Linear Algebra: wk6 Finding vectors multiplication that looks like projection

Bill Chung

December 28, 2021

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

```r
#6.2 Exampel 1
u1 <- c(3,1,1)
u2 <- c(-1,2,1)
u3 <- c(-0.5, -2, 7/2)

print(t(u1)%*%u2)
```

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

```r
print(t(u3)%*%u2)
```

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

```r
print(t(u1)%*%u3)
```

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

```r
#page 399, example 2
y <- c(6,1,-8)

A <- cbind(u1,u2,u3)
print(A)
```

```
##      u1 u2   u3
## [1,]  3 -1 -0.5
## [2,]  1  2 -2.0
## [3,]  1  1  3.5
```

- is $\vec{y}$ in $C(\mathbb{A})$?

```
Rank(A)
```

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

Since $\mathbb{A}$ is full rank, we can get $\curvearrowleft$ the following way

```
x <- inv(A)%*%y
print(A%*%x)
```

```
##      [,1]
## [1,]    6
## [2,]    1
## [3,]   -8
```

```
print("++++++++++++++++++++++++++++++++++")
```

```
## [1] "++++++++++++++++++++++++++++++++++"
```

```
############################################
# PROJECTION
############################################

# since each column vector of A are orthogonal we can use projection
# as well
x1 <- y%*%u1/(Norm(u1)^2)
x2 <- y%*%u2/(Norm(u2)^2)
x3 <- y%*%u3/(Norm(u3)^2)

# then using these coordinate you can get the following result as well
x <- c(x1, x2, x3)
print(A%*%x)
```

```
##      [,1]
## [1,]    6
## [2,]    1
## [3,]   -8
```

Orthogonal projection

- Very important concept and may take a few days of practice.
- see page 340.
- Suppose you have $\vec{u}$ and denote its subspace by L, and you have $\vec{y}$ that is not in the span of $\vec{u}$

projecting $\vec{y}$ onto L

$$\text{proj}_L \vec{y} = \hat{y} = \frac{\vec{y}\vec{u}}{\vec{u}\vec{u}}\vec{u}$$

```
# Example 3 (see slide 8)
y <- c(7,6)
u <- c(4,2)

hat_y <- y%*%u/(Norm(u)^2)*u
```

```
## Warning in y %*% u/(Norm(u)^2) * u: Recycling array of length 1 in array-vector arithmetic is depreca
##   Use c() or as.vector() instead.
```

```
residual <- y - hat_y

print(y)
```

```
## [1] 7 6
```

```
print(hat_y + residual)
```

```
## [1] 7 6
```

```
#####################################################
# what is the relationship between hat_y and residuel?
#####################################################

print(round(hat_y %*% residual,3))
```

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

```
# Example 3 (see slide 8)
# the same problem, but solved without using Norm()
y <- c(7,6)
u <- c(4,2)

#+++++++++++++++++++++++++++++++++++++++++++++
#what would be the physical meaning of this?
#see I wonder by Sam. =)
#+++++++++++++++++++++++++++++++++++++++++++++
print((y%*%u)/(u%*%u))
```

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

```
hat_y <- (y%*%u)/(u%*%u)*u
```

```
## Warning in (y %*% u)/(u %*% u) * u: Recycling array of length 1 in array-vector arithmetic is depreca
##   Use c() or as.vector() instead.
```

```r
print(hat_y)
```

```
## [1] 8 4
```

```r
residual <- y - hat_y

##############################################
# Will this always be zero?
# Why or why not?
##############################################
print(hat_y%*%residual)
```

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

```r
# example 6, see slide 13
# Orthonomal colums
#######################################################
#  Special property of matrix with orthonormal columns
#######################################################

u1 <- c(1/sqrt(2), 1/sqrt(2), 0)
u2 <- c(2/3, -2/3, 1/3)
U <- cbind(u1, u2)
x <- c(sqrt(2), 3)

##########################
print("Printing the norm of the vectors")
```

```
## [1] "Printing the norm of the vectors"
```

```r
print(Norm(u1))
```

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

```r
print(Norm(u2))
```

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

```r
print("++++++++++++++++++++++++++")
```

```
## [1] "++++++++++++++++++++++++++"
```

```r
print(round(t(U)%*%U,2))
```

```
##    u1 u2
## u1  1  0
## u2  0  1
```

```r
print("=======================")
```

```
## [1] "======================="
```

```r
##################################################
RHS <- U%*%x
print(U%*%x)
```

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

```r
print("++++++++++++++++++++++++++++++++++++++")
```

```
## [1] "++++++++++++++++++++++++++++++++++++++"
```

```r
print(Norm(U%*%x))
```

```
## [1] 3.316625
```

```r
print(Norm(x))
```

```
## [1] 3.316625
```

```r
print("+++++++++++++++++++++++++++++++++++++++")
```

```
## [1] "+++++++++++++++++++++++++++++++++++++++"
```

```r
##################################################
print(t(U)%*%RHS)
```

```
##        [,1]
## u1 1.414214
## u2 3.000000
```

```r
print(x)
```

```
## [1] 1.414214 3.000000
```

