Proceedings of the 3rd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/2025.GL31275

A Study on Matrix Computation: From Theory to Practice

Huining Yang

New Channel Qingdao, Qingdao, China
kryhn993@outlook.com

Abstract. Matrix computation is the basic equipment of modern scientific computation. It provides a necessary tool for solving linear systems and revising models, and analyzing large datasets. In this work, we do a study about the matrix computation, properties, and decomposition techniques. And we emphasize their academic support and practical meanings. Before introducing the methods of the Gram-Schmidt process and QR decomposition, we begin with the basic matrix operations and features. Based on these principles, we explore two main applications: Ordinary Least Squares (OLS) regression for statistical modelling and Principal Component Analysis (PCA) for dimensionality reduction. Moreover, we also discuss the real-world applications in different fields. This work aims to connect the theories of matrix operations with real applications, and it provides a structured perspective for modern data analysis and experimental operations.

Keywords: matrix computation, decomposition, ordinary least squares, principal component analysis

1. Introduction

Matrix computation plays an important role in the wide scientific and engineering training. It is the main structure of solving basic computational tasks such as linear systems, altering complex models, and analyzing large datasets. Its meanings extend to many fields, including machine learning, statistics, physics, and economics. In these works, a matrix provides an efficient structure for displaying and using data. Matrix supports academic analysis, and it can be used in several applications. It makes matrix operations a necessary tool in academic research and industry. A deep understanding of computing is important to develop digital algorithms, ensure the stability and accuracy of calculations, and promote the creativity of solving problems. It is important to study matrix operations because it has a wide impact. This type of research provides valuable suggestions for theory and the real world [1].

In this work, we make a clear and detailed introduction to crucial matrix computation and its real applications. First, we explain the basic matrix operations, such as addition and multiplication, which are the basis of further operations. Then, we check the key matrix features, including determinants, eigenvalues and eigenvectors. These concepts are crucial for many mathematical and engineering applications. To build these basic concepts, we explore the orthogonalization techniques, focusing on the Gram-Schmidt process. We also cover the decomposition methods, such as QR decomposition, widely used in data analyzing. Based on this theoretical foundation, we have

DOI: 10.54254/2753-8818/2025.GL31275

further explored two real applications: ordinary least squares (OLS) regression and principal component analysis (PCA). Ordinary least squares (OLS) regression is the basic method used in statistical modeling, and principal component analysis (PCA) is a widely used tool in reducing the number of variables in data. These two technologies both make full use of matrix operations. Finally, we discussed how OLS and PCA are used in the real world. It suggests the importance of them in the field of finance, machine learning, and scientific operation. By making some examples to explain how these methods help to solve practical problems. We emphasize the role matrix operation plays in the data analyzing field nowadays.

2. Matrix computation

A matrix is a basic mathematical structure used to arrange numbers and organize them. It is a rectangular grid made of rows and columns. This simple form makes it a useful tool in many areas, including mathematics, engineering, and computer science. Formally, an $m \times n$ matrix A is represented as: $\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$, which can be also written as (a_{ij}) or $[a_{ij}]$, or $(a_{ij})_{1\leq i\leq m,\, 1\leq j\leq n}$. Each part in the matrix comes from a certain group of numbers; the most common is the real numbers ($\mathbb{R}$) or complex numbers ($\mathbb{C}$). Matrix forms the base of many crucial technologies in mathematics and operations. They are the core of many fields, such as linear algebra, numerical analysis, optimization, and machine learning. Matrix is used to solve linear equations, represent geometric transformations and store data in the machine study model. Developing the ability to operate matrix is important for advancing work in these areas [2].

2.1. Basic matrix properties and operations

Matrix addition and subtraction are basic operations in linear algebra. They complete by element by element, but it just suitable for matrix has the same size. If two matrices A and B are both $m \times n$, then we can find their sum by adding and subtracting the elements in the position of the matrices. For example, if $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$, then:

$$A \pm B = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \pm \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{bmatrix}.$$

These operations are commutative and associative, meaning that $A+B=B+A$ and $(A+B)+C=A+(B+C)$ for any three matrices A , B , and C of the same size. However, subtraction is not commutative, as $A-B \neq B-A$ in general. In the wide range of computational applications, such as image analysis and data mining, which always appear in matrix operations. Especially in machine learning, it is the core of processes like parameter updates during training [3].

