Matrix - Basic

Definitions

Given N×M matrix A, then M×N matrix B is :

• *the transpose of A*	*if*	$B_{m,n}$	=	$A_{n,m}$	
• *the conjugate transpose of A*	*if*	$B_{m,n}$	=	$\overline{A_{n,m}}$	*(i.e. complex conjugate)*
• *the left inverse of A*	*if*	BA	=	I	*(note : left inverse is not denoted by* A^{-1}*)*
• *the right inverse of A*	*if*	AB	=	I	*(note : right inverse is not denoted by* A^{-1}*)*

Example that is positive definite but asymmetric :

$$[x \quad y]\begin{bmatrix} +1 & +1 \\ -1 & +1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = x^2 + xy - xy + y^2 = x^2 + y^2 > 0$$

Given N×N matrix A, then it is :

• *orthogonal*	*if*	AA^T	=	I	
• *symmetric*	*if*	A^T	=	A	
• *Hermitian*	*if*	A^T	=	$\overline{A}$	
• *positive definite*	*if*	XAX^T	>	0	*for all non zero 1×N row matrix X*
• *positive semi definite*	*if*	XAX^T	≥	0	*for all non zero 1×N row matrix X*
• *negative definite*	*if*	XAX^T	<	0	*for all non zero 1×N row matrix X*
• *negative semi definite*	*if*	XAX^T	≤	0	*for all non zero 1×N row matrix X*
• *symmetic positive definite*	*if*	*A is both symmetric*			*and positive definite*
• *symmetic positive semi definite*	*if*	*A is both symmetric*			*and positive semi definite*
• *symmetic negative definite*	*if*	*A is both symmetric*			*and negative definite*
• *symmetic negative semi definite*	*if*	*A is both symmetric*			*and negative semi definite*

Please note that :

- *product of diagonal matrices is also diagonal,*
- *product of upper triangular matrices is also upper triangular,*
- *product of lower triangular matrices is also lower triangular,*
- *product of orthonormal basis is also orthonormal basis,* *(read the proof in later section)*
- *product of symmetric matrix is* ***not necessarily symmetric****,*
- *product of positive definite matrix is* ***not necessary positive definite****.*

Given N×N matrix A, then :

• *minor of A is*	$M_{n,m}$	=	$\det(A_{n,m})$	*where* $A_{n,m}$ *is submatrix without row n & column m*
• *cofactor of A is*	$C_{n,m}$	=	$(-1)^{n+m} M_{n,m}$	
• *cofactor matrix of A is*	C	=	$(C_{n,m})_{n,m\in[1,N]}$	
• *adjugate matrix of A is*	$adj(A)$	=	C^T	

Given N×N matrix A, then the followings are equivalent.

- *A is invertible.*
- *A is non singular.*
- *A has non zero determinant.*
- *A can be written as a row echelon form.*
- *A can be written as a product of elementary matrices.*
- *There is exactly one solution for AX = B.*
- *There is exactly one solution for AX = 0, which is trival solution.*

If A is linearly dependent, then there exists (inf) non trival solution to AX = 0.
If A is linearly independent, then there exists only trival solution to AX = 0.

Property of transpose and inverse

	$(A^T)^T$	=	A	$(A^{-1})^{-1}$	=	A	
	$(A+B)^T$	=	$A^T + B^T$	$(A+B)^{-1}$	=	$A^{-1} - (I + A^{-1}B)^{-1}A^{-1}BA^{-1}$	*(proof in the next section)*
	$(AB)^T$	=	$B^T A^T$	$(AB)^{-1}$	=	$B^{-1}A^{-1}$	
	$(cA)^T$	=	$c(A^T)$	$(cA)^{-1}$	=	A^{-1}/c	
	$\det(A^T)$	=	$\det(A)$	$\det(A^{-1})$	=	$1/\det(A)$	
	$(A^T)^{-1}$	=	$(A^{-1})^T$				
proof	$(A^{-1})^T A^T$	=	$(AA^{-1})^T$	=	I		

Inverse of matrix sum

$$(A+B)^{-1} = A^{-1} - (I + A^{-1}B)^{-1}A^{-1}BA^{-1}$$

(Proof) We start with :

$$\begin{aligned}
(I+P)^{-1} &= (I+P)^{-1}(I+P-P) \\
&= (I+P)^{-1}(I+P) - (I+P)^{-1}P \\
&= I - (I+P)^{-1}P \\
(A+B)^{-1} &= (AI + AA^{-1}B)^{-1} \\
&= (A(I + A^{-1}B))^{-1} \\
&= (I + A^{-1}B)^{-1}A^{-1} \\
&= (I - (I + A^{-1}B)^{-1}A^{-1}B)A^{-1} \qquad \textit{by putting } P = A^{-1}B \textit{ into } I - (I+P)^{-1}P \\
&= A^{-1} - (I + A^{-1}B)^{-1}A^{-1}BA^{-1}
\end{aligned}$$