Discussion

- $\mathbb{U}$ transformed a vector in R^2 to R^3.
- The size of vector changed, but the norm of the vector did not change.
- Recall that $\mathbb{A}^{-1}\vec{b}$ only works when $\mathbb{A}$ is singular.
- But notice, when $\mathbb{U}$ has orthonormal columns, we can use $\mathbb{U}^T$ to transfrom the `RHS` be to row space!

```r
#page 351, example 3
u1 <- c(2,5,-1)
u2 <- c(-2,1,1)
y <- c(1,2,3)

#since the norm is not 1
#you still need to normalize it
print(Norm(u1))
```

```
## [1] 5.477226
```

```r
U <- cbind(u1,u2)

#using Gram matrix
print("using gram matrix")
```

```
## [1] "using gram matrix"
```

```r
x_hat<- inv(t(U)%*%U)%*%t(U)%*%y
print(U%*%x_hat)
```

```
##      [,1]
## [1,] -0.4
## [2,]  2.0
## [3,]  0.2
```

```r
# using projection
print("using projection")
```

```
## [1] "using projection"
```

```r
x1 <- y%*%u1/(Norm(u1)^2)
x2 <- y%*%u2/(Norm(u2)^2)
x <- rbind(x1,x2)
print(U%*%x)
```

```
##      [,1]
## [1,] -0.4
## [2,]  2.0
## [3,]  0.2
```

Orthogonal complement subspace

$$R(\mathbb{A})^{\perp} = N(\mathbb{A})$$

$$C(\mathbb{A})^{\perp} = N(\mathbb{A}^T)$$

- Orthogonal basis for subspace $\mathbb{W}$ is a basis for $\mathbb{W}$ that is also an orthogonal set
- $\mathbb{U} \in R^{mbyn}$ has orthogonal columns if and only if $\mathbb{U}^T\mathbb{U} = \mathbb{I}$

Angle between vectors

- This concept can be extended to beyond R^3

$$\vec{u}\vec{v} = ||\vec{u}||||\vec{v}||cos(\theta)$$

Difference between projecting onto orthogonal basis vs basis

- Explain what will be difference

More on matrix with orthonormal columns

- Orthonormal columns

$$\mathbb{U}\vec{x} = ||\vec{x}||$$
$$(\mathbb{U}\vec{x})(\mathbb{U}\vec{y}) = \vec{x}\vec{y}$$
$$(\mathbb{U}\vec{x})(\mathbb{U}\vec{y}) = 0 \text{ if and only if} \vec{x}\vec{y} = 0$$

Orthogonal decomposition

- Projecting $\vec{y}$ on the the orthogonal basis or orthogonal complement subsapce (i.e., this is linear regression)
- Given baiss, you can create orthonormal basis that spans the same space. `The Gram-schmidt process`

```
# see the example 1 from Chapter 6
# page 362

r1 <- c(4,0)
r2 <- c(0,2)
r3 <- c(1,1)

#your feature
A <- rbind(r1,r2,r3)

#your response
b <- c(2,0,11)
```

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

- Above set of equations require set of assumptions. can you identify them?

```r
# see the example 1 from Chapter 6
# page 362 continue

# this tells you the linear combination of
# column vectors of A that will get you y_hat

x <- inv(t(A)%*%A)%*%t(A)%*%b

#########################################
#what is the physical meaning of this x?
# x is the least sqaure solution
#########################################
print(x)
```

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

```r
print("++++++++++++++++++++++++++++++")
```

```
## [1] "++++++++++++++++++++++++++++++"
```

```r
print(b)
```

```
## [1]  2  0 11
```

```r
print("++++++++++++++++++++++++++++++")
```

```
## [1] "++++++++++++++++++++++++++++++"
```

```r
###########################################
# predict the value
############################################
y_hat <-A%*%x
print(y_hat)
```

```
##    [,1]
## r1    4
## r2    4
## r3    3
```

```r
print("------------------------------")
```

```
## [1] "------------------------------"
```

```r
residual <- b - y_hat
print(residual)
```

```
##    [,1]
## r1   -2
## r2   -4
## r3    8
```

```
########################################
# Why is this value zero? can anyone explain?
########################################
print(round(t(y_hat)%*%residual,4))
```

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

```
# see the example 2 from Chapter 6
# page 363 continue

v1 <- c(1,1,1,1,1,1)
v2 <- c(1,1,0,0,0,0)
v3 <- c(0,0,1,1,0,0)
v4 <- c(0,0,0,0,1,1)
b  <- c(-3,-1,0,2,5,1)

A <- cbind(v1,v2,v3,v4)

# Will the gram matrix invertible?
print(Rank(A))
```