Matrix multiplication goes beyond the easy operations. The principle of this process is to multiply each row of the matrix by each column of the second matrix. Then, summing the results to form a new matrix. If matrix A of size $m \times p$ and matrix B of $p \times n$, their product $C = AB$

DOI: 10.54254/2753-8818/2025.GL31275

is an $m \times n$ matrix. Each element in c_{ij} is found by taking the i -th row of A and the j -th column of B . Formally, $C = AB,$ with

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj},\ 1 \leq i \leq m,\ 1 \leq j \leq n.$$

There is special situation is the identity matrix. It is a square matrix that has 1s on the main diagonal and 0s in other positions. The identity matrix acts as the multiplicative neutral element, meaning that for any conformable matrix A , the product satisfies $AI = IA = A$, preserving the original matrix. Matrix multiplication is associative, meaning that $(AB)C = A(BC)$ whenever the products are defined, and distributive over addition, i.e., $A\,(B\, +\, C)\ =\ AB\, +\, AC$. However, it is not commutative in general, meaning that $AB \neq BA$ in most cases. This operation plays a crucial role in various scientific and engineering fields. In computer graphics, matrix multiplication is used to apply geometric transformations such as translations, rotations, and scaling to objects. For instance, a rotation matrix can be multiplied by a vector representing a point's coordinates. Then we can compute its new position after rotation [4].

While basic matrix operations like addition and multiplication help manipulate data and solve linear systems, determinants provide critical insights into a matrix's properties. They help determine whether a matrix is invertible, assess the uniqueness of solutions in linear systems, and evaluate geometric transformations such as scaling and orientation. The determinant of a matrix is a scalar value calculated for a square matrix (a matrix that has the same number of rows and columns). The determinant of a matrix can be positive, negative, or zero. The determinant of matrix A can be represented as $det\,(A)$ or $|\mathrm{A}|$. Considering a simple case, if matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, then the determinant of the matrix A can be calculated by the $\det\,(A) = (ad - bc)$.

The computation of determinant can be extended to higher dimension square matrices recursively through Laplace expansion. For matrix $A \in \mathbb{R}^{n\times n}$,

$$det\,(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij},$$

where M_{ij} is the determinant of the $(n-1) \times (n-1)$ minor obtained by removing the i -th row and j -th column of $A.$ The determinant is particularly useful in solving systems of linear equations and in defining matrix invertibility, where A is invertible if and only if $det(A) \neq 0$. For instance, if matrix $A = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$, then

$$det\,(A) = a_1 \begin{vmatrix} b_2 & b_3 \\ c_2 & c_3 \end{vmatrix} - a_2 \begin{vmatrix} b_1 & b_3 \\ c_1 & c_3 \end{vmatrix} + a_3 \begin{vmatrix} b_1 & b_2 \\ c_1 & c_2 \end{vmatrix}.$$

Thus,

DOI: 10.54254/2753-8818/2025.GL31275

$$det\left(A\right)=a_1\left(b_2c_3-b_3c_2\right)-a_2\left(b_1c_3-b_3c_1\right)+a_3\left(b_1c_2-b_2c_1\right),$$

by using the case we've illustrated for 2×2 matrices.

2.2. Transpose and inverse matrix

The transpose of a matrix is an operation that switches its rows and columns by flipping it along the main diagonal [5]. Given an $m\times n$ matrix A ,its transpose, denoted as A^T , is an $n\times m$ matrix where the element at row i and column j in A moves to row j and column i in A^T . Mathematically, this is expressed as:

$$\left(A^T\right)_{ij}=A_{ji},\ 1\le i\le n,1\le j\le m.$$

The transpose operation has several useful properties. First, taking the transpose twice returns the original matrix, i.e., $(AT)^T=A$. Second, the transpose distributes over addition, meaning $(A+B)^T=A^T+B^T$, and scalar multiplication follows $(cA)^T=cA^T$ for any scalar c. A particularly important property is that the transpose of a product reverses the order of multiplication: $(BA)^T=B^TA^T$. Additionally, a matrix is said to be symmetric if it is equal to its transpose, i.e., $A=A^T$, which is a fundamental property in many mathematical and physical applications.

