Modification of negative eigenvalues to create positive definite matrices and approximation of standard errors of correlation estimates

L. R. Schaeffer

Centre for Genetic Improvement of Livestock
Department of Animal & Poultry Science
University of Guelph
Guelph, Ontario, Canada

August 25, 2010

Introduction

Current methods of estimating covariance matrices (REML and Bayesian methods) for multiple trait methods, provide estimated matrices that are always positive definite (pd), if the starting matrices are pd. Starting matrices are often derived from the literature, from different sources of information and may not necessarily be pd (i.e. all eigenvalues are positive). An easy approach is to use a diagonal starting matrix, in which all diagonals are positive. Alternatively, one may use a phenotypic covariance matrix reduced by some constant, or a procedure where off-diagonals are gradually reduced until the matrix is pd. Whichever approach is used, the starting matrix needs to be checked to ensure that it is pd.

Hayes and Hill (1981) presented the "bending" procedure to modify the eigenvalues of non-pd matrices, and Jorjani et al.(2003) gave a weighted bending procedure. Finally, Meyer and Kirkpatrick (2010) presented bending using a penalized maximum likelihood method. There have been many other techniques applied to the same problem. In these procedures all of the eigenvalues are modified, both positive and negative eigenvalues. Consequently, the newly re-formed matrix could be quite different from the original matrix. The method in this paper only modifies the negative eigenvalues, and does not have any optimal properties other than the new matrix is pd.

The pd matrix is then utilized in one of the computational versions of REML, and at convergence the resulting estimated matrix should be pd. The first question is what are the standard errors of the correlation estimates from the matrix. Some REML software provides large sample variances of covariance estimates. A sampling method using an inverted Wishart distribution for the matrix of estimated variances and covariances is described to obtain a very rough approximation for standard errors of estimated correlations.

Program codes in the R language are given, but remember that, in R, a task may be solved in many different ways with all solutions giving the same results. Thus, the code given is for descriptive purposes, and not necessarily for computational efficiency. Also, the code does not check for any error conditions that could be encountered, and assumes the user knows how the functions work.

PD Function

There are many ways to change the negative eigenvalues of a matrix to be positive. The concept in this method is that the negative eigenvalues should be very small positive values in decreasing order.

First, sum the negative eigenvalues to give S, square the total, multiply by 100, add one to it, and call this quantity W. Let P be the smallest absolute value of the eigenvalues. If the negative eigenvalues, E[i] are numbered $i = 1$ to m where E[1] is the smallest negative value, and E[m] is the largest negative value, and m is the number of negative eigenvalues. The i^{th} negative eigenvalue is made positive by the formula,

```
NEWvalue[i] =  P * ( (S-E[i])*(S-E[i]) / W )
```

To illustrate, let E[1] = -.09, E[2]=-.46, and E[3]=-3.50, then

```
P = .09,
S = -4.05, and
W = 1641.25.  # equals 100(4.05 squared) plus 1
```

The new eigenvalues would be calculated as follows:

```
E[1] = P * ( (-4.05+.09) *(-3.96) )/ 1641.25 = .00085992
E[2] = P * ( (-4.05+.46) *(-3.59) )/ 1641.25 = .00070674
E[3] = P * ( (-4.05+3.50)*(-0.55) )/ 1641.25 = .00001659
```

Re-create the matrix using the new eigenvalues and the original eigenvectors.

An R function to compute new eigenvalues and to recreate a pd matrix is as follows:

```
PDFORCE = function(Q){
N = nrow(Q)
HC = Q
D = eigen(Q)
E = D$values
U = D$vectors
v = as.numeric(E < 0)
m = sum(v) # number of negative values
if(m > 0){
S = sum(v*E)*2
W = (S*S*100)+1
P = E[N - m] # smallest positive value
k = N - m + 1
for(i in k:N){
C = E[i]
E[i] = P * (S-C)*(S-C)/W
}
HC = U %*% diag(E) %*% t(U)
}
return(HC) }
```

To demonstrate the procedure, use the example matrix from Jorjani et al.(2003) where

$$\mathbf{Q} = \begin{pmatrix} 100 & 95 & 80 & 40 & 40 \\ 95 & 100 & 95 & 80 & 40 \\ 80 & 95 & 100 & 95 & 80 \\ 40 & 80 & 95 & 100 & 95 \\ 40 & 40 & 80 & 95 & 100 \end{pmatrix}.$$

The eigenvalues are

$$\mathtt{E} = \begin{pmatrix} 399.48 & 98.52 & 23.65 & -3.12 & -18.52 \end{pmatrix}.$$

After using `PDFORCE`, the new eigenvalues were

$$\mathtt{newE} = \begin{pmatrix} 399.48 & 98.52 & 23.65 & .02287 & .00065 \end{pmatrix},$$

and the re-constructed matrix was

$$\mathbf{Q}_2 = \begin{pmatrix} 103.17 & 90.83 & 79.47 & 44.54 & 37.07 \\ 90.83 & 106.50 & 94.18 & 74.07 & 44.54 \\ 79.47 & 94.18 & 102.33 & 94.18 & 79.47 \\ 44.54 & 74.07 & 94.18 & 106.50 & 90.83 \\ 37.07 & 44.54 & 79.47 & 90.83 & 103.17 \end{pmatrix},$$

which is similar to the one obtained by Jorjani et al. (2003) after four iterations without weighting factors. Their purpose, however, was to find a pd matrix that was "better" in some sense. The purpose in this paper was to find a reasonable pd matrix that can be used as input to REML software for estimating covariance matrices from data, or for deriving estimated breeding values from a multiple trait BLUP analysis.

