JSS Mahavidyapeetha
Sri Jayachamarajendra College of Engineering (SJCE),
Mysore – 570 006
An Autonomous Institute Affiliated to
Visvesvaraya Technological University, Belgaum

"Determine the projection matrix P spammed over a given space and also to determine the matrices that cannot be diagonalised"

Report submitted in partial fulfillment of
curriculum prescribed for the Linear Algebra
for the award of the degree of
BACHELOR OF ENGINEERING IN COMPUTER SCIENCE
AND ENGINEERING by

Nithin Prabhu G - 4JC15CS068
Mrudula N M - 4JC15CS137

Submitted to,
Dr. Roopamala T D
Associate Professor,
Department of Computer Science & Engineering,
SJCE, Mysore

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Contents

Chapter 1

INTRODUCTION

1.1 Matrix

A matrix (plural: matrices) is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns. The individual items in an $m \times n$ matrix A, often denoted by $a_{i,j}$, where max i = m and max j = n, are called its elements or entries.

1.1.1 Projection Matrix

A projection matrix P is a n cross n square matrix that gives a vector space projection from R^n to a subspace W. The columns of P are the projections of the standard basis vectors, and W is the image of P. A square matrix P is a projection matrix iff $P^2 = P$.

A projection matrix P is orthogonal iff $P = P^*$, where P^* denotes the adjoint matrix of P. A projection matrix is a symmetric matrix iff the vector space projection is orthogonal. In an orthogonal projection, $< v, Pw >=< v_W, Pw >=< Pv, w >$.

1.1.2 Invertible Matrices

In linear algebra, an n-by-n square matrix A is called invertible (also nonsingular or nondegenerate) if there exists an n-by-n square matrix B such that
$AB = BA = I_n$

where I_n denotes the n-by-n identity matrix and the multiplication used is ordinary matrix multiplication. If this is the case, then the matrix B is uniquely determined by A and is called the inverse of A, denoted by $A^{-1}1$.

A square matrix that is not invertible is called singular or degenerate. A square matrix is singular if and only if its determinant is 0. Singular matrices are rare in the sense that a square matrix randomly selected from a continuous uniform distribution on its entries will almost never be singular.

1.2 Determinant

The determinant $det(A)$ or $|A|$ of a square matrix A is a number encoding certain properties of the matrix. A matrix is invertible if and only if its determinant is nonzero. Its absolute value equals the area (in R_2) or volume (in R_3) of the image of the unit square (or cube), while its sign corresponds to the orientation of the corresponding linear map: the determinant is positive if and only if the orientation is preserved.

The determinant of 2-by-2 matrices is given by

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

The determinant of 3-by-3 matrices involves 6 terms (rule of Sarrus). The more lengthy Leibniz formula generalises these two formulae to all dimensions.

Adding a multiple of any row to another row, or a multiple of any column to another column, does not change the determinant. Interchanging two rows or two columns affects the determinant by multiplying it by -1. Using these operations, any matrix can be transformed to a lower (or upper) triangular matrix, and for such matrices the determinant equals the product of the entries on the main diagonal; this provides a method to calculate the determinant of any matrix. Finally, the Laplace expansion expresses the determinant in terms of minors, that is, determinants of smaller matrices. This expansion can be used for a recursive definition of determinants (taking as starting case the determinant of a 1-by-1 matrix, which is its unique entry, or even the determinant of a 0-by-0 matrix, which is 1), that can be seen to be equivalent to the Leibniz formula. Determinants can be used to solve linear systems using Cramer's rule, where the division of the determinants of two related square matrices equates to the value of each of the system's variables.

1.3 Eigenvalues and eigenvectors

In linear algebra, an eigenvector or characteristic vector of a linear transformation is a nonzero vector that only changes by a scalar factor when that linear transformation is applied to it. More formally, if T is a linear transformation from a vector space V over a field F into itself and v is a vector in V that is not the zero vector, then v is an eigenvector of T if $T(v)$ is a scalar multiple of v. This condition can be written as the equation

$Tv = \lambda v$

where λ is a scalar in the field F, known as the eigenvalue, characteristic value, or characteristic root associated with the eigenvector v.

If the vector space V is finite-dimensional, then the linear transformation T can be represented as a square matrix A, and the vector v by a column vector, rendering the above mapping as a matrix multiplication on the left hand side and a scaling of the column vector on the right hand side in the equation

$Av = \lambda v$

There is a correspondence between n by n square matrices and linear transformations from an n-dimensional vector space to itself. For this reason, it is equivalent to define eigenvalues and eigenvectors using either the language of matrices or the language of linear transformations.

Geometrically an eigenvector, corresponding to a real nonzero eigenvalue, points in a direction that is stretched by the transformation and the eigenvalue is the factor by which it

is stretched. If the eigenvalue is negative, the direction is reversed. Now consider the linear transformation of n-dimensional vectors defined by an n by n matrix A,

$Av = w$

or

$$\begin{bmatrix} A_{11} & A_{12} & ... & A_{1n} \\ A_{21} & A_{22} & ... & A_{2n} \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ A_{n1} & A_{n2} & ... & A_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ . \\ . \\ . \\ v_n \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ . \\ w_n \end{bmatrix}$$

where, for each row,
$w_i = A_{i1}v_1 + A_{i2}v_2 + \cdots + A_{in}v_n = \sum_{j=1}^{n} A_{ij}v_j$.
If it occurs that v and w are scalar multiples, that is if
$Av = w = \lambda v$,—(1)
then v is an eigenvector of the linear transformation A and the scale factor λ is the eigenvalue corresponding to that eigenvector. Equation (1) is the eigenvalue equation for the matrix A.
Equation (1) can be stated equivalently as
$(A - \lambda I)v = 0$,
where I is the n by n identity matrix.

