Matrix Decomposition

Introduction

Decomposition	*definition*	*A*	*output*
LU decomposition	$A = LU$	*square*	*triangular LU*
LU decomposition with partial pivoting	$PA = LU$	*square*	*triangular LU*
LU decomposition with full pivoting	$PAQ = LU$	*square*	*triangular LU*
LDU decomposition	$A = LDU$	*square*	*triangular LU and diagonal D*
Cholesky decomposition	$A = LL^T = U^T U$	*symmetric*	*triangular LU*
Cholesky decomposition without sqrt	$A = LDL^T = U^T DU$	*symmetric*	*triangular LU and diagonal D*
QR decomposition	$A = QR$	*rectangular*	*orthonormal Q and triangular R*
LQ decomposition	$A = LQ$	*rectangular*	*orthonormal Q and triangular L*
SVD decomposition	$A = USV^T$	*rectangular*	*orthonormal U,V & diagonal S*
Eigen decomposition	$A = X\Lambda X^T$	*square*	*diagonal Λ*
		symmetric	*diagonal Λ and orthonormal X*

Rank decomposition

Suppose A is a Y×X matrix, the row span (column span) and row null space (column null space) are defined as :

$$span_{row}(A) = \{wA \quad s.t. \quad w = 1\times Y \quad row\_matrix\}$$ *i.e. set of linear combination of A's rows*

$$span_{column}(A) = \{Aw \quad s.t. \quad w = X\times 1 \quad column\_matrix\}$$ *i.e. set of linear combination of A's columns*

$$null_{row}(A) = \{z \quad s.t. \quad Az = 0 \quad where \quad z = X\times 1 \quad column\_matrix\}$$ *i.e. set of z that is orthogonal to A's rows*

$$null_{column}(A) = \{z \quad s.t. \quad zA = 0 \quad where \quad z = 1\times Y \quad row\_matrix\}$$ *i.e. set of z that is orthogonal to A's columns*

$$rank_{row}(A) = Dim(span_{row}(A))$$
$$rank_{columm}(A) = Dim(span_{column}(A))$$
$$nullity_{row}(A) = Dim(null_{row}(A))$$
$$nullity_{columm}(A) = Dim(null_{column}(A))$$

Row rank decomposition

Suppose A is a Y×X matrix (by treating A as row data matrix in X dimensional space), row rank decomposition is :

$$\begin{bmatrix} A_1 \\ A_2 \\ \dots \\ A_Y \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,K} \\ w_{2,1} & w_{2,2} & \dots & w_{2,K} \\ \dots & \dots & \dots & \dots \\ w_{Y,1} & w_{Y,2} & \dots & w_{Y,K} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ \dots \\ R_K \end{bmatrix}$$

where A_y and R_k are 1×X row vectors

s.t.
$$\begin{aligned} span_{row}(A) &= \{uA \quad s.t. \quad u = 1\times Y \quad row\_matrix\} \\ &= \{vR \quad s.t. \quad v = 1\times K \quad row\_matrix\} \\ &= span_{row}(R) \end{aligned}$$

and there exists no W (except δ_k) such that R_k = WR (or in a formal way, there exists no non trival solution to WR = 0). Thus row data matrix A is decomposed into a linearly independent (but not necessarily orthogonal) basis R (with K row vectors in X dimensional space), having the same row span as A, while K is called row rank.

Column rank decomposition

Suppose A is a Y×X matrix (by treating A as column data matrix in Y dimensional space), column rank decomposition is :

$$\begin{bmatrix} A_1 & A_2 & \dots & A_X \end{bmatrix} = \begin{bmatrix} C_1 & C_2 & \dots & C_L \end{bmatrix} \begin{bmatrix} w'_{1,1} & w'_{2,1} & \dots & w'_{X,1} \\ w'_{1,2} & w'_{2,2} & \dots & w'_{X,2} \\ \dots & \dots & \dots & \dots \\ w'_{1,K} & w'_{2,K} & \dots & w'_{X,L} \end{bmatrix}$$

where A_x and C_l are 1×Y column vectors

s.t.
$$\begin{aligned} span_{column}(A) &= \{Aw \quad s.t. \quad w = X\times 1 \quad column\_matrix\} \\ &= \{Cv \quad s.t. \quad v = L\times 1 \quad column\_matrix\} \\ &= span_{column}(C) \end{aligned}$$

and there exists no W (except δ_k) such that C_k = CW (or in a formal way, there exists no non trival solution to CW = 0). Thus column data matrix A is decomposed into a linearly independent (but not necessarily orthogonal) basis C (with L column vectors in Y dimensional space), having the same column span as A, while L is called column rank.

Property 1 : Row rank equals to column rank

Lets prove that row rank equals to column rank. It is the same as proving the row rank of A equals to the row rank of A^T.

$$A = WR \qquad \text{i.e. row space of } A \text{ is the span of rows in } R$$

$$A^T = R^T W^T \qquad \text{i.e. row space of } A^T \text{ is the span of rows in } W^T \qquad \text{(equation 1)}$$

from equation 1, we have :

$$\text{row rank of } A^T \le \text{row rank of } W^T = \text{num of row in } W^T = K = \text{row rank of } A \qquad \text{(equation 2)}$$

putting A^T into A of equation 2, we have :

$$\text{row rank of } A \le \text{row rank of } A^T$$

$$\Rightarrow \text{row rank of } A = \text{row rank of } A \qquad \text{(Note : In general nullity}_{row} \ne \text{nullity}_{column}\text{)}$$

Property 2 : Rank null space theorem

Rank null space theorem states that sum of rank and nullity equals to dimension of data space, i.e.

$$rank(A) + nullity_{row}(A) = rank_{row}(A) + nullity_{row}(A) = X$$
$$rank(A) + nullity_{column}(A) = rank_{column}(A) + nullity_{column}(A) = Y$$

(Proof – step 1) Suppose rank of a Y×X row data matrix A is K, since row rank equals to column rank, A can be written as [A_1, A_2], where A_1 is a Y×K matrix of K independent column vectors and A_2 is a Y×(X-K) matrix of (X-K) column vectors dependent on A_1, hence A_2 can always be written as linear combination of A_1, where W is a K×(X-K) matrix.

$$A_2 = A_1 W \quad \Rightarrow \quad A = [A_1, A_2] = [A_1, A_1 W]$$

if $$Z = \begin{bmatrix} -W \\ I \end{bmatrix} \qquad \text{where } I \text{ is a (X-K)×(X-K) identity matrix}$$

then $$AZ = [A_1, A_1 W]\begin{bmatrix} -W \\ I \end{bmatrix} = 0 \qquad \text{hence columns of } Z \text{ are X-K linearly independent solutions to } Az = 0$$

(Proof – step 2) Suppose u is a solution for Az = 0, we have :

$$Au = 0$$

$$Au = [A_1, A_1 W]\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0 \qquad \text{where } u_1 \text{ is K×1 column matrix, } u_2 \text{ is (X-K)×1 column matrix.}$$

$$A_1(u_1 + W u_2) = 0$$

$$u_1 + W u_2 = 0 \qquad \text{since } A_1 \text{ is linear independent, i.e. the only solution is trival solution}$$

$$u_1 = -W u_2$$

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -W u_2 \\ u_2 \end{bmatrix} = \begin{bmatrix} -W \\ I \end{bmatrix} u_2 = Z u_2$$

Thus any solution to Az = 0 can be expressed as linear combination of columns in Z, i.e. u = Zu_2. Hence the dimension of null space of row data matrix A is X-K, in other words, rank(A) + $nullity_{row}$(A) = K+(X-K) = X. Pls repeat for column nullity.

Property 3 : Rank of A^TA equals to rank of A

$$rank(A) = rank(A^T A) \qquad \text{(equation 3)}$$

$$\Rightarrow rank(A^T) = rank(AA^T) \qquad \text{(by property 1)}$$

(Proof of equation 3)

For all $z \in nullity_{row}(A)$

$$Az = 0$$
$$A^T A z = A^T 0 = 0$$
$$z \in nullity_{row}(A^T A)$$
$$nullity_{row}(A) \subseteq nullity_{row}(A^T A)$$

For all $z \in nullity_{row}(A^T A)$

$$A^T A z = 0$$
$$z^T A^T A z = z^T 0 = 0$$
$$(Az)^T Az = 0 \qquad \text{(i.e. } Az = 0\text{)}$$
$$z \in nullity_{row}(A)$$
$$nullity_{row}(A^T A) \subseteq nullity_{row}(A)$$

Combining the above results, we have :

$$nullity_{row}(A) = nullity_{row}(A^T A)$$

$$rank(A) = rank(A^T A) \qquad \text{(by property 2, since both A and } A^TA \text{ live in dimension X)}$$