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

```
print("===================")
```

```
## [1] "==================="
```

```
# is b in C(A)
Ab <- cbind(A,b)
print(Rank(Ab))
```

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

Discussion

1. Will the gram matrix be invertible?
2. Is $\vec{b}$ in C($\mathbb{A}$)?
3. How can we get $\hat{\vec{x}}$?

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

4. Note that this process is different from when you have ∞ number of solution

```
###########################################
# Create gram matrix, but this is singular
###########################################
G <- t(A)%*%A

# creating gram matrix requir multiplying RHS
# by AT
RHS <- t(A)%*%b

hyperplane <- cbind(G,RHS)
rref(hyperplane)
```

```
##    v1 v2 v3 v4
## v1  1  0  0  1  3
## v2  0  1  0 -1 -5
## v3  0  0  1 -1 -2
## v4  0  0  0  0  0
```

$$
\begin{aligned}
x_1 &= 3 - x_4 \\
x_2 &= -5 + x_4 \\
x_3 &= -2 + x_4 \\
x_4 &= \text{free}
\end{aligned}
$$

```
shift <- c(3,-5,-2,0)
basis <- c(-1,1,1,1)

#One solution
print(A%*%shift)
```

```
##      [,1]
## [1,]   -2
## [2,]   -2
## [3,]    1
## [4,]    1
## [5,]    3
## [6,]    3
```

```
#Another solution
print("===========================")
```

```
## [1] "==========================="
```

```
x_another <- shift + 0.01*basis
print(A%*%x_another)
```

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

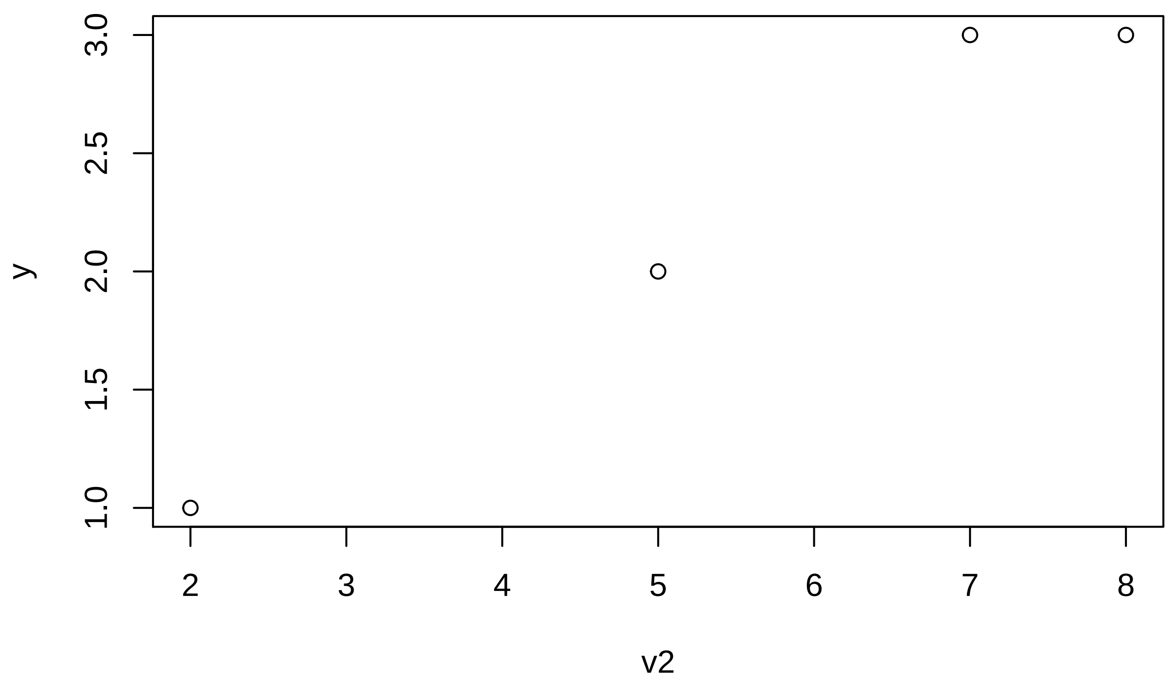
```
## [3,]    1
## [4,]    1
## [5,]    3
## [6,]    3
```

```
# page 370 example 1

v1 <- c(1,1,1,1)
v2 <- c(2,5,7,8)
y <- c(1,2,3,3)

A<- cbind(v1,v2)

plot(v2,y)
```




```
# fractions() is function in MASS
fractions(inv(t(A)%*%A)%*%t(A)%*%y)
```

```
##    [,1]
## v1  2/7
## v2 5/14
```

The general lienar model

- See page 317, is the following model linear?

$$y = \beta_0 + \beta_1 + \beta_2 x^2$$

- In the equation above β_0 is the constant term, what would be the effect of leaving this constant out of the model? Explain the effect using the following terms: `hyperplane` and `subspace`