Linear Algebra: wk3 rref

Bill Chung

January 26, 2023

```
library(far)
library(MASS)
library(pracma)
```

Concepts

Space, subspace

- Domain, codomain (Range, C($mathbbA$))

Domain, codomain, Range

- You will see the following notation from time to time

$$T : R^n \to R^m$$

- the above notation is saying that `matrix T will be used to multiply vector with size of n and the resulting vector will have size m`
- And we will get into the details later.
- Vector resize within a space which consist of so many subspaces.
- When you put vectors into a matrix, you get two space, I call them `input` and `output space`. Input space can be divided into `row space` and `nullspace`, and `output space` can be divided into `column space` and `left null space`
- Think of domain as `row space` and `codomain` as `output space` and `range` as `column space`

Rank nullity theorem

If $\mathbb{A}$ has n columns, then $\text{Rank}(\mathbb{A}) + \dim \text{Nul}(\mathbb{A}) = n$

- see page 156 for the invertible matrix theorem (continued)

Invertible Linear Transformation

- A linear transformation $\mathbb{T}: R^n \rightarrow R^n$ is said to be `invertible` if there exists a funciton $\mathbb{S}: R^n \rightarrow R^n$ such that

$$\mathbb{S}(\mathbb{T}(\vec{x})) = \vec{x} \text{ for all } \vec{x} \text{ in } R^n$$
$$\mathbb{T}(\mathbb{S}(\vec{x})) = \vec{x} \text{ for all } \vec{x} \text{ in } R^n$$

where $\dim(\mathbb{A})$ = n by n, $\vec{x}, \vec{b} \in R^n$ - $\mathbb{A}$ is the standard matrix for $\mathbb{T}$

$$\mathbb{A}\vec{x} = \vec{b}$$

- $\mathbb{A}^{-1}$ is the standard matrix for $\mathbb{S}$

$$\mathbb{A}^{-1}\vec{b} = \vec{x}$$

```
r1 <- c(0,1,-4)
r2 <- c(2,-3,2)
r3 <- c(5,-8,9)
A <- rbind(r1,r2,r3)
print(A)
```

```
##    [,1] [,2] [,3]
## r1    0    1   -4
## r2    2   -3    2
## r3    5   -8    9
```

```
print(rref(A))
```

```
##    [,1] [,2] [,3]
## r1    1    0    0
## r2    0    1    0
## r3    0    0    1
```

- Above matrix is invertible matrix based on `rref()`
- Inverse transformation `undo` the transformation

```
x <- c(3,6,9)

#to use the same notation
T <- A
b<- T%*%x
print("Before the transformation")
```

```
## [1] "Before the transformation"
```

```
print(x)
```

```
## [1] 3 6 9
```

```
print("After the transformaiton")
```

```
## [1] "After the transformaiton"
```

```
print(T%*%x)
```

```
##    [,1]
## r1  -30
## r2    6
## r3   48
```

When $\mathbb{A}$ is not a square matrix

- With respect to $\mathbb{A}$, $\vec{b}$ is in your `range` and $\vec{x}$ is in `domain`.
- $\mathbb{A}$ transform vectors in domain to range.
- $\mathbb{A}^{-1}$ can transform values in range back to domain, when the $\mathbb{A}$ involved 1-1 transformation.

```
#chapter 1.1 example 3

r1 <- c(0,3,-6,6,4,-5)
r2 <- c(3,-7,8,-5,8,9)
r3 <- c(3,-9,12,-9,6,15)

A <- rbind(r1,r2,r3)
print(A)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6]
## r1    0    3   -6    6    4   -5
## r2    3   -7    8   -5    8    9
## r3    3   -9   12   -9    6   15
```

```
Rank(A)
```

```
## [1] 3
```

```
rref(A)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6]
## r1    1    0   -2    3    0  -24
## r2    0    1   -2    2    0   -7
## r3    0    0    0    0    1    4
```

```
B <- A[,c(1,2,5)]
print(B)
```

```
##    [,1] [,2] [,3]
## r1    0    3    4
## r2    3   -7    8
## r3    3   -9    6
```

```r
inv(B)
```

```
##      r1   r2        r3
## [1,] -5  9.0 -8.666667
## [2,] -1  2.0 -2.000000
## [3,]  1 -1.5  1.500000
```

Exercise.