LU (and LDU) decomposition

LU (and LDU) decomposition is a matrix form of Gaussian elimination. It is applicable to square matrix only, it can decompose a N×N matrix A into a lower triangular matrix L and an upper triangular matrix U, both with size N×N. The number of elements in A is N², while the total number of elements in L and U is 2×(1+2+3+…+N) = N×(N+1), thus given a matrix A, the corresponding matrix L and U are underdetermined, we need to add some constraints on L or U so as to obtain a unique LU decomposition. For example, we can constrain all diagonal elements of L to be one. From remark 4 and 5 in 'Elementary Operation' document, square matrix can (usually) be decomposed as applying a sequence of elementary row operations on an upper triangular matrix.

$$A = E_{N,N-1}E_{N,N-2}\ldots E_{4,3}E_{4,2}E_{4,1}E_{3,2}E_{3,1}E_{2,1}U$$

Or in recursive form :

$$A = E_{N,N-1}A_{N,N-1}$$

$$A_{n,m} = \begin{bmatrix} E_{n,m-1}A_{n,m-1} & & m>1 \\ E_{n-1,n-2}A_{n-1,n-2} & n>2 & m=1 \\ E_{2,1}U & n=2 & m=1 \end{bmatrix} \qquad \forall n\in[2,N] \text{ and } \forall m\in[1,n-1]$$

and elementary addition matrices are :

$$E_{n,m} = E_{add}(n,m,-p/q)$$

p = *element on the nth row and mth column in matrix* $A_{n,m}$

q = *element on the mth row and mth column in matrix* $A_{n,m}$

which means adding multiple (i.e. –p/q) of row m to row n, this is valid only for non zero q in all intermediate steps. This is Gaussian elimination. Besides, since n is always larger than m, all elementary row additions (i.e. $E_{n,m}$ $\forall n\in[1,N]$ *and* $\forall m\in[1,n-1]$*) are lower triangular matrices with unit diagonal elements, and hence the cascade of elementary row additions is also a lower triangular matrix with unit diagonal elements, i.e.*

$$L = E_{N,N-1}E_{N,N-2}\ldots E_{4,3}E_{4,2}E_{4,1}E_{3,2}E_{3,1}E_{2,1} \qquad \forall n\in[2,N] \text{ and } \forall m\in[1,n-1] \Rightarrow n>m$$

where $L = (l_{n,m})_{\forall n\in[1,N],\forall m\in[1,N]}$

s.t. $l_{n,m} = 0 \qquad \forall n,m\in[1,N]$ *such that* $n<m$

s.t. $l_{n,m} = 1 \qquad \forall n,m\in[1,N]$ *such that* $n=m$

$\Rightarrow A = LU$

This is known as the LU decomposition, which is valid only if we have non zero value for q in all intermediate steps. What happens if this requirement is not satisfied? In the first step of LU decomposition, i.e. $E_{2,1}U$*, the value of q is* $a_{1,1}$*, suppose it is zero, then either* $l_{1,1}$ *or* $u_{1,1}$ *must be zero (recall :* $a_{1,1} = l_{1,1}\times u_{1,1}$*). If* $l_{1,1}$ *is zero, the determinant of L is zero, and if* $u_{1,1}$ *is zero, the determinant of U is zero (recall : the determinant of a lower or upper triangular matrix is the product of its diagonal elements), which is impossible, as the determinant of A can be non zero in general (recall : the determinant of matrix product is the product of individual's determinant). Thus, we can proceed LU decomposition for matrix A having zero* $a_{1,1}$ *by (method 1) reordering the rows of A to avoid zero diagonal element, (method 2) reordering both the rows and columns of A to avoid zero diagonal element, or (method 3) decomposing matrix A into 3 matrices LDU without reordering A. Method 1 is known as LU decomposition with partial pivoting, method 2 is known as LU decomposition with full pivoting, while method 3 is known as LDU decomposition.*

LU decomposition with partial (or full) pivoting

The reordering of rows and columns are done by multiplying matrix A with row permutation matrix P or (and) column permutation matrix Q. Please read 'Permutation' document for permutation matrix.

$PA = LU$ *(partial pivoting)* *where P is a N×N row permutation matrix*

$PAQ = LU$ *(full pivoting)* *where Q is a N×N column permutation matrix*

LDU decomposition

LDU decomposition decomposes A into three N×N matrices : LDU, where L and U respectively are the lower and upper triangular matrices, both with all diagonal elements constrainted to 1, while D is a diagonal matrix with configurable diagonal elements. Hence the total number of elements in LDU is : 2×(1+2+3+…+(N–1)) + N = (N–1)×N+N = N², which is the same as that in matrix A. The philosophy of LDU decomposition is to separate the diagonal elements of A into D, so that D can capture both zero and non zero diagonal elements in A.

$$A = LDU \qquad \text{where } d_{n,m} = \begin{bmatrix} a_{n,n} & \text{if} & n=m \\ 0 & \text{if} & n\neq m \end{bmatrix}$$

Cholesky decomposition

*Cholesky decomposition is a special case of LU decomposition, which is applicable to N×N matrix A that is both (1) symmetric **(or Hermitian in general)** and (2) semi positive definite only. Symmetric and semi positive definite matrix is very common in different types of least square problem. Cholesky decomposition is faster than LU decomposition (nearly twice as efficient) as it can make use of the symmetric property of matrix A. With Cholesky decomposition, matrix A can be written as the product of a lower triangular matrix with the transpose of itself on the right, or as the product of an upper triangular matrix with the transpose of itself on the left. The diagonal elements of L and U are **not necessarily one**. The total number of elements in L or U is (1+2+3+…+N) = N(N–1)/2. Please note that : L = U^T.*

$$
\begin{aligned}
A &= LL^T \\
&= U^T U \\
&= \begin{bmatrix} l_{1,1} & 0 & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & 0 & \dots & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{N,1} & l_{N,2} & l_{N,3} & \dots & l_{N,N} \end{bmatrix}
\begin{bmatrix} l_{1,1} & l_{2,1} & l_{3,1} & \dots & l_{N,1} \\ 0 & l_{2,2} & l_{3,2} & \dots & l_{N,2} \\ 0 & 0 & l_{3,3} & \dots & l_{N,3} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & l_{N,N} \end{bmatrix} \\
&= \begin{bmatrix} l_{1,1}^2 & symmetric & symmetric & \dots & symmetric \\ l_{2,1}l_{1,1} & l_{2,1}^2 + l_{2,2}^2 & symmetric & \dots & symmetric \\ l_{3,1}l_{1,1} & l_{3,1}l_{2,1} + l_{3,2}l_{2,2} & l_{3,1}^2 + l_{3,2}^2 + l_{3,3}^2 & \dots & symmetric \\ \dots & \dots & \dots & \dots & \dots \\ l_{N,1}l_{1,1} & l_{N,1}l_{2,1} + l_{N,2}l_{2,2} & l_{N,1}l_{3,1} + l_{N,2}l_{3,2} + l_{N,3}l_{3,3} & \dots & l_{N,1}^2 + l_{N,2}^2 + \dots + l_{N,N}^2 \end{bmatrix}
\end{aligned}
$$

*The implementation is straight forward. For all n starting from 1 to N, and for all m starting from 1 to n, we solve for the value of $l_{n,m}$ based on a set of previously found $l_{i,j}$, this method is known as **bootstrapping**. The bootstrapping order is :* $1_{1,1} \to 1_{2,1} \to 1_{2,2} \to 1_{3,1} \to 1_{3,2} \to 1_{3,3} \to \dots \to 1_{n-1,1} \to 1_{n-1,2} \to \dots \to 1_{n-1,n-1} \to 1_{n,1} \to 1_{n,2} \to \dots \to 1_{n,m-1} \to 1_{n,m}$.

$$
a_{n,m} = \sum_{k=1}^{m} l_{n,k} l_{m,k} = \sum_{k=1}^{m-1} l_{n,k} l_{m,k} + l_{n,m} l_{m,m} \qquad \forall n \in [1,N] \text{ and } \forall m \in [1,n]
$$

$$
\Rightarrow \begin{cases} l_{n,n} = \sqrt{a_{n,n} - \sum_{k=1}^{n-1} l_{n,k} l_{m,k}} & \forall n \in [1,N] \text{ and } m = n \quad \textit{(All } l_{i,j} \textit{ on RHS have been found.)} \\ l_{n,m} = \left(a_{n,m} - \sum_{k=1}^{m-1} l_{n,k} l_{m,k} \right) \dfrac{1}{l_{m,m}} & \forall n \in [1,N] \text{ and } m < n \quad \textit{(All } l_{i,j} \textit{ on RHS have been found.)} \end{cases}
$$