The transpose is widely used in linear algebra, statistics, and machine learning. For example, in inner product spaces, transposing a column vector turns it into a row vector, which makes certain matrix operations possible. In machine learning, transposing is often used to adjust how data is arranged. This helps with tasks like matrix multiplication in neural networks and regression analysis.

The inverse of a square matrix A is another matrix, written as $A^{-1}A^{-1}A^{-1}$, that satisfies the basic identity:

$$AA^{-1}=A^{-1}A=I$$

Here, I is the identity matrix with the same size as A . A matrix has an inverse only if it is square $(n\times n)$ and has full rank. This means that its determinant is nonzero: $det(A)\ne 0$. If these conditions are not set up, the matrix is called singular and does not have an inverse.

Gaussian elimination is a systematic algorithm often used to solve linear systems. It can also be used to find the inverse of a matrix. The main idea is to transform a given $n\times n$ matrix A into the identity matrix I by using elementary row operations. At the same time, you apply the same operations to the identity matrix in order to find A^{-1} . The process begins by combining the original matrix A with the identity matrix I to form $[AI]$. In the forward elimination step, row operations are used to turn A into an upper triangular matrix by choosing pivot elements and eliminating the entries below them. If the pivot element is zero, rows are swapped to ensure that the pivot is not zero. To create zeros below the diagonal, each column is processed by subtracting right multiples of the pivot row. This process continues until the left side of the augmented matrix becomes upper triangular. Once this form is reached, the matrix is ready for the next phase of row operations. This continues until the matrix is in upper triangular form. Next, scale the matrix and the pivot becomes 1. Then use the backward substitution to eliminate the pivots above. These processes transform the

DOI: 10.54254/2753-8818/2025.GL31275

augmented matrix into an identity matrix. In other words, $A^{-1}[A|I] = [I|A^{-1}]$. If the matrix cannot be turned into the identity matrix, the matrix is singular, and it does not have an inverse. Gaussian elimination is a widely used method to find inverse. It often uses in numerical computations and solving linear systems of equations.

2.3. Eigenvalues and eigenvectors

Eigenvalues and eigenvectors are main concepts in linear algebra. They help us understand linear transformations and solve systems of linear equations. These ideas are used in many fields, such as physics, computer science, engineering, machine learning and statistics. This section explains what eigenvalues and eigenvectors are. It also shows how to calculate them and why they matter in both theory and real-world problem [2].

Given a square matrix A of size $n \times n$, an eigenvector v and its eigenvalue λ satisfy the equation $Av = \lambda v$. Here, A is the transformation matrix or a linear operator, and v is the eigenvector that is a non-zero vector. In simple terms, applying the matrix A to v just stretches or shrinks it. The direction of ν stays the same. The scaling factor λ is the eigenvalue. During this transformation, the direction of the eigenvector remains unchanged; it is just stretched or compressed by the factor λ .

To find the eigenvalues and eigenvectors of a matrix A , we start by calculating the eigenvalues. The eigenvalues are the solutions to the characteristic equation. This equation comes from the determinant of $A - \lambda I$, where λ is a scalar. I is the identity matrix with the same size as A . The characteristic equation is obtained by setting the determinant of $A - \lambda I$ equal to zero, i.e., $det(A - \lambda I) = 0$. This results in a polynomial equation in λ , and solving this equation gives the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, which are the roots of the polynomial. Once the eigenvalues are found, we can compute the eigenvectors. For each eigenvalue λ , we substitute it into the equation $(A - \lambda I)v = 0$, which is a system of linear equations. Solving this system allows us to identify the non-zero eigenvectors that are in the eigenspace for each eigenvalue. The eigenvalue multiplicity affects the number of linearly independent eigenvectors in the eigenspace, which is the set of all scaled versions of the eigenvector.

2.4. Gram-schmidt process and QR decomposition

Converting a set of linearly independent vectors into orthogonal vectors that are mutually perpendicular is achieved through the Gram-Schmidt process. The process is done by progressively subtracting the projections of previous vectors from the current vector to guarantee orthogonality, and then normalizing the vectors if desired. The result is a set of orthogonal (or orthonormal, if normalized) vectors [5].

Given a set of linearly independent vectors $a_1, a_2, \ldots, a_n$, the goal is to generate an orthogonal set of vectors $q_1, q_2, \ldots, q_n$. First, we start by normalizing the first vector a_1 to obtain the first orthogonal vector q_1 :

$$q_1 = \frac{a_1}{|| a_1 ||}$$

where $|| \cdot ||$ computes the l^2 -norm of the vector. Next, for the second vector a_2 , we subtract the projection of a_2 onto q_1 to ensure that q_2 is orthogonal to q_1 by:

DOI: 10.54254/2753-8818/2025.GL31275

$$q_2 = \frac{a_2 - \langle a_2, q_1 \rangle q_1}{\|a_2 - \langle a_2, q_1 \rangle q_1\|}$$

where $\langle a, b \rangle$ denotes the dot product. For the third vector a_3 , we subtract the projections of a_3 onto both q_1 and q_2 to ensure that q_3 is orthogonal to both q_1 and q_2 :

$$q_3 = \frac{a3 - \langle a3, q_1 \rangle q_1 - \langle a3, q_2 \rangle q_2}{\|a3 - \langle a3, q_1 \rangle q_1 - \langle a3, q_2 \rangle q_2\|}$$

This process continues for the remaining vectors. Afterward, the set of vectors $q_1, q_2, \ \ldots, q_n$ will be orthogonal.

QR decomposition is a method used in linear algebra to factor a matrix into two components: a Q matrix (orthogonal or unitary matrix) and an R matrix (upper triangular matrix). Specifically, given a matrix A , QR decomposition expresses it as:

$$A = QR$$

Here, the Q matrix is orthogonal, meaning its columns are orthonormal vectors, and it satisfies $Q^T Q = I$, where I is the identity matrix. The R matrix is upper triangular and represents the projection of A onto the orthonormal basis formed by the columns of Q .

The process of QR decomposition involves finding two matrices, Q and R , such that their product reconstructs the original matrix A . This can be done using several methods, such as the Gram-Schmidt process that introduced above. In the Gram-Schmidt process, the columns of A are iteratively orthogonalized to form the columns of Q , and the corresponding coefficients are used to construct the matrix R . The Q matrix contains orthonormal vectors, while R is an upper triangular matrix whose entries are determined by the projections of the original matrix.

For instance, in the Gram-Schmidt process, given matrix A with columns $a_1, a_2, \ldots, a_n$, the first column of Q is obtained by normalizing a_1 by $q_1 = \frac{a_1}{\| a_1 \|}$ Subsequent columns of Q are formed by subtracting projections onto the previously found columns. The matrix R is then calculated as:

$$R = Q^T A$$

QR decomposition is commonly utilized in the solution of linear systems, specifically overdetermined systems in least squares problems. The numerical stability of Q, which is orthogonal, ensures that the solution process minimizes errors. The QR algorithm and other eigenvalue algorithms use QR decomposition as a fundamental tool to converge to a triangular form, revealing the eigenvalues of a matrix.

3. Applications

3.1. Ordinary least square

The principal method of estimating the coefficients of a linear model is based on Ordinary least squares regression, which minimizes the residual sum of squares (RSS) between the observed and

DOI: 10.54254/2753-8818/2025.GL31275

predicted values [6]. The linear model is given by:

$$y = X\beta + \epsilon$$

where $y \in \mathbb{R}^n$ is the vector of observed dependent variable values, $X \in \mathbb{R}^{n\times p}$ is the design matrix (each row corresponds to an observation, and each column corresponds to an independent attribute), $\beta \in \mathbb{R}^p$ is the vector of unknown regression coefficients that needs to be estimated, $\epsilon \in \mathbb{R}^n$ is the vector of residuals or errors, assumed to satisfy $E[\epsilon] = 0$ and $Var(\epsilon) = \sigma^2 I$.

