SOME APPLICATIONS OF THE PSEUDOMATIC OF A MATRIX

T. N. E. Greville

1. INTRODUCTION

In a previous note [1] attention was called to the notion of the pseudoinverse of a rectangular or singular matrix introduced by E. H. Moore [2, 3] and later rediscovered independently by Bjerhammar [4, 5] and Penrose [6]. It is the purpose of the present note to point out two specific applications of the pseudoinverse. Among other possible uses not discussed here is its application to bivariate interpolation.

As a preliminary it will be useful to redefine the pseudoinverse. An $m \times n$ matrix $A$ of rank $r > 0$ can be expressed as a product

$$A = BC,$$

where $B$ is $m \times r$ and $C$ is $r \times n$, and both are of rank $r$. Then the pseudoinverse of $A$ is given by

$$A^\dagger = C^T(CC^T)^{-1}(B^TB)^{-1}B^T,$$

where the superscript $T$ denotes the transpose. To complete the definition, we define the pseudoinverse of a zero matrix as equal to its transpose.

It will be noted that, for the particular cases $n = r$ and $m = r$, (1) reduces to $A = BI$ and $A = IC$, respectively, and (2) therefore reduces to (1) and (2) of [1]. Equation (2) can therefore be written in the form

$$A^\dagger = C^tB^t.$$

The various properties of the pseudoinverse as given in [1] are now easily derived. In particular it will be convenient to recall three of these: (i) the pseudoinverse is unique, (ii) for a nonsingular matrix it reduces to the ordinary inverse, and (iii) $(A^\dagger)^\dagger = A$.

2. POLYNOMIALS ORTHOGONAL OVER DISCRETE DOMAINS

In a recent note [8] Dent and Newhouse have described a recursive procedure for obtaining orthogonal polynomials over a discrete domain, making use of a
method of matrix inversion by means of submatrices given by Fraser, Duncan, and Collar [11]. We shall point out certain advantages that result from a modification of their procedure involving the introduction of the pseudoinverse.

Given \( n \) distinct abscissas \( x_i, \ i = 1, 2, \ldots, n \), let \( y_i \) denote a given ordinate corresponding to \( x_i \), and consider the problem of fitting a polynomial of degree \( k < n \). It can be shown [9] that there is a unique polynomial \( f(x) \) of the required degree which provides the best fit to the given ordinates in the sense of least squares. Let \( q_j(x), j = 0, 1, \ldots, n, \) be a sequence of known polynomials such that \( q_j(x) \) is of proper degree \( j \). In general, we shall be interested in the case \( q_j(x) = x^j \). It is evident that \( f(x) \) has a unique representation in the form

\[
f(x) = \sum_{j=0}^{k} c_j q_j(x),
\]

where the coefficients \( c_j \) are to be determined. Let \( y \) denote the vector whose elements are the given ordinates, and let \( Q_k, k = 0, 1, \ldots, n, \) denote the \( n \times (k + 1) \) matrix \( (q_j(x_i)) \). Parenthetically we remark that the matrix \( S_{k+1} \) of Dent and Newhouse is \( Q_k^T Q_k \). The least-square fitting problem may now be restated as the problem of finding the vector \( d_k \) which is the "best" solution of the matrix equation

\[
Q_k d_k = y
\]

in the sense that the length of \( y - Q_k d_k \) is a minimum. Bjerhammar [4, 5] and Penrose [12] have shown that this solution is

\[
d_k = Q_k^T y.
\]

Since \( k < n \), the columns of \( Q_k \) are linearly independent. Thus, taking \( B = Q_k \) and \( C = I_{k+1} \), the identity matrix of order \( k + 1 \), in (2) gives

\[
Q_k^T = (Q_k^T Q_k)^{-1} Q_k^T.
\]

We note that \( Q_k^T Q_k = I_{k+1} \), while \( M_k = Q_k Q_k^T \) is the smoothing matrix [13] which gives the fitted ordinates in terms of the given ones, since

\[
M_k y = Q_k d_k.
\]

