Assignment 1

EE4013 - C & Data Structures

S.Abdur Rahman Nawaz - EE18BTECH11052

IIT HYDERABAD

August 25, 2021

Q: Consider a matrix **P** whose only eigenvectors are the multiples of $\begin{bmatrix} 1 \\ 4 \end{bmatrix}$.

Consider the following satements.

1. **P** does not have an inverse.
2. **P** has a repeated eigenvalue.
3. **P** cannot be diagonalized.

Which of the following options is correct?

- A. only (i) and (iii) are necessarily true.
- B. only (ii) is necessarily true.
- C. only (i) and (ii) are necessarily true.
- D. only (ii) and (iii) are necessarily true.

Solution

We can observe that all the eigenvectors are linearly dependent.
Following are the properties/theorms that we will use,

1. Eigenvectors from different eigenvalues are linearly independent.
2. An n x n matrix is diagonalizable iff there are n linearly independent eigenvectors.

Sinice all the eigenvectors are linearly dependent, using property (1) we conclude that **P** has repeated eigenvalues.
Using property (2) we can conclude that **P** cannot be diagonalized.
Hence (D) is the right answer.

We use Gaussian Elimination to find if the eigenvectors are linearly independent.

- First stack all the eigenvectors to form a matrix.
- Apply Gaussian elimination to convert the matrix into its row echelon form.
- While converting to the row echelon form, we can keep track of the rank.

Algorithm for Gaussian Elimination:

1. Let the input matrix be A[ROWS][COLS]. Initialize rank as the total number of columns.
2. For each row, do the following,
   I. if A[row][row] is non zero, make all the elements of the current column as zero except A[row][row] by applying the appropriate transformations.
   II. if A[row][row] is zero, two cases arise,
      i. If there is any row below it with non zero element in the same column, the swap the rows.
      ii. if all the elements below in the current columns are zero, then swap the current column with the last column and reduce the rank by 1.

Code

```
int compute_rank(float A[ROWS][COLS]) {

    int rank = COLS;
    display(A);

    for(int row=0;row<rank;row++){

        /*if diagonal element is non-zero,
        make all elements in the column
        0 except A[row][row].*/

        if(A[row][row]){

            for(int i=0;i<ROWS;i++){

                if(i!=row){
                    float temp = A[i][row]/A[row][row];

                    for(int j=0;j<rank;j++)
                        A[i][j] -= temp*A[row][j];
                }
            }
        }

        /*else swap the row with any other
        row below it that has a non zero
        zero element in the same column.*/

        /*if there is non non zero element
        then just swap with the last row.*/

        else{
            bool flag = true;

            for(int i=row+1;i<ROWS;i++){
                if(A[i][row]){
                    flag = false;
                    swap(A, row, i, rank);
                    break;
                }
            }

            if(flag){
                rank--;

                for(int i=0;i<ROWS;i++)
                    A[i][row] = A[i][rank];
            }

            row--;
        }
        display(A);
    }

    return rank;
}
```

Example: let the matrix A be :

$$\begin{bmatrix} 2 & 1 & -3 & 4 \\ 5 & 1 & -3 & -2 \\ 3 & 1 & -3 & 0 \end{bmatrix} \tag{1}$$

Step 1: Since A[0][0] is non zero, we make all the entries in the current column as 0 by performing the transformations, $R_2 = R_2 - 2.5R_1$ & $R_3 = R_3 - 1.5R_1$.
Matrix A after step 1:

$$\begin{bmatrix} 2 & 1 & -3 & 4 \\ 0 & -1.5 & 4.5 & -12 \\ 0 & -0.5 & 1.5 & -6 \end{bmatrix} \tag{2}$$

Step 2: Since A[1][1] is non-zero, we make all the entries in the current column as 0 by performing the transformations, $R_1 = R_1 + (2/3)R_2$ & $R_3 = R_3 - (1/3)R_2$.