<u>Cholesky decomposition with diagonal matrix</u>

*In some steps of Cholesky decomposition, there are square roots, which are slow. The square roots can be eliminated by introducing an extra diagonal matrix, i.e. A = LDL^T = U^{T}DU, while the diagonal elements of L and U **must be one**.*

$$
\begin{aligned}
A &= LDL^T \\
&= U^T DU \\
&= \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{2,1} & 1 & 0 & \dots & 0 \\ l_{3,1} & l_{3,2} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{5,1} & l_{5,2} & l_{5,3} & \dots & 1 \end{bmatrix}
\begin{bmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & d_N \end{bmatrix}
\begin{bmatrix} 1 & l_{2,1} & l_{3,1} & \dots & l_{N,1} \\ 0 & 1 & l_{3,2} & \dots & l_{N,2} \\ 0 & 0 & 1 & \dots & l_{N,3} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \\
&= \begin{bmatrix} d_1 & symmetric & symmetric & \dots & symmetric \\ l_{2,1}d_1 & l_{2,1}^2 d_1 + d_2 & symmetric & \dots & symmetric \\ l_{3,1}d_1 & l_{3,1}l_{2,1}d_1 + l_{3,2}d_2 & l_{3,1}^2 d_1 + l_{3,2}^2 d_2 + d_3 & \dots & symmetric \\ \dots & \dots & \dots & \dots & \dots \\ l_{N,1}d_1 & l_{N,1}l_{2,1}d_2 + l_{N,2}d_2 & l_{N,1}l_{3,1}d_1 + l_{N,2}l_{3,2}d_2 + l_{N,3}d_3 & \dots & l_{N,1}^2 d_1 + l_{N,2}^2 d_2 + \dots + l_{N,N-1}^2 d_{N-1} + d_N \end{bmatrix}
\end{aligned}
$$

The implementation is similar.

$$
a_{n,m} = \sum_{k=1}^{m} l_{n,k} l_{m,k} d_k = \begin{bmatrix} \sum_{k=1}^{m-1} l_{n,k} l_{m,k} d_k + l_{n,m} l_{m,m} d_m = \sum_{k=1}^{m-1} l_{n,k} l_{m,k} d_k + d_m & m = n \\ \sum_{k=1}^{m-1} l_{n,k} l_{m,k} d_k + l_{n,m} l_{m,m} d_m = \sum_{k=1}^{m-1} l_{n,k} l_{m,k} d_k + l_{n,m} d_m & m < n \end{bmatrix}
$$

$$
\Rightarrow \begin{cases} d_n = a_{n,n} - \sum_{k=1}^{m-1} l_{n,k} l_{m,k} d_k & \forall n \in [1,N] \quad \textit{(All } l_{i,j} \textit{ on RHS have been found.)} \\ l_{n,m} = \left(a_{n,m} - \sum_{k=1}^{m-1} l_{n,k} l_{m,k} d_k \right) \dfrac{1}{d_m} & \forall n \in [1,N] \text{ and } m < n \quad \textit{(All } l_{i,j} \textit{ on RHS have been found.)} \end{cases}
$$

QR (and LQ) decomposition

QR (and LQ) decomposition performs orthonormalization based on Gram Schmidt process, the former is applicable to rectangular matrix A with size M×N (i.e. N column vectors in dimension M), while the latter is applicable to rectangular matrix A with size N×M (i.e. N row vectors in dimension M). Q matrix is a set of K orthonormal vectors, i.e. { Q_1, Q_2, Q_3, ..., Q_K } such that $Q_i^TQ_j = \delta_{ij}$ $\forall i, j\in[1, K]$ for QR decomposition, while $Q_iQ_j^T = \delta_{ij}$ $\forall i, j\in[1, K]$ for LQ decomposition, where K is known as the rank of matrix A. [Recall : column vector means representing a vector by a column matrix and row vector means representing a vector by a row matrix.]

Case 1 : M×N matrix A with column vectors, when N > M ≥ K or when M > N ≥ K

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,N} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{M,1} & a_{M,2} & a_{M,3} & \cdots & a_{M,N} \end{bmatrix} = \begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,K} \\ q_{2,1} & q_{2,2} & q_{2,3} & \cdots & q_{2,K} \\ q_{3,1} & q_{3,2} & q_{3,3} & \cdots & q_{3,K} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{M,1} & \underset{M\times 1}{q_{M,2}} & q_{M,3} & \cdots & q_{M,K} \end{bmatrix} \begin{array}{cccccccc} & & K\times K & & & & K\times(N-K) & \\ \hline r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{K,1} & r_{K+1,1} & \cdots & r_{N,1} \\ 0 & r_{2,2} & r_{2,3} & \cdots & r_{K,2} & r_{K+1,2} & \cdots & r_{N,2} \\ 0 & 0 & r_{3,3} & \cdots & r_{K,3} & r_{K+1,3} & \cdots & r_{N,3} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & r_{K,K} & r_{K+1,K} & \cdots & r_{N,K} \end{array}$$

$$\begin{bmatrix} A_1 & A_2 & A_3 & \cdots & A_N \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 & Q_3 & \cdots & Q_K \end{bmatrix} \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots & r_{K,1} & r_{K+1,1} & \cdots & r_{N,1} \\ 0 & r_{2,2} & r_{2,3} & \cdots & r_{K,2} & r_{K+1,2} & \cdots & r_{N,2} \\ 0 & 0 & r_{3,3} & \cdots & r_{K,3} & r_{K+1,3} & \cdots & r_{N,3} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & r_{K,K} & r_{K+1,K} & \cdots & r_{N,K} \end{bmatrix}$$

i.e.	A	=	QR			
where	A	=	*M×N matrix*	A_n	=	*M×1 column vector*
	Q	=	*M×K matrix*	Q_k	=	*M×1 column vector (orthonormal)*
	R	=	*K×N matrix*			

Case 2 : N×M matrix A with row vectors, when N > M ≥ K or when M > N ≥ K

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,M} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,M} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N,1} & a_{N,2} & a_{N,3} & \cdots & a_{N,M} \end{bmatrix} = \begin{array}{ccccc|l} l_{1,1} & 0 & 0 & \cdots & 0 & \leftarrow K\times K \\ l_{2,1} & l_{2,2} & 0 & \cdots & 0 & \\ l_{3,1} & l_{3,2} & l_{3,3} & \cdots & 0 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ l_{K,1} & l_{K,2} & l_{K,3} & \cdots & l_{K,K} & \\ l_{K+1,1} & l_{K+1,2} & l_{K+1,3} & \cdots & l_{K+1,K} & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ l_{N,1} & l_{N,2} & l_{N,3} & \cdots & l_{N,K} & \leftarrow (N-K)\times K \end{array} \begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,M} \\ q_{2,1} & q_{2,2} & q_{2,3} & \cdots & q_{2,M} \\ q_{3,1} & q_{3,2} & q_{3,3} & \cdots & q_{3,M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{K,1} & q_{K,2} & q_{K,3} & \cdots & q_{K,M} \end{bmatrix} \leftarrow 1\times M$$

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \cdots \\ A_N \end{bmatrix} = \begin{bmatrix} l_{1,1} & 0 & 0 & \cdots & 0 \\ l_{2,1} & l_{2,2} & 0 & \cdots & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ l_{K,1} & l_{K,2} & l_{K,3} & \cdots & l_{K,K} \\ l_{K+1,1} & l_{K+1,2} & l_{K+1,3} & \cdots & l_{K+1,K} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ l_{N,1} & l_{N,2} & l_{N,3} & \cdots & l_{N,K} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \cdots \\ Q_N \end{bmatrix}$$

i.e.	A	=	LQ			
where	A	=	*N×M matrix*	A_n	=	*1×M row vector*
	Q	=	*K×M matrix*	Q_k	=	*1×M row vector (orthonormal)*
	L	=	*N×K matrix*			

Now lets find Q matrix, R matrix and L matrix using Gram Schmidt process. The Q matrix is :

$$Q_1 = \frac{A_1}{\|A_1\|}$$

$$Q_2 = \frac{A_2 - proj_{Q_1}(A_2)}{\|A_2 - proj_{Q_1}(A_2)\|}$$

$$Q_3 = \frac{A_3 - proj_{Q_1}(A_3) - proj_{Q_2}(A_3)}{\|A_3 - proj_{Q_1}(A_3) - proj_{Q_2}(A_3)\|}$$

$$Q_4 = \frac{A_4 - proj_{Q_1}(A_4) - proj_{Q_2}(A_4) - proj_{Q_3}(A_4)}{\|A_4 - proj_{Q_1}(A_4) - proj_{Q_2}(A_4) - proj_{Q_3}(A_4)\|}$$

while the R matrix and L matrix are :

$$\begin{aligned} r_{k,n} &= l_{n,k} && \forall k \in [1,K] \text{ and } \forall n \in [1,N] && \textit{(equation 1)} \\ &= \langle A_n, Q_k \rangle && \forall k \in [1,K] \text{ and } \forall n \in [1,N] \\ &= A_n{}^T Q_k && \textit{for column vector} \\ &= A_n Q_k{}^T && \textit{for row vector} \end{aligned}$$

If A_n is a column vector, then Q_k is a column vector, and

$$\begin{aligned} proj_{Q_k}(A_n) &= Q_k \frac{Q_k{}^T A_n}{Q_k{}^T Q_k} = Q_k(Q_k{}^T A_n) && \text{since } Q_k{}^T Q_k = 1 \\ \|A_n\| &= A_n{}^T A_n \end{aligned}$$