Inverse of matrix block

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

(Proof) Suppose we have :

$$\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

$$\begin{aligned}
X_{11} &= AA^{-1} + AA^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} - B(D - CA^{-1}B)^{-1}CA^{-1} \\
&= I + B(D - CA^{-1}B)^{-1}CA^{-1} - B(D - CA^{-1}B)^{-1}CA^{-1} \\
&= I \\
X_{12} &= -AA^{-1}B(D - CA^{-1}B)^{-1} + B(D - CA^{-1}B)^{-1} \\
&= -B(D - CA^{-1}B)^{-1} + B(D - CA^{-1}B)^{-1} \\
&= 0 \\
X_{21} &= CA^{-1} + CA^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} - D(D - CA^{-1}B)^{-1}CA^{-1} \\
&= CA^{-1} + (CA^{-1}B - D)(D - CA^{-1}B)^{-1}CA^{-1} \\
&= CA^{-1} - CA^{-1} \\
&= 0 \\
X_{22} &= -CA^{-1}B(D - CA^{-1}B)^{-1} + D(D - CA^{-1}B)^{-1} \\
&= (D - CA^{-1}B)(D - CA^{-1}B)^{-1} \\
&= I
\end{aligned}$$

Sherman Morrison formula

Sherman Morrison formula states that if u and v are row vectors, $u^T v$ is the outer product, then :

$$(A + \beta u^T v)^{-1} = A^{-1} - \frac{\beta A^{-1}u^T vA^{-1}}{1 + \beta vA^{-1}u^T} \qquad \textit{Note : } 1 + \beta vA^{-1}u^T \textit{ is a scalar.}$$

(Proof) By direct multiplication :

$$\begin{aligned}
(A + \beta u^T v)\left(A^{-1} - \frac{\beta A^{-1}u^T vA^{-1}}{1 + \beta vA^{-1}u^T}\right) &= (A + \beta u^T v)A^{-1} - (A + \beta u^T v)\frac{\beta A^{-1}u^T vA^{-1}}{1 + \beta vA^{-1}u^T} \\
&= I + \beta u^T vA^{-1} - \frac{\beta u^T vA^{-1} + \beta^2 u^T vA^{-1}u^T vA^{-1}}{1 + \beta vA^{-1}u^T} \\
&= I + \beta u^T vA^{-1} - \frac{\beta u^T(1 + \beta vA^{-1}u^T)vA^{-1}}{1 + \beta vA^{-1}u^T} \\
&= I + \beta u^T vA^{-1} - \frac{(1 \cancel{+ \beta vA^{-1}u^T})\beta u^T vA^{-1}}{1 \cancel{+ \beta vA^{-1}u^T}} \qquad \textit{Trick : } 1 + \beta vA^{-1}u^T \textit{ is a scalar.} \\
&= I + \beta u^T vA^{-1} - \beta u^T vA^{-1} \\
&= I \qquad \textit{This is useful in recursive least square.}
\end{aligned}$$

Matrix differentiation

<u>*Jacobian, Hessian and Taylor series*</u>
Given function $F : \Re^M \to \Re^N$,

$$F(X) = \begin{bmatrix} f_1(X) \\ f_2(X) \\ f_3(X) \\ \dots \\ f_N(X) \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, \dots, x_M) \\ f_2(x_1, x_2, \dots, x_M) \\ f_3(x_1, x_2, \dots, x_M) \\ \dots \\ f_N(x_1, x_2, \dots, x_M) \end{bmatrix} \quad \textit{where} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_M \end{bmatrix}$$

Jacobian is defined as :

$$J(F(X)) = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 & \partial f_1/\partial x_3 & \dots & \partial f_1/\partial x_M \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 & \partial f_2/\partial x_3 & \dots & \partial f_2/\partial x_M \\ \partial f_3/\partial x_1 & \partial f_3/\partial x_2 & \partial f_3/\partial x_3 & \dots & \partial f_3/\partial x_M \\ \dots & \dots & \dots & \dots & \dots \\ \partial f_N/\partial x_1 & \partial f_N/\partial x_2 & \partial f_N/\partial x_3 & \dots & \partial f_N/\partial x_M \end{bmatrix} \quad \textit{which is 1st order derivative}$$

