A NUMERICAL PROCEDURE FOR FINDING
THE POSITIVE DEFINITE PATTERNED MATRIX
CLOSEST TO A POSITIVE DEFINITE MATRIX

by

H. Hu and I. Olkin

Technical Report No. 274
July 1990

Prepared Under the Auspices
of
National Science Foundation Grant
DMS 87-08083
Ingram Olkin, Project Director

Department of Statistics
Stanford University
Stanford, California
A NUMERICAL PROCEDURE FOR FINDING
THE POSITIVE DEFINITE PATTERNED MATRIX
CLOSEST TO A POSITIVE DEFINITE MATRIX

by

H. Hu and Ingram Olkin
Northern Illinois University and Stanford University

Technical Report No. 274

Prepared Under the Auspices
of
National Science Foundation Grant
DMS 87-08083
Ingram Olkin, Project Director

Department of Statistics
Stanford University
Stanford, California
A NUMERICAL PROCEDURE FOR FINDING
THE POSITIVE DEFINITE PATTERNED MATRIX
CLOSEST TO A POSITIVE DEFINITE MATRIX

by

H. Hu and I. Olkin
Northern Illinois University and Stanford University

ABSTRACT

Patterned covariance matrices arise naturally from models in the physical, biological, psychological and social sciences. When the underlying data arises from a multivariate normal distribution, maximum likelihood estimates of the population covariance matrix can be obtained numerically, or in some special cases, as closed form expressions or via an iterative procedure. Without the assumption of normality we address the problem of obtaining an estimator that has the appropriate pattern and is close to the sample covariance matrix.

Key words: multivariate normal distribution, least squares, patterned covariance matrices.
A NUMERICAL PROCEDURE FOR FINDING
THE POSITIVE DEFINITE PATTERNED MATRIX
CLOSEST TO A POSITIVE DEFINITE MATRIX

H. Hu and I. Olkin

A variety of patterned covariance matrices arise naturally from models in the physical, biological, psychological, and social sciences. For example, measurements on the petal of a flower, or on the tentacles of a starfish might exhibit a covariance matrix pattern of the form

\[
\Sigma = \sigma^2 \begin{pmatrix}
1 & \rho_1 & \rho_2 & \rho_3 & \cdots & \rho_2 & \rho_1 \\
1 & \rho_1 & \rho_2 & \rho_3 & \rho_2 & \rho_1 \\
1 & \rho_1 & \rho_4 & \rho_3 & \cdots & \rho_3 & \rho_1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
1 & \rho_1 & 1 & \cdots & 1 & \rho_1 \\
1 & 1 & 1 & \cdots & 1 & 1 \\
\end{pmatrix},
\]

which is a circulant. (The lower portion of a symmetric matrix is not displayed.) Here the measurements \(x_1, x_2, \ldots, x_p\) have the property that the relation of \(x_i\) to \(x_{i+j}\) depends only on \(j\). Thus the covariance is \(\text{Cov}(x_i, x_{i+j}) = \sigma^2 \rho_j\), where \(\rho_{p+j} = \rho_{p-j} = \rho_j\). This model also arises if meteorological measurements, say, are taken at the vertices of a regular polygon. (For a discussion of the statistical analysis for such models see Olkin and Press, 1969; Olkin, 1973; Goodenough, Gandini, Olkin, Pizzamiglio, Thayer and Witkin, 1977.)

Relations between items on a psychological test can also be modeled in terms of a patterned structure. In a sequence of papers, Guttman (1954, 1955) introduced a set of patterned covariance matrices which he termed simplex and raddex structures. The models that motivated these patterns were based on studies of psychological tests with certain characteristics that produced patterns. These papers set the stage for considerable research on such models in the educational and psychological literature.