If A_n is a row vector, then Q_k is a row vector, and

$$\begin{aligned} proj_{Q_k}(A_n) &= \frac{A_n Q_k{}^T}{Q_k Q_k{}^T} Q_k = (A_n Q_k{}^T) Q_k && \text{since } Q_k Q_k{}^T = 1 \\ \|A_n\| &= A_n A_n{}^T \end{aligned}$$

In general, for ***non orthogonal basis*** *Z, the projection is defined as :*

If A_n is a column vector, then Z_k is a column vector, and

$$proj_{Z_k}(A_n) = Z_k(Z_k{}^T Z_k)^{-1} Z_k{}^T A_n = PA_n$$

If A_n is a row vector, then Z_k is a row vector, and

$$proj_{Q_k}(A_n) = A_n Z_k{}^T (Z_k Z_k{}^T)^{-1} Z_k = A_n P$$

explanation :

$Z_k{}^T A_n$	*inner product*
$(Z_k{}^T Z_k)^{-1} Z_k{}^T A_n$	*normalized inner product*
$Z_k(Z_k{}^T Z_k)^{-1} Z_k{}^T A_n$	*normalized vector*
$Z_k(Z_k{}^T Z_k)^{-1} Z_k{}^T = P$	*projection matrix (col)*

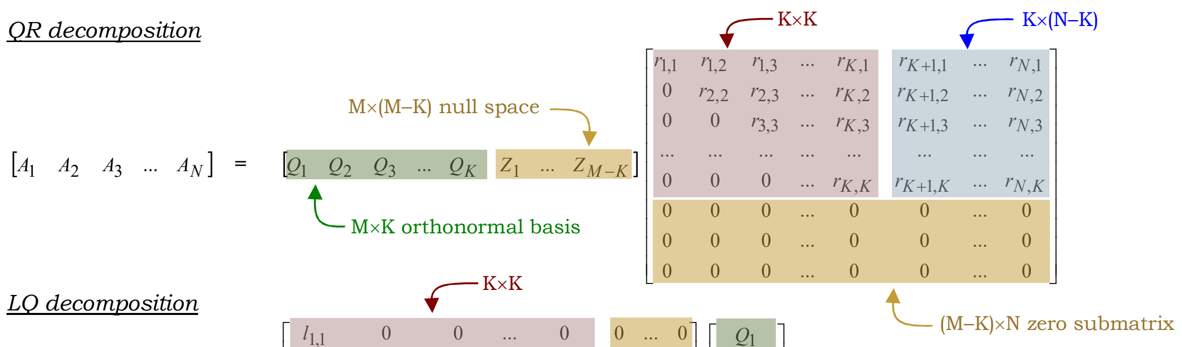
Can you plot it geometrically?

QR decomposition and LQ decomposition are in fact related by a transpose. Suppose A is M×N, then A^T is N×M.

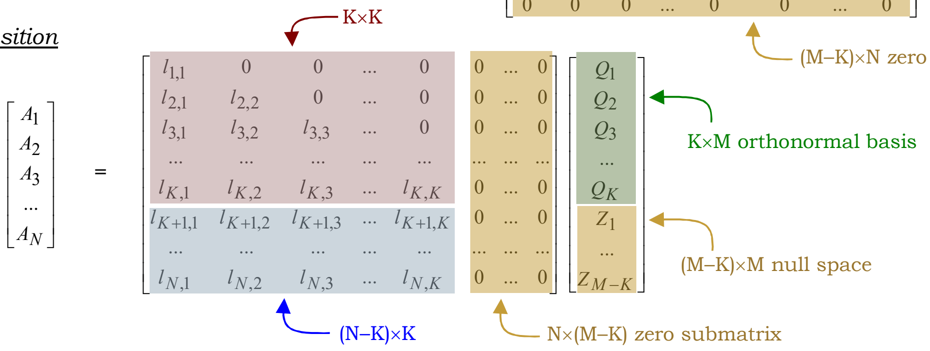
$$\begin{aligned} A &= QR && \textit{QR decomposition} \\ A^T &= LQ^T && \textit{LQ decomposition} \\ A^T &= R^T Q^T && \textit{transpose of QR decomposition} \\ \textit{thus} \quad L &= R^T && \textit{hence it is consistent with equation 1} \end{aligned}$$

Gram Schmidt process cannot help to find the null space, yet we can still generalize QR and LQ decomposition by including the null space (found by other methods) into the Q matrix, then we have (Z stands for the M–K dimensional null space) :

QR decomposition



LQ decomposition



Eigen decomposition

Lets define eigenvector (eigenfunction) and eigenvalue first. Eigenvector X of a ***linear transformation*** *(which maps a N dimensional vector to another N dimensional vector, and is usually represented as a N×N matrix A) is a vector, which becomes a multiple of itself after applying the linear transformation, thus eigenvector are scaled in magnitude only (with no change in direction) after applying the linear transformation. The scale in magnitude is called eigenvalue. This idea can be generalized to a more universal set of* ***linear operators****. For example, if L is a linear differential operator which maps a continuous function to another continuous function, then eigenfunctions f is defined as functions, when operated by linear differential operator L, returns multiple of itself (please read the 'Green function' document). Both eigenvector and eigenfunction share common properties.*

$$\begin{aligned} AX &= \lambda X & Lf &= \lambda f \\ X &= \text{eigenvector (}N\times 1\text{ column matrix)} & f &= \text{eigenfunction} \\ \lambda &= \text{eigenvalue (scalar)} & \lambda &= \text{eigenvalue} \end{aligned}$$

Property 1
Multiple of eigenvector (eigenfunction) is also an eigenvector (eigenfunction).

$$\begin{aligned} \text{If} \quad AX &= \lambda X \\ \text{then} \quad A(cX) &= cA(X) && \text{since A is a linear transformation} \\ &= c\lambda X && \text{since X is an eigenvector} \\ &= \lambda(cX) && \text{hence proved} \\ \text{If} \quad Lf &= \lambda f \\ \text{then} \quad L(cf) &= cL(f) && \text{since L is a linear operator} \\ &= c\lambda f && \text{since f is an eigenfunction} \\ &= \lambda(cf) && \text{hence proved} \end{aligned}$$

Property 2
Weighted sum of eigenvectors (eigenfunctions) having ***same eigenvalue*** *is also an eigenvector (eigenfunction) having the same eigenvalue.*

$$\begin{aligned} \text{If} \quad AX_1 &= \lambda X_1 \\ \text{and} \quad AX_2 &= \lambda X_2 \\ \text{then} \quad A(w_1X_1 + w_2X_2) &= w_1A(X_1) + w_2A(X_2) && \text{since A is a linear transformation} \\ &= w_1\lambda X_1 + w_2\lambda X_2 \\ &= \lambda(w_1X_1 + w_2X_2) && \text{hence proved} \\ \text{If} \quad Lf_1 &= \lambda f_1 \\ \text{and} \quad Lf_2 &= \lambda f_2 \\ \text{then} \quad L(w_1f_1 + w_2f_2) &= w_1Lf_1 + w_2Lf_2 && \text{since L is a linear operator} \\ &= w_1\lambda X_1 + w_2\lambda X_2 \\ &= \lambda(w_1X_1 + w_2X_2) && \text{hence proved} \end{aligned}$$

Property 3 From now on, lets focus on eigenvectors only (ignore eigenfunction).
If A is an ***identity matrix****, then all vectors are eigenvectors. Here is the one line proof.*

$$IX = X = \lambda X \qquad \text{which is true for all X, where } \lambda = 1$$

Property 4
If A is a ***diagonal matrix****, then all N unit vectors in space $\Re^N$ are eigenvectors, with eigenvalues being equivalent to the diagonal elements of A. Suppose I_n be the nth column vector of identity matrix I, i.e. I_n is a unit vector in space $\Re^N$:*

$$AI_n = \begin{bmatrix} a_{1,1} & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & a_{n,n} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \dots & a_{N,N} \end{bmatrix} \begin{bmatrix} 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dots \\ a_{n,n} \\ \dots \\ 0 \end{bmatrix} = a_{n,n} \begin{bmatrix} 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{bmatrix}$$

Property 5
If A is a ***triangular matrix*** *(either lower or upper), then its diagonal elements are the N eigenvalues. The proof starts with solving the characteristic function for eigenvalues (please refer to implemention method 1).*

$$\begin{aligned} \det(A - \lambda I) &= 0 \\ \textstyle\prod_{n=1}^{N}(A - \lambda I)_{n,n} &= 0 && \text{since } A - \lambda I \text{ is also a triangular matrix} \\ \textstyle\prod_{n=1}^{N}(a_{n,n} - \lambda) &= 0 && \Rightarrow \quad \lambda = a_{n,n} \quad \forall n \in [1, N] \end{aligned}$$