When M=1, Hessian is defined as :

$$H(f(X)) = \begin{bmatrix} \partial^2 f/\partial x_1 \partial x_1 & \partial^2 f/\partial x_1 \partial x_2 & \partial^2 f/\partial x_1 \partial x_3 & \dots & \partial^2 f/\partial x_1 \partial x_M \\ \partial^2 f/\partial x_2 \partial x_1 & \partial^2 f/\partial x_2 \partial x_2 & \partial^2 f/\partial x_2 \partial x_3 & \dots & \partial^2 f/\partial x_2 \partial x_M \\ \partial^2 f/\partial x_3 \partial x_1 & \partial^2 f/\partial x_3 \partial x_2 & \partial^2 f/\partial x_3 \partial x_3 & \dots & \partial^2 f/\partial x_3 \partial x_M \\ \dots & \dots & \dots & \dots & \dots \\ \partial^2 f/\partial x_M \partial x_1 & \partial^2 f/\partial x_M \partial x_2 & \partial^2 f/\partial x_M \partial x_3 & \dots & \partial^2 f/\partial x_M \partial x_M \end{bmatrix} \quad \textit{which is 2nd order derivative}$$

Please note that Jacobian is $N \times M$ matrix, while Hessian is $M \times M$ symmetric matrix. Besides, don't confuse Hessia with Hermitian. When M=1, we can rewrite Taylor series in terms of Jacobian and Hessian as :

$$f(X + \Delta X) = f(X) + J(f(X))\Delta X + \frac{1}{2}\Delta X^T H(f(X))\Delta X$$
$$J(f(X)) = 1 \times M \text{ row matrix}$$
$$H(f(X)) = M \times M \text{ matrix}$$

<u>*Two fundalmental derivative formulae*</u>
Suppose X is a $M \times 1$ column matrix, F is a $N \times 1$ column matrix, G is a $L \times 1$ column matrix. F and G are functions that map from M to N dimensional space and from M to L dimensional space respectively. Derivative of scalar s with respect to X and derivative of column vector v (with size $N \times 1$) with respect to X are :

$$s = F^T AG + B \qquad \frac{ds}{dX} = (F^T A)J_G + (AG)^T J_F = F^T AJ_G + G^T A^T J_F \qquad \textit{formula for scalar}$$

$$v = AG + B \qquad \frac{dv}{dX} = AJ_G \qquad \textit{formula for vector}$$

where :

$$\frac{dF}{dX} = J_F \qquad N \times M \qquad F : \Re^M \to \Re^N \qquad \text{dependent on } X$$

$$\frac{dG}{dX} = J_G \qquad L \times M \qquad G : \Re^M \to \Re^L \qquad \text{dependent on } X$$

$$\frac{dX}{dX} = I \qquad M \times M$$

$$\frac{dC}{dX} = 0 \qquad N \times M \qquad C : \Re^M \to \Re^N \qquad \text{independent on } X$$

Here is a summary of specialised cases. The derivative of s follows the convention of Jacobian matrix, hence it is a $1 \times M$ row matrix, however it is sometimes more convenient to be expressed as a $M \times 1$ column matrix in optimization algorithms and regression algorithms, in that case, we need to take transpose on the result matrix.

	scalar s	F	A	G	ds/dX	non symmetric A	symmetric A (N=L)	identity A (N=L)
general	$F^T AG + B$	$N,1$	N,L	$L,1$	$1,M$	$F^T AJ_G + G^T A^T J_F$	$F^T AJ_G + G^T AJ_F$	$F^T J_G + G^T J_F$
$F = G$	$F^T AF + B$	$N,1$	N,N	$N,1$	$1,M$	$F^T (A + A^T) J_F$	$2F^T AJ_F$	$2F^T J_F$
$F = X$	$X^T AG + B$	$M,1$	M,L	$L,1$	$1,M$	$X^T AJ_G + G^T A^T$	$X^T AJ_G + G^T A$	$X^T J_G + G^T$
$G = X$	$F^T AX + B$	$N,1$	N,M	$M,1$	$1,M$	$F^T A + X^T A^T J_F$	$F^T A + X^T AJ_F$	$F^T + X^T J_F$
$F = G = X$	$X^T AX + B$	$M,1$	M,M	$M,1$	$1,M$	$X^T (A + A^T)$	$2X^T A$	$2X^T$
$F = I$	$AG + B$	-	$1,L$	$L,1$	$1,M$	AJ_G	-	-
$F = I, G = X$	$AX + B$	-	$1,M$	$M,1$	$1,M$	A	-	-
$G = I$	$F^T A + B$	$N,1$	$N,1$	-	$1,M$	$A^T J_F$	-	-
$G = I, F = X$	$X^T A + B$	$M,1$	$M,1$	-	$1,M$	A^T	-	-

	vector v	A	G	dv/dX	derivative	
general	$AG + B$	N,L	$L,1$	N,M	AJ_G	(the same as above when N = 1)
$G = X$	$AX + B$	N,M	$M,1$	N,M	A	(the same as above when N = 1)

where derivative ds/dX can either be expressed as a row matrix or a column matrix :