We note also that the sum of the squared differences between the given ordinates and the fitted ones, which Forsythe denotes by \( \delta_k \), is given by\(^8\)

\[
\delta_k = y^T (I - M_k) y.
\]

Now, let \( q_k \) denote the last column of \( Q_k \) and consider the vector \( p_k = q_k - M_{k-1} q_k \). Since \( Q_{k-1}^T p_k = 0 \) by (3), we see that \( p_k \) is orthogonal to every column of \( Q_{k-1} \). Moreover, its elements are ordinates (corresponding to the abscissas \( x_i \)) of a polynomial \( p_k(x) \) of proper degree \( k \). For \( k < n \), \( p_k(x) \) is uniquely determined by these \( n \) ordinates, while \( q_n \) is necessarily a linear combination of the columns of \( Q_{n-1} \), so that \( p_n = 0 \), and therefore

\[
p_n(x) = h \prod_{i=1}^{n} (x - x_i),
\]

\(^8\) See [9] for explanation of the use of these quantities in judging the degree of polynomial best suited to the given data.
where \( h \) is arbitrary. The polynomials \( p_k(x), k = 0, 1, \cdots, n, \) are an orthogonal set over the discrete domain \( x_1, x_2, \cdots, x_n \). If the \( q_k(x) \) are monic polynomials (in particular, if \( q_k(x) = x^k \), the \( p_k(x) \) are the same polynomials found by Dent and Newhouse. Otherwise, they differ at most by a constant factor.

We have
\[
p_k = q_k - Q_{k-1}a_k
\]
where
\[
a_k = Q_{k-1}^t q_k
\]
is the same as the \( A_k \) of Dent and Newhouse. Thus, if \( a_{ki}, i = 0, 1, \cdots, k - 1, \) are the elements of \( a_k \),
\[
(4) \quad p_k(x) = q_k(x) - \sum_{i=0}^{k-1} a_{ki} q_i(x)
\]
as given by them.

To recapitulate, if we can find a convenient method of obtaining the pseudo-inverses \( Q_k \) for \( k = 0, 1, \cdots, n \), then the problem is practically solved, for:
\[
d_k = Q_k^t y \text{ gives the coefficients by which the fitted polynomial of degree } k \text{ is expressed in terms of the known polynomials } q_i(x).
\]
\[
M_k y = Q_k d_k \text{ gives the ordinates of the fitted polynomial corresponding to the given ordinates.}
\]
\[
y^T (I - M_k) y = y^T (y - Q_k d_k) \text{ gives the sum of the squared residuals when a polynomial of degree } k \text{ is fitted by least squares.}
\]
\[
a_k = Q_{k-1}^t q_k \text{ gives the coefficients by which } p_k(x), \text{ the orthogonal polynomial of degree } k, \text{ is expressed in terms of the known polynomials } q_i(x) \text{ in accordance with (4).}
\]
\[
p_k = q_k - Q_{k-1} a_k \text{ gives the ordinates of } p_k(x) \text{ corresponding to the given abscissas.}
\]

3. MULTILINEAR REGRESSION COEFFICIENTS

Let a variate \( y \) depend on \( n \) variates \( x^{(1)}, x^{(2)}, \cdots, x^{(n)} \), and let it be required to determine the coefficients \( a_j \) in the regression equation
\[
y = \sum_{j=1}^{n} a_j x^{(j)}.
\]
It is assumed that corresponding numerical values \( y_i, x^{(j)}_i \) are given for \( i = 1, 2, \cdots, m. \) If \( y \) denotes the column-vector whose \( i \)th component is \( y_i \), \( a \) the column-vector whose \( j \)th component is \( a_j \) and \( X \) the matrix \( (x^{(j)}_i) \), the regression coefficients are given by
\[
(5) \quad a = X^t y.
\]

If the columns of \( X \) are linearly independent,\(^9\) as will usually be the case, the least squares regression equation is unique. Otherwise, there will be many solutions which yield the minimum value for the sum of the squared residuals.