In this model, $X\beta$ represents the predicted values of y , meaning that each observation in y is approximated as a linear combination of the columns of X weighted by the corresponding coefficients in β . The term ϵ accounts for the deviation between the actual observed values and the predicted ones, capturing randomness, measurement errors, and unobserved factors that influence y . The goal of OLS is to determine the optimal coefficient vector β by minimizing the RSS, given by:

$$S(\beta) = \sum_{i=1}^{n} (y_i - x_i\beta)^2 = (y - X\beta)^T (y - X\beta)$$

Here, the subscript i denotes the i -th observation, where y_i is the observed value and x_i is the corresponding feature vector. The prediction $x_i\beta$ is a weighted sum of features, and the residual $\epsilon_i = y_i - x_i\beta$ captures the difference between actual and predicted values. RSS aggregates these squared residuals for all n observations, with the matrix representation providing a concise representation.

To obtain the optimal estimator $\widehat{\beta}$, we minimize $S(\beta)$ by taking its derivative with respect to β and setting it to zero:

$$\frac{\partial S(\beta)}{\partial \beta} = -2X^T\left(y - X\beta\right) = 0$$

Rearranging gives the normal equation:

$$X^T X\widehat{\beta} = X^T y$$

Solving for $\widehat{\beta}$, we obtain:

$$\widehat{\beta} = \left(X^T X\right)^{-1} X^T y$$

provided that $X^T X$ is invertible. This solution ensures that $\widehat{\beta}$ is the unique coefficient vector that minimizes the sum of squared residuals. The function to be minimized in this context is:

DOI: 10.54254/2753-8818/2025.GL31275

$$\min_{\beta}\sum_{i=1}^{n}\left(y_i-\sum_{j=1}^{n}x_{ij}\beta_j\right)^2$$

where each term $\left(y_i-\sum_{j=1}^{n}x_{ij}\beta_j\right)^2$ represents the squared vertical distance between the observed data point and the regression line. The function creates a convex quadratic surface that looks like a paraboloid, ensuring that the derivatives disappear at a unique minimum.

From a geometric perspective, OLS projects y onto the column space of X, making $X\widehat{\beta}$ the closest point in this space to y. The residual vector $y-\ X\widehat{\beta}$ is orthogonal to this space, satisfying:

$$X^T\left(y-\ X\widehat{\beta}\right)=0$$

Since each row vector x_i of X consists of feature values for observation i, it can be expressed as:

$$x_i=\left[x_{i1}\ x_{i2}\ \cdots x_{in}\right]\begin{bmatrix}\beta_1\\ \beta_2\\ \vdots\\ \beta_n\end{bmatrix}=\sum_{j=1}^{n}x_{ij}\beta_j$$

which states that the inner product of the residuals and the columns of X is zero, confirming that no further linear adjustment can reduce the residual sum of squares.

However, when X^TX is nearly singular, meaning it is close to being non-invertible due to multicollinearity or numerical instability, an alternative computational approach is required. QR decomposition provides a numerically stable solution by decomposing X into an orthogonal matrix Q and an upper triangular matrix R, so that $X=\ QR$ where $Q^TQ=I$. Substituting this into the normal equation $X^TX\beta=X^Ty$, we get

$$(QR)^T\left(QR\right)\beta=(QR)^Ty$$

which simplifies to

$$R^TQ^TQR\beta=R^TQ^Ty.$$

Since $Q^TQ=I$, this reduces to

$$R^TR\beta=R^TQ^Ty.$$

Multiplying both sides by $\left(R^T\right)^{-1}$, we obtain

DOI: 10.54254/2753-8818/2025.GL31275

$$R\beta = Q^T y.$$

Since R is upper triangular, β can be efficiently solved using back-substitution. OLS provides an optimal way to estimate coefficients under standard assumptions, while QR decomposition ensures numerical stability when direct inversion of $X^T X$ is problematic. The geometric meaning of OLS is simple but powerful. It shows that OLS works like an orthogonal projection. The data is projected onto a subspace in a high-dimensional space.

3.2. Principal Component Analysis

Principal Component Analysis (PCA) helps to reduce the dimensions of high-dimensional data. It can remain the original variance as it possible. PCA is used to explore data, and visualize and preprocess. It can help eliminate the redundancy and increase its computation rate.