- Get 5 matrices from class
- Given a matrix and a vector
    - Show different ways of spanning the given vector

```r
print(A)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6]
## r1    0    3   -6    6    4   -5
## r2    3   -7    8   -5    8    9
## r3    3   -9   12   -9    6   15
```

```r
b <- c(1,5,9)

Rank(A)
```

```
## [1] 3
```

```r
H <- cbind(A,b)
Rank(A)
```

```
## [1] 3
```

```r
rref(A)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6]
## r1    1    0   -2    3    0  -24
## r2    0    1   -2    2    0   -7
## r3    0    0    0    0    1    4
```

```r
B  <- A[,c(1,2,5)]
print(B)
```

```
##    [,1] [,2] [,3]
## r1    0    3    4
## r2    3   -7    8
## r3    3   -9    6
```

```r
x_B<- inv(B)%*%b
x_B
```

```
##       [,1]
## [1,]  -38
## [2,]   -9
## [3,]    7
```

```r
B%*%x_B
```

```
##    [,1]
## r1    1
## r2    5
## r3    9
```

```r
dim(A)
```

```
## [1] 3 6
```

```r
A%*%c(-38,-9,0,0,7,0)
```

```
##    [,1]
## r1    1
## r2    5
## r3    9
```

```r
r1 <- c(0,3,-6,6,4,-5)
r2 <- c(3,-7,8,-5,8,9)
r3 <- c(3,-9,12,-9,6,15)
r4 <- c(1,3,5,6,7,-1,10)

A <- rbind(r1,r2,r3,r4)
```

```
## Warning in rbind(r1, r2, r3, r4): number of columns of result is not a multiple
## of vector length (arg 1)
```

```r
print(A)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## r1    0    3   -6    6    4   -5    0
## r2    3   -7    8   -5    8    9    3
## r3    3   -9   12   -9    6   15    3
## r4    1    3    5    6    7   -1   10
```

```r
dim(A)
```

```
## [1] 4 7
```

```r
Rank(A)
```

```
## [1] 4
```

```r
rref(t(A))
```

```
##      r1 r2 r3 r4
## [1,]  1  0  0  0
## [2,]  0  1  0  0
## [3,]  0  0  1  0
## [4,]  0  0  0  1
## [5,]  0  0  0  0
## [6,]  0  0  0  0
## [7,]  0  0  0  0
```

```r
rref(A)
```

```
##    [,1] [,2] [,3]       [,4] [,5]       [,6]      [,7]
## r1    1    0    0  2.5384615    0 -21.538462 2.3846154
## r2    0    1    0  1.5384615    0  -4.538462 1.3846154
## r3    0    0    1 -0.2307692    0   1.230769 0.6923077
## r4    0    0    0  0.0000000    1   4.000000 0.0000000
```

```r
B <- A[,c(1,2,3,5)]
Rank(B)
```

```
## [1] 4
```

```r
dim(B)
```

```
## [1] 4 4
```

```r
inv(B)
```

```
##               r1         r2         r3            r4
## [1,] -4.84615385  8.3076923 -8.0256410  1.538462e-01
## [2,] -0.84615385  1.3076923 -1.3589744  1.538462e-01
## [3,]  0.07692308 -0.3461538  0.3205128  7.692308e-02
## [4,]  1.00000000 -1.5000000  1.5000000 -8.326673e-17
```

```r
cond(B)
```

```
## [1] 286.6405
```

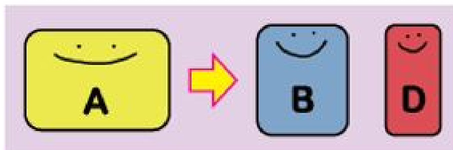
```r
b <- c(0.1, -1.5, 1,-1000)

x_B <- inv(B)%*%b

B%*%x_B
```

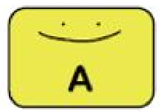
```
##         [,1]
## r1      0.1
## r2     -1.5
## r3      1.0
## r4 -1000.0
```