- *use row matrix if you want to have the same convention as Jacobian*
- *use column matrix if you want to use it in algorithms (e.g. least square, Newton, Gauss Newton etc)*

Intuition of matrix product

Matrix can be classified into the following types :

- *row data matrix*
- *column data matrix*
- *linear transformation*
- *linear combination*

Suppose A is a data matrix with size Y×X, then it denotes Y row vectors, each belongs to X dimensional space, if A is treated as a row data matrix, and it denotes X column vectors, each belongs to Y dimensional space, if A is treated as a column data matrix. In general, A can be interpreted as both row and column data matrix.

If A = $Y \times X$ *data matrix*
= *row data matrix*

then $$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \dots \\ A_Y \end{bmatrix}$$

where A_y = $1 \times X$ *row vector*

If A = $Y \times X$ *data matrix*
= *column data matrix*

$$A = \begin{bmatrix} A_1 & A_2 & A_3 & \dots & A_X \end{bmatrix}$$

where A_x = $Y \times 1$ *column vector*

Here is the convention adopted in my documents, when A is a row data matrix, then it has a size of N×M, when A is a column data matrix, then it has a size of M×N, both represent N vectors in M dimensional space (N can be greater than or smaller than M).

Please differentiate the following notations :

- *row data matrix* — *rectangular matrix with one row as a vector (each row is a **row vector**)*
- *row matrix* — *rectangular matrix with one row only*
- *row vector* — *a row in a **row data matrix***
- *column data matrix* — *rectangular matrix with one column as a vector (each column is a **column vector**)*
- *column matrix* — *rectangular matrix with one column only*
- *column vector* — *a column in a **column data matrix***

*Please note that we **never** classify linear transformation (and linear combination) into row or column matrix. Different meanings for matrix product are classified as the following :*

- *product between a data matrix and a linear transformation (include : shearing, rotation, dimension scaling)*
- *product between a data matrix and a linear combination (include : data scaling, permutation, elementary addition)*
- *product between two data matrices as inner product*
- *product between two data matrices as covariance*

If	A	=	$N \times M$ *row data matrix*	*If*	A	=	$M \times N$ *column data matrix*
then	A	=	$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \dots \\ A_N \end{bmatrix}$		A	=	$\begin{bmatrix} A_1 & A_2 & A_3 & \dots & A_N \end{bmatrix}$
where	A_n	=	$1 \times M$ *row vector*	*where*	A_n	=	$M \times 1$ *column vector*

Part 1 : Linear transformation $\Re^M \to \Re^K$

$$A' = AF \qquad\qquad A' = FA$$

then $A' = \begin{bmatrix} A'_1 \\ A'_2 \\ A'_3 \\ \dots \\ A'_N \end{bmatrix}$ *then* $A' = \begin{bmatrix} A'_1 & A'_2 & A'_3 & \dots & A'_N \end{bmatrix}$

A'_n	=	$1 \times K$ *row vector*	A'_n	=	$K \times 1$ *column vector*
	=	*linear transformation of* ***one*** A_n		=	*linear transformation of* ***one*** A_n
F	=	$M \times K$ *linear transformation*	F	=	$K \times M$ *linear transformation*
	=	*performs mapping* $\Re^M \to \Re^K$		=	*performs mapping* $\Re^M \to \Re^K$

Special cases for both row data matrix A and column data matrix A

- *when K = M* $\Rightarrow$ *perform shearing in $\Re^M \to \Re^M$*
- *when K = M and F is orthonormal* $\Rightarrow$ *perform rotation with no dimensional scaling*
- *when K = M and F is diagonal* $\Rightarrow$ *perform dimension scaling with different scales $f_{m,m}$ (remark 1)*

If F is orthonormal, it can always be written as a matrix with cosines and sines (i.e. rotation).
For example, if F denotes a rotation θ in xy plane, followed by a rotation ϕ in yz plane, we have :

$$F = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta\cos\phi & \sin\theta\sin\phi \\ \sin\theta & \cos\theta\cos\phi & \cos\theta\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

Part 2 : Linear combination

$$A' = WA \qquad\qquad A' = AW$$

then $A' = \begin{bmatrix} A'_1 \\ A'_2 \\ A'_3 \\ \dots \\ A'_K \end{bmatrix}$ *then* $A' = \begin{bmatrix} A'_1 & A'_2 & A'_3 & \dots & A'_K \end{bmatrix}$

