

Linear Algebra Review

1 Vectors

1.1 Definition

A real n -vector is an ordered n -tuple of real numbers of the form

$$v = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

We say that $v \in R^n$ if v is a real n -vector.

1.2 Addition

Addition of two or more vectors of the same size is defined as follows

$$a + b = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{pmatrix}$$

The addition is commutative and associative.

1.3 Scalar multiplication

Scalar multiplication of a vector is defined as follows

$$kv = k \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} ka_1 \\ ka_2 \\ \vdots \\ ka_n \end{pmatrix}$$

A linear combination of n vectors is defined by $v = kw_1 + kw_2 + \cdots + kw_n$ where v is a vector of the same dimensionality as w_1, w_2, \dots , and w_n .

1.4 Inner Product

The *inner product* (also known as *dot product*) of two n -dimensional vectors a and b is defined as $a^T b$ which is a scalar.

$$a^T b = a \cdot b = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n = \sum_{i=1}^n a_i b_i$$

1.5 Complexity

The addition operation requires $O(n)$ addition operations for two vectors of size n . The multiplication by a scalar requires $O(n)$ multiplication operation for a vector of size n . The inner products requires $O(n)$ multiplication operations plus one addition operations.

2 Matrices

2.1 Definition

An $m \times n$ matrix A is an array of mn numbers with m rows and n columns

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

The matrix A is also written as $A = (a_{ij})$. The size of A is $m \times n$.

2.2 Matrix Addition and multiplying by a scalar

Matrices of the same size can be added, and matrices can be multiplied by scalars, according to the following rules:

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ b_{21} & \cdots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix} = \begin{pmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & \cdots & a_{2n} + b_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{pmatrix}$$

$$kA = \begin{pmatrix} ka_{11} & ka_{12} & \cdots & ka_{1n} \\ ka_{21} & ka_{22} & \cdots & ka_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ka_{m1} & ka_{m2} & \cdots & ka_{mn} \end{pmatrix}$$

2.2.1 Complexity

The addition operation requires $O(mn)$ addition operations for two matrices of size $m \times n$. The multiplication by a scalar requires $O(mn)$ multiplication operation for a matrix of size $m \times n$.

2.3 Matrix Multiplication

If $A = (a_{ij})$ is an $m \times n$ matrix, and $B = (b_{jk})$ is an $n \times p$ matrix, then the product AB is given by $C = (c_{ik})$ where

$$c_{ik} = \sum_{j=1}^n a_{ij}b_{jk}$$

2.3.1 Complexity

If both A and B are square $n \times n$ matrices then the complexity of multiplication is $O(n^3)$.

2.4 Matrix Inversion

Let A be a square $n \times n$ matrix. The inverse of A , if it exists, is a matrix A^{-1} such that

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

Not every matrix A has an inverse. A has an inverse if $|A| \neq 0$. In this case we say that A is invertible.

2.4.1 Complexity

The inverse of the matrix can be obtained by using *the Gauss Elimination* method which is $O(n^3)$. There are other methods that perform a little faster, for example *the Strassen's method* that requires $O(n^{\log_2 7}) = O(n^{2.807})$ operations.

3 Eigenvalues and Eigenvectors

3.1 Definition

Let A be a complex square $n \times n$ matrix. Then if λ is a complex number and x is a nonzero complex column vector of size n satisfying $Ax = \lambda x$, we call x an eigenvector of A , while λ is called an eigenvalue of A . We also say that x is an eigenvector corresponding to the eigenvalue λ .

3.2 Characteristic Equation and Characteristic Polynomial

The equation $\det(A - \lambda I) = 0$ is called *the characteristic equation* of A , while the polynomial $\det(A - \lambda I)$ is called *the characteristic polynomial* of A . Hence the eigenvalues of A are the roots of the characteristic polynomial of A .

3.3 The Diagonal form of a Matrix

Suppose the $n \times n$ matrix A has n linearly independent eigenvectors. Then if these vectors are chosen to be the columns of a matrix S , it follows that $S^{-1}AS$ is a diagonal matrix Λ , with the eigenvalues of A along its diagonal:

$$S^{-1}AS = \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$$

3.4 The Trace of a Matrix

Let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of an $n \times n$ matrix A with corresponding eigenvectors v_1, \dots, v_n . The trace of A $\text{Tr}(A)$ is the sum of its eigenvalues.

$$\text{Tr}(A) = \sum_i \lambda_i$$

3.5 Complexity

Solving the eigenvalue problem involves the following steps

1. Compute the determinant of $A - \lambda I$.
2. Find the roots of the characteristic polynomial. The n roots are the eigenvalues.
3. For each eigenvalue solve the equation $(A - \lambda I)x = 0$. Since the determinant is zero, there are solutions other than $x = 0$. those are the eigenvectors.