How to find basis of fundamental four subspaces



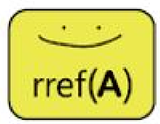

```
library(pracma)
```

```
print(A)
```



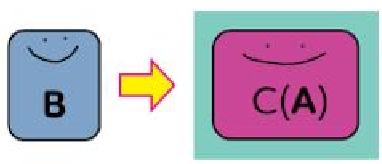
$$\begin{bmatrix} 1.1 & 1.1 & 3.3 & 1.1 \\ -1.2 & 3.3 & 5.4 & 7.8 \\ 1.3 & 2.2 & 5.7 & 3.1 \end{bmatrix}$$

```
print(rref(A))
```

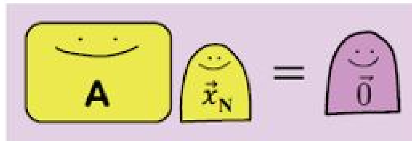


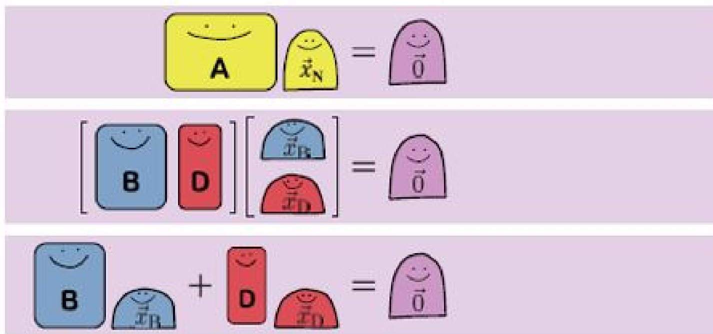
$$\begin{bmatrix} 1 & 0 & 1 & -1 \\ -0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
B <- A [ , c(1,2)]
print(B)
```

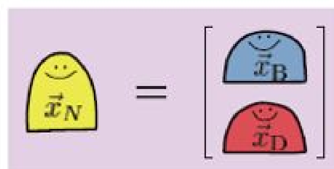


$$\begin{bmatrix} 1.1 & 1.1 \\ -1.2 & 3.3 \\ 1.3 & 2.2 \end{bmatrix}$$

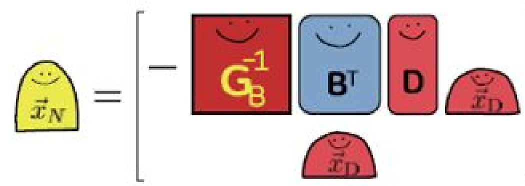




$$\vec{x}_N = \begin{bmatrix} \vec{x}_B \\ \vec{x}_D \end{bmatrix}$$



$$\vec{x}_N = \begin{bmatrix} \vec{x}_B \\ \vec{x}_D \end{bmatrix} = \begin{bmatrix} -G_B^{-1} B^T D \vec{x}_D \\ \vec{x}_D \end{bmatrix}$$



$$A\vec{x}_N = \vec{0}$$

$$B\vec{x}_B + D\vec{x}_D = 0$$

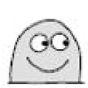
$$B\vec{x}_B = -D\vec{x}_D$$

$$B^T B\vec{x}_B = -B^T D\vec{x}_D$$

$$G_B\vec{x}_B = -B^T D\vec{x}_D$$

$$G_B^{-1} G_B\vec{x}_B = -G_B^{-1} B^T D\vec{x}_D$$

$$\vec{x}_B = -G_B^{-1} B^T D\vec{x}_D$$



```r
r1 <- matrix(c(1.1,1.1,3.3,1.1), nrow =1)
r2 <- matrix(c(-1.2,3.3,5.4,7.8), nrow =1)
r3 <- matrix(c(1.3,2.2,5.7,3.1), nrow =1)
A <- rbind(r1,r2,r3)
print(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  1.1  1.1  3.3  1.1
## [2,] -1.2  3.3  5.4  7.8
## [3,]  1.3  2.2  5.7  3.1
```

```r
print(rref(A))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    1   -1
## [2,]    0    1    2    2
## [3,]    0    0    0    0
```

```r
Rank(A)
```

```
## [1] 2
```

```r
B <- A[,c(1,2)]
print(B)
```

```
##      [,1] [,2]
## [1,]  1.1  1.1
## [2,] -1.2  3.3
## [3,]  1.3  2.2
```

```r
D <- A[,-c(1,2)]
print(D)
```

```
##      [,1] [,2]
## [1,]  3.3  1.1
## [2,]  5.4  7.8
## [3,]  5.7  3.1
```

```r
GB <- t(B)%*%B
print(GB)
```

```
##      [,1]  [,2]
## [1,] 4.34  0.11
## [2,] 0.11 16.94
```

```r
print(-inv(GB)%*%t(B)%*%D)
```

```
##      [,1] [,2]
## [1,]   -1    1
## [2,]   -2   -2
```

```r
I <- diag(2)
print(I)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

```r
print(rbind(-inv(GB)%*%(t(B)%*%D),I))
```

```
##      [,1] [,2]
## [1,]   -1    1
## [2,]   -2   -2
## [3,]    1    0
## [4,]    0    1
```

```r
N <- rbind(-inv(GB)%*%(t(B)%*%D),I)
print(round(A%*%N,2))
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    0    0
## [3,]    0    0
```

Projection Matrix (preview)

$$\mathbb{I} - \mathbb{P} = \mathbb{P}^{\perp}$$

```r
r1 <- matrix(c(1.1,1.1,3.3,1.1), nrow =1)
r2 <- matrix(c(-1.2,3.3,5.4,7.8), nrow =1)
r3 <- matrix(c(1.3,2.2,5.7,3.1), nrow =1)
A <- rbind(r1,r2,r3)
print(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  1.1  1.1  3.3  1.1
## [2,] -1.2  3.3  5.4  7.8
## [3,]  1.3  2.2  5.7  3.1
```

```r
Rank(A)
```

```
## [1] 2
```

```r
rref(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    1   -1
## [2,]    0    1    2    2
## [3,]    0    0    0    0
```

```r
b <- A[,c(1)] + A[,c(2)] + c(2.2,2.1,0)
b
```

```
## [1] 4.4 4.2 3.5
```

```r
B <- A[,c(1,2)]

Ab <- cbind(B,b)

Rank(Ab)
```

```
## [1] 3
```

$$
\begin{aligned}
\mathbb{B}\vec{x}_b + \mathbb{B}^{\perp}\vec{x}_b^{\perp} &= \vec{b} \\
\mathbb{B}^T\mathbb{B}\vec{x}_b &= \mathbb{B}^T\vec{b} \\
(\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T\mathbb{B}\vec{x}_b &= (\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T\vec{b} \\
\vec{x}_b &= (\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T\vec{b}
\end{aligned}
$$

- this is projection matrix

$$\mathbb{P} = \mathbb{B}(\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T$$

```r
print(B)
```

```
##      [,1] [,2]
## [1,]  1.1  1.1
## [2,] -1.2  3.3
## [3,]  1.3  2.2
```

```r
print(b)
```

```
## [1] 4.4 4.2 3.5
```

```r
P <-

x_B <- inv(t(B)%*%B)%*%t(B)%*%b

P<- B%*%inv(t(B)%*%B)%*%t(B)

#the best you can do within C(A)
estimate <- B%*%x_B

#this is residual not error
residual <- b-B%*%inv(t(B)%*%B)%*%t(B)%*%b

#b
estimate + residual
```

```
##      [,1]
## [1,]  4.4
## [2,]  4.2
## [3,]  3.5
```

```r
estimate
```

```
##          [,1]
## [1,] 2.766667
## [2,] 3.966667
## [3,] 4.666667
```

```r
P%*%b
```

```
##          [,1]
## [1,] 2.766667
## [2,] 3.966667
## [3,] 4.666667
```

```r
# breaking it down
P_ortho <- diag(3)-P
P%*%b + P_ortho%*%b
```

```
##      [,1]
## [1,]  4.4
## [2,]  4.2
## [3,]  3.5
```

```r
b
```

```
## [1] 4.4 4.2 3.5
```