Matrix A after step 2:

$$\begin{bmatrix} 2 & 0 & 0 & -4 \\ 0 & -1.5 & 4.5 & -12 \\ 0 & 0 & 0 & -2 \end{bmatrix} \tag{3}$$

Step 3: Since A[2][2] is zero and there is no rows remaining below, we remove the current column, by copying the last column here and reduce the rank by 1. And then process this row again.

Matrix A after step 3:

$$\begin{bmatrix} 2 & 0 & -4 & -4 \\ 0 & -1.5 & -12 & -12 \\ 0 & 0 & -2 & -2 \end{bmatrix} \tag{4}$$

Step 4: Since A[2][2] is non zero now, we make all the entries in the current column as 0 by performing the transformations, $R_1 = R_1 - 2R_3$ & $R_2 = R_2 - 6R_3$.

Matrix A after step 4:

$$\begin{bmatrix} 2 & 0 & 0 & -4 \\ 0 & -1.5 & 0 & -12 \\ 0 & 0 & -2 & -2 \end{bmatrix} \tag{5}$$

Hence the rank is 3 i.e, there are 3 linearly independent columns in the given matrix.

Proof for Property (1):

- Eigenvectors from different eigenvalues are linearly independent.

We will solve the proof for 2 eigenvectors, but the proof can be extended for any number of eigenvetors.

Let v_1 and v_2 be two eigenvectors for the matrix $\mathbf{P}$ and λ_1, λ_2 be the corresponding distinct eigenvalues.

Taking the linear combination,

$$\alpha_1 v_1 + \alpha_2 v_2 = 0 \tag{6}$$

We need to show that $\alpha_1 = \alpha_2 = 0$
Multiply both sides with $\mathbf{P}$ to equation (1)

$$\alpha_1 \mathbf{P} v_1 + \alpha_2 \mathbf{P} v_2 = 0 \tag{7}$$

$$\alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2 = 0 \tag{8}$$

Now, multiply equation (1) with λ_1,

$$\lambda_1 \alpha_1 v_1 + \lambda_1 \alpha_2 v_2 = 0 \tag{9}$$

Subtract (4) from (3),

$$\alpha_2(\lambda_2 - \lambda_1) v_2 = 0 \tag{10}$$

Since $\lambda_2 - \lambda_1 \neq 0$, and since $v_2 \neq 0$ (because v_2 is an eigenvector), then $\alpha_2 = 0$.

Using this on the original linear combination $\alpha_1 v_1 + \alpha_2 v_2 = 0$, we conclude that $\alpha_1 = 0$ as well (since $v_1 \neq 0$).

So v_1 and v_2 are linearly independent.

Proof for Property 2:

- An n x n matrix is diagonalizable iff there are n linearly independent eigenvectors.

Let P be a matrix and P^{-1} exists.

$$\leftrightarrow \exists Q, QP = PQ = I \tag{11}$$

$$\leftrightarrow \exists Q, P^T Q^T = Q^T P^T = I \tag{12}$$

$$\leftrightarrow (P^T)^{-1} exists \tag{13}$$

$$\leftrightarrow rank(P^T) = n \tag{14}$$

$\leftrightarrow$ Rows of P^T are linearly independent

$\leftrightarrow$ Columns of P are linearly independent

$(\exists P, \exists$ diagonal D, $AP = PD)$
$\leftrightarrow$ columns of P are eigenvectors of A.

If there are n linearly independent eigenvectors, make them columns of P. Then $AP = PD$ (D is diagonal) and P^{-1} exists, so $D = P^{-1}AP$. Therefore, A is diagonalizable.

If A is diagonalizable, there is P such that P^{-1} exists and $AP = PD$ (D is diagonal). Therefore columns of P are linearly independent and they are eigenvectors of A. Therefore, A is diagonalizable.

Appendix

- C implementation for Gaussian Elimination Algorithm can be found here.