Eigenspace, eigenbasis and eigen decomposition

*Given a linear transformation A, the number of eigenvectors can be infinity, more than N or less than N. Eigenvectors are not necessarily orthogonal. Lets group the eigenvectors according to their eigenvalues, i.e. eigenvectors having the same eigenvalue λ_m are grouped to form a set. According to property 2, any span of the set is also an eigenvector having the same eigenvalue (it belongs to the same set too). Now we name the set as the eigenspace of A associated with eigenvalue λ_m. There are infinity eigenvectors in that eigenspace, yet we can orthonormalize the eigenspace into $\gamma_A(\lambda_m)$ orthonormal eigenvectors, where $\gamma_A(\lambda_m)$ is known as geometric multiplicity, which equals to the dimension of the eigenspace. For simplicity, from now on, the term 'eigenvector' refers only to eigenvector after orthonormalization process (i.e. we **purify** the entire set eigenvector by removing eigenvectors that are dependent on the others). Besides, there may be multiple eigenspaces for A in general. Suppose A denotes a linear transformation from $\Re^N$ to $\Re^N$, and suppose there are M eigenspaces, each is associated with an eigenvalue λ_m, $\forall m\in[1,M]$, then :*

- *eigenvectors from the same eigenspace are **orthonormal** to each other (because of orthonormalization)*
- *eigenvectors from different eigenspaces are **not necessaily orthonormal** to each other*
- *sum of geometric multiplicity over all eigenspaces gives the total number of 'purified' eigenvectors (must be $\leq$ N)*

$$1 \leq \sum_{m=1}^{M}\gamma_A(\lambda_m) \leq N \qquad \text{where} \quad M \leq N$$

$$1 = \gamma_A(\lambda_m) \quad \forall m\in[1,M] \qquad \text{when} \quad M = N$$

If the total number of eigenvectors is exactly N, then the eigenvectors form a set called eigenbasis. Please note that : (1) eigenbasis does not necessarily exist, (2) eigenvectors in eigenbasis are not necessarily orthonormal, hence eigenbasis is just a basis (not an orthogonormal basis). If eigenbasis does exist, then we can define eigen decomposition by grouping all eigenvectors as rows (or as columns) of N×N matrix X, and grouping all eigenvalues as diagonal elements of N×N diagonal matrix Λ.

For row eigenvector X

$$XA = \Lambda X$$
$$\Rightarrow \quad A = X^{-1}\Lambda X$$
$$\text{where} \quad X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \ldots \\ X_N \end{bmatrix}$$
$$X_n = 1\times N \text{ row matrix}$$
$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ 0 & 0 & \lambda_3 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \lambda_N \end{bmatrix}$$

For column eigenvector X

$$AX = X\Lambda$$
$$\Rightarrow \quad A = X\Lambda X^{-1}$$
$$\text{where} \quad X = \begin{bmatrix} X_1 & X_2 & X_3 & \ldots & X_N \end{bmatrix}$$
$$X_n = N\times 1 \text{ column matrix}$$
$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ 0 & 0 & \lambda_3 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \lambda_N \end{bmatrix}$$

*Note : Matrix A in the LHS case and in the RHS case are the same matrix, while matrix X in the LHS case and in the RHS case are related by a transpose. Hence the **LHS case and the RHS case are identical**.*

Interpretation 1 (linear transformation)

For both LHS and RHS cases above, we treat A and X as transformation matrices (hence we don't need to classify them as row or column matrix), eigen decomposition can be interpreted as decomposing the linear transformation A into a shearing X (or X^{-1}), followed by a dimension scaling Λ, and finally with an inverse shearing X^{-1} (or X).

Interpretation 2 (linear combination)

For the LHS case above, if we treat A and X as row data matrices, then eigenvalue decomposition can be interpreted as a linear combination of rows in X (i.e. eigenvectors) with weights specified in $X^{-1}\Lambda$, if we treat A and X^{-1} as column data matrices, then eigenvalue decomposition can be interpreted as a linear combination of columns in X^{-1} (i.e. eigenvectors) with weights specified in ΛX.

For the RHS case above, if we treat A and X as column data matrices, then eigenvalue decomposition can be interpreted as a linear combination of columns in X (i.e. eigenvectors) with weights specified in ΛX^{-1}, if we treat A and X^{-1} as row data matrices, then eigenvalue decomposition can be interpreted as a linear combination of rows in X^{-1} (i.e. eigenvectors) with weights specified in XΛ.

Interpretation 3 (inner product or covariance)

For both LHS and RHS cases above, A can also be interpreted as inner product within X^{-1} or covariance (weighted 2nd moment, to be more precise) of X.

What does it mean physically if the eigenbasis is orthonormal?

Following interpretation 1 above, when eigenbasis is orthonormal, the shearing X becomes a rotation in $\Re^N$ space without any scaling (since orthonormal transformation is a matrix with sine and cosine, which means rotation). As there are N^2 elements in matrix A, an orthonormal eigenbasis means some constrains on the degree of freedom of A. Yet, what is the constrain exactly? Lets consider a N=2 case.

$$\begin{aligned}
A &= \begin{bmatrix} \cos\vartheta & +\sin\vartheta \\ -\sin\vartheta & \cos\vartheta \end{bmatrix}\begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_y \end{bmatrix}\begin{bmatrix} \cos\vartheta & -\sin\vartheta \\ +\sin\vartheta & \cos\vartheta \end{bmatrix} \\
&= \begin{bmatrix} \cos\vartheta & +\sin\vartheta \\ -\sin\vartheta & \cos\vartheta \end{bmatrix}\begin{bmatrix} \lambda_x\cos\vartheta & -\lambda_x\sin\vartheta \\ +\lambda_y\sin\vartheta & \lambda_y\cos\vartheta \end{bmatrix} \\
&= \begin{bmatrix} \lambda_x\cos^2\vartheta+\lambda_y\sin^2\vartheta & (\lambda_y-\lambda_x)\sin\vartheta\cos\vartheta \\ (\lambda_y-\lambda_x)\sin\vartheta\cos\vartheta & \lambda_x\sin^2\vartheta+\lambda_y\cos^2\vartheta \end{bmatrix} \quad = \quad A^T
\end{aligned}$$

Hence when eigenbasis is orthonormal, A can be replicated by a rotation plus a scale plus a inverse rotation, resulting in a symmetric A, which gives us inspiration to derive property 6.

Property 6 – for symmetric matrix

In general, eigenvectors from different eigenspaces are not necessarily orthonormal to each other. However they must be orthonormal when linear transformation A is symmetric. Suppose A is a N×N symmetric matrix, while X_n and X_m (where $n,m\in[1,N]$ such that $n\neq m$) are two N×1 column eigenvectors with different eigenvalues λ_n and λ_m respectively, they are picked from different eigenspaces, then we have :

$$\begin{aligned}
AX_n &= \lambda_n X_n & &\Rightarrow & X_n &= A^{-1}(\lambda_n X_n) \\
AX_m &= \lambda_m X_m & &\Rightarrow & X_m &= (AX_m)/\lambda_m \\
X_n^T X_m &= (\lambda_n/\lambda_m)(A^{-1}X_n)^T(AX_m) \\
&= (\lambda_n/\lambda_m)X_n^T(A^{-1})^T AX_m \\
&= (\lambda_n/\lambda_m)X_n^T(A^{-1})^T A^T X_m & &\text{since A is symmetric, i.e. } A^T = A \\
&= (\lambda_n/\lambda_m)X_n^T(AA^{-1})^T X_{mj} \\
&= (\lambda_n/\lambda_m)X_{ni}^T X_m \\
0 &= (1-\lambda_n/\lambda_m)X_n^T X_m \\
X_n^T X_m &= 0 & &\text{since } \lambda_i \neq \lambda_j \text{ as they belong to different eigenspaces}
\end{aligned}$$

Property 7 – for symmetric positive definite matrix

When a square matrix is both symmetric and positive definite, then its eigen values must be all positive. Suppose A is a N×N symmetric and positive definite matrix, then its eigenvectors must form orthonormal basis and we have :

$$\begin{aligned}
& A &&= X^T\Lambda X && \text{where } X^TX = I \\
\text{and}\quad & y^TAy &&> 0 && \text{for all non zero } N\times 1 \text{ column matrix } y \\
\Rightarrow\quad & y^TX^T\Lambda Xy &&> 0 \\
& (Xy)^T\Lambda Xy &&> 0 \\
& z^T\Lambda z &&> 0 && \text{for some } N\times 1 \text{ column matrix z, such that } z = Xy \\
& \textstyle\sum_{n=1}^N \Lambda_{n,n} z_n^2 &&> 0 \\
\Rightarrow\quad & \Lambda_{n,n} &&> 0 && \text{for all } n
\end{aligned}$$

Property 8A – for square root of matrix

Square root is defined for symmetric positive definite matrix only :

$$A = A^{1/2}A^{1/2} \qquad \text{for symmetric positive definite matrix A}$$

Since A is symmetric positive definite, all its eigenvalues should be positive.

$$\begin{aligned}
& A &&= X\Lambda X^{-1} \\
& &&= X\Lambda^{1/2}\Lambda^{1/2}X^{-1} && \text{as we define } \Lambda = \Lambda^{1/2}\Lambda^{1/2}\text{, i.e. } \Lambda_{n,n} = (\Lambda_{n,n}^{1/2})^2 \\
& &&= X\Lambda^{1/2}X^{-1}X\Lambda^{1/2}X^{-1} \\
& &&= A^{1/2}A^{1/2} \\
\text{i.e.}\quad & A^{1/2} &&= X\Lambda^{1/2}X^{-1}
\end{aligned}$$

Therefore A and $A^{1/2}$ share the same eigenvectors, while having eigenvalues related by square root.

