Linear Algebra Review

BDSI 2019

Holly Hartman

6/20/2019

Scalars and Vectors

Scalars - Single number, represents magnitude

$$x = [a] = a; a \in \mathbb{R}$$

Vectors - Sequence of numbers, represents direction and magnitude

$$y = [a, b, \cdots] =< a, b, \cdots >; a, b, \cdots \in \mathbb{R}$$

Vectors can be notated $y, \boldsymbol{y}, \overrightarrow{y}$.

Matrices

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 4 \\ 6 & 2 & 1 \end{bmatrix}$$

$\boldsymbol{A}$ is a 2×3 matrix (2 rows, 3 columns)

$$B = \begin{bmatrix} 2 & 6 & 6 \\ 4 & 5 & 3 \\ 9 & 6 & 2 \end{bmatrix}$$

$\boldsymbol{B}$ is a 3×3 matrix (3 rows, 3 columns)

Entries identified by subscripts. $\boldsymbol{B}_{2,3} = 3$ and $\boldsymbol{A}_{1,2} = 2$,

Matrices in R

```r
A = matrix(c(3, 2, 4, 6, 2, 1), nrow = 2, byrow = T)
dim(A)
```

```
## [1] 2 3
```

```r
A[1, 2]
```

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

Matrix notation

While vectors are typically lower case, matrices are typically upper case. Matrices can be written many ways to indicate that they are a matrix.

$$X$$

$$\boldsymbol{X}$$

$$\mathbf{X}$$

$$\boldsymbol{\mathsf{X}}$$

$$\mathbb{X}$$

I will use $\boldsymbol{X}$ to distinguish between scalars and matrices.

Types of Square Matrices

Square - same number of rows as columns

$$\begin{bmatrix} 2 & 6 & 6 \\ 4 & 5 & 3 \\ 9 & 6 & 2 \end{bmatrix}$$

Diagonal - Only non-zero values on the diagonal

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

Types of Square Matrices

Symmetric - If matrix is $\boldsymbol{B}$, entry $\boldsymbol{B}_{ij} = \boldsymbol{B}_{ji}$

$$\begin{bmatrix} 2 & 4 & 9 \\ 4 & 5 & 6 \\ 9 & 6 & 3 \end{bmatrix}$$

Identity - Diagonal matrix where all entries are 1. Notation: $\boldsymbol{I}_n$

$$\boldsymbol{I}_3 \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Let $\boldsymbol{B}$ be $r \times c$, then $I_r\boldsymbol{B} = \boldsymbol{B}$ and $\boldsymbol{B}I_c = \boldsymbol{B}$.

Types of Square Matrices

Upper triangular matrix - $\boldsymbol{B}_{ij} = 0$ if $i > j$

$$\begin{bmatrix} 2 & 4 & 9 \\ 0 & 5 & 6 \\ 0 & 0 & 3 \end{bmatrix}$$

Lower triangular matrix - $\boldsymbol{B}_{ij} = 0$ if $i < j$

$$\begin{bmatrix} 2 & 0 & 0 \\ 4 & 5 & 0 \\ 9 & 6 & 3 \end{bmatrix}$$

Matrix algebra

Addition - matrices need same dimensions

$$\boldsymbol{A}+\boldsymbol{A}=\begin{bmatrix}3 & 2 & 4\\ 6 & 2 & 1\end{bmatrix}+\begin{bmatrix}3 & 2 & 4\\ 6 & 2 & 1\end{bmatrix}=\begin{bmatrix}6 & 4 & 8\\ 12 & 4 & 2\end{bmatrix}$$

Multiplication - If $\boldsymbol{A}$ is $a \times n$ then $\boldsymbol{B}$ must be $n \times b$. Final matrix will be $a \times b$.

