Handin8

April 30, 2022

By Christian Amstrup Petersen, Student number: 202104742

[17]:
```
import numpy as np
```

We're given the following matrix:

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

A The characteristic polynomial

We here want to find the characteristic polynomial p for A. We get this by using proposition 21.15, so:

$$p(A) = det(A - \lambda I_3)$$

Following definition 21.12, we notice that parts of the equation is nulified, due to $A_{01} = 0$ and $A_{02} = 2$, so we are left with:

$$p(A) = (2 - \lambda) det \begin{bmatrix} 1 - \lambda & 1 \\ 1 & 1 - \lambda \end{bmatrix}$$

$$= (2 - \lambda)(\lambda^2 - 2\lambda)$$

$$= -\lambda^3 + 4\lambda^2 - 4\lambda$$

B Eigenvalues

We're asked to show that 0 and 2 are Eigenvalues to A. To determine our eigenvalues normally, we would have to solve for the roots of $p(A)$ (See prop. 21.2), but we are now asked to verify instead. This simplifies the process, as we can just set λ equal to one of our given values, and checking if $p(A)$ then equals 0. For $\lambda = 0$:

$$-0^3 + 4 \cdot 0^2 - 4 \cdot 0 = 0$$

We can conclude that 0 is an eigenvalue for A. Now for $\lambda = 2$:

$$-2^3 + 4 \cdot 2^2 - 4 \cdot 2 = 0$$

$$-8 + 16 - 8 = 0$$

Therefore 2 is also an eigenvalue for A.

C More(!) Eigenvalues! (Maybe)

As stated above, we could find the Eigenvalues by solving $p(A)$ for its roots, so let us just do that:

[4]:
```
coeff = [-1, 4, -4, 0]
eigen = np.roots(coeff)
eigen
```

[4]:
```
array([2., 2., 0.])
```

Although we have a dublicate root of 2, this changes nothing for what we proved in assignment b, and we can thus conclude, that only $\lambda_0 = 0$ and $\lambda_1 = 2$ (and an added $\lambda_2 = 2$) are eigenvalues for A.

D Eigen-va... I mean -vectors

We now turn our eyes to definition 21.1. Using our λ_0, λ_1 and λ_2, we now want to determinte our eigenvectors v_0, v_1, and v_2. Firstly, for also later use, we denote the following:

We take the following formula from the definition:

$$(A - \lambda_x I_n)v_x = 0$$

From doing some 'række'-operations on any given $(A - \lambda_x I_3)$, we can contruct an echelon-matrix, from which we can create our v_x. So for $lambda_0$, we just get $(A - \lambda_0 I_3) = A$, and due to $A_{R1} = A_{R2}$, we subtract one from the other and therefore get:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} v_0 = 0$$

$$\Rightarrow v_0 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

And for λ_1 and λ_2, we get:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

Here we split up the calculations for v_1 and v_2. Firstly

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} v_1 = 0$$

Here we eliminate one of the remaining rows by adding the other to it, so we're left with:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix} v_1 = 0$$

$$v_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

For v_2, we use a piece of advice from Mr. Swann, as we have a row of free variables, and we can simply set the related entrance in v_2 to 1, and leave the rest as zeros, so we get:

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

$$\Rightarrow v_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

We can now denote this collection as

$$V = [v_0|v_1|v_2]$$

Taking a look at prop. 21.5, stating that eigenvectors are lineary independent, we fulfill the first requirement in def. 19.1, for V being a basis in $\mathbb{R}^3$. For the other requirement, we know that we're dealing a with a kvadratic matrix, and thus use prop. 19.4, so we can conclude that V is a basis for $\mathbb{R}^3$.

E The V, the upside one, and the inverted

We now want to chooce fitting variables for the equation:

$$A = V \Lambda V^{-1}$$

But this is just the equation seen in prop 21.6, and to all we need to do is introduce

$$\Lambda = diag(\lambda_0, \lambda_1, \lambda_2) = diag(0, 2, 2),$$

and use numpy to calculate the inverted V:

[16]:
```
V = np.array([[0.,0.,1.], [1.,1.,0.], [-1.,1.,0.]])
np.linalg.inv(V)
```

[16]:
```
array([[ 0. ,  0.5, -0.5],
       [ 0. ,  0.5,  0.5],
       [ 1. ,  0. ,  0. ]])
```

So

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 0.5 & -0.5 \\ 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} = A$$

F It's not a Markov-chain

For this part of the handin, we're asked to decide A^k for all k. And thanks to dear Mr. Swann and his wonderful notes, it's a fairly easy job. Taking a look just above formula 20.3, we get the formula:

$$A^k = TC^kT^{-1},$$

which resembles our little equation from assignment (e) in this handin, with the only requirement for this to hold, being that C is a diagonal matrix. Following this formula, switching out with our own values, we get that

$$A^k = V\Lambda^kV^{-1}$$