Correlation Estimates

Standard errors of estimates of correlations may not always be available from a covariance component analysis. The following approach is an approximation based on the Gibbs sampling technique. Let **V** be the estimated covariance matrix of order N, assumed to be pd (otherwise PDFORCE should be applied), and the degrees of freedom in estimating this matrix was ndf, which is generally much larger than N. The idea is to generate ndf random samples of N variates with covariance matrix equal to **V**, and to calculate the sum of squares and crossproducts of those random variates. Then feed that matrix of crossproducts, SS, into an inverted Wishart random matrix generator to obtain an sample value of **V**. Calculate the correlations among the sample values and compute the standard deviation of the estimates over a few thousand samplings of **V**.

One of the necessary operations needed is a function to convert a covariance matrix into a correlation matrix. A simple R function for this purpose is as follows:

```
CORMAT = function(Q){
D = sqrt(diag(Q))
B = diag(1/D)
HC = B %*% Q %*% B
HC }
```

The function is used in the following manner.

```
R = CORMAT(M)
CORMAT(M)
```

The inverted Wishart generator (`riwish( )`), is found in the R package "MCMCpack". An R function to generate samples and calculate standard deviations of the correlation estimates is as follows:

```
CORSEE = function(nsam,ndf,V){
N = nrow(V)
T = t(chol(V))
m = (N * (N+1))/2
W = matrix(data=c(0),nrow=nsam,ncol=m)
for(i in 1:nsam) {
SS = V*0
for(j in 1:ndf) {
x=matrix(data=rnorm(N,0,1),ncol=1)
z = T %*% x
SS = SS + z %*% t(z) }
V2 = riwish(ndf,SS)
W[i, ]=hsmat(CORMAT(V2)) }
C = sqrt(diag(cov(W)))
return(C) }
```

The function `hsmat( )` converts a full stored, symmetric matrix into a half-stored vector, in order to save space. The function `CORSEE` was applied to $\mathbf{Q}_2$, from the previous section. The correlations in this matrix were:

$$Cor(\mathbf{Q}_2) = \begin{pmatrix} 1.00 & .87 & .77 & .42 & .36 \\ .87 & 1.00 & .90 & .70 & .42 \\ .77 & .90 & 1.00 & .90 & .77 \\ .42 & .70 & .90 & 1.00 & .87 \\ .36 & .42 & .77 & .87 & 1.00 \end{pmatrix}.$$

Standard errors were approximated using `nsam=500` and `ndf` equal to 50, 100, or 500, (Table 1).

Table 1. Approximate standard errors of correlation estimates.

Element	ndf=50	ndf=100	ndf=500
1,2	.058	.038	.015
1,3	.084	.059	.025
1,4	.165	.118	.053
1,5	.181	.123	.057
2,3	.037	.027	.011
2,4	.106	.074	.034
2,5	.170	.118	.051
3,4	.040	.028	.013
3,5	.090	.060	.025
4,5	.055	.035	.016

The size of the standard errors depend on `ndf`, but also on the magnitude of the correlation estimate. Larger correlation estimates tend to have smaller standard errors because they are close to the boundary of +1, and moderate correlation estimates have larger standard errors.

Discussion

Two common problems with covariance matrices were addressed in this paper. Firstly, for covariance matrices that were not pd, a procedure was given to modify the eigenvalues to be positive and to re-construct the matrix as pd. The reader may use the R function provided to try other ideas for changing negative eigenvalues to be positive. However, the purpose of the approach was to provide a pd matrix that could be used as input to other software for either estimating covariance matrices or for applying to multiple trait BLUP.

The second problem was that of deriving standard errors on correlation estimates with an approximate Gibbs sampling technique. A similar approach could be taken for obtaining standard errors on heritability estimates, if needed. The methods presented were entirely ad hoc with the goal of being simple and quick. Theoretically better approximations are most certainly likely, but may not be necessary.

References

Hayes, J. F., W. G. Hill. 1981. Modification of estimates of parameters in the construction of genetic selection indices ('bending'). Biometrics 37:483-493.

Jorjani, H., L. Klei, U. Emanuelson. 2003. A simple method for weighted bending of genetic (co)variance matrices. J. Dairy Sci. 86:677-679.

Meyer, K., M. Kirkpatrick. 2010. Better estimates of genetic covariance matrices by "bending" using penalized maximum likelihood. Genetics 185:1097-1110.