19 : APPENDIX

An Index Notation for Multivariate Analysis

1. Tensor Products and Formal Orderings

The algebra of multivariate statistical analysis is, predominantly, the algebra of tensor products, of which the Kronecker product of matrices is a particular instance.

> The Kronecker product of the $t \times n$ matrix $B = [b_{lj}]$ and the $s \times m$ matrix $A = [a_{ki}]$ is defined as an $ts \times nm$ matrix $B \otimes A = [b_{lj}A]$ whose ljth partition is $b_{lj}A$.

In many statistical texts, this definition provides the basis for subsequent algebraic developments. The disadvantage of using the definition without support from the theory of multilinear algebra is that the results which are generated often seem purely technical and lacking in intuitive appeal.

Another approach to the algebra of tensor products relies upon the algebra of abstract vector spaces. Thus

> The tensor product $\mathcal{U} \otimes \mathcal{V}$ of two finite-dimensional vector spaces $\mathcal{U}$ and $\mathcal{V}$ may be defined as the dual of the vector space of all bilinear functionals on $\mathcal{U}$ and $\mathcal{V}$.

This definition facilitates the development of a rigorous abstract theory. In particular, $\mathcal{U} \otimes \mathcal{V}$, defined in this way, already has all the features of a vector space. However, the definition also leads to acute technical difficulties when we seek to represent the resulting algebra in terms of coordinate vectors and matrices.

The approach which we shall adopt here is to define tensor products in terms of formal products. According to this approach, a definition of $\mathcal{U} \otimes \mathcal{V}$ may be obtained by considering the set of all objects of the form $\sum_i \sum_j x_{ij}(u_i \otimes v_j)$, where $u_i \otimes v_j$, which is described as an elementary or decomposable tensor product, comprises an ordered pair of elements taken from the two vector spaces. If the latter are coordinate vector spaces, then, of course, their elements will be ordered sets of numbers.

The cost of this approach is that, in theory, we have to impose the properties of a vector space one-by-one on the set of objects which we have defined. These properties are no longer inherited from the parent spaces $\mathcal{U}$ and $\mathcal{V}$.

Since we are interested in coordinate-specific representations of multiple tensor products, and since part of our aim is to understand how best to represent the latter within a computer, we shall devote the remainder of the present section to the question of formal orderings. We may begin with the definition of a Cartesian product:

> If $\mathcal{A}$ and $\mathcal{B}$ are two sets of objects, then their Cartesian product $\mathcal{A}\times\mathcal{B}$ is the set of all pairs (a, b) with $a \in \mathcal{A}$ and $b \in \mathcal{B}$.

Notice that the definition entails an ordering of the elements a and b within the pair (a, b), which serves, at least, to indicate the origin of the elements. However, the ordering might also correspond to a non-symmetric relationship between the sets $\mathcal{A}$ and $\mathcal{B}$. In that case, (a, b) is distinguished from (b, a) and only the first of these is admitted to the definition.

Imagine that $\mathcal{A} = \{a_i; i = 1, \ldots, m\}$ and $\mathcal{B} = \{b_j; j = 1, \ldots, n\}$ are themselves ordered sets and that $a_i \in \mathcal{A}$ precedes $b_j \in \mathcal{B}$ in any pair (a_i, b_j). Then a so-called lexicographic or "dictionary" ordering is induced over the set of pairs which is arranged in the following fashion:

$$\Big\{(a_1, b_1), (a_1, b_2), \ldots, (a_1, b_n), (a_2, b_1), \ldots, (a_2, b_n), \ldots, (a_m, b_n)\Big\}. \tag{1}$$

If $a' = [a_1, \ldots, a_n]$ and $b' = [b_1, \ldots b_m]$ are coordinate vectors, and if (a_i, b_j) stands for the product of two numbers, then what is displayed above amounts to the elementary tensor product $a' \otimes b'$.