Property 8B – for power of matrix
Power of a N×N matrix A (i.e. cascading linear transformation) is :

$$\begin{array}{lcllcl} A & = & X\Lambda X^{-1} & & & \\ A^2 & = & (X\Lambda X^{-1})(X\Lambda X^{-1}) & & = & X\Lambda^2 X^{-1} \\ A^3 & = & (X\Lambda X^{-1})(X\Lambda X^{-1})(X\Lambda X^{-1}) & & = & X\Lambda^3 X^{-1} \\ A^N & = & X\Lambda^N X^{-1} & & & \end{array}$$

Property 9 – for similarity
Similarity transformation from a N×N matrix A to a N×N matrix B is defined as :

$$B = QAQ^{-1} \qquad \text{where Q is N×N orthogonal basis}$$

then the eigen decomposition of B is related to the eigen decomposition of A by :

$$\begin{array}{llcll} & A & = & X\Lambda X^{-1} & \text{where X is N×N eigenbasis of A (not necessarily orthonormal)} \\ & B & = & Y\Delta Y^{-1} & \text{where Y is N×N eigenbasis of B (not necessarily orthonormal)} \\ \Rightarrow & B & = & QAQ^{-1} & \\ & & = & QX\Lambda X^{-1}Q^{-1} & \\ & & = & (QX)\Lambda(QX)^{-1} & \\ & & = & Y\Delta Y^{-1} & \\ \Rightarrow & Y & = & QX & \text{product of orthonormal is also orthonormal} \\ \text{and} & \Delta & = & \Lambda & \end{array}$$

$$\begin{array}{llcll} \text{since} & \det(A) & = & \prod_{n=1}^{N}\Lambda_n & \\ \& & \det(B) & = & \prod_{n=1}^{N}\Delta_n & \text{thus we have : } \det(A) = \det(B) \\ \text{since} & tr(A) & = & \sum_{n=1}^{N}\Lambda_n & \\ \& & tr(B) & = & \sum_{n=1}^{N}\Delta_n & \text{thus we have : } tr(A) = tr(B) \end{array}$$

When eigenbasis (eigen decomposition) exists, we can derive it using the following methods.

Implementation method 1 : Characteristic equation
once eigenvalues are found, eigenvectors can be found by solving system of linear equation

$$\begin{array}{lcll} AX & = & \lambda X & \\ (A-\lambda I)X & = & 0 & \text{(equation 2)} \\ \det(A-\lambda I) & = & 0 & \end{array}$$

resulting in a polynomial of λ in degree N, which can be factorized and grouped into different M eigenspaces.

$$\begin{array}{lcl} f(\lambda) & = & c_N\lambda^N + c_{N-1}\lambda^{N-1} + ... + c_3\lambda^3 + c_2\lambda^2 + c_1\lambda + c_0 \\ & = & c_N(\lambda-\lambda_1)^{\gamma_A(\lambda_1)}(\lambda-\lambda_2)^{\gamma_A(\lambda_2)}...(\lambda-\lambda_M)^{\gamma_A(\lambda_M)} \end{array}$$

After obtaining M roots for λ using numerical method, we substitute the values of λ_m back into equation 2 (for all m from 1 to M), resulting in a system of ***linearly dependent*** *equations, in which all equations pass through the origin, and intersect along a line, a plane or a hyperplane. The intersecting hyperplane is the* ***eigenspace*** *corresponding to eigenvalue λ_m, while the dimension of the intersecting hyperplane gives the* ***geometric multiplicity*** *associated with λ_m. We then need to orthonormalize the eigenspace to give* ***eigenvectors****. Recall : the equations do not intersect at a single point, if they do, this point must be the origin, and we deliberately make the system linearly dependent by setting determinant zero (i.e. ∞ solutions). Thus never solve X by performing LU (or other) decomposition on equation 2.*

Implementation method 2 : Power method
Suppose there exists an eigenbasis for linear transformation A, if X and Λ are N×N matrix containing the N column eigenvectors X_n ∀n∈[1,N] and N eigenvalues λ_n ∀n∈[1,N], then power method starts with any initial guess V, which can always be expressed as a linear combinations of eigenbasis, i.e. V is a N×1 column vector such that :

$$\begin{array}{llcl} & V & = & XW \\ & & = & \text{any initial guess} \\ \text{where} & X & = & [X_1 \quad X_2 \quad X_3 \quad ... \quad X_N] \end{array}$$

and $W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \dots \\ w_N \end{bmatrix}$

Power method is an iterative method which eventually gives the dominant eigenvalue as :

$$
\begin{aligned}
V^{(1)} &= AV \\
V^{(t)} &= AV^{(t-1)} \\
\text{then} \quad \lambda_1 &= \lim_{t\to\infty} V_1^{(t)} / V_1^{(t-1)} \\
&= \lim_{t\to\infty} V_2^{(t)} / V_2^{(t-1)} \\
&= \lim_{t\to\infty} V_m^{(t)} / V_m^{(t-1)} \qquad \forall m \in [1, N] \qquad \text{remark : pick the } m^{th} \text{ element}
\end{aligned}
$$

$$
\begin{aligned}
\text{(Proof)} \quad V &= XW \\
&= \textstyle\sum_{n=1}^N w_n X_n \\
V^{(1)} &= \textstyle\sum_{n=1}^N w_n AX_n = \sum_{n=1}^N w_n \lambda_n X_n \qquad \text{since } AX_n = \lambda_n X_n \\
V^{(2)} &= \textstyle\sum_{n=1}^N w_n \lambda_n AX_n = \sum_{n=1}^N w_n \lambda_n{}^2 X_n \qquad \text{since } AX_n = \lambda_n X_n \\
V^{(t)} &= \textstyle\sum_{n=1}^N w_n \lambda_n{}^{t-1} X_n = \sum_{n=1}^N w_n \lambda_n{}^t X_n \\
\lim_{t\to\infty} \frac{V_m^{(t)}}{V_m^{(t-1)}} &= \lim_{t\to\infty} \frac{\sum_{n=1}^N w_n \lambda_n{}^t X_{n,m}}{\sum_{n=1}^N w_n \lambda_n{}^{t-1} X_{n,m}} \qquad \text{remark : pick the } m^{th} \text{ element} \\
&= \frac{w_1 \lambda_1{}^t X_{1,m}}{w_1 \lambda_1{}^{t-1} X_{1,m}} \qquad \text{since } \lambda_1 \text{ is dominant eigenvalue} \\
&= \lambda_1
\end{aligned}
$$

Question : How about other eigenvalues?

<u>Implementation method 3 : QR algorithm (Don't confuse with QR decomposition)</u>
QR decomposition starts with $A^{(1)} = A$, and in each iteration t, we apply QR decomposition on $A^{(t)}$, such that $A^{(t)} = Q^{(t)}R^{(t)}$, and then generate $A^{(t+1)}$ as the product of QR matrices in reverse order, i.e. $A^{(t+1)} = R^{(t)}Q^{(t)}$, by repeating the process, $A^{(t)}$ will eventually become a similarity transformation of an upper triangular matrix U. The diagonal elements of the upper triangular matrix U give all the N eigenvalues.

$$
\begin{aligned}
A^{(t+1)} &= R^{(t)}Q^{(t)} \\
&= (Q^{(t)})^{-1}Q^{(t)}R^{(t)}Q^{(t)} \qquad \text{since } Q^{(t)T}Q^{(t)} = I \text{ as } Q^{(t)} \text{ is orthonormal} \\
&= (Q^{(t)})^{-1}A^{(t)}Q^{(t)} \\
&= (Q^{(t)})^{-1}(Q^{(t-1)})^{-1}A^{(t-1)}Q^{(t-1)}Q^{(t)} \\
&= (Q^{(t)})^{-1}(Q^{(t-1)})^{-1}\dots(Q^{(2)})^{-1}(Q^{(1)})^{-1}AQ^{(1)}Q^{(2)}\dots Q^{(t-1)}Q^{(t)} \\
&= (Q^{(1)}Q^{(2)}\dots Q^{(t-1)}Q^{(t)})^{-1}AQ^{(1)}Q^{(2)}\dots Q^{(t-1)}Q^{(t)} \\
&= Q^{-1}AQ \\
\text{where} \quad Q &= Q^{(1)}Q^{(2)}\dots Q^{(t-1)}Q^{(t)} \qquad \text{product of orthonormal is orthonormal} \\
A &= QA^{(t+1)}Q^{-1} \qquad \text{which is true for all t} \\
&= QA^{(\infty)}Q^{-1} \\
&= QUQ^{-1} \qquad \text{see remark (We cannot conclude here as U is not diagonal.)}
\end{aligned}
$$

$$
\begin{aligned}
\text{eigen value of } A &= \text{eigen value of } QUQ^{-1} \\
&= \text{eigen value of } U \qquad \text{see property 9} \\
&= (u_{n,n} : n \in [1, N]) \qquad \text{see property 5}
\end{aligned}
$$

Remark :
Prove that $A^{(t)}$ becomes an upper triangular matrix as t tends to infinity for QR algorithm. Please read 'Orthogonal Bases and the QR Algorithm', written by Peter J Oliver.