PCA finds new axes, called principal components. They are orthogonal and capture the most variance in the data. The first component holds the most variance and the second holds the next most, and so on. This transformation uses eigenvalue decomposition on the covariance matrix. For a dataset X with $\boldsymbol{m}$ samples and $\boldsymbol{n}$ features, the covariance matrix is calculated as

$$C = \frac{1}{m} X^T X$$

Here, C is an $n \times n$ symmetric matrix that shows the correlations between features. The eigenvectors and eigenvalues of $\boldsymbol{C}$ satisfy:

$$Cv_i = \lambda_i v_i$$

Here, v_i is the i -th principal component (eigenvector), and λ_i is the eigenvalue, which shows the amount of variance it explained. The eigenvalues are listed in descending order, meaning the components with higher values capture more of the data's variation. The size of λ_i determines how much of the total variance is explained by its corresponding principal component. This means that the first few components capture most of the dataset’s information, helping to reduce dimensionality.

Once the principal components are found, the original dataset is projected onto the new coordinate system. This transformation is:

$$X^{'} = X_c V_k$$

Here, X' is the reduced dataset, Xc is the mean-centered data, and Vk contains the top k principal components. This projection keeps key patterns while cutting down dimensions. It lines up with the directions of highest variance to make the data clearer and easier to handle.

PCA helps find a simpler way to look at complex data. It picks the directions where the data spreads out the most. Even if the data has many features, it often lies on a smaller shape, and PCA helps us see that. It removes repeated information, shows patterns more clearly, and makes tasks like grouping or sorting the data easier. The new version of the data is easier to understand and work with.

DOI: 10.54254/2753-8818/2025.GL31275

While PCA is a linear technique, many real-world datasets show nonlinear patterns that standard PCA cannot capture. To deal with non-linear patterns in data, researchers have developed extensions like kernel PCA [7] and deep learning methods such as autoencoders [8]. Kernel PCA improves the standard PCA by applying the non-linear mapping function $\emptyset(x)$. This will project the data into higher higher-dimensional space, and it make the linear model easier to find. In this space, the covariance matrix is replaced by the kernel matrix:

$$K_{ij} = \phi(x_i)^T \phi(x_j)$$

This helps kernel PCA find the linear model the regular PCA may miss. The kernel function is chosen to control how the data is transformed. It plays an important role in revealing hidden and complex structures.

The polynomial kernel captures non-linear relationships by increasing the product of two dots to power, representing:

$$K(x_i,\ x_j) = \left(x_i^T x_j + c\right)^d$$

The Gaussian Radial Basis Function (RBF) kernel maps the data to an infinite-dimensional space. It seems complex, but it helps RBF find the non-linear model. And it is useful for RBF in image recognition. The way the sigmoid kernel works is like the activation functions in neural networks. So, it makes it suitable for the model which need to imitate decision boundaries. On the other hand, the cosine similarity kernel is not relative to the distance. Instead, it measures the angle between vectors. It is useful for analysing the high-dimensional tasks, because direction is more important than distance.

Choosing the right kernel can make kernel PCA find the complex and hidden models that are ignored by standard PCA. It makes it very useful for pulling out the important features, cutting down the noise and improving the ability of the machine to study the models. It is a big upgrade to traditional PCA, and it becomes the first method in data science and model recognition [9].

3.3. Real-world applications

3.3.1. PCA in low-rank approximation

Principal Component Analysis (PCA) is very suitable for understanding the large high-dimensional datasets by focusing on explaining the key components of most of the variance [10]. It is the first method to retain the main information while reducing the dimension. In the process of compressing the data, PCA is helpful to simplify the large dataset by using fewer parts to save storage space and computing power. For example, the video streaming platform can PCA to compress HD videos, focusing on the key visual elements like edges, and discarding the unimportant details. This can not only save space but also raise the speed of data transfer. It can play smoothly even when the bandwidth is limited. It is useful for different mobile streaming under different net conditions. It also plays an important role in efficient data usage in the user experience.

