OpenMx Matrices and Operators

Sarah Medland

Matrices: the building blocks

- Many types

? Denotes a free element

mxMatrix(type="Zero", nrow=2, ncol=3, name="a")

```
0 0 0
0 0 0
```

mxMatrix(type="Unit", nrow=2, ncol=3, name="a")

```
1 1 1
1 1 1
```

mxMatrix(type="Ident", nrow=3, ncol=3, name="a")

```
1 0 0
0 1 0
0 0 1
```

mxMatrix(type="Diag", nrow=3, ncol=3, free=TRUE, name="a")

```
? 0 0
0 ? 0
0 0 ?
```

mxMatrix(type="Sdiag", nrow=3, ncol=3, free=TRUE, name="a")

```
0 0 0
? 0 0
? ? 0
```

mxMatrix(type="Stand", nrow=3, ncol=3, free=TRUE, name="a")

```
1 ? ?
? 1 ?
? ? 1
```

mxMatrix(type="Symm", nrow=3, ncol=3, free=TRUE, name="a")

```
? ? ?
? ? ?
? ? ?
```

mxMatrix(type="Lower", nrow=3, ncol=3, free=TRUE, name="a")

```
? 0 0
? ? 0
? ? ?
```

mxMatrix(type="Full", nrow=2, ncol=4, free=TRUE, name="a")

```
? ? ? ?
? ? ? ?
```

Matrix operations – classic vx OpenMx

Name	Mathematical	Classic Mx	OpenMx	Conformability
Inverse	$^{-1}$	~	solve()	r=c
Transpose	‘	‘	t()	None
Power	^	^	^ or %^%	None
Star	*	*	%*%	$c_A = r_B$
Dot	.	.	*	$r_A = r_B \,\&\, c_A = c_B$
Kronecker	$\otimes$	@	%x%	none
Quadratic	&	&	%&%	$c_A = r_B = c_B$
Division	%	%	/	$r_A = r_B \,\&\, c_A = c_B$
Addition	+	+	+	$r_A = r_B \,\&\, c_A = c_B$
Subtraction	-	-	-	$r_A = r_B \,\&\, c_A = c_B$
Adhesion	_ or |	_ or |	rbind or cbind	$r_A = r_B \text{ or } c_A = c_B$

Inverse

- mxAlgebra(expression=solve(*matrix_name), name=“A"),*
- Only square matrices may be inverted, but they may be either symmetric or non-symmetric. The inverse of matrix A is usually written A^{-1} and implies that $AA^{-1} = A^{-1}A = I$ where I is the identity matrix.
- If the inverse does not exist (possibly due to rounding errors), OpenMx will terminate with an error message. Some precautions can be taken to avoid this, such as supplying starting values that allow inversion, or putting boundary constraints on parameters to prevent their taking values that would lead to a singular matrix.

$$solve\begin{bmatrix} a & c \\ b & d \end{bmatrix} = \frac{1}{ad - cb}\begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$$

Transpose

- mxAlgebra(expression=t(*matrix_name*) , *name="A"),*
- Any matrix may be transposed. The transpose of A is written A'. The order of the matrix changes from r×c to c×r, as the rows become the columns and vice-versa.

$$t\begin{bmatrix}1 & 3\\ 2 & 4\end{bmatrix}=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$$

$$t\begin{bmatrix}1 & \\ 2 & 3\end{bmatrix}=\begin{bmatrix}1 & 2\\ & 3\end{bmatrix}$$

Power 1

- *mxAlgebra(expression=(A^2) , name="a"),*
- All the elements of a matrix may be raised to a power using the ^ symbol.
- When a matrix is raised to the power of a matrix, this operator works the same way as the Dot product (see below), but elements of the first matrix are raised to the power of those in the second matrix instead of multiplied by them.

$$\begin{bmatrix}a & c\\ b & d\end{bmatrix}\text{^}2=\begin{bmatrix}a^2 & c^2\\ b^2 & d^2\end{bmatrix}$$

$$\begin{bmatrix}a & c\\ b & d\end{bmatrix}\text{^}\begin{bmatrix}e & g\\ f & h\end{bmatrix}=\begin{bmatrix}a^e & c^g\\ b^f & d^h\end{bmatrix}$$

Power 2

- *mxAlgebra(expression= (A %^% 2), name="a"),*
- *Essentially, this operator works the same way as the Kronecker product (see below), but elements of the first matrix are raised to the power of those in the second matrix instead of multiplied by them. It is possible to use negative powers and non-integer exponents to indicate reciprocal functions and roots of elements, but it is not possible to raise a negative number to a non-integer power.*

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} \%\hat{}\% \begin{bmatrix} e & g \\ f & h \end{bmatrix} = \begin{bmatrix} a^e & a^g & c^e & c^g \\ a^f & a^h & c^f & c^h \\ b^e & b^g & d^e & d^g \\ b^f & b^h & d^f & d^h \end{bmatrix}$$

Multiplication

- *mxAlgebra(expression=(A %*% B), name="C"),*
- *%*% or 'Star' is the ordinary form of matrix multiplication (usually written as *).*
- *The elements of A(m×n) and B(n×p) are combined to form the elements of matrix C(m×p) using the formula* $C_{ij} = \sum_{k=1}^{n} A_{ik} \times B_{kj}$.
- *Matrices multiplied in this way must be conformable for multiplication. This means that the number of columns in the first matrix must equal the number of rows in the second matrix. For example, the matrix product A%*%B*

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \%*\% \begin{bmatrix} g & h \\ i & j \end{bmatrix} = \begin{bmatrix} a\times g + b\times i & a\times h + b\times j \\ c\times g + d\times i & c\times h + d\times j \\ e\times g + f\times i & e\times h + f\times j \end{bmatrix} = \begin{bmatrix} ag + bi & ah + bj \\ cg + di & ch + dj \\ ge + fi & eh + fj \end{bmatrix}$$

Dot product

- *mxAlgebra(expression=(A * B), name="C"),*
- *Dot is another type of matrix multiplication, which is done element by element (usually written as .).*
- *For two matrices to be multiplied in this way, they must have the same dimensions. Elements of the dot product are described by the formula Cij = Aij×Dij.*
- *For example, the dot product A*D is*

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} * \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} a\times g & b\times h \\ c\times i & d\times j \\ e\times k & f\times l \end{bmatrix}$$

Kronecker product

- *mxAlgebra(expression=(A %x% B), name="C");* ⊗
- The right Kronecker product of two matrices A B is formed by multiplying each element of A by the matrix B. If A is of order (m×n) and B is of order (p×q), then the result will be of order mp×nq.
- There are no conformability criteria for this type of product. ⊗
- For example A %x% B is

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \otimes \begin{bmatrix} g & h \\ i & j \end{bmatrix} = \begin{bmatrix} a\times g & a\times h & b\times g & b\times h \\ a\times i & a\times j & b\times i & b\times j \\ c\times g & c\times h & d\times g & d\times h \\ c\times i & c\times j & d\times i & d\times j \\ e\times g & e\times h & f\times g & f\times h \\ e\times i & e\times j & f\times i & f\times j \end{bmatrix}$$

Quadratic product

- *mxAlgebra(expression=(A %&% B), name="C"),*
- Many structural equation and other statistical models use quadratic products of the form ABA', and the quadratic operator (%&%) is both a simple and efficient way to implement quadratics. Note that E can be any shape, but to be conformable for quadratic product the matrix B must be square and have the same number of columns as the matrix E.
- For example, the quadratic product E%&%B

$$\begin{bmatrix} a & b \end{bmatrix} \%\&\% \begin{bmatrix} g & h \\ i & j \end{bmatrix} = \begin{bmatrix} a & b \end{bmatrix} \%*\% \begin{bmatrix} g & h \\ i & j \end{bmatrix} \%*\% \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a^2 g + abi + abd + b^2 i \end{bmatrix}$$

Element division

- *mxAlgebra(expression=(A / B), name="C"),*
- / does element by element division. For two matrices to be divided in this way, they must have the same dimensions. Elements of the result, C are described by the formula $\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{D}_{ij}$.
- For example, the division A/D is

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \quad / \quad \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} \quad = \quad \begin{bmatrix} a \div g & b \div h \\ c \div i & d \div j \\ e \div k & f \div l \end{bmatrix}$$

Addition

- *mxAlgebra(expression=(A + B), name="C"),*
- Addition of matrices is performed *element by element. For two matrices to be added, they* must have the same dimensions. Elements of the sum, C are described by the formula $\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{D}_{ij}$.
- For example

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} + \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} a+g & b+h \\ c+i & d+j \\ e+k & f+l \end{bmatrix}$$

Subtraction

- *mxAlgebra(expression=(A - B), name="C"),*
- Subtraction of matrices is performed *element by element. For one matrix to be subtracted* from another, they must have the same dimensions. Elements of the difference, C are described by the formula $\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{D}_{ij}$.
- For example

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} - \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} a-g & b-h \\ c-i & d-j \\ e-k & f-l \end{bmatrix}$$

Horizontal Adhesion

- *mxAlgebra(expression=cbind (A , B), name="C"),*
- cbind() allows partitioning of matrices. Its operation is called horizontal adhesion because cbind(A,D) is formed by sticking D onto the right hand side of A. For two matrices to be adhered in this way, they have to have the same number of rows. If A *(m×n) and D (m×p) are adhered, the* result C is of order *(m×(n+p)).*
- For example

$$cbind\left(\begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix},\begin{bmatrix}7\\ 8\\ 9\end{bmatrix}\right)=\begin{bmatrix}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{bmatrix}$$

Vertical Adhesion

- *mxAlgebra(expression=rbind (A , B), name="C"),*
- rbind() allows partitioning of matrices. Its operation is called vertical adhesion because rbind(A,D) is formed by sticking D underneath A. For two matrices to be adhered in this way, they must have the same number of columns. If A *(m×n) and D (p×n) are adhered, the* result C is of order *((m+p)×n).*
- For example

$$rbind\left(\begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix},\begin{bmatrix}5 & 6 & 7 & 8\end{bmatrix}\right)=\begin{bmatrix}1 & 2 & 3 & 4\\ 5 & 6 & 7 & 8\end{bmatrix}$$