$$\boldsymbol{AB}=\begin{bmatrix}3 & 2 & 4\\ 6 & 2 & 1\end{bmatrix}\begin{bmatrix}2 & 6 & 6\\ 4 & 5 & 3\\ 9 & 6 & 2\end{bmatrix}=$$

$$\begin{bmatrix}3\times 2+2\times 4+4\times 9 & 3\times 6+2\times 5+4\times 6 & 3\times 6+2\times 3+4\times 2\\ 6\times 2+2\times 4+1\times 9 & 6\times 6+2\times 5+1\times 6 & 6\times 6+3\times 3+1\times 2\end{bmatrix}$$

Matrix algebra in R

```r
A = matrix(c(3, 2, 4, 6, 2, 1), nrow = 2, byrow = T)
A + A
```

```
##      [,1] [,2] [,3]
## [1,]    6    4    8
## [2,]   12    4    2
```

```r
B = matrix(c(2, 6, 6, 4, 5, 13, 9, 6, 2), nrow = 3,
    byrow = T)
A %*% B
```

```
##      [,1] [,2] [,3]
## [1,]   50   52   52
## [2,]   29   52   64
```

Bad matrix algebra in R

```
A * B
```

```
## Error in A * B: non-conformable arrays
```

Bad matrix algebra in R

```
(A.bad = matrix(c(3, 2, 4, 6, 2, 1, 2, 3, 4), nrow = 3,
    byrow = T))
```

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

```
(B.bad = matrix(c(2, 6, 6, 4, 5, 13, 9, 6, 2), nrow = 3,
    byrow = T))
```

```
##      [,1] [,2] [,3]
## [1,]    2    6    6
## [2,]    4    5   13
## [3,]    9    6    2
```

Bad matrix algebra in R

```
A.bad * B.bad
```

```
##      [,1] [,2] [,3]
## [1,]    6   12   24
## [2,]   24   10   13
## [3,]   18   18    8
```

```
(A.bad * B.bad) == (A.bad %*% B.bad)
```

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

Transpose

Transpose - columns become rows and rows become columns. Notation varies. Can be either $\boldsymbol{A}^T$ or $\boldsymbol{A}'$. An $n \times p$ matrix becomes $p \times n$

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 4 \\ 6 & 2 & 1 \end{bmatrix}$$

$$\boldsymbol{A}' = \begin{bmatrix} 3 & 6 \\ 2 & 2 \\ 4 & 1 \end{bmatrix}$$

Transpose in R

```r
A
```

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

```r
t(A)
```

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

Rules of operations - Commutative Laws

$\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}$

$a\boldsymbol{B} = \boldsymbol{B}a$

Note: $\boldsymbol{AB} \neq \boldsymbol{BA}$ except in special cases.

Let $\boldsymbol{A}$ be a square matrix: $\boldsymbol{AI}_n = \boldsymbol{I}_n\boldsymbol{A} = \boldsymbol{A}$

Rules of operations - Distributive laws

$$\boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C}) = \boldsymbol{AB}+\boldsymbol{AC}$$

$$(\boldsymbol{B}+\boldsymbol{C})\boldsymbol{A} = \boldsymbol{BA}+\boldsymbol{CA}$$

$$a(\boldsymbol{B}+\boldsymbol{C}) = a\boldsymbol{B}+a\boldsymbol{C} = (\boldsymbol{B}+\boldsymbol{C})a$$

Rules of operations - Associative Laws

$(A + B) + C = A + (B + C)$

$(AB)C = A(BC)$

Rules of operations - Transpose Laws

$(\boldsymbol{A}+\boldsymbol{B})' = \boldsymbol{A}' + \boldsymbol{B}'$

$(\boldsymbol{A}\boldsymbol{B})' = \boldsymbol{B}'\boldsymbol{A}'$

$(a\boldsymbol{B})' = a\boldsymbol{B}' = \boldsymbol{B}'a$

If $\boldsymbol{A}$ is symmetric $\boldsymbol{A}^T = \boldsymbol{A}$.