PCA is widely used in scientific computation and machine learning. In the field of fluid dynamics and quantum mechanics, it is helpful for large, sparse matrices. This reduces the costs of computations. At the same time, it maintains the accuracy. In the machine study, PCA improves the

DOI: 10.54254/2753-8818/2025.GL31275

sorting, clustering, and anomaly detection. It deletes the redundant data, emphasizes the relevant features, and improves the efficiency and generalization [11].

In the sociological field, PCA is helpful to analyze the potential structure in the complex dataset, such as it can identify the potential dimensions of social attitude and values in large surveys. For example, in a study of education inequality, PCA can turn many indicators such as parental education, income and school type into a smaller set, in order to compare the difference between structures.

In the future, PCA can support more studies, such as identity and social class, by investigating the deeper patterns and data in survey or social media. With the growth of the number of datasets and its complexity, PCA will become the main method of reducing the noise while emphasizing the meaningful trend of populations. It can track how cultural values vary with time.

3.3.2. Ordinary Least Square for regression analysis

Ordinary Least Squares (OLS) regression is commonly used in marketing research to analyse consumer behaviour and market trend. A typical application is assessing how advertising impacts the sales. For example, a company may build an OLS model to measure the impact of changing the ad spending. The businesses can optimize marketing policy and allocate resources efficiently. OLS is essential in understanding how marketing and investing impact sales. It makes company is able to adjust budgets and set efficient ad and enhance profitability. It also proves the value in analysing price elasticity, predicting customer demand, and modelling consumer sentiment. It demonstrates its flexibility and importance in marketing analytics [12].

Except for the field of marketing, OLS regression is a main tool of sociological. It was widely used in analysing the effect of social factors such as income, race and education make impact to health, mobility and political participation. For example, researchers may use OLS to estimate how socioeconomic factors affect the way people get access to higher education, in order to help inform policies with greater equity.

On the other hand, OLS regression plays a bigger role in causal analysis. Using better data and methods, such as regularization and instrumental variables, in order to understand how the social policies affect real-world outcomes, such as how welfare affects job access and how school fundings affect long-term achievement. As sociological research becomes more data-driven, both PCA and OLS will be powerful tools for understanding inequality, public opinion, and social change.

4. Conclusion

In this work, we provided a comprehensive overview of the method of matrix computation. And we explore the key operation, properties and various decompositions. We also explore how these methods are applied, focusing on OLS regression and PCA. What's more, we emphasize their real-world applications to prove their importance. By connecting the theory and experiments, this research clearly explains the key role the matrix computations play in data analysing and modelling tasks.

In future work, we will explore more advanced matrix computation, including the numerical approximation methods and regularization strategies of OLS to make them more flexible in complex models. We will also explore the methods of research in non-linear dimensionality, such as manifold learning and autoencoders. Moreover, we will explore the technologies of emerging matrix-based, such as low-rank approximations and tensor decompositions in order to improve the efficiency and scalability in large computations.

DOI: 10.54254/2753-8818/2025.GL31275

References

[1] Golub, G. H., & Van Loan, C. F. (2013). Matrix computations. JHU press.
[2] Bronson, R., Costa, G. B., Saccoman, J. T., & Gross, D. (2023). Linear algebra: Algorithms, applications, and techniques.
[3] Stoll, M. (2020). A literature survey of matrix methods for data science. GAMM-Mitteilungen, 43(3), e202000013.
[4] Salomon, D. (2007). Transformations and projections in computer graphics. Springer Science & Business Media.
[5] Strang, G. (2022). Introduction to linear algebra. Wellesley-Cambridge Press.
[6] Van Huffel, S., & Zha, H. (1993). 10 The total least squares problem.
[7] Schölkopf, B., Smola, A., & Müller, K. R. (1997, October). Kernel principal component analysis. In International conference on artificial neural networks (pp. 583-588). Berlin, Heidelberg: Springer Berlin Heidelberg.
[8] Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv: 1606.05908.
[9] Bakır, G. H., Weston, J., & Schölkopf, B. (2004). Learning to find pre-images. Advances in neural information processing systems, 16(449-456), 1.
[10] Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv: 1404.1100.
[11] Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.
[12] Seyda Deligonul, Z., Chabowski, B. R., Seggie, S. H., Xu, S., & Tamer Cavusgil, S. (2009).