Vedaank Tiwari
9-24-17
EE16A

HW 4

1 Finding Null Spaces

1A The maximum possible numbers of linearly independent vectors you can pick from these vector columns, is limited by the number of rows. If there are fewer rows then columns, the number of linearly independent (max) vectors is the number of rows, otherwise it is just the number of columns.

1B
$$A = \begin{bmatrix} 1 & 1 & 0 & -2 & 3 \\ 0 & 0 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$
The minimum number of vectors spanning the column space is 2

$$Col(A) = \text{span}\left\{ \begin{matrix} 1 \\ 0 \\ 0 \end{matrix}, \begin{matrix} 1 \\ 0 \\ 0 \end{matrix}, \begin{matrix} 0 \\ 2 \\ 0 \end{matrix}, \begin{matrix} -2 \\ -2 \\ 0 \end{matrix}, \begin{matrix} 3 \\ 2 \\ 0 \end{matrix} \right\} = \text{span}\left\{ \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \begin{matrix} 0 \\ 2 \\ 0 \end{matrix} \right\}$$

1C
$$\underset{(3\times 5)}{A =} \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{matrix} \\ \left[\begin{matrix} 1 & 1 & 0 & -2 & 3 \\ 0 & 0 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \middle| \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}\right] \end{array}$$
$$x_1 + x_2 - 2x_4 + 3x_5 = 0$$
$$2x_3 - 2x_4 + 2x_5 = 0$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{matrix} -x_2 + 2x_4 + 3x_5 \\ x_2 \\ x_4 - x_5 \\ x_4 \\ x_5 \end{matrix} = x_2 \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 2 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} 3 \\ 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}$$

$$Nul(A) = \text{span}\left\{ \overset{\vec{v}_1}{\begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}} \overset{\vec{v}_2}{\begin{bmatrix} 2 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}} \overset{\vec{v}_3}{\begin{bmatrix} 3 \\ 0 \\ -1 \\ 0 \\ 1 \end{bmatrix}} \right\}$$

Linearly Independent

Basis $\Rightarrow \{\vec{v}_1\ \vec{v}_2\ \vec{v}_3\} = \boxed{3} \leftarrow$ nullity

1D

$$B = \begin{bmatrix} 2 & -4 & 4 & 8 \\ 1 & -2 & 3 & 6 \\ 2 & -4 & 5 & 10 \\ 3 & -6 & 7 & 14 \end{bmatrix} \begin{matrix} \frac{1}{2} \\ R1-2R2 \\ R1-R3 \\ 3R1-2R4 \end{matrix} \quad \begin{bmatrix} 1 & -2 & 2 & 4 \\ 0 & 0 & 5 & -4 \\ 0 & 0 & -1 & -2 \\ 0 & 0 & -2 & -4 \end{bmatrix} \begin{matrix} R1+2R3 \\ 5R3+R2 \\ -1R3 \\ -2R3+R4 \end{matrix}$$

$$B = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 \end{matrix} \\ \begin{bmatrix} 1 & -2 & 0 & 0 \\ 0 & 0 & 0 & -14 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix}$$

$x_1 = 2x_2$

$-14x_4 = 0 \quad x_4 = 0$

$x_3 = -2x_4$

$$x = \begin{bmatrix} 2x_2 \\ x_2 \\ 0 \\ 0 \end{bmatrix} = x_2 \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{Null}(B) = \text{Span}\left\{ \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

2. Traffic Flows

2A $t_1 = B \to A$ $\quad t_1 + t_3 = 0$

$t_2 = C \to B$ $\quad t_2 - t_1 = 0$

$t_3 = C \to A$ $\quad -t_2 - t_3 = 0$

If $t_1 = 10$, $t_3 = -10$, $t_2 = 10$

Yes because this system is conservative, it is possible to figure out the flows on the other roads.

2B

A $t_1 + t_3 - t_4 = 0$

B $t_2 - t_1 = 0$ $\quad t_1 = t_2$

C $t_5 - t_3 - t_2 = 0$

D $t_4 - t_5 = 0$ $\quad t_4 = t_5$

$$t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_1 \\ t_5 - t_2 \\ t_4 \\ t_4 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_1 \\ t_4 - t_1 \\ t_4 \\ t_4 \end{bmatrix}$$

$AD = t_4$ $BA = t_1$ - Berkeley

$CB = t_2$ $BA = t_1$ - Stanford

$$= t_1 \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} + t_4 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Subspace / Basis $= \left\{ \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}$