The previous steps are not practical for large n since it involves finding the roots of high order polynomials. Alternatively, the methods for actually computing the eigenvalues and eigenvectors of a matrix A usually are preceded by a reduction step, in which the matrix A is transformed to a *similar* matrix B having the same eigenvalues as A . The matrix B has a simpler structure than A , so the standard methods for computing the eigenvalues and eigenvectors are computationally less expensive when applied to B than when applied to A . The reduction is performed in $O(n^3)$ operations for an $n \times n$ matrix. An iterative algorithms like the **QR** method is then applied to the reduced matrix for computing the eigenvalues. The method is expensive since a complete step for a dense $n \times n$ matrix requires $O(n^3)$ operations. The speed of convergence of the iterative algorithm depends on the structure of the matrix.

4 Singular Value Decomposition (SVD)

4.1 Definition

If A is an $m \times n$ matrix, then A can be rewritten as

$$A = UDV^T$$

where

$$U = [u_1 \cdots u_m]$$

is an $m \times m$ matrix and

$$V = [v_1 \cdots v_n]$$

is an $n \times n$ matrix and

$$D = \text{diag}(\sigma_1, \cdots, \sigma_p), p = \min\{m, n\}$$

is an $m \times n$ matrix with

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$$

The matrices U, V are orthogonal matrices. The elements $\sigma_i (i = 1 : \min\{m, n\})$ on the main diagonal of D are called the *singular values* of A . The smallest singular value of a square matrix A gives the distance of A to the *nearest* singular matrix.

4.2 Properties

The following relations hold for each $\sigma_i (i = 1 : \min\{m, n\})$. For this reason, we call (u_i, v_i) a *singular vector pair*

$$\begin{aligned}Av_i &= \sigma_i u_i, \\ A^T u_i &= \sigma_i v_i\end{aligned}$$

The vectors u_i and v_i are the i^{th} eigenvectors of the matrices AA^T and $A^T A$ respectively.

4.3 Complexity

The singular values and the singular value decomposition of an $m \times n$ matrix A can be computed by a method closely related to the QR method.

5 Systems of linear Equations

5.1 Definition

A linear equation in n unknowns x_1, x_2, \dots, x_n is an equation of the form

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n = b$$

where a_1, a_2, \dots, a_n, b are given real numbers.

A system of n linear equations in n unknowns x_1, x_2, \dots, x_n is a family of linear equations

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n\end{aligned}$$

We can write this system in the form $Ax = b$, where

$$A = (a_{ij}), x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

We wish to determine if such a system has a solution, that is to find out if there exist numbers x_1, x_2, \dots, x_n which satisfy each of the equations simultaneously. The solution is unique if $|A| \neq 0$. We say that the system is *consistent* if it has a solution. Otherwise the system is called *inconsistent*.

5.2 The condition number of a matrix

The condition number c of a matrix A is defined as the ratio between its largest and its smallest eigenvalue.

$$c = \lambda_{max}/\lambda_{min}$$

It attempts to measure the sensitivity to roundoff errors or the vulnerability of a solution for solving $Ax = b$. That is, when A and b are slightly changed, how great is the effect on $x = A^{-1}b$. The matrix A is called ill-conditioned if it is sensitive to roundoff errors.

5.3 Solving Linear Equations

There are several algorithms used for solving a system of linear equations. *The Gauss elimination* and *The Gauss Jordan* methods are used for small dense matrices (a 100×100 matrix is small in this sense). The complexity of both methods is $O(n^3)$. Many problems in practice require the solution of very large systems of linear equations in which the matrix A is sparse. In this case the usual elimination methods can not be applied.

5.3.1 The Gauss Elimination Method

The method is closely related to that of computing the inverse A^{-1} of the matrix A provided this inverse exists. The system can be solved by reducing the matrix A to a triangular matrix using the standard row operations. There are three types of elementary row operations that can be performed:

1. Interchanging two rows.
2. Multiplying a row by a nonzero scalar.
3. Adding a multiple of one row to another row.

5.3.2 Iterative Methods for large systems of linear equations

Iterative methods like *the Jacobi*, *the Gauss Siedel* and *the SOR* methods begin with an initial vector $x^{(0)}$ and generate a sequence of vectors

$$x^{(0)} \longrightarrow x^{(1)} \longrightarrow x^{(2)} \longrightarrow \dots$$

which converge toward the desired solution x . An individual iteration step $x^{(i)} \longrightarrow x^{(i+1)}$ requires an amount of work which is comparable to the multiplication of the matrix A with a vector. Iterative methods converges very slowly and there are certain properties that the matrix A must satisfy in order to guarantee the convergence of these methods towards the exact solution in a finite number of steps.

For more information about the topics covered in this review refer to [1],[2],[3].

References

- [1] K. Matthews. *Elementary Linear Algebra Lecture notes*. www.numbertheory.org/book/, 1991.
- [2] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, second edition edition, 1993.
- [3] G. Strang. *Linear Algebra and Its Applications*. HBJ, third edition edition, 1988.