Matrices in R

```r
t(A %*% B)
```

```
##      [,1] [,2]
## [1,]   50   29
## [2,]   52   52
## [3,]   52   64
```

```r
t(B) %*% t(A)
```

```
##      [,1] [,2]
## [1,]   50   29
## [2,]   52   52
## [3,]   52   64
```

Matrices in R

```
t(A %*% B) == t(B) %*% t(A)
```

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

Determinants

Determinant - scalar that can be computed from a matrix

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$det(\boldsymbol{A}) = |\boldsymbol{A}| = ad - bc$

```r
(A = matrix(c(3, 2, 4, 6), nrow = 2, byrow = T))
```

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

```r
det(A)
```

```
## [1] 10
```

Determinants

```
(A = matrix(c(3, 2, 4, 6, 4, 6, 9, 3, 5), nrow = 3,
    byrow = T))
```

```
##      [,1] [,2] [,3]
## [1,]    3    2    4
## [2,]    6    4    6
## [3,]    9    3    5
```

```
det(A)
```

```
## [1] -18
```

Matrix Inverse

The inverse of a matrix, $\boldsymbol{A}^{-1}$ is such that $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = I$.

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\boldsymbol{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{det(\boldsymbol{A})} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Inverse matrices in R

```r
(A = matrix(c(3, 2, 4, 6), nrow = 2, byrow = T))
```

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

```r
solve(A)
```

```
##      [,1] [,2]
## [1,]  0.6 -0.2
## [2,] -0.4  0.3
```

Inverse matrices in R

```r
B = matrix(c(2, 6, 6, 4, 5, 13, 9, 6, 2), nrow = 3,
    byrow = T)

det(B)
```

```
## [1] 392
```

```r
solve(B)
```

```
##             [,1]        [,2]         [,3]
## [1,] -0.17346939  0.06122449  0.122448980
## [2,]  0.27806122 -0.12755102 -0.005102041
## [3,] -0.05357143  0.10714286 -0.035714286
```

Most square matrices are invertible

```r
count1 <- 0  #To count number of errors
for (i in 1:1000) {
    # loop 1000 times
    n <- sample(100, 1)  #How big the matrix is
    A <- matrix(runif(n^2, min = -100, max = 100),
        nrow = n)  #Randomly generate a matrix
    temp <- try(solve(A), silent = T)  #Catch any error that occurs
    if (inherits(temp, "try-error")) {
        # If an error occured
        count1 <- count1 + 1  #add to count1
    }
}
count1
```

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

What does a non-invertible matrix look like?

- Not square

```
A <- matrix(runif(6, min = -100, max = 100), nrow = 3)
solve(A)
```

```
## Error in solve.default(A): 'a' (3 x 2) must be square
```

- Determinant $= 0$

```
A <- matrix(runif(6, min = -100, max = 100), nrow = 3)
A <- cbind(A[, 1] * 2, A)
round(det(A), digits = 10)
```

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

```
solve(A)
```

```
## Error in solve.default(A): Lapack routine dgesv: system is exactly singu
```

Matrix algebra with inverses

$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$

$(\boldsymbol{ABCD}...)^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}\boldsymbol{C}^{-1}\boldsymbol{D}^{-1}...$

$\left(\boldsymbol{A}^{-1}\right)^{T} = (\boldsymbol{A}^{T})^{-1}$

Matrices in Statistics

Linear regression formula is written:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots \beta_p x_{ip} + e_i$$

There are n y_i measurements, p βs, and n ϵs

Matrices in Statistics

This can be written much more concisely with matrix notation:

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

Least squares regression

Recall that the least squares estimate of $\boldsymbol{\beta}$ is:

$$\hat{\beta} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

(If this is unfamiliar, review the Linear Regression slides from Emily)

Let's see this in action in R

We will use a data set on fertility and socioeconomic indicators for provinces of Switzerland around 1888.

```r
library(datasets)
names(swiss)
```

```
## [1] "Fertility"        "Agriculture