A'_k	=	*1×M row vector*	A'_k	=	*M×1 column vector*
	=	*linear combination of* ***all*** A_n		=	*linear combination of* ***all*** A_n
		(weighted by $w_{k,n}$ *)*			*(weighted by* $w_{n,k}$ *)*
W	=	$K \times N$ *linear combination*	W	=	$N \times K$ *linear combination*

Special cases for both row data matrix A or column data matrix A

- *when K = N* $\Rightarrow$ *generate N row (or column) vectors from N row (or column) vectors*
- *when K = N and W is diagonal* $\Rightarrow$ *scaling N row (or column) vectors by $w_{n,n}$ (remark 1)*
- *when K = N and W is permutation* $\Rightarrow$ *equivalent to row (or column) permutation*
- *when K = N and W is elem addition* $\Rightarrow$ *equivalent to elementary row (or column) addition*
- *when K $\neq$ N and W is a delta* $\Rightarrow$ *equivalent to picking a desired row (or column) from A*

Exampe 1 : W = permutation matrix (please read 'permutation' document for definition of permutation matrix)

$$A' = P_{\sigma_1}P_{\sigma_2}P_{\sigma_3}...P_{\sigma_T}A \quad = \text{ cascaded permutation } (\sigma_T \circ ... \circ \sigma_3 \circ \sigma_2 \circ \sigma_1) \text{ on row data matrix } A$$

$$A' = AP^T_{\sigma_T}...P^T_{\sigma_3}P^T_{\sigma_2}P^T_{\sigma_1} \quad = \text{ cascaded permutation } (\sigma_T \circ ... \circ \sigma_3 \circ \sigma_2 \circ \sigma_1) \text{ on column data matrix } A$$

Exampe 2 : W = elementary addition operation (please read 'elementary operation' document for definitions of E & F)

$$A' = E_T E_{T-1}....E_3E_2E_1A \quad = \text{ cascaded elementary row addition on row data matrix } A$$

$$A' = AF_1F_2F_3....F_{T-1}F_T \quad = \text{ cascaded elementary column addition on column data matrix } A$$

Example 3 : W = delta matrix

$$A' = DA$$

$$= \begin{bmatrix} \delta_{\pi(1)} \\ \delta_{\pi(2)} \\ \delta_{\pi(3)} \\ ... \\ \delta_{\pi(K)} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ ... \\ A_N \end{bmatrix} = \begin{bmatrix} A_{\pi(1)} \\ A_{\pi(2)} \\ A_{\pi(3)} \\ ... \\ A_{\pi(K)} \end{bmatrix} \quad = \text{ picking the } k^{th} \text{ row from row data matrix } A$$

$$\delta_x = [0 \quad ... \quad 1 \quad ... \quad 0] \quad = 1\times N \text{ row matrix with value one at position } x\text{, and value zero otherwise}$$

$$A' = AD$$

$$= [A_1 \quad A_2 \quad A_3 \quad ... \quad A_N]\times$$

$$[\delta_{\pi(1)} \quad \delta_{\pi(2)} \quad ... \quad \delta_{\pi(K)}]$$

$$= [A_{\pi(1)} \quad A_{\pi(2)} \quad ... \quad A_{\pi(K)}] \quad = \text{ picking the } k^{th} \text{ column from column data matrix } A$$

$$\delta_x = \begin{bmatrix} 0 \\ ... \\ 1 \\ ... \\ 0 \end{bmatrix} \quad = N\times 1 \text{ column matrix with value one at position } x\text{, and value zero otherwise}$$

Remark 1 : Dimension scaling vs data scaling
Both linear transformation and linear combination become scaling when F (or W) is diagonal, yet they represent different scaling : the former means ***dimension scaling****, while the latter means* ***data scaling****.*

- *scaling in linear transformation* ⇒ $f_{m,m}$ *= scale for the* m^{th} *dimension in A* *(called dimension scaling)*
- *scaling in linear combination* ⇒ $w_{n,n}$ *= scale for the* n^{th} *data in A* *(called data scaling)*

Remark 2 : Cascade of linear transformation and linear combination
Both linear transformation and linear combination can be cascaded. Please note that linear transformation matrix and linear combination matrix are different in associativity for row data matrix A and column data matrix A. Suppose we perform T linear transformations in sequence, and if the dimension after the t^{th} *linear transformation is* M_t*, then we have :*

$A' = AF_1F_2F_3...F_T$ — *for row data matrix A (left associative)* — F_1 *is* $M\times M_1$ *and* F_t *is* $M_{t-1}\times M_t$

$A' = F_T...F_3F_2F_1A$ — *for column data matrix A (right associative)* — F_1 *is* $M_1\times M$ *and* F_t *is* $M_t\times M_{t-1}$

Suppose we perform T linear transformations in sequence, and if the number of data after the t^{th} *linear combination is* N_t*, then we have :*

$A' = W_T...W_3W_2W_1A$ — *for row data matrix A (right associative)* — W_1 *is* $N_1\times N$ *and* W_t *is* $N_t\times N_{t-1}$

$A' = AW_1W_2W_3...W_T$ — *for column data matrix A (left associative)* — W_1 *is* $N\times N_1$ *and* W_t *is* $N_{t-1}\times N_t$

Part 3 : Inner product (projection)

Suppose A and B are both N×M row data matrices, while W is a M×M weight matrix (different weights for different dimensions), then we have inner product defined as :

$$X = AWB^T = \langle A, B \rangle_W$$

$$x_{n_1,n_2} = \sum_{m=1}^{M} w_m a_{n_1,m} b_{n_2,m} \qquad \textit{if W is diagonal}$$

Suppose A and B are both M×N column data matrices, while W is a M×M weight matrix (different weights for different dimensions), then we have inner product defined as :

$$X = A^T WB = \langle A, B \rangle_W$$

$$x_{n_1,n_2} = \sum_{m=1}^{M} w_m a_{m,n_1} b_{m,n_2} \qquad \textit{if W is diagonal}$$

Inner product *is usually used to find* ***projection or orthogonality*** *between two vectors.*

Part 4 : Outer product (covariance matrix)

Suppose A and B are both N×M row data matrices, while W is a N×N weight matrix (different weights for different data or observations), then we have outer product defined as :

$$X = A^T WB$$

$$x_{m_1,m_2} = \sum_{n=1}^{N} w_n a_{n,m_1} b_{n,m_2} \qquad \textit{if W is diagonal}$$

Suppose A and B are both M×N column data matrices, while W is a N×N weight matrix (different weights for different data or observations), then we have outer product defined as :

$$X = AWB^T$$

$$x_{m_1,m_2} = \sum_{n=1}^{N} w_n a_{m_1,n} b_{m_2,n} \qquad \textit{if W is diagonal}$$

Outer product *is usually used to find* ***covariance matrix*** *between two dimensions (or between two random variables).*

Part 5 : Replication and summation by all one matrix (a special case of linear combination)

Row matrix A with size 1×M can be replicated to generate row data matrix B with size N×M by B = lA, where l is N×1 all one column matrix, and $B_n = A$ for all $n\in[1,N]$. Column matrix A with size M×1 can be replicated to generate column data matrix B with size M×N by $B = Al^T$, where l^T is 1×N all one row matrix, and $B_n = A$ for all $n\in[1,N]$.

Given N×M row data matrix A, the sum of row data is given by l^TA. Given M×N column data matrix A, the sum of column data is given by Al. Given a square matrix weight matrix W (not necessarily diagonal), then l^TWl gives the sum of weight.

$$l^T A = \sum_{n=1}^{N} A_n \qquad \textit{for row data matrix A}$$

$$Al = \sum_{n=1}^{N} A_n \qquad \textit{for column data matrix A}$$

$$l^T Wl = \sum_{n=1}^{N} \sum_{m=1}^{N} w_{n,m}$$

<u>*More about covariance matrix*</u>

Suppose A is a N×M row data matrices, then we have covariance defined as :

$$C = \frac{(A - l\bar{A})^T W (A - l\bar{A})}{l^T W l}$$ *if W is diagonal, where w_n = weight of row vector A_n*

$$c_{m_1,m_2} = \text{cov}(m_1, m_2)$$ *covariance between dimension m_1 and m_2*

where $$\bar{A} = \frac{l^T W A}{l^T W l} = \frac{\sum_{n=1}^{N} w_n A_n}{\sum_{n=1}^{N} w_n}$$ *average of all A_n, which is a 1×M row matrix*

Suppose A is a M×N column data matrices, then we have covariance defined as :

$$C = \frac{(A - \bar{A} l^T) W (A - \bar{A} l^T)^T}{l^T W l}$$ *if W is diagonal, where w_n = weight of column vectorA_n*

$$c_{m_1,m_2} = \text{cov}(m_1, m_2)$$ *covariance between dimension m_1 and m_2*

where $$\bar{A} = \frac{A W l}{l^T W l} = \frac{\sum_{n=1}^{N} w_n A_n}{\sum_{n=1}^{N} w_n}$$ *average of all A_n, which is a M×1 column matrix*

Lets simplify.

$$C = \frac{(A - 1\bar{A})^T W (A - 1\bar{A})}{1^T W 1}$$

$$= \frac{A^T W A - A^T W 1 \bar{A} - \bar{A}^T 1^T W A + \bar{A}^T 1^T W 1 \bar{A}}{1^T W 1}$$

$$= \frac{A^T W A - \bar{A}^T 1^T W 1 \bar{A}}{1^T W 1}$$ *since* $A^T W 1 \bar{A} = \bar{A}^T 1^T W A = \bar{A}^T 1^T W 1 \bar{A}$ *(1)*

$$= \frac{A^T W A}{1^T W 1} - \bar{A}^T \bar{A}$$ *since* $1^T W 1$ *is a scalar and thus* $\bar{A}^T 1^T W 1 \bar{A} = (1^T W 1)(\bar{A}^T \bar{A})$

Equation (1) can be proved by :

$$\bar{A} = \frac{1^T W A}{1^T W 1} \Rightarrow 1^T W A = (1^T W 1)\bar{A} \quad (2)$$

and $$\bar{A}^T = \frac{A^T W 1}{1^T W 1} \Rightarrow A^T W 1 = \bar{A}^T (1^T W 1) \quad (3)$$

Therefore (1) is proved :

$$\bar{A}^T 1^T W A = \bar{A}^T 1^T W 1 \bar{A}$$ *using (2)*

$$A^T W 1 \bar{A} = \bar{A}^T 1^T W 1 \bar{A}$$ *using (3)*

Orthonormal basis

What are independent, orthogonal and orthonormal? Suppose X and Y are two $1\times N$ row vectors, then

- *X and Y are independent if* $Y \neq cX$
- *X and Y are orthogonal if* $XY^T = 0$
- *X and Y are orthonormal if* $XY^T = 0$ *and* $XX^T = YY^T = 1$

A set of independent vectors which spans a space $\Re^N$ is known as the ***basis*** *of the space, while a set of orthonormal vectors which spans a space $\Re^N$ is known as the* ***orthonormal basis*** *of the space. Suppose Q is a $N\times N$ matrix denoting orthonormal basis, then we have :*

$$QQ^T = I \qquad \text{for row vectors in } Q \qquad \text{(known as row orthonormal)}$$
$$Q^TQ = I \qquad \text{for column vectors in } Q \qquad \text{(known as column orthonormal)}$$

Inverse of orthonormal

Suppose Q is a $N\times N$ orthonormal basis, with row vectors, i.e.

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \cdots \\ Q_N \end{bmatrix} \qquad \text{where } Q_n \text{ is row vector } \forall n\in[1,N]$$

$$\begin{aligned} QQ^T &= (Q_nQ_m^T)_{n,m\in[1,N]} \\ &= (\delta_{n,m})_{n,m\in[1,N]} \\ &= I \\ \Rightarrow \quad Q^{-1} &= Q^T \end{aligned}$$

Suppose Q is a $N\times N$ orthonormal basis, with column vectors, i.e.

$$Q = [Q_1 \quad Q_2 \quad Q_3 \quad \ldots \quad Q_N] \qquad \text{where } Q_n \text{ is column vector } \forall n\in[1,N]$$

$$\begin{aligned} Q^TQ &= (Q_n^TQ_m)_{n,m\in[1,N]} \\ &= (\delta_{n,m})_{n,m\in[1,N]} \\ &= I \\ \Rightarrow \quad Q^{-1} &= Q^T \end{aligned}$$

If Q is row orthonormal basis, then Q must be column orthonormal basis, and vice versa. The proof is simple.

$$QQ^{-1} = Q^{-1}Q = I \qquad \text{since Q is orthonormal, lets put } Q^{-1} = Q^T$$
$$\Rightarrow \quad QQ^T = Q^TQ = I \qquad \text{hence we have : row orthonormal} \Leftrightarrow \text{column orthonormal}$$

Besides the inverse of orthonormal is also orthonormal, since :

$$\begin{aligned} (Q^{-1})(Q^{-1})^T &= (Q^{-1})(Q^T)^T && \text{since Q is orthonormal, i.e. } Q^{-1} = Q^T \\ &= (Q^{-1})Q \\ &= I && \text{hence we have : Q is orthonormal} \Leftrightarrow Q^{-1} \text{ is orthonormal} \end{aligned}$$

Hence these are equivalent : Q is row orthonormal, Q is column orthonormal, inverse Q is orthonormal. ***(Note : we don't need to specify whether Q is row orthonormal or column orthonormal, as they are equivalent.)***

Product between two different orthonormal basis

Suppose U and V are two orthonormal basis, lets prove that Q is also an orthonormal basis.

$$\begin{aligned} Q &= UV && \text{note : there is no transpose} \\ QQ^T &= UV(UV)^T \\ &= UVV^TU^T \\ &= UU^T && \text{since V is orthonormal : } VV^T = I \\ &= I && \text{since U is orthonormal : } UU^T = I \end{aligned}$$

Hence the product of an orthonormal basis with another orthonormal basis is also an orthonormal basis. This property will be used twice in deriving the QR algorithm.

Projection matrix

Lets consider the column vector case (since it matches with the convention in least square). Suppose A is a $M\times N$ column data matrix (i.e. N vectors in $\Re^M$), and B is a $M\times 1$ column data matrix in the same space, then the projection of B on the space spanned by the columns of A is P_AB, where P_A is called the projection matrix :

$$\begin{aligned} proj_A(B) &= P_AB \\ P_A &= A(A^TA)^{-1}A^T \end{aligned}$$

The proof involves two steps : (1) prove that P_AB lies in the column span of A (i.e. show that there exists W, such that linear combination AW equals to P_AB) and (2) vector $B-proj_A(B)$ is orthogonal to all columns in A (i.e. $A^T(B-proj_A(B))=0$).

(1)
$$\begin{aligned} P_AB &= A(A^TA)^{-1}A^TB \\ &= AW \end{aligned}$$
where $W = (A^TA)^{-1}A^TB$ $\quad\Rightarrow$ *P_AB lies in the column span of A*

(2)
$$\begin{aligned} A^T(B-P_AB) &= A^T(B-A(A^TA)^{-1}A^TB) \\ &= A^TB-A^TA(A^TA)^{-1}A^TB) \\ &= A^TB-A^TB \\ &= 0 \end{aligned}$$
$\Rightarrow$ B-*P_AB is orthogonal to all columns of A*

Besides, there is one more requirement on projection matrix : projection of projection is itself, i.e.

$$\begin{aligned} proj_A(proj_A(B)) &= proj_A(B) \\ P_AP_A...P_AB &= P_AP_AB = P_AB \end{aligned}$$
i.e. $P_AP_A...P_A = P_AP_A = P_A$

$$\begin{aligned} \Rightarrow\quad P_AP_A...P_A &= \underbrace{(A(A^TA)^{-1}A^T)(A(A^TA)^{-1}A^T)...(A(A^TA)^{-1}A^T)}_{T} \\ &= (A(A^TA)^{-1}A^TA(A^TA)^{-1}A^T)\underbrace{...(A(A^TA)^{-1}A^T}_{T-2}) \\ &= \underbrace{(A(A^TA)^{-1}A^T)...(A(A^TA)^{-1}A^T)}_{T-1} = A(A^TA)^{-1}A^T \end{aligned}$$

In general A is not an orthogonal basis. What happens if A is an orthogonal basis? In this case, A^TA is a diagonal matrix :

$$\begin{aligned} A^TA &= diag(\lambda_1,\lambda_2,...\lambda_N) && \textit{which is } N\times N \quad \textit{where } \lambda_n = A_n{}^TA_n = mag(A_n) \\ (A^TA)^{-1} &= diag(\lambda_1^{-1},\lambda_2^{-1},...\lambda_N^{-1}) && \textit{which is } N\times N \\ (A^TA)^{-1}A^TB &= diag(\lambda_1^{-1},\lambda_2^{-1},...\lambda_N^{-1})A^TB && \textit{which is } N\times 1 \\ ((A^TA)^{-1}A^TB)_n &= (A^TB)_n/\lambda_n \\ &= (A_n{}^TA_n)^{-1}(A^TB)_n \\ &= (A_n{}^TA_n)^{-1}(A_n{}^TB) \end{aligned}$$

where $A_n{}^TB$ = *inner product*

$(A_n{}^TA_n)^{-1}(A_n{}^TB)$ = *inner product with normalization*

Similarly, we can derive the projection for row vectors. Here is a summary.

<u>Summary</u>

For row data matrix A and single row data B, then the projection of B on space spanned by the rows of A is :

$$\begin{aligned} proj_A(B) &= BP_A \\ &= BA^T(AA^T)^{-1}A \end{aligned}$$
where $P_A = A^T(AA^T)^{-1}A$

For column data matrix A and single column data B, then the projection of B on space spanned by the columns of A is :

$$\begin{aligned} proj_A(B) &= P_AB \\ &= A(A^TA)^{-1}A^TB \end{aligned}$$
where $P_A = A(A^TA)^{-1}A^T$