is done in one

TABLE 1. Correlation Matrix of the Doll Data

	1	2	3	4	5	6
1. Right hand grip	1.000	0.885	0.525	0.579	0.455	0.620
2. Left hand grip	0.885	1.000	0.570	0.595	0.570	0.620
3. Standing height	0.525	0.570	1.000	0.805	0.630	0.430
4. Sitting height	0.580	0.595	0.805	1.000	0.680	0.475
5. Weight	0.455	0.570	0.630	0.680	1.000	0.390
6. Vital capacity	0.620	0.620	0.430	0.475	0.390	1.000

TABLE 2. Four Solutions of the Doll Data



iteration and gives a minimum loss of .417045. For the second solution we set the diagonal equal to zero. We now need 10 iterations and find a loss

of .007540. Third, we define a block of the first three and a block of the second three variables, and we make the within-block weights zero. This amounts to computing a singular value decomposition of the off-diagonal block. After 4 iterations, we have .007148. And finally we use the same block weights, but now make the diagonal equal to one. This finds .015852 after 12 iterations.

5. Code

```
ipSymLS<-function(
          target,
          w=matrix(1,dim(target)[1],dim(target)[2]),
          ndim=2,
          init=FALSE,
          itmax=100,
          eps=1e-6,
          verbose=FALSE) {
if (is.matrix(init)) x<-init
else {
          z<-eigen(target); x<-z$vectors[,1:ndim]; x<-x
             %*%diag(sqrt(z$values[1:ndim]))
          }
n<-dim(target)[1]; xx<-tcrossprod(x); oloss<-sum(w*(
    target-xx)^2); itel<-1
repeat {
     for (i in 1:n) {
          ai<-crossprod(x,w[i,]*x)-w[i,i]*outer(x[i,],x[
             i,])
          bi<-colSums((w[i,]*target[i,])*x)-w[i,i]
             *target[i,i]*x[i,]
          if (w[i,i] == 0) x[i,]<-solve(ai,bi)
          else {
               li<-sum(x[i,]^2); zi<-x[i,]/sqrt(li); wi
                  <-w[i,i]; ci<-target[i,i]
```

```
                a4<-wi; a3<-0; a2<-2*(sum(zi*(ai%*%zi)
                    )-(wi*ci)); a1<--4*sum(zi*bi)
                b1<-a1/(2*a4); b2<-a2/(3*wi); bb<-as.
                    complex((b1^4)+2*(b1^2)*(b2^3))
                s<-(b1^2)+(b2^3)+sqrt(bb); kf<-s^(1/3)
                    +(b2^2)*s^(-1/3)+b2; li<-Re(-b1/kf)
                x[i,]<-li*zi; aa<-2*(li^2)*ai; bb<-2
                    *li*bi; ei<-eigen(aa)
                l<-ei$values; k<-ei$vector; kb<-drop(
                    crossprod(k,bb)); ml<-min(l)
                ul<-ml-sqrt(sum(kb^2)); uu<-ml-sqrt(
                    sum(kb[which(l==ml)]^2)); u<-uu
                fu<-function(u) sum((kb/(l-u))^2)-1
                gu<-function(u) 2*sum((kb^2)/((l-u)^3)
                    )
                repeat {
                        if (abs(fu(u))<1e-6) break()
                        u<-u-fu(u)/gu(u)
                        }
                x[i,]<-li*(k%*%(kb/(l-u)))
                }
        }
    xx<-tcrossprod(x); nloss<-sum(w*(target-xx)^2)
    if (verbose)
            cat("Iteration: ",formatC(itel,digits=6,
                width=6),
                    "Previous Loss: ",formatC(
                        oloss,digits=6,width=12,
                        format="f"),
                    "Current Loss: ",formatC(nloss
                        ,digits=6,width=12,format="
                        f"),
                    "\n")
```

```
    if (((oloss-nloss) < eps) || (itel == itmax))
        break()
    oloss<-nloss; itel<-itel+1
    }
return(x)
}
```

References

Andrew R. Conn, Nicholas I.M. Gould, and Philippe L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2000.

William W. Hager. Minimizing a Quadratic over a Sphere. *SIAM Journal of Optimization*, 12:188–208, 2001.

D.J. Jeffrey. Formulae, Algorithms, and Quartic Extrema. *Mathematics Magazine*, 70:349–356, 1997.

C. Spearman. *The Abilities of man*. The Macmillan Company, New York, N.Y., 1927.

Department of Statistics, University of California, Los Angeles, CA 90095-1554

E-mail address, Jan de Leeuw: deleeuw@stat.ucla.edu

URL, Jan de Leeuw: http://gifi.stat.ucla.edu