Chapter 2

PROBLEM STATEMENT

1. Design a program to find the projection matrix P onto the space spanned by 2 vectors.

2. Design a program to check which of the matrices cannot be diagonalised.

Chapter 3

ALGORITHM

3.0.1 To find the projection matrix P onto the space spanned by 2 vectors

1. Give 2 input vectors whose projection matrix has to be found.
2. Find the transpose of those matrices.
3. Then find the dot product of obtained transpose with the matrices.
4. Similarly repeat these steps to obtain projection matrix P given by
   P = A * (A * A^T) A^T

3.0.2 To check which of the matrices cannot be diagonalised

1. Let A be the $n \times n$ matrix that you want to diagonalize.
2. Find the characteristic polynomial $p(t)$ of A.
3. Find eigenvalues λ of the matrix A and their algebraic multiplicities from the characteristic polynomial $p(t)$.
4. For each eigenvalue λ of A, find a basis of the eigenspace E_λ.
   If there is an eigenvalue λ such that the geometric multiplicity of λ, $dim(E_\lambda)$, is less than the algebraic multiplicity of λ, then the matrix A is not diagonalizable. If not, A is diagonalizable, and proceed to the next step.
5. If we combine all basis vectors for all eigenspaces, we obtained n linearly independent eigenvectors $v_1 v_2 \dots v_n$.
6. Define the nonsingular matrix
   $S = [v_1 v_2 \dots v_n]$.
7. Define the diagonal matrix D, whose (i,i)-entry is the eigenvalue λ such that the i-th column vector v_i is in the eigenspace E_λ.
8. Then the matrix A is diagonalized as
   $S^{-1}AS = D$.

Chapter 4

PYTHON CODE

problem1.py

```
import numpy as np
a1 = np.matrix([[1],[1],[0]]);
a2 = np.matrix('1;1;-1');

    a = np.matrix('1 1;0 1;1 -1');

    temp = np.dot(np.transpose(a) , a)

    print("The projection matrix P is given by, P = A * (A * A^T) A^T")

    print(' A * A^T = ');
print(temp)

    det = temp[0,0] * temp [1,1] - temp[0,1] * temp[1,0];

    temp[0,0],temp[1,1] = temp[1,1],temp[0,0];
temp[0,1] = -1 * temp[0,1];
temp[1,0] = -1 * temp[1,0];

    temp = (1/det) * temp

    print(" ( A * A^T ) inverse = ")
print(temp)

    temp = np.dot(a,temp)

    print(" A * (A * A^T) = ")
print(temp)

    temp = np.dot(temp,np.transpose(a))
print(" P = ")
print(temp)
```

program2.py

```
import numpy as np

    a = np.matrix('2 2;-2 -2')

    y1,y2 = np.linalg.eigvals(a);

    print('Matrix A : ')
print(a);

    if(y1 == y2 ) or isinstance(y1, complex) or isinstance(y2, complex):
print('The matrix A is not diagonalisable')

    else:
print('The matrix A is diagonalisable')

    print('The eigen values of A are : ')
print(y1);
print(y2);

    print('Matrix A : ')
a = np.matrix('2 0 ; 2 -2')

    y1,y2 = np.linalg.eigvals(a);
print(a);

    if(y1 == y2) or isinstance(y1, complex) or isinstance(y2, complex):
print('The matrix A is not diagonalisable')

    else:
print('The matrix A is diagonalisable')
print('The eigen values of A are : ')
print(y1);
print(y2);
print('Matrix A : ')

    a = np.matrix('2 0; 2 2')

    y1,y2 = np.linalg.eigvals(a);
print(a);
if(y1 == y2) or isinstance(y1, complex) or isinstance(y2, complex):
print('The matrix A is not diagonalisable')
else:
print('The matrix A is diagonalisable')
print('The eigen values of A are : ')
print(y1);
print(y2);
```

Chapter 5

APPLICATIONS

Some of applications of matrices in computer science :

- In computer based applications, matrices play a vital role in the projection of three dimensional image into a two dimensional screen, creating the realistic seeming motions.
- Stochastic matrices and Eigen vector solvers are used in the page rank algorithms which are used in the ranking of web pages in Google search.
- The matrix calculus is used in the generalization of analytical notions like exponentials and derivatives to their higher dimensions.
- One of the most important usages of matrices in computer side applications are encryption of message codes.
- Matrices and their inverse matrices are used for a programmer for coding or encrypting a message.
- A message is made as a sequence of numbers in a binary format for communication and it follows code theory for solving. Hence with the help of matrices, those equations are solved.
- With these encryptions only, internet functions are working and even banks could work with transmission of sensitive and private data's.

Chapter 6

REFERENCES

1. *https://en.wikipedia.org/wiki/Matrix_(mathematics)*
2. *https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors*
3. **Linear Algebra and Its Applications, Fourth Edition (2007)**
   by *Gilbert Strang*