A lexicographic ordering of this nature arises when we map an $m \times n$ matrix $X = [x_{ij}]$ into a sequence of memory locations within a computer. If the elements of the matrix are mapped row-by-row into the memory, then the sequence of addresses will correspond to the following sequence of matrix elements:

$$x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, \ldots, x_{2n}, \ldots, x_{mn}. \tag{2}$$

In order to access the value of x_{ij}, we should need to construct a *pointer* to the corresponding memory address. If the addresses were numbered sequentially from 1 to mn, then the pointer would indicate the $\{(i-1)n+j\}$th address. We must emphasise that the pointer serves only to locate the value of x_{ij}. It does not reveal the nature of the abstract object—a matrix in this case—to which the element belongs. For this purpose, we need a further device.

Now consider the matter of forming the product of the ordered set $\mathcal{A} \times \mathcal{B}$ with another ordered set $\mathcal{C} = \{c_k; k = 1, \ldots, p\}$. In the simplest case, we can define the product to be the set $(\mathcal{A} \times \mathcal{B}) \times \mathcal{C}$ comprising all pairs $((a_i, b_j), c_k)$

with $(a_i, b_j) \in \mathcal{A} \times \mathcal{B}$; and $c_k \in \mathcal{C}$. If the elements in the pairs bear a symmetric relationship to each other—such as a relationship of orthogonality for example—then we may remove the inner parentheses so as to write the elements of the product of the sets as ordered triples (a_i, b_j, c_k).

The set of all such triples is ordered according to a lexicographic scheme comprising the three indices i, j and k. The value of element (a_i, b_j, c_k) may be extracted from the corresponding array within a computer by pointing to the $\{(i-1)np + (j-1)p + k\}$th address. It should be clear how such a scheme can be extended to encompass multiple products of any order.

Matters become considerably more complicated if the relationship between elements of a pair is non-symmetric, and if both (a_i, b_j) and (b_j, a_i) are admitted when $a_i \in \mathcal{A}$ and $b_j \in \mathcal{B}$. In that case, a distinction arises between $(c_k, (a_i, b_j))$ and $((b_j, a_i), c_k)$. Moreover, the notation becomes ambiguous unless a_i and b_j bear marks of their origins. Such ambiguity can be avoided by placing parentheses around the initial element a_i. On removing the commas, which have become redundant, we get $({}_3 c_k ({}_2 ({}_1 a_i)_1 b_j)_2)_3$ and $({}_3 ({}_2 b_j ({}_1 a_i)_1)_2 c_k)_3$. These expressions are to be interpreted in view of the ordering of the sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ in which the elements in the triples originate. Thus the element a_i in the innermost parenthesis originates in the first set $\mathcal{A}$, the element b_j within the next parentheses originates in the second set $\mathcal{B}$, and so on. The subscripts on the parentheses are only for legibility.

The next point to notice about the triples is that the elements b_j and c_k have acquired attributes by virtue of their position within non-symmetric relationships. Thus, in the product $({}_3 c_k ({}_2 ({}_1 a_i)_1 b_j)_2)_3$, for example, c_k is a *left* element whereas b_j is a *right* element. However, the innermost element a_i has neither of these attributes. In order to attribute a position to a_i, we make use of a dummy element denoted by an zero. Thus, in place of (a_i), we may write $((0)a_i)$ if a_i is a right element or else $(a_i(0))$ if it is a left element.

Within the computer, the set of the triples will continue to be stored in a one-dimensional array which conveys no information about the nature of the object to which the elements belong. The array must be accompanied by a *bridge* which links it to the abstract object. Such a bridge can be constructed using the parenthesis notation. Consider, for example, the triple $({}_3 c_k ({}_2 ({}_1 (0) a_i)_1 b_j)_2)_3$. If the elements a_i, b_j and c_k are replaced by their respective indices i, j and k, then we obtain a string of integers and parentheses, $({}_3 k ({}_2 ({}_1 (0) i)_1 j)_2)_3$. This serves two purposes. One the one hand, the numbers i, j, k enable us to construct a pointer to a memory address within the one-dimensional computer array which contains the value of the element. On the other hand, the parentheses serve to define the nature of the object to which the element belongs.

The bridge has two more components. The first of these is a dimensions vector which records the upper limits of the indices. This information is needed in the construction of the pointer. The second is a permutation vector which allows pairs of indices to swap their positions within the nest of parentheses without severing their connections, via the pointer, with the corresponding memory locations.

Example. The parenthesis notation, it must be admitted, is not easy to read by eye. In practice, it is intended to be read by a computer. To understand the effect of the notation in a simple context, let us consider once more the $m \times n$ matrix $X = [x_{ij}]$ whose elements are arrayed in lexicographic order under (2).

Imagine that the array is accompanied by the bridge $(j((0)i))$. Then we should understand that the index i of the element x_{ij}, which is a *right* index within the bridge, is a *row* index of a matrix. The index j, which is a *left* index in the bridge, is a *column* index of a matrix.

Imagine, instead, that the array is accompanied by the bridge $((i(0))j)$. Then it is implied that the element x_{ij} is to be found in the jth row and the ith column of the matrix X' which is the transpose of X.

Finally, let us imagine that the array is accompanied by $(j(i(0)))$. Then x_{ij} is an element of a long column vector $X^c = [x_{11}, x_{21}, \ldots, x_{mn}]'$ constructed by slicing the matrix X vertically and joining the columns end-to-end. We may notice that the elements are arrayed in a reversed lexicographic order with the index j as the principal classifier. An indication of this reversal is provided by the fact that, when we read from left to right, the indices in the bridge are also in reversed order. If the array were accompanied by the bridge $(((0)i)j)$, then we should understand that x_{ij} is an element of a long row vector X^r whose elements are arrayed in the same lexicographic order as the addresses of the memory cells which store their values.

The objects X, X', X^r and X^c all share the same elements. There may be numerous occasions when we wish to transform one of these objects into another. The foregoing example makes it clear that such transformations can be performed by using the bridge alone. There is no need to alter or disturb the memory locations in which the values of the elements are stored.

2. The Index Notation and the Vectorisation Operations

In this section, we shall begin to develop a serviceable index notation which conveys the same information as the bridge notation but which is intended for human manipulation rather than machine calculation. We start by recapitulating some of the results from the end of the previous section.

Consider the $m \times n$ matrix $X = [x_{ij}]$ which has the scalar x_{ij} in the ith row and the jth column. Let e_i be a column vector of order m with a unit in the ith position and with zeros elsewhere, and let e^j be a row vector of order n with a unit in the jth position and with zeros elsewhere. Then $e_i \otimes e^j$ is a matrix of order $m \times n$ with a unit in the ijth position and zeros elsewhere, and we have

$$X = \sum_i \sum_j x_{ij}(e_i \otimes e^j). \tag{3}$$

We propose to write this as

$$X = (x_{ij}e_i^j). \tag{4}$$

Here we have $e_i^j = e_i \otimes e^j$. The expression $x_{ij}e_i^j$ is surrounded by parentheses, and these are to indicate that summation takes place in respect of both of the indices associated with the basis elements e_i^j.

We might be tempted to flout convention by writing the scalar element x_{ij} as x_i^j so that the positions of its indices would conform with those of the basis elements. In this way, we would avoid giving precedence to either of the indices. However, a convention which gives precedence to one or other of the indices is precisely what is required for the purpose of reducing a two-dimensional array to a linear sequence in which the elements are liable to stored within a computer.

Naturally, the expression in (4) may be specialised by the omission of one or other of the indices to give a column vector or a row vector. Thus (x_ie_i) is the column vector $[x_1, \ldots, x_m]'$, whilst (x_je^j) is the row vector $[x_1, \ldots, x_n]$. An easy mnemonic springs to mind: an raised index in the *roof* signifies a *row* whereas a lowered index in the *cellar* signifies a *column.*

We may also construct row and column vectors from the matrix X by the processes of vectorisation. We define X^c to be a column vector whose generic element x_{ij} is contained in the $\{(j-1)m+i\}$th position. Since $e_{ji} = e_j \otimes e_i$ is an $mn \times 1$ column vector with a unit in that position and zeros elsewhere, it follows that $X^c = \sum_i \sum_j x_{ij}(e_j \otimes e_i)$ or, equivalently,

$$X^c = (x_{ij}e_{ji}). \tag{5}$$

A comparison of this with the expression under (4) shows that $(x_{ij}e_i^j)^c = (x_{ij}e_{ji})$. Thus, when we convert the superscript j to a subscript, we move it to the head of the existing string of subscripts. The indices of the elements of X^c follow a reversed lexicographic ordering.

These conventions conform with those of the bridge notation of the previous section. Thus, whereas the matrix X is associated with the bridge notation $(j((0)i))$ wherein i is a row index and j is a column index, the long column vector X^c is associated with the notation $(j(i(0)))$. The mapping from X to X^c is accomplished by the simple commutation which changes $((0)i)$ into $(i(0))$.

The alternative way of vectorising the matrix X is to create a row vector X^r of order $1\times mn$ whose generic element x_{ij} is contained in the $\{(i-1)n+j\}$th position. Since $e^{ij} = e^i \otimes e^j$ is a row vector with a unit in that position an with zeros elsewhere, we have

$$X^r = (x_{ij}e^{ij}), \tag{6}$$

and thus we have the convention that $(x_{ij}e_i^j)^r = (x_{ij}e^{ij})$. When we convert the subscript i to a superscript, we move it to the head of the existing string of superscripts. The indices of the elements in X^r follow a direct lexicographic ordering. In the bridge notation, X^r is associated with $(((0)i)j)$.

It remains only to consider the operation of transposition. This is summarised by the identity $(e_i \otimes e^j)' = (e_j \otimes e^i)$ which leads us to the expression

$$X' = (x_{ij}e_i^j)' = (x_{ij}e_j^i). \tag{7}$$

It follows that

$$(X')^c = (X^r)' \quad \text{and} \quad (X^c)' = (X')^r. \tag{8}$$

The transformation of $X^c = (x_{ij}e_{ji})$ into $(X')^c = (x_{ij}e_{ij})$ is effected by an operator know variously as the vec-permutation matrix, according to Henderson and Searle [4], the commutation matrix, according to Magnus and Neudecker [8], or the tensor commutator, according to Pollock [9].

In the bridge notation, X^c is associated with $(j(i(0)))$ whereas $(X')^c$ is associated with $(i(j(0)))$. The swapping of the positions of i and j is accompanied by a corresponding interchange within the permutation vector. It is notable that the transformation of X into X', which corresponds to the transformation of $(j((0)i))$ into $((i(0))j)$, does not affect the permutation vector.

Example. Consider the equation

$$y_{tj} = \mu + \gamma_t + \delta_j + \varepsilon_{tj} \tag{9}$$

wherein $t = 1, \ldots, T$ and $j = 1, \ldots, M$. This relates to a two-way analysis of variance. For a concrete interpretation, we may imagine that y_{tj} is an observation taken at time t in the jth region. Then the parameter γ_t represents an effect which is common to all observations taken at time t, whilst the parameter δ_j represents a characteristic of the jth region which prevails through time.

In ordinary matrix notation, the set of TM equations becomes

$$Y = \mu\iota_T\iota_M' + \gamma\iota_M' + \iota_T\delta' + \mathcal{E}, \tag{10}$$

where $Y = [y_{tj}]$ and $\mathcal{E} = [\varepsilon_{tj}]$ are matrices of order $T \times M$, $\gamma = [\gamma_1, \ldots, \gamma_T]'$ and $\delta = [\delta_1, \ldots, \delta_M]'$ are vectors of orders T and M respectively, and ι_T and ι_M are vectors of units whose orders are indicated by their subscripts.

In terms of the index notation, the TM equations are represented by

$$(y_{tj}e_t^j) = \mu(e_t^j) + (\gamma_t e_t^j) + (\delta_j e_t^j) + (\varepsilon_{tj}e_t^j). \tag{11}$$

As an illustration, we may consider the case where $T = M = 3$. Then equations (10) and (11) represent the following structure:

$$\begin{aligned}\begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} = \mu \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} &+ \begin{bmatrix} \gamma_1 & \gamma_1 & \gamma_1 \\ \gamma_2 & \gamma_2 & \gamma_2 \\ \gamma_3 & \gamma_3 & \gamma_3 \end{bmatrix} \\ &+ \begin{bmatrix} \delta_1 & \delta_2 & \delta_3 \\ \delta_1 & \delta_2 & \delta_3 \\ \delta_1 & \delta_2 & \delta_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix}.\end{aligned} \tag{12}$$

3. Multiple Tensor Products

The tensor product entails an associative operation which combines matrices or vectors of any order. Let $B = [b_{lj}]$ and $A = [a_{ki}]$ be arbitrary matrices of orders $t \times n$ and $s \times m$ respectively. Then their tensor product $B \otimes A$, which is also know as a Kronecker product, is defined in terms of the index notation by writing

$$(b_{lj} e_l^j) \otimes (a_{ki} e_k^i) = (b_{lj} a_{ki} e_{lk}^{ji}). \tag{13}$$

Here e_{lk}^{ji} stands for a matrix of order $st \times mn$ with a unit in the row indexed by lk—the $\{(l-1)s + k\}$th row—and in the column indexed by ji—the $\{(j-1)m + i\}$th column—and with zeros elsewhere. We may notice that the indices lk are not ordered relative to the indices ji. That is to say, we have

$$\begin{aligned} e_{lk}^{ji} &= e_l \otimes e_k \otimes e^j \otimes e^i \\ &= e^j \otimes e^i \otimes e_l \otimes e_k \\ &= e^j \otimes e_l \otimes e_k \otimes e^i \\ &= e_l \otimes e^j \otimes e^i \otimes e_k \\ &= e_l \otimes e^j \otimes e_k \otimes e^i \\ &= e^j \otimes e_l \otimes e^i \otimes e_k. \end{aligned} \tag{14}$$

The virtue of the index notation is that it makes no distinction amongst these various products on the RHS—unless a distinction can be found between such expressions as $e^{j\ i}_{\ l\ k}$ and $e_l^{\ j}{}_k^{\ i}$.

4. Compositions

In order to demonstrate the rules of matrix composition, let us consider the matrix equation

$$Y = AXB', \tag{15}$$

which we shall construe as a mapping from X to Y. In the index notation, this is written as

$$\begin{aligned}(y_{kl}e_k^l) &= (a_{ki}e_k^i)(x_{ij}e_i^j)(b_{lj}e_j^l)\\ &= (\{a_{ki}x_{ij}b_{lj}\}e_k^l).\end{aligned}\tag{16}$$

Here we have

$$\{a_{ki}x_{ij}b_{lj}\} = \sum_i\sum_j a_{ki}x_{ij}b_{lj};\tag{17}$$

which is to say that the braces surrounding the expression on the LHS are to indicate that summations are taken with respect to the repeated indices i and j. The operation of composing two factors depends upon the cancellation of a superscript (column) index, or string of indices, in the leading factor with an equivalent subscript (row) index, or string of indices, in the following factor.

The matrix equation of (15) can be vectorised in a variety of ways. In order to represent the mapping from $X^c = (x_{ij}e_{ji})$ to $Y^c = (y_{kl}e_{lk})$, we may write

$$\begin{aligned}(y_{kl}e_{lk}) &= (\{a_{ki}x_{ij}b_{lj}\}e_{lk})\\ &= (a_{ki}b_{lj}e_{lk}^{ji})(x_{ij}e_{ji}).\end{aligned}\tag{18}$$

Notice that the product $a_{ki}b_{lj}$ within $(a_{ki}b_{lj}e_{lk}^{ji})$ does not need to be surrounded by braces since it contains no repeated indices. Nevertheless, there would be no harm in writing $\{a_{ki}b_{lj}\}$.

The matrix $(a_{ki}b_{lj}e_{lk}^{ji})$ is decomposable. That is to say

$$\begin{aligned}(a_{ki}b_{lj}e_{lk}^{ji}) &= (b_{lj}e_l^j)\otimes(a_{ki}e_k^i)\\ &= B\otimes A;\end{aligned}$$

and, therefore, the vectorised form of equation (15) is

$$\begin{aligned}Y^c &= (AXB')^c\\ &= (B\otimes A)X^c.\end{aligned}\tag{19}$$

Example. The equation under (11), which relates to a two-way analysis of variance, can be vectorised to give

$$(y_{tj}e_{jt}) = \mu(e_{jt}) + (e_{jt}^t)(\gamma_t e_t) + (e_{jt}^j)(\delta_j e_j) + (\varepsilon_{tj}e_{jt}).\tag{20}$$

Using the notation of the Kronecker product, this can also be rendered as

$$\begin{aligned}Y^c &= \mu(\iota_M\otimes\iota_T) + (\iota_M\otimes I_T)\gamma + (I_M\otimes\iota_T)\delta + \mathcal{E}^c\\ &= X\beta + \mathcal{E}^c.\end{aligned}\tag{21}$$

In comparing (20) and (21), we see, for example, that $(e_{jt}^{\,t}) = (e_j) \otimes (e_t^t) = \iota_M \otimes I_T$. We recognise that (e_t^t) is the sum over the index t of the matrices of order T which have a unit in the tth diagonal position and zeros elsewhere; and this sum amounts, of course, to the identity matrix of order T.

By vectorising our example under (12), we get

$$\begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{12} \\ y_{22} \\ y_{32} \\ y_{13} \\ y_{23} \\ y_{33} \end{bmatrix} = \left[\begin{array}{c|ccc|ccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{c} \mu \\ \hline \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \hline \delta_1 \\ \delta_2 \\ \delta_3 \end{array}\right] + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \varepsilon_{31} \\ \varepsilon_{12} \\ \varepsilon_{22} \\ \varepsilon_{32} \\ \varepsilon_{13} \\ \varepsilon_{23} \\ \varepsilon_{33} \end{bmatrix}. \tag{22}$$

5. Matrix Differential Calculus

We shall now use the index notation to examine three alternative definitions of the derivative of a matrix function $Y = Y(X)$ with respect to its matrix argument X. We shall establish relationships amongst these definitions and we shall argue that only one of them is viable. The other definitions, which are widely used in multivariate statistical analysis, are not consistent with the classical representation of linear algebra via matrix theory, and they lead to serious practical difficulties which do not arise when the appropriate definition is adopted.

According to the first definition, the derivative is a partitioned matrix $[\partial Y/\partial x_{ij}]$ whose ijth partition is derived from the matrix Y by replacing each element y_{kl} by its derivative $\partial y_{kl}/\partial x_{ij}$. Thus the elements of the matrix derivative have the same disposition as the elements $x_{ij}y_{kl}$ of the product $X \otimes Y = (x_{ij}e_i^j) \otimes (y_{kl}e_k^l)$, which we can also write as

$$X \otimes Y = (x_{ij}y_{kl}e_{ik}^{jl}). \tag{23}$$

This definition has been studied by Rogers [10] amongst others; and he has ascribed to it the notation $\varepsilon Y/\varepsilon X$. Graham [3] has also adopted it as the basic definition in his textbook.

According to the second definition, the derivative of Y with respect to X is a partitioned matrix $[\partial y_{kl}/\partial X]$ whose klth partition is derived from the matrix X by replacing the elements x_{ij} by the derivatives $\partial y_{kl}/\partial x_{ij}$. The elements of the matrix derivative therefore have the same disposition as the elements of

$$Y \otimes X = (x_{ij}y_{kl}e_{ki}^{lj}). \tag{24}$$

This is probably the most commonly used definition. It has been employed extensively by both Rogers [10] and Balestra [1] in their treatises of matrix differential calculus; and it was adopted by MacRae [5] in a seminal article to which many authors have referred.

From another point of view, the two definitions already considered follow directly from the definitions of Dwyer and MacPhail [2] who considered the forms $\partial Y/\partial x_{ij}$ and $\partial y_{kl}/\partial X$ without arranging them into partitioned matrices.

According to the third definition, the derivative of Y with respect to X is a matrix $\partial Y^c/\partial X^c$ whose elements $\partial y_{kl}/\partial x_{ij}$ have the same arrangements as the elements $y_{kl}x_{ij}$ in the product

$$Y^c \otimes (X^c)' = (x_{ij}y_{kl}e_{lk}^{ji}). \tag{25}$$

This final definition, which we may call the vectorial definition, has been used by Neudecker [8] and by the present author—Pollock [9]—amongst many others. Nel [7] has ascribed to this derivative the notation $\partial \mathrm{vec} Y/\partial \mathrm{vec}' X$.

The relationships between the three definitions are revealed by juxtaposing their expressions:

$$\begin{aligned}
\left[\frac{\partial Y}{\partial x_{ij}}\right] &= \left(\frac{\partial y_{kl}}{\partial x_{ij}}\; e_{ik}^{jl}\right),\\
\left[\frac{\partial y_{kl}}{\partial X}\right] &= \left(\frac{\partial y_{kl}}{\partial x_{ij}}\; e_{ki}^{lj}\right),\\
\left[\frac{\partial Y^c}{\partial X^c}\right] &= \left(\frac{\partial y_{kl}}{\partial x_{ij}}\; e_{lk}^{ji}\right).
\end{aligned} \tag{26}$$

The first two derivatives are seen to differ from each other only in respect of the orders within the pair of column indices j, l and within the pair of row indices k, i. The third derivative $\partial Y^c/\partial X^c$ differs from the other two in a more complicated way which requires the conversion of the basis vectors e_i and e^l into e^i and e_l respectively.

6. Chain Rules

To illustrate the contention that the vectorial definition is the appropriate one, we shall consider the problem of defining a chain rule for matrix derivatives.

Let $X = X(Z)$ and $Y = Y(X)$ be two matrix transformations whose composition is $Y = Y(Z)$. Then the vectorial definition gives rise to a rule in the form of

$$\frac{\partial Y^c}{\partial Z^c} = \frac{\partial Y^c}{\partial X^c}\,\frac{\partial X^c}{\partial Z^c}, \tag{27}$$

which entails nothing more than the multiplication of the forms $\partial Y^c/\partial X^c$ and $\partial X^c/\partial Z^c$ according to the ordinary rules of matrix algebra. Chain rules obeying the normal algebra of matrix compositions are not available for the other definitions which we have considered.

To provide the simplest example of the chain rule under (27), let us consider the following matrix equations:

$$\begin{aligned} Y &= AXB' \qquad \text{or} \qquad (y_{kl}e_k^l) = (a_{ki}e_k^i)(x_{ij}e_i^j)(b_{lj}e_j^l), \\ X &= CZD' \qquad \text{or} \qquad (x_{ij}e_i^j) = (c_{if}e_i^f)(z_{fu}e_f^u)(d_{ju}e_u^j). \end{aligned} \tag{28}$$

The composition of the two mappings gives

$$Y = (AC)Z(BD)' \qquad \text{or} \qquad (y_{kl}e_k^l) = (\{a_{ki}c_{if}\}e_k^f)(z_{fu}e_f^u)(\{b_{lj}d_{ju}\}e_u^l). \tag{29}$$

The vectorised versions of the two equations under (28) are given by

$$\begin{aligned} Y^c &= (B \otimes A)X^c \qquad \text{or} \qquad (y_{kl}e_{lk}) = (\{b_{lj}a_{ki}\}e_{lk}^{ji})(x_{ij}e_{ji}), \\ X^c &= (D \otimes C)Z^c \qquad \text{or} \qquad (x_{ij}e_{ji}) = (\{d_{ju}c_{if}\}e_{ji}^{uf})(z_{fu}e_{uf}), \end{aligned} \tag{30}$$

and that of their composition under (29) is given by

$$Y^c = (BD \otimes AC)Z^c \qquad \text{or} \qquad (y_{kl}e_{lk}) = (\{b_{lj}d_{ju}\}\{a_{ki}c_{if}\}e_{lk}^{uf})(z_{fu}e_{uf}). \tag{31}$$

Next, by referring to the definition under (26), we find that $\partial Y^c/\partial X^c = (\{\partial y_{kl}/\partial x_{ij}\}e_{lk}^{ji}) = (b_{lj}a_{ki}e_{lk}^{ji})$. In this manner, we can easily confirm that

$$\begin{aligned} \frac{\partial Y^c}{\partial X^c} &= B \otimes A, \\ \frac{\partial X^c}{\partial Z^c} &= D \otimes C, \\ \frac{\partial Y^c}{\partial Z^c} &= BD \otimes AC. \end{aligned} \tag{32}$$

Finally, by confirming that

$$\begin{aligned} (B \otimes A)(D \otimes C) &= (b_{lj}a_{ki}e_{lk}^{ji})(d_{ju}c_{if}e_{ji}^{uf}) \\ &= (\{b_{lj}d_{ju}\}\{a_{ki}c_{if}\}e_{lk}^{uf}) \\ &= (\{b_{lj}d_{ju}\}e_l^u) \otimes (\{a_{ki}c_{if}\}e_k^f) = BD \otimes AC, \end{aligned} \tag{33}$$

we verify the chain rule in question.

To obtain chain rules for the alternative definitions, we are obliged to invent special operations of composition which do not accord with the usual matrix algebra. Consider the following derivatives:

$$\begin{aligned} \left[\frac{\partial y_{kl}}{\partial X}\right] &= (b_{lj}a_{ki}e_{ki}^{lj}) = (A')^c \otimes B^r, \\ \left[\frac{\partial x_{ij}}{\partial Z}\right] &= (d_{ju}c_{if}e_{if}^{ju}) = (C')^c \otimes D^r, \\ \left[\frac{\partial y_{kl}}{\partial Z}\right] &= (\{a_{ki}c_{if}\}\{b_{lj}d_{ju}\}e_{kf}^{lu}) = ([AC]')^c \otimes (BD)^r. \end{aligned} \tag{34}$$

A chain rule is obtained by defining a star product of the derivatives such that

$$\left[\frac{\partial y_{kl}}{\partial Z}\right] = \left[\frac{\partial y_{kl}}{\partial X}\right] * \left[\frac{\partial x_{ij}}{\partial Z}\right]. \tag{35}$$

In terms of our example, this becomes

$$\begin{aligned} ([A']^c \otimes B^r) * ([C']^c \otimes D^r) &= ([AC]')^c \otimes (BD)^r \quad \text{or} \\ (b_{lj}a_{ki}e^{lj}_{ki}) * (d_{ju}c_{if}e^{ju}_{if}) &= (\{a_{ki}c_{if}\}\{b_{lj}d_{ju}\}e^{lu}_{kf}). \end{aligned} \tag{36}$$

In place of the ordinary convention of matrix multiplication, which implies that

$$(e^{ji}_{lk})(e^{uf}_{ji}) = (e^{uf}_{lk}), \tag{37}$$

we have a new convention of star products to the effect that

$$(e^{lj}_{ki}) * (e^{ju}_{if}) = (e^{lu}_{kf}). \tag{38}$$

This star product is a generalisation of a product which MacRae [5] used in defining a rule for the composition of a derivative $\partial y/\partial X$ of a scalar function $y = y(X)$ with the derivative $[\partial x_{ij}/\partial Z]$ of a matrix function $X = X(Z)$.

The generalised star product has the manifest disadvantage that it cannot be extended in any simple way to accommodate the composition of multiple tensor products. By contrast, the usual rules for matrix manipulation extend easily to such cases. For example, for triple products or matrices, we have the simple rule that

$$(A \otimes B \otimes C)(D \otimes E \otimes F) = AD \otimes BE \otimes CF. \tag{39}$$

References

[1] P. Balestra, 1976, La Dérivation Matricielle, Sirey, Paris.

[2] P.S. Dwyer and M.S. MacPhail, 1948, Symbolic matrix derivatives, Ann. Math. Statis. 19: 517–534.

[3] A. Graham, 1981, Kronecker Products and Matrix Calculus with Applications, Ellis Horwood, Chichester.

[4] H.V. Henderson and S.R. Searle, 1981, The vec-permutation matrix, the vec operator and Kronecker products: A review, Linear and Multilinear Algebra 9: 271–288.

[5] E.C. MacRae, 1974, Matrix derivatives with an application to an adaptive linear decision problem, Ann. Statist. 2: 337–346.

[6] J.R. Magnus and H. Neudecker, 1979, The commutation matrix: Some properties and applications, Ann. Statis. 2: 337–346.

[7] D.G. Nel, 1980, On matrix differentiation in statistics, South African Statistical Journal 14: 137–193.

[8] N. Neudecker, 1969, Some theorems on differentiation with special reference to Kronecker matrix products. J. Amer. Statist, Assoc. 64: 953–963.

[9] D.S.G. Pollock, 1979, The Algebra of Econometrics, Wiley, Chichester.

[10] G.S. Rogers, 1980, Matrix Derivatives, Marcel Dekker, New York.