NE250_HW03_mnegus-prob4

October 15, 2017

1 NE 250 – Homework 3

1.1 Problem 4

10/20/2017

```
In [1]: import numpy as np
        old_settings = np.seterr(divide='raise')
```

a.) We are given the matrix:

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

```
In [2]: A = np.array([[1,1,-1,3],[1,2,-4,-2],[2,1,1,5],[-1,0,-2,-4]])
```

The inverse, $\mathbf{A}^{-1}$ of a square matrix, $\mathbf{A}$, is equal to the adjugate of the matrix, $\mathbf{A}^{\dagger}$ divided by the determinant of $\mathbf{A}$.

$$\mathbf{A}^{-1} = \frac{\mathbf{A}^{\dagger}}{\det \mathbf{A}}$$

```
In [3]: def invert(matrix):
            adjugate_matrix = adjugate(matrix)
            det_matrix = determinant(matrix)
            try:
                inverse_matrix = adjugate_matrix/det_matrix
                return inverse_matrix
            except FloatingPointError:
                return 'The matrix has a determinant of zero; it is not invertible.
```

The adjugate of a square matrix, $\mathbf{A}^{\dagger}$, is the transpose of the cofactor matrix, $\mathbf{C_A}$.

$$\mathbf{A}^{\dagger} = \mathbf{C}_{\mathbf{A}}^{T}$$

```
In [4]: def adjugate(matrix):
            cofactor_matrix = cofactor(matrix)
            adjugate_matrix = transpose(cofactor_matrix)
            return adjugate_matrix

        def transpose(matrix):
            transpose_matrix = np.empty_like(matrix)
            for i in range(len(matrix)):
                for j in range(len(matrix[0])):
                    transpose_matrix[j,i] = matrix[i,j]
            return transpose_matrix
```

The cofactor of a square matrix, $\mathbf{C_A}$ is the signed matrix of minors, $\mathbf{M_A}$.

$$\mathbf{C}_{\mathbf{A},ij} = (-1)^{i+j}\,\mathbf{M_A}$$

```
In [5]: def cofactor(matrix):
            minors_matrix = minors(matrix)
            cofactor_matrix = np.copy(minors_matrix)
            for i in range(len(cofactor_matrix)):
                for j in range(len(cofactor_matrix[0])):
                        cofactor_matrix[i,j] *= (-1)**(i+j)
            return cofactor_matrix
```

The matrix of minors of a square matrix, $\mathbf{M_A}$ is quite literally a matrix of the minors of $\mathbf{A}$.

```
In [6]: def minors(matrix):
            minors_matrix = np.empty_like(matrix)
            for i in range(len(minors_matrix)):
                for j in range(len(minors_matrix[0])):
                    minors_matrix[i,j] = minor(matrix,i,j)
            return minors_matrix
```

The minor of matrix element $\mathbf{A}_{ij}$ is the determinant of submatrix formed with the rows and columns other than i and j.

```
In [7]: def minor(matrix,i,j):
            submatrix = np.copy(matrix)
            submatrix = np.delete(submatrix,i,axis=0)
            submatrix = np.delete(submatrix,j,axis=1)
            minor_ij = determinant(submatrix)
            return minor_ij
```

Finally, the determinant of a matrix is either, $ad - bc$ for a 2×2 matrix, or the sum of signed minors in a row, i of square matrix of order > 2 mutiplied by the values of the minor's respective j.

```
In [8]: def determinant(matrix):
            assert len(matrix) == len(matrix[0])
```

```
    if len(matrix) == 2:
        return matrix[0,0]*matrix[1,1]-matrix[0,1]*matrix[1,0]
    else:
        signed_minors = []
        for j in range(len(matrix[0])):
            if (j+2)%2 == 1:
                sign = -1
            else: sign = 1
            signed_minors.append(matrix[0,j]*sign*minor(matrix,0,j))
        return sum(signed_minors)
```

We can use this all together to find the inverse of **A**.

```
In [9]: print(invert(A))
```

```
The matrix has a determinant of zero; it is not invertible.
```

b.) We are given the matrix:

$$\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$$

and we know that the eigenvalue λ and eigenvector $\vec{v}$ obey the rule

$$\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \vec{v} = \lambda \vec{v}$$

Equivalently,

$$\left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \lambda \mathbb{1} \right) \vec{v} = 0$$

We want the non-trivial solution to this equation, when $\vec{v} \neq \vec{0}$. $\vec{v} = \vec{0}$ when $\left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \lambda \mathbb{1} \right)$ is invertible, so we will instead assert that $\left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \lambda \mathbb{1} \right)$ is not invertible. By definition, this means

$$\det \left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \lambda \mathbb{1} \right) = 0$$

or

$$\det \begin{bmatrix} 3 - \lambda & -1 \\ -1 & 3 - \lambda \end{bmatrix} = 0$$

We can solve this now for lambda:

$$\begin{aligned} (3 - \lambda)^2 - (-1)^2 &= 0 \\ 9 - 6\lambda + \lambda^2 - 1 &= 0 \\ 8 - 6\lambda + \lambda^2 &= 0 \end{aligned}$$

and using the quadratic formula we find

$$\lambda = \frac{-(-6) \pm \sqrt{(-6)^2 - 4(8)}}{2}$$
$$\lambda = \frac{6 \pm \sqrt{36 - 32}}{2}$$
$$\lambda = \frac{6 \pm 2}{2}$$
$$\lambda = 3 \pm 1$$

$$\boxed{\lambda = 2, 4}$$

We can use this eigenvalue to solve for $\vec{v}$.

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \vec{v} = 0 \text{ and } \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \vec{v} = 0$$

This gives the equations

$$\lambda = 2 : \begin{cases} v_1 - v_2 = 0 \\ v_2 - v_1 = 0 \end{cases} \qquad \lambda = 4 : \begin{cases} -v_1 - v_2 = 0 \end{cases}$$

For $\boxed{\lambda = 2,\ \vec{v} = \begin{bmatrix} v_0 \\ v_0 \end{bmatrix}}$, and for $\boxed{\lambda = 4,\ \vec{v} = \begin{bmatrix} v_0 \\ -v_0 \end{bmatrix}}$.