\(^9\) Linear dependence would indicate that at least one of the variates \( x^{(j)} \) is completely determined by the remaining ones, and therefore redundant.
Of these possible solutions, (5) then gives the one for which the sum of the
squares of the coefficients $a_j$ is smallest.

From (5) it follows that the vector

$$ y' = Xa = XX^Ty $$

gives the values of the variate $y$ predicted by the regression equation, while

$$ y^T(y - y') = y^T(I - XX^T)y $$

is the sum of the squares of the residuals.

4. RECURSIVE ALGORITHM FOR THE PSEUDoinverse

Let $a_k$ denote the $k$th column of a given matrix $A$, and let $A_k$ denote the sub-
matrix consisting of the first $k$ columns. As previously pointed out by Dent and
Newhouse, there are substantial advantages in using a recursive procedure for
obtaining $A_k^\dagger$ from $A_{k-1}^\dagger$. In fitting a polynomial this makes it unnecessary to
decide in advance the degree of polynomial to be fitted, or to start over from
scratch if an unfortunate choice is made. Instead, one fits polynomials of suc-
cessively higher degree and can stop when it appears that the most suitable
degree has been reached. Though the advantage is less clear-cut in the regression
application, one can attempt to arrange the variables in decreasing order of their
probable importance in the regression equation, and can note how much the
coefficients change as the less significant variables are introduced, and, if desired,
the reduction at each step in the standard error of estimate.

In order to derive the desired recursive procedure, let us consider $A_k$ in the
partitioned form

$$(A_{k-1} \quad a_k),$$

and similarly partition $A_k^\dagger$ in the form

$$ A_k^\dagger = \begin{pmatrix} B_k \\ b_k \end{pmatrix}. $$

Multiplication then gives

$$(7) \quad A_kA_k^\dagger = A_{k-1}B_k + a_kb_k. $$

As shown in [1], $A_kA_k^\dagger$ is symmetric, and also, when used as a left multiplicator,
it leaves unchanged any matrix with columns in the column-space of $A_k$. It
follows that, as a right multiplicator, it leaves unchanged any matrix with rows in
the transposed column-space of $A_k$. Now [1, p. 40], $A_{k-1}^\dagger$ has rows in the trans-
posed column-space of $A_{k-1}$ (which is contained in that of $A_k$). Therefore

$$ A_{k-1}^\dagger A_kA_k^\dagger = A_{k-1}^\dagger. $$

By similar reasoning, $A_{k-1}^\dagger A_{k-1}$ as a left multiplicator leaves unchanged any
matrix with columns in the transposed row-space of $A_{k-1}$. Now, since $A_k^\dagger$ has
columns in the transposed row-space of $A_k$, a moment’s reflection will convince
the reader that $B_k$ (as a submatrix of $A_k^\dagger$) has columns in the transposed row-space of $A_{k-1}$ (submatrix of $A_k$). Therefore,

$$A_{k-1}^\dagger A_{k-1}B_k = B_k.$$ 

It follows that multiplying (7) on the left by $A_{k-1}^\dagger$ gives

$$A_{k-1}^\dagger = B_k + A_{k-1}^\dagger a_kb_k.$$ 

Thus we may write

$$A_k^\dagger = \begin{pmatrix} A_{k-1}^\dagger - d_k b_k \\ b_k \end{pmatrix},$$

where

$$d_k = A_{k-1}^\dagger a_k$$

and $b_k$ remains to be determined.

From (6) and (8) we obtain

$$A_kA_k^\dagger = A_{k-1}A_{k-1}^\dagger - A_{k-1} d_kb_k + a_kb_k = A_{k-1}A_{k-1}^\dagger + c_kb_k,$$

where

$$c_k = a_k - A_{k-1} d_k.$$ 

Multiplying (11) on the left by $A_{k-1}^\dagger$ and making use of (9) and the relations given in [1], we obtain

$$A_{k-1}^\dagger c_k = 0.$$ 

Now the rows of $A_{k-1}^\dagger$ are in the transposed column-space of $A_{k-1}$, and, in fact, they span that space (since the equation $A_{k-1}A_{k-1}^\dagger A_{k-1} = A_{k-1}$ shows that $A_{k-1}^\dagger$ is not of lower rank than $A_{k-1}$). Therefore (12) shows that $c_k$ is orthogonal to the column-space of $A_{k-1}$.

It is necessary now to consider two cases, according to whether $c_k = 0$ or not. From (11) we see that $c_k = 0$ implies that $a_k$ is in the column-space of $A_{k-1}$; in other words, $A_k$ and $A_{k-1}$ have the same rank. Let us first deal with the case $c_k \neq 0$, and let us consider the matrix

$$P_k = A_{k-1}A_{k-1}^\dagger + c_k c_k^\dagger.$$ 

Now, (11) shows that $c_k$ is in the column-space of $A_k$, and it follows that $c_k^\dagger$ is in the transposed column-space of $A_k$. It follows from (13) that the rows of $P_k$ are in the transposed column-space of $A_k$. From (2), taking $B = c_k$ and $C$ as the identity matrix of order one, it is easily verified that $c_k^\dagger$ is a scalar multiple of $c_k^\dagger$, and that

$$c_k^\dagger c_k = 1.$$ 

Further, multiplying (11) on the left by $c_k^\dagger$ and making use of the fact that $c_k$ is orthogonal to the column-space of $A_{k-1}$ gives

$$c_k^\dagger a_k = 1,$$
and it follows from (13) that

$$P_k a_k = A_{k-1} d_k + c_k = a_k$$

by (11), while we observe also that

$$P_k A_{k-1} = A_{k-1}.$$ 

Thus, (6) shows that

$$P_k A_k = A_k.$$

We see then that $P_k$ has both the properties which uniquely determine the left identity matrix [1, p. 39] of $A_k$, and therefore

$$P_k = A_k A_k^\dagger.$$ 

From (13), (14) and (10) we see that

$$c_k c_k^\dagger = c_k b_k,$$

since both are equal to $A_k A_k^\dagger - A_{k-1} A_{k-1}^\dagger$. Multiplying (15) on the left by $c_k^\dagger$ gives

$$b_k = c_k^\dagger.$$ 

Turning now to the case $c_k = 0$, (11) shows that we then have

$$a_k = A_{k-1} d_k.$$ 

Let $G_k$ denote the submatrix of $A_k A_k^\dagger$ obtained by deleting the last row and the last column. Then it follows from (8) and (6) that

$$G_k = A_{k-1}^\dagger A_{k-1} - d_k b_k A_{k-1}.$$ 

The first term of the right member is symmetric [1, p. 39], as is also $G_k$, being a principal minor of a symmetric matrix. It follows that $d_k b_k A_{k-1}$ is symmetric. Since $b_k A_{k-1}$ is a one-rowed matrix, this implies\(^{10}\) that

$$b_k A_{k-1} = h d_k^\tau,$$

where $h$ is some scalar.

From (8), (6), (17) and (18) we have

$$A_k^\dagger A_k = \left( A_{k-1}^\dagger A_{k-1} - h d_k d_k^\tau \right) (I \quad d_k).$$ 

Now, (9) shows that $d_k$ is in the column-space of $A_{k-1}^\dagger$, which is the transposed row-space of $A_{k-1}$. It follows that

$$A_{k-1}^\dagger A_{k-1} d_k = d_k.$$ 

Thus (19) becomes

$$A_k^\dagger A_k = \begin{pmatrix} A_{k-1}^\dagger A_{k-1} - h d_k d_k^\tau & d_k - h d_k d_k^\tau d_k \\ h d_k^\tau d_k & h d_k^\tau d_k \end{pmatrix}.$$ 

\(^{10}\) If $d_k = 0$, then (since $c_k = 0$), (11) implies $a_k = 0$. Equating the last row of $A_k^\dagger A_k$ to the transpose of its last column gives $b_k A_{k-1} = 0$, so that the conclusion still holds.
In view of the symmetry of this matrix and the fact that \(d_k^T d_k\) is a scalar, we have
\[
(h d_k^T)^T = h d_k = d_k - h(d_k^T d_k) d_k,
\]
and solving for \(h\) gives
\[
(20) \quad h = (1 + d_k^T d_k)^{-1}.
\]
Since the rows of \(A_k^\dagger\) are in the transposed column-space of \(A_k\), \(b_k\) is in that space, which, in this case is identical with the transposed column-space of \(A_{k-1}\). Thus \(b_k A_{k-1} A_{k-1}^\dagger = b_k\), and therefore multiplying (18) on the right by \(A_{k-1}^\dagger\) gives
\[
(21) \quad b_k = h d_k^T A_{k-1}^\dagger.
\]
On substituting (20) this gives
\[
(22) \quad b_k = (1 + d_k^T d_k)^{-1} d_k^T A_{k-1}^\dagger.
\]
The desired recursive procedure for obtaining \(A_k^\dagger\) from \(A_{k-1}^\dagger\) then consists in applying formulas (9), (11), (16) or (22), and (8), in that order. In order to initiate the process, we note that \(A_1^\dagger\) is a zero vector if \(a_1\) is a zero vector; otherwise it can be computed from (2).

5. "STREAMLINED" ALGORITHM FOR STATISTICAL APPLICATIONS

For the purpose of statistical applications, some "streamlining" of the algorithm can be effected by noting that in these situations it is unnecessary to obtain the pseudoinverse explicitly. Rather, what is wanted is the "best" (in the sense of least squares) solution \(x = A^\dagger \alpha\) of an inconsistent system \(Ax = \alpha\). The algorithm can be modified to give \(A_k^\dagger \alpha\) for \(k = 1, 2, \ldots\) successively. To this end it is convenient to define a matrix \(A'\) obtained by enlarging \(A\) through the addition of two columns on the right: (i) the vector \(a\) and (ii) a total column, which is the sum of all the preceding column vectors. Then (8) gives
\[
(23) \quad A_k^\dagger A' = \begin{pmatrix} A_{k-1}^\dagger A' - d_k(b_k A') \\ b_k A' \end{pmatrix}.
\]
The penultimate column of this matrix is \(A_k^\dagger \alpha\), while the final column should be the sum of the preceding column vectors if the arithmetic has been correctly performed. Moreover, (9) shows that \(d_k\) is the \(k\)th column of \(A_{k-1}^\dagger A'\).

In order to obtain \(b_k A'\) for use in (23) we must first compute the right member of (11). If this vector vanishes, (22) shows that
\[
(24) \quad b_k A' = (1 + d_k^T d_k)^{-1} d_k^T A_{k-1}^\dagger A'.
\]
If (11) does not vanish, it gives \(c_k\), and, in view of (16) and (2), we have
\[
(25) \quad b_k A' = (c_k^T c_k)^{-1} c_k^T A'.
\]
If we first compute the vector \(c_k^T A'\), we note that its \(k\)th element is \(c_k^T a_k\). Multiplying (11) on the left by \(c_k^T\) and noting, as previously shown, that \(c_k\) is

\[\text{Provided } d_k \neq 0.\] If \(d_k = 0\), (21) shows that \(b_k = 0\), and (22) still holds.
orthogonal to the column-space of $A_{k-1}$, we obtain $c_k^Tc_k = c_k^T a_k$. It follows from (25) that $b_k A'$ is obtained from the computed vector $c_k^T A'$ upon "normalizing" it by dividing by its $k$th element. With these explanations, (11), (25) or (24), and (23) constitute the recursive procedure desired.

REFERENCES