In the context of demographic and migration studies, Piazza and Cavalli-Sforza (1983), Cavalli-Sforza and Piazza (1985), have developed tree models that can be interpreted in
terms of patterns. Examples of such patterns are

\[
(2) \quad \Sigma = \begin{pmatrix}
\alpha & \beta & \gamma & \delta \\
\alpha & \gamma & \delta \\
\alpha & \delta \\
\alpha & 
\end{pmatrix}, \quad \Sigma = \begin{pmatrix}
\alpha & \beta & \beta & \gamma & \delta \\
\alpha & \beta & \gamma & \delta \\
\alpha & \gamma & \delta \\
\alpha & \beta \\
\alpha
\end{pmatrix}.
\]

Given a \( p \times n \) data matrix \( X = (x_{ij}) \), where each column is an observation on a \( p \)-dimensional vector from a population with covariance matrix \( \Sigma \), we form the \( p \times p \) sample cross-product matrix \( S = (s_{ij}) \), where \( s_{ij} = \sum_{\alpha=1}^{N}(x_{i\alpha} - \bar{x}_i)(x_{j\alpha} - \bar{x}_j) \) and \( \bar{x}_i = \sum_{\alpha} x_{i\alpha} / N \). When the underlying distribution is normal, then \( S/(N - 1) \) is an unbiased estimator and \( S/N \) is the maximum likelihood estimator of \( \Sigma \). However, when \( \Sigma \) has a pattern as in (1) or (2), then the estimate \( \tilde{\Sigma} \) should exhibit a similar pattern. Only a few patterns that arise in practice lead to closed form maximum likelihood estimates. The theory of patterned covariance matrices that arise from group invariance has been studied in a series of papers by Andersson (1975), Andersson, Brøns and Jensen (1983), and Andersson & Perlman (1984). In general we must rely on numerical methods to obtain maximum likelihood solutions. This involves finding the

\[
(3a) \quad \max_{\Omega} \{ |\Sigma|^{m/2} \exp -\frac{1}{4} \text{tr} \Sigma^{-1} S \}
\]

or equivalently, of

\[
(3b) \quad \max_{\Omega} \{ \log |\Sigma| - \text{tr} \Sigma^{-1} V \}, \quad V = S/n,
\]

over the set \( \Omega = \{ \Sigma: \Sigma > 0, \Sigma \in P \} \), where \( \Sigma > 0 \) means that \( \Sigma \) is positive definite, \( P \) is the set of matrices having a particular pattern, and \( n = N - 1 \). Several multivariate packages contain algorithms for resolving (3); these usually fall out as special cases in generalized factor-analytic models.

When the pattern in \( \Sigma \) is linear in the sense that

\[
(4) \quad \Sigma = \alpha_1 G_1 + \cdots + \alpha_k G_k,
\]

or
\( \Sigma^{-1} = \alpha_1 G_1 + \cdots + \alpha_k G_k, \)

where each \( G_i \) is a matrix of constants, Anderson (1970) has provided an iterative procedure that converges to the maximum likelihood estimator.

The examples (1) or (2) illustrate linear patterns for which there is no closed form solution, so that the procedure of Anderson (1970) can be used to obtain the maximum likelihood estimator of \( \Sigma \). However, if we remove the underlying assumption of normality, then we need to consider other criteria for estimating \( \Sigma \) based on a sample covariance matrix \( V \).

The problem now can be stated as finding the matrix \( \tilde{\Sigma} \) closest to a given matrix \( V \) for \( \tilde{\Sigma} \) in a particular set \( \Omega \). There is a long history of results dealing with special sets \( \Omega \), the most prominent being the class of positive definite matrices. For a discussion of results on finding matrices closest to a given matrix see Halmos (1972), Marshall and Olkin (1979), and Higham (1988). However, in none of these is there a consideration of a variety of patterns. Indeed, there are so many patterns of interest, we cannot expect to obtain a closed form expression.

One general numerical procedure is to use least squares, that is,

\[
\min_{\tilde{\Sigma} \in \Omega} \| V - \tilde{\Sigma} \|.
\]

Here \( V \) is a sample covariance matrix and is assumed to be positive definite with probability one; this assumption is made throughout. Also, we use the Euclidean norm throughout.

In the case that \( \Sigma \) has a linear structure (4), which we label as \( \mathcal{L} \), then

\[
\min_{\tilde{\Sigma} \in \mathcal{L}} \| V - \tilde{\Sigma} \|
\]

is resolved simply to yield the minimizer

\[
\tilde{\Sigma} = \tilde{\alpha}_1 G_1 + \cdots + \tilde{\alpha}_k G_k,
\]

where \( \tilde{\alpha}_i \) is the mean of the elements of \( V \) corresponding to \( G_i \). This follows by writing \( \| V - \Sigma \| \) as a sum of squares about each parameter. For example, if

\[
\Sigma = \begin{pmatrix}
\alpha + \beta & \alpha & \alpha \\
\alpha & \alpha + \gamma & \beta \\
\alpha & \beta & \gamma
\end{pmatrix}
= \alpha G_1 + \beta G_2 + \gamma G_3,
\]
where
\[ G_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \]
then the minimizer \( \hat{\Sigma} \) from (6) is
\[ \hat{\alpha} = (\nu_{11} + 2\nu_{12} + 2\nu_{13})/5, \quad \hat{\beta} = (\nu_{11} + \nu_{22} + 2\nu_{23})/4, \quad \hat{\gamma} = (\nu_{22} + \nu_{33})/2. \]

If the resulting matrix \( \hat{\Sigma} > 0 \), then it provides the closest patterned positive definite matrix to \( \Sigma \). However, \( \hat{\Sigma} \) obtained in this way need not always yield a positive definite matrix. An example in which a \( \hat{\Sigma} \) obtained this way is not positive definite is
\[
\Sigma = \begin{pmatrix} \alpha & \alpha & \alpha \\ \alpha & \beta & \beta \\ \alpha & \beta & \gamma \end{pmatrix}, \quad V = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \hat{\Sigma} = \begin{pmatrix} 7/5 & 7/5 & 7/5 \\ 7/5 & 4/3 & 4/3 \\ 7/5 & 4/3 & 1 \end{pmatrix}.
\]

Thus, we need to find a patterned positive definite \( \hat{\Sigma} \) of the type given by \( \Sigma \) that is close to a positive definite \( V \). To resolve this we use an algorithm developed by Hu (1989) that yields the positive definite minimizer of \( \| V - \Sigma \| \) over the set \( \{ \Sigma : \Sigma \in \mathcal{L}, \Sigma > 0 \} \).

What this algorithm does is add a constraint
\[ u^T \Sigma u \geq \Delta, \quad \text{for all } \| u \| = 1, \]
where \( \Delta \) normally is chosen to be small, e.g., 0.1, 0.01, 0.001. The value of \( \Delta \) determines how close the patterned positive definite matrix \( \hat{\Sigma} \) is to \( V \). For example, when \( V \) is given by (9), the algorithm yields
\[ \hat{\Sigma} = \begin{pmatrix} 1.259 & 1.259 & 1.259 \\ 1.410 & 1.410 & \end{pmatrix}, \quad \Delta = 0.1, \quad D = 4.214 \]
\[ (10) \]
\[ \hat{\Sigma} = \begin{pmatrix} 1.327 & 1.327 & 1.327 \\ 1.336 & 1.336 & 1.355 \end{pmatrix}, \quad \Delta = 0.01, \quad D = 4.019 \]
\[ \hat{\Sigma} = \begin{pmatrix} 1.335 & 1.335 & 1.335 \\ 1.332 & 1.332 & 1.333 \end{pmatrix}, \quad \Delta = 0.001, \quad D = 4.002, \]
where \( D = \| V - \hat{\Sigma} \|^2 \) is a measure of the distance of the target \( \hat{\Sigma} \) to the given matrix \( V \). Of course, the smaller the constraint threshold \( \Delta \), the smaller the distance.
Some applications call for fixing $\text{tr} \, \tilde{\Sigma}$ to be equal to $\text{tr} \, V$. For example (9) this leads to solutions

\begin{equation}
\tilde{\alpha} = 1.696, \quad \tilde{\beta} = 1.826, \quad \tilde{\gamma} = 2.478,
\end{equation}

regardless of the choice of $\Delta$. These results are to be compared to the solutions (10).

**Comments on the Algorithm**

The set of $n \times n$ positive definite matrices is an open set, so that the optimal solution of

\begin{equation}
\min \| V - \Sigma \|^2
\end{equation}

subject to $\Sigma \in \mathcal{L}$, $\Sigma > 0$ may not exist, i.e., $\{ \Sigma \in \mathcal{L} \} \cap \{ \Sigma > 0 \}$ may be empty or $\inf\{\| V - \Sigma \|^2 : \Sigma \in \mathcal{L}, \Sigma > 0 \}$ cannot be attained. In order to obtain an approximate solution of (12), the algorithm "solves" the semi-infinite quadratic program

\begin{equation}
\min \| V - \Sigma \|^2
\end{equation}

subject to $\Sigma \in \mathcal{L}$, $\Sigma^T = \Sigma$, $y^T \Sigma y \geq \Delta$ for all $y$ satisfying $y^T y = 1$, where $\Delta$ is a small positive number and $y^T \Sigma y \geq \Delta$ is a linear constraint for any fixed $y$. Let $\mathcal{F}_1 = \{ \Sigma : \Sigma \in \mathcal{L}, \Sigma^T = \Sigma, y^T \Sigma y \geq \Delta \}$ for all $y$ satisfying $y^T y = 1$ be the feasible region of (13) and $\mathcal{F} = \{ \Sigma : \Sigma \in \mathcal{L}, \Sigma > 0 \}$ be the feasible region of (12). $\mathcal{F}_1$ is a closed convex set contained in $\mathcal{F}$ and $\mathcal{F}_1$ tends to $\mathcal{F}$ as $\Delta$ tends to zero. If $\mathcal{F}$ is nonempty, then $\mathcal{F}_1$ is nonempty for all sufficiently small $\Delta$, and if $\mathcal{F}_1$ is nonempty, then the optimal solution of (13) exists and is unique. To solve (13), the algorithm generates and solves a sequence of ordinary convex quadratic programs, $QP(k)$ for $k = 0, 1, \ldots$, whose solutions converge to the optimal solution of (13). At the $k$-th iteration, if the optimal solution $\Sigma^{(k)}$ of $QP(k)$ is positive definite, then $\Sigma^{(k)}$ is an approximate solution of (12) and the algorithm stops. If $\Sigma^{(k)}$ is not positive definite, then find a unit eigenvector $y^{(k)}$ of $\Sigma^{(k)}$ corresponding to its smallest eigenvalue, add the constraint $y^{(k)^T} \Sigma y^{(k)} \geq \Delta$ to $QP(k)$, and increase $k$. The algorithm can be formally described as follows.
The Algorithm.

Step 0: Let $k := 0$; let $QP(k)$ be the quadratic program $\min \| V - \Sigma \|^2$, subject to $\Sigma \in \mathcal{L}$, $\Sigma^T = \Sigma$.

Step 1: If $QP(k)$ has no feasible solution, then (13) has no feasible solution, stop. Otherwise, find the optimal solution $\Sigma^{(k)}$ of $QP(k)$.

Step 2: If $\Sigma^{(k)}>0$, then $\Sigma^{(k)}$ is an approximate solution of (12), stop. Otherwise, let $y^{(k)}$ be a unit eigenvector corresponding to the smallest eigenvalue of $\Sigma^{(k)}$; form $QP(k+1)$ by adding the constraint $y^{(k)^T}\Sigma y^{(k)} \geq \Delta$ to $QP(k)$; $k := k + 1$, go to Step 1.

It is shown (Hu (1989)) that for any fixed $\delta > 0$ the algorithm terminates in finitely many iterations and an upper bound on the total number of iterations depends on the value of $\delta$, the smaller the $\delta$, the bigger the upper bound. In the case (13) is infeasible, one may decrease $\delta$ and restart the algorithm. If $\delta$ is very small and (13) is infeasible, then it is likely that (12) is infeasible.

Computational Results. We have coded the algorithm in FORTRAN. We use the NAG library subroutine F02ABF to calculate eigenvalues and eigenvectors, and the subroutine LSSOL from the Department of Operations Research of Stanford University to solve $QP(k)$. The program was executed on a micro VAX machine.

We illustrate the method by carrying out several examples obtained by simulations.

For the pattern

$$
\Sigma = \begin{pmatrix}
\alpha & \beta & \gamma & \delta \\
\alpha & \gamma & \delta & \\
\alpha & \delta & & \\
\alpha & & & \\
\end{pmatrix}
$$

with $\alpha = 6.5000$, $\beta = 5.400$, $\gamma = 6.200$, $\delta = 4.900$ and generated sample data

$$
V = \begin{pmatrix}
6.5597 & 5.6264 & 6.4309 & 4.9427 \\
6.9568 & 6.6318 & 4.9867 & \\
6.9370 & 5.0249 & & \\
6.4133 & & & \\
\end{pmatrix}
$$

we obtain after two iterations

$$
\tilde{\alpha} = 6.79, \quad \tilde{\beta} = 5.68, \quad \tilde{\gamma} = 6.43, \quad \tilde{\delta} = 4.98.
$$
Note that if one uses the "natural" estimator in which \( \bar{\alpha} \) is the mean of the diagonal elements of \( V \), \( \bar{\beta} \) is \( \nu_{12} \), \( \bar{\gamma} \) is the mean of \( \nu_{13} \) and \( \nu_{23} \), and \( \bar{\delta} \) is the mean of \( \nu_{14} \), \( \nu_{24} \), and \( \nu_{34} \), that the resulting matrix is not positive definite.

The solution, if we demand that the trace be held fixed to equal \( \text{tr} \ V \) is

\[
\bar{\alpha} = 6.72, \quad \bar{\beta} = 5.71, \quad \bar{\gamma} = 6.38, \quad \bar{\delta} = 4.98.
\]

The second patterned matrix in (2) provides a good test case. Here we need only specify the parameters \( \alpha, \beta, \gamma, \delta \).

For the generated sample data

\[
V = \begin{bmatrix}
6.5977 & 5.5620 & 5.2479 & 6.2808 & 5.0818 \\
6.8833 & 5.3926 & 6.4562 & 5.2618 \\
7.2024 & 6.3566 & 4.9889 \\
6.9383 & 5.7070 & \\
4.9490 & 
\end{bmatrix}
\]

we obtain after 4 iterations

\[
\bar{\alpha} = 6.61675, \quad \bar{\beta} = 5.50737, \quad \bar{\gamma} = 6.23458, \quad \bar{\delta} = 5.11556.
\]

For this example the objective value is 1.8776, the distance is 3.7553, \( \Delta = 10^{-9} \), CPU time is 11.87 sec.

With the additional constraint that the trace be fixed to be equal to \( \text{tr} \ V \), the solution is obtained after 5 iterations to be

\[
\bar{\alpha} = 6.51414, \quad \bar{\beta} = 5.51784, \quad \bar{\gamma} = 6.16999, \quad \bar{\delta} = 5.12204.
\]

The objective value is 1.91745, the distance is 3.83490, \( \Delta = 10^{-9} \), and the CPU time is 13.61 sec.
References


8