Yes, it is possible to determine all the traffic flow with the Berkeley student's suggestion, but the Stanford student suggestion tells us absolutely nothing. It is possible to generate every other vector in the system as they are in the span of t_1 and t_4

2C

$$\begin{bmatrix} 1 & 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix}$$

we can generate B by observing the traffic flow system presented in figure # 2

B (incidence matrix)

Each column of B represents the constraint that all the traffic in this system is conserved, No new cars enter & leave the system

2D A $t_1 + t_3 - t_4$ $\quad t_3 = t_4 - t_1$

B $t_2 - t_1$ $\quad t_2 = t_1$ $\quad t =$

C $t_5 - t_3 - t_2$

D $t_4 - t_5$ $\quad t_4 = t_5$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_1 \\ t_4 - t_1 \\ t_4 \\ t_4 \end{bmatrix}$$

Subspace basis = nullspace =

$$\text{Span}\left\{ \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\} = t_1 \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} + t_4 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

dimension 2 =

2E

$$\begin{bmatrix} 1 & 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} R1+R2 \qquad \begin{bmatrix} 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} R2+R3$$

$$\begin{bmatrix} 1 & 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} R3+R4 \qquad \left[\begin{array}{ccccc|c} 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$t_1 + t_3 - t_4 = 0$

$t_2 + t_3 - t_4 = 0$

$t_5 = t_4$

$t_2 = -t_3 + t_4$

$t_3 = t_4 - t_1$

$$t = \begin{bmatrix} t_1 \\ t_4 - t_3 \\ t_3 \\ t_4 \\ t_4 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_1 \\ t_4 - t_1 \\ t_4 \\ t_4 \end{bmatrix} = t_1 \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} + t_4 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \text{Span}\left\{ \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

Same as 2D ✓

dimension → minimum entries needed

2F In a more general road network graph G, with incidence matrix B_G. If B is k dimensional null space, this does NOT mean flows along any k roads is always sufficient to recover the exact flows although it is possible to recover the exact flows using k leaves minimum, just like the example of the stanford student if, those measurements are equivalent, of they originate from one source or have the same source or destination. Because if we know the similar information it does not give us enough information to determine the solution. Therefore, the stanford student case when he selects

2G G → network of n roads with incidence matrix B_G - k dim null

The exact traffic flows when one has two linearly independent measurements according to the basis of G and the measurements gives us more information. We need M to be at least the same as the dimension of the basis otherwise it is not possible to generate the entire flow of traffic. If two valid measurements are the same, recovery will fail, so they must be distinct (unrelated) and thereby linearly independent.

Yes it is possible to re-expressed this in terms of the null space of M and B_G as dim(Null) + dim(Col) = the dimension of the matrix G.

2H We can represent measurements on these flows using transition matrices. Using this matrix and finding a basis, we can see whether we have enough data to determine the overall flow of the entire system. It depends on the rank of our basis or the number of linearly independent columns needed in the basis. (which we can find from the null space of the transition matrix.)

2I No, if the incidence matrix has a k dimensional null space, this does not mean we can always pick a set of k roads such that measuring the flow along these roads is sufficient to recover the exact flows. For example recovery will fail if there are two valid flows with the same measurements, as this tells us no new information (when t_1 and t_2 $Mt_1 = Mt_2$). Therefore we need k distinct road measurements to recover the exact flows. If they are not distinct, that means the two columns are not linearly independent, which is why we cannot recover the exact flows.

3A $\begin{array}{cc} & \begin{matrix}1 & 2\end{matrix} \\ \begin{matrix}1\\2\end{matrix} & \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\end{array} = A$

3B There is exactly one 'one hop' paths from webpage 1 to webpage 2. There are no two hop paths from webpage 1 to webpage 2. There is exactly one three hop path from webpage 1 to webpage 2.

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

3C Eigen values is A

$A\vec{v} = \lambda\vec{v}$

λ is an eigenvalue so A

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

$\lambda^2 - 1 = 0$ $\quad \frac{1}{2} = 0.5$

$x^2 = 1$

1

Scores = 0.5, 0.5

3D $\begin{array}{cc} & \begin{matrix}1 & 2 & 3 & 4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}0 & 1 & 1 & 1\\0 & 0 & 1 & 1\\0 & 0 & 0 & 1\\1 & 0 & 1 & 0\end{bmatrix}\end{array} = A$

3E Zero 1 hop paths

One 2 hop paths. $1 \to 4 \to 2$

One 3 hop path $1 \to 4 \to 3 \to 2$

3F Eigenvals & A

$A\vec{v} = \lambda v$

use ipython.. eigenvalues $\rightarrow$ $[2.339, \ -0.5, \ -0.1165, \ -0.7227]$

1 2 3 4

λ= 12.5

$$
\text{36}\quad
\begin{array}{c|c}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} &
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array} = A
$$

3H There are no paths from webpage 1 to webpage 3

3I Eigenvalues of A.