Singular value decomposition (SVD)

Singular value decomposition decomposes a rectangular matrix A into a left singular vector basis U, a right singular vector basis V and a scaling matrix S, such that U and V are both orthonormal, while S is diagonal. Note that eigenbasis in eigen decomposition is not necessarily orthonormal, yet both left singular basis and right singular basis in singular value decomposition are always orthonormal.

For row vectors in A

$$A = USV^T$$

where
$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \dots \\ U_N \end{bmatrix}$$

U_n = *1×N row matrix*

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \dots \\ V_M \end{bmatrix}$$

V_n = *1×M row matrix*

$$S = \begin{bmatrix} s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 \\ 0 & 0 & s_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s_M \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

= *N×M matrix (if N>M)*

For column vectors in A

$$A = V^T SU$$

where
$$U = \begin{bmatrix} U_1 & U_2 & U_3 & \dots & U_N \end{bmatrix}$$

U_n = *N×1 column matrix*

$$V = \begin{bmatrix} V_1 & V_2 & V_3 & \dots & V_M \end{bmatrix}$$

V_n = *M×1 column matrix*

$$S = \begin{bmatrix} s_1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & s_3 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & s_M & 0 & \dots & 0 \end{bmatrix}$$

= *M×N matrix (if N>M)*

Note : Matrix A in the LHS case and in the RHS case are related by a transpose, while matrices (U,V) in the LHS case and in the RHS case are also related by a transpose. Hence the ***LHS case and the RHS case are identical****.*

Interpretation 1 (linear transformation)

For the LHS case above, SVD can be interpreted either as generating row data matrix A by applying dimensional scaling S, followed by rotation V on orthonormal row data matrix U. For the RHS case above, SVD can be interpreted either as generating column data matrix A by applying dimensional scaling S, followed by rotation V on orthonormal column data matrix U. For linear transformation interpretation, matrices A, U, V are treated as row data matrices for LHS case (or as column data matrices for RHS case), however we cannot find the null space with this interpretation, so we go for another interpretation.

Interpretation 2 (linear combination)

When N<M, there exists null space, which can be only found using linear combination interpretation. We rewrite V^T as row data matrix, i.e. Z_n denotes the nth row of V^T.

$$V^T = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ \\ Z_M \end{bmatrix} = \begin{bmatrix} Z' \\ Z'' \end{bmatrix}$$

where Z' is K×M and Z" is (M-K)×M

(Case 1) In the LHS case, row data matrix A is treated as the linear combination of row data matrix Z with weight US, Z" provides the null space of row data matrix A, as since $AZ_n^T = 0$ for all $Z_n \in Z''$.

$$A = U \begin{bmatrix} s_1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & s_3 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & s_N & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ \dots \\ Z_N \\ Z_{N+1} \\ \dots \\ Z_M \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} Z'' \end{matrix}$$

$$\Rightarrow \quad AZ_n^T = U\begin{bmatrix} s_1Z_1Z_n^T \\ s_2Z_2Z_n^T \\ s_3Z_3Z_n^T \\ \dots \\ s_NZ_NZ_n^T \\ 0 \\ \dots \\ 0 \end{bmatrix} = U\begin{bmatrix} 0 \\ 0 \\ \dots \\ s_n \\ \dots \\ 0 \\ 0 \end{bmatrix} \quad \text{if } n \le N$$

$$= U\begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad \text{if } n > N$$

By definition, Z" is null space.

(Case 2) In the RHS case, column data matrix A is treated as the linear combination of column data matrix Z with weight SU^T, Z" provides the null space of column data matrix A, as since Z_nA = 0 for all Z_n∈Z".

$$A = [Z_1^T \quad Z_2^T \quad Z_3^T \quad \dots \quad Z_N^T \quad \underbrace{Z_{N+1}^T \quad \dots \quad Z_M^T}_{Z''^T}]\begin{bmatrix} s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 & 0 & \dots & 0 \\ 0 & 0 & s_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & s_N \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} U^T$$

$$\Rightarrow \quad Z_nA = [s_1Z_nZ_1^T \quad s_2Z_nZ_2^T \quad s_3Z_nZ_3^T \quad \dots \quad s_NZ_nZ_N^T \quad 0 \quad \dots \quad 0]\, U^T$$
$$= [0 \quad 0 \quad \dots \quad s_n \quad \dots \quad 0 \quad 0]\, U^T \quad \text{if } n \le N$$
$$= [0 \quad 0 \quad 0 \quad \dots \quad 0]\, U^T \quad \text{if } n > N$$

By definition, Z" is null space.

<u>Implementation of SVD</u>
Singular value decomposition is closely related to eigen decomposition.

given	A	=	USV^T	*(S is sorted)*
	AA^T	=	$X\Lambda X^{-1}$	*(Λ is sorted)*
	A^TA	=	$Y\Delta Y^{-1}$	*(Δ is sorted)*
then	U	=	X	
	V	=	Y	
	$s_{n,n}$	=	$\sqrt{\Lambda_{n,n}}$	
		=	$\sqrt{\Delta_{n,n}}$	

(Proof)

$$\begin{aligned} AA^T &= (USV^T)(USV^T)^T \\ &= (USV^T)(VS^TU^T) \\ &= USS^TU^T && \textit{(V is orthonormal)} \\ &= USS^TU^{-1} && \textit{(U is orthonormal)} \\ &= X\Lambda X^{-1} \\ \Rightarrow \quad X &= U \\ \text{and} \quad \Lambda_{n,n} &= s_{n,n}^2 \end{aligned}$$

$$\begin{aligned} A^TA &= (USV^T)^T(USV^T) \\ &= (VS^TU^T)(USV^T) \\ &= VS^TSV^T && \textit{(U is orthonormal)} \\ &= VS^TSV^{-1} && \textit{(V is orthonormal)} \\ &= Y\Delta Y^{-1} \\ \Rightarrow \quad Y &= V \\ \text{and} \quad \Delta_{n,n} &= s_{n,n}^2 \end{aligned}$$

Hence singular value decomposition can be implemented by eigen decomposition.

Application : solving AX = B

Solving AX=B by matrix inverse of A^TA or Cramer's rule is slow, $O(N^3)$. A more efficient method is to decompose either the square matrix A^TA or the rectangular matrix A into product of triangular matrices, and then apply (1) forward substitution for lower triangular matrix and (2) backward substitution for upper triangular matrix to solve for the unknown X. Suppose A is a N×M matrix (i.e. N row vectors in Mth dimensional space), while B is a N×1 matrix (i.e. N observations) :

$$\begin{aligned} AX &= B && \textit{for QR decomposition and SVD decomposition on matrix A} \\ A^TAX &= A^TB && \textit{for LU decomposition and Cholesky decomposition on matrix } A^TA \end{aligned}$$

(1)
$$\begin{aligned} A^TAX &= A^TB \\ LUX &= A^TB && \textit{by LU decomposition} \\ UX &= L^{-1}A^TB && \textit{by forward substitution} && \textit{(don't do matrix inverse)} \\ X &= U^{-1}L^{-1}A^TB && \textit{by backward substitution} && \textit{(don't do matrix inverse)} \end{aligned}$$

(2)
$$\begin{aligned} A^TAX &= A^TB \\ LL^TX &= A^TB && \textit{by Cholesky decomposition} \\ L^TX &= L^{-1}A^TB && \textit{by forward substitution} && \textit{(don't do matrix inverse)} \\ X &= (L^T)^{-1}L^{-1}A^TB && \textit{by backward substitution} && \textit{(don't do matrix inverse)} \end{aligned}$$

(3)
$$\begin{aligned} AX &= B \\ QRX &= B && \textit{by QR decomposition} \\ RX &= Q^TB && \textit{inverse of Q is its transpose} \\ X &= R^{-1}Q^TB && \textit{by backward substitution} && \textit{(don't do matrix inverse)} \end{aligned}$$

(4)
$$\begin{aligned} AX &= B \\ USV^TX &= B && \textit{by SVD decomposition} \\ SV^TX &= U^TB && \textit{inverse of U is its transpose} \\ V^TX &= S^{-1}U^TB && \textit{inverse of S is diagonal matrix with element-wise inverse} \\ X &= VS^{-1}U^TB && \textit{inverse of V is its transpose} \end{aligned}$$

(5)
$$\begin{aligned} A^TAX &= A^TB \\ X &= (A^TA)^{-1}(A^TB) && \textit{by direct matrix inverse of } A^TA\textit{, which is } O(N^3) \end{aligned}$$

(6)
$$\begin{aligned} A^TAX &= A^TB \\ X &= \ldots && \textit{by Cramer's rule, which is again } O(N^3) \end{aligned}$$

Remark : Forward substitution is used to solve LX = C, while backward substitution is used to solve UX = C.

Forward substitution | *Backward substitution*

$$\begin{bmatrix} l_{1,1} & 0 & 0 & \ldots & 0 \\ l_{2,1} & l_{2,2} & 0 & \ldots & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ l_{M,1} & l_{M,2} & l_{M,3} & \ldots & l_{M,M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \ldots \\ x_M \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \ldots \\ c_M \end{bmatrix} \quad \textit{and} \quad \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \ldots & u_{1,M} \\ 0 & u_{2,2} & u_{1,3} & \ldots & u_{2,M} \\ 0 & 0 & u_{3,3} & \ldots & u_{3,M} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & u_{M,M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \ldots \\ x_M \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \ldots \\ c_M \end{bmatrix}$$

$$\sum_{k=1}^{m} l_{m,k}x_k = c_m \quad \textit{and} \quad \sum_{k=m}^{M} u_{m,k}x_k = c_m$$

$$l_{m,m}x_m = c_m - \sum_{k=1}^{m-1} l_{m,k}x_k \quad \textit{and} \quad u_{m,m}x_m = c_m - \sum_{k=m+1}^{M} u_{m,k}x_k$$

$$x_m = \left(c_m - \sum_{k=1}^{m-1} l_{m,k}x_k\right)\frac{1}{l_{m,m}} \quad \textit{and} \quad x_m = \left(c_m - \sum_{k=m+1}^{M} u_{m,k}x_k\right)\frac{1}{u_{m,m}}$$

Bootstrap the value of x_m for m starting from 1 to M for forward substitution in LX = C, and bootstrap the value of x_m for m starting from M to 1 for backward substitution in UX = C. This bootstrapping technique is useful, we have used it once in deriving Cholesky decomposition.

Application : finding determinant for square matrix A

Finding determinant by applying Leibniz formula or Laplace expansion is slow, O(N³). A more efficient method is to decompose the square matrix into product of triangular matrices, and then apply (1) the property that determinant of matrix product equals to product of individual's determinant, and (2) determinant of triangular matrix is the product of its diagonal elements. Suppose A is a N×N square matrix :

(1)
$$\begin{aligned}\det(A) &= \det(LU) && \textit{by LU decomposition}\\ &= \det(L)\det(U)\\ &= \textstyle\prod_{n=1}^{N} l_{n,n}\prod_{n=1}^{N} u_{n,n}\end{aligned}$$

(2)
$$\begin{aligned}\det(A) &= \det(LL^T) && \textit{by Cholesky decomposition}\\ &= \det(L)\det(L^T)\\ &= (\textstyle\prod_{n=1}^{N} l_{n,n})^2\end{aligned}$$

(3)
$$\begin{aligned}\det(A) &= \det(QR) && \textit{by QR decomposition}\\ &= \det(Q)\det(R)\\ &= \det(R) && \textit{since determinant of orthonormal basis = volume of unit hypercube = 1}\\ &= \textstyle\prod_{n=1}^{N} r_{n,n}\end{aligned}$$

(4)
$$\begin{aligned}\det(A) &= \det(X\Lambda X^{-1}) && \textit{by eigen decomposition}\\ &= \det(X)\det(\Lambda)\det(X^{-1})\\ &= \det(\Lambda) && \textit{since determinant of orthonormal basis = volume of unit hypercube = 1}\\ &= \textstyle\prod_{n=1}^{N} \lambda_n\end{aligned}$$

(5)
$$\begin{aligned}\det(A) &= \det(USV^T) && \textit{by SVD decomposition}\\ &= \det(U)\det(S)\det(V^T)\\ &= \det(S) && \textit{since determinant of orthonormal basis = volume of unit hypercube = 1}\\ &= \textstyle\prod_{n=1}^{N} s_{n,n}\end{aligned}$$

(6)
$$\det(A) = \sum_{\sigma\in S_N}\operatorname{sgn}(\sigma)\prod_{n=1}^{N} a_{\sigma(n),n} \qquad \textit{by Leibniz formula}$$

(7)
$$\det(A) = \sum_{q=1}^{N} a_{p,q}C_{p,q} \qquad \textit{by Laplace expansion (where } C_{p,q} \textit{ is cofactor and } p\in[1,N]\textit{)}$$

Application : principal component analysis

Principal component analysis can be solved by eigen decomposition or singular value decomposition. Given a N×M row data matrix A, which has ***zero mean****, our objective is to find an orthonormal linear transformation P, so that B=AP is a N×K row data matrix, achieving objectives (1) diagonalized covariance matrix and (2) largest variance along the 1st coordinate, second largest variance along the 2nd coordinate … etc. Therefore A lies in M dimensional space, while B lies in K dimensional space (K≤M), in other words, B extracts only the K most important dimensions. P is M×K matrix, it denotes a mapping $\Re^M \to \Re^K$, it can be regarded as K column vectors P_k, each lies in M dimensional space, each P_k is known as the principal component of A.*

$$B = AP$$

Covariance in transformed domain is :

$$\begin{aligned}\operatorname{cov}_B &= \frac{1}{N}B^T B\\ &= \frac{1}{N}(AP)^T(AP)\\ &= P^T(\frac{1}{N}A^T A)P\\ &= P^T(\operatorname{cov}_A)P\end{aligned}$$

where cov_A = *M×M matrix*
cov_B = *K×K matrix*

The rest can be implemented by either (1) eigen decomposition or (2) singular value decomposition.

Method 1 : by eigen decomposition
By applying eigen decomposition on cov$_A$, we have :

cov_B	=	$P^T(X^T \Lambda X)P$	*where rows in X are eigenvectors, both X and Λ are M×M matrices*
	=	$P'^T \Lambda P'$	*where* $P' = XP$ *product of orthonormal basis is also orthonormal*
	=	Λ	*where* $P = X^T$

Hence by setting P as the transpose of X (where X is row eigenvector matrix), then we can diagonalize the covariance matrix of B. Since columns of P are the principal components, while columns of P equal to the rows of X, thus eigenbasis gives the set of principal axes for row data matrix A. We have completed the first part of the proof, yet we still need to prove why the 1st component of B has the largest variance, suppose K = 1, then cov$_B$ is a scalar, we have :

$\arg\max_{P, P^T P = I} \mathrm{cov}_B$	=	$\arg\max_P (\mathrm{cov}_B - \eta(P^T P - I))$	*where η is Lagrange multiplier*
	=	$\arg\max_P (P^T(\mathrm{cov}_A)P - \eta(P^T P - I))$	
$2P^T(\mathrm{cov}_A) - 2\eta P^T$	=	0	*take derivative wrt P and set it to zero*
$P^T(\mathrm{cov}_A)$	=	ηP^T	
$(\mathrm{cov}_A)P$	=	ηP	*take transpose, and cov$_A$ is symmetric*

which is the definition of eigenvector, with P as the column eigenvector, and η as the eigenvalue. (Interesting!!! Lagrange multiplier is exactly the eigenvalue.) How about the 2nd component of B? We can repeat the above procedures, with K = 2, and maximize (cov$_B$)$_{2,2}$ under constrain that P cannot be the eigenvector with the largest eigenvalue, we will then end up with exactly the same formulae above. Since P cannot be the eigenvector with the largest eigenvalue, P can only be the eigenvector with the 2nd largest eigenvalue.

Method 2 : by singular value decomposition
Since principal component equals to eigenbasis of A^{T}A, and the latter equals to the right singular basis of A, we have :

A	=	USV^T
V_m	=	*principal components of A*
	=	*1×M row vector*

Application : low rank approximation
Low rank approximation of a large data matrix is a minimization problem which finds a N×M matrix A' having limited rank K to approximate a given N×M matrix A, under a particular matrix norm definition. The basic one is Frobenius norm.

$$A'_{opt} = \arg\min_{A'} \|A - A'\|_F \qquad \text{such that } rank(A') \le K$$

For Frobenius norm, there is an analytic solution in terms of singular value decomposition of A.

A'_{opt}	=	$\arg\min_{A'} \|A - A'\|_F$	
	=	$\arg\min_{A'} \sqrt{\sum_{n=1}^{\min(N,M)} s_{Dn}^2}$	*where s$_{Dn}$ are the singular values of* $A - A' = U_D S_D V_D^T$

This function can be minimized by :

A	=	USV^T	=	$U\mathrm{diag}(s_1, s_2, ..., s_N)V^T$	*where s$_n$ are the singular values of A in descending order*
A'_{opt}	=	$US'V^T$	=	$U\mathrm{diag}(s_1, s_2, ..., s_K, 0, 0, ..., 0)V^T$	*thus* $A - A' = U(S - S')V^T$ *, i.e.* $U = U_D$ *and* $V = V_D$

The objective becomes :

$$\min_{A'} \|A - A'\|_F = \sqrt{\sum_{n=1}^{\min(N,M)} s_{Dn}^2} = \sqrt{\sum_{n=K+1}^{\min(N,M)} s_n^2}$$

Reference
[1] Orthogonal Bases and the QR Algorithm, Peter J Oliver, University of Minnesota.