$A\vec{v} = \lambda \vec{v}$

$[-1, 1, -1.325, 1.1780, -1.780]$

use python

$\boxed{1: 0.5 \quad 2\; 0.5, \quad 3: -1.325, \quad 4\; -1.325 \quad 5 \quad 1.1780}$

4 List the vectors that span the null space of A and dimension

$$
A = \begin{bmatrix}
1 & 3 & -4 & -3 & 1 \\
0 & 1 & -3 & 1 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$x_1 + 3x_2 - 4x_3 - 3x_4 - x_5$

$x_2 - 3x_3 + x_4 = 0 \quad x_2 = 3x_3 - x_4$

$$
\begin{aligned}
x_1 &= -3(3x_3 - x_4) + 4x_3 + 3x_4 - x_5 \\
&= -9x_3 + 3x_4 + 4x_3 + 3x_4 - x_5 \\
&\quad -5x_3 + 6x_4 - x_5
\end{aligned}
$$

$$
x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -5x_3 + 6x_4 - x_5 \\ 3x_3 - x_4 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
$$

4 (cont)

$$= x_3 \begin{bmatrix} -5 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 6 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\text{Null}(A) = \text{span} \left\{ \begin{array}{ccc} \vec{v}_1 & \vec{v}_2 & \vec{v}_3 \\ \begin{bmatrix} -5 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 6 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array} \right\}$$

Dimension of Null(A) = 3 = nullity

Basis = $\{\vec{v}_1, \vec{v}_2, \vec{v}_3\}$

5 Worked on this HW alone.

In [1]:
```
import numpy as np
matrix_3f = np.array([[0, 1, 1, 1],
                      [0, 0, 1, 1],
                      [0, 0, 0, 1],
                      [1, 0, 1, 0]])
np.linalg.eig(matrix_3f)
```

Out[1]:
```
(array([ 1.83928676+0.j        , -1.00000000+0.j        ,
        -0.41964338+0.60629073j, -0.41964338-0.60629073j]),
 array([[  6.75507948e-01+0.j        ,  -4.10399876e-17+0.j        ,
           6.09542210e-01+0.j        ,   6.09542210e-01-0.j        ],
        [  4.37593286e-01+0.j        ,  -2.18879934e-16+0.j        ,
           1.07340727e-01+0.52464311j,   1.07340727e-01-0.52464311j],
        [  2.83472437e-01+0.j        ,  -7.07106781e-01+0.j        ,
          -4.32666281e-01+0.1847799j ,  -4.32666281e-01-0.1847799j ],
        [  5.21387098e-01+0.j        ,   7.07106781e-01+0.j        ,
           6.95352019e-02-0.33986321j,   6.95352019e-02+0.33986321j]]))
```

In [3]:
```
import numpy as np
matrix_3i = np.array([[0, 1, 0, 0, 0],
                      [1, 0, 0, 0, 0],
                      [0, 0, 0, 0, 1],
                      [0, 0, 1, 0, 0],
                      [0, 0, 1, 1, 0]])
np.linalg.eig(matrix_3i)
```

Out[3]:
```
(array([ 1.00000000+0.j        , -1.00000000+0.j        ,
         1.32471796+0.j        , -0.66235898+0.56227951j,
        -0.66235898-0.56227951j]),
 array([[ 0.70710678+0.j        , -0.70710678+0.j        ,
          0.00000000+0.j        ,  0.00000000+0.j        ,  0.00000000-0.j
   ],
        [ 0.70710678+0.j        ,  0.70710678+0.j        ,
          0.00000000+0.j        ,  0.00000000+0.j        ,  0.00000000-0.j
   ],
        [ 0.00000000+0.j        ,  0.00000000+0.j        ,
         -0.54843176+0.j        ,  0.43441848-0.3687798j ,
          0.43441848+0.3687798j ],
        [ 0.00000000+0.j        ,  0.00000000+0.j        ,
         -0.41399889+0.j        , -0.65586562+0.j        , -0.65586562-0.j
   ],
        [ 0.00000000+0.j        ,  0.00000000+0.j        ,
         -0.72651740+0.j        , -0.08038366+0.48852922j,
         -0.08038366-0.48852922j]]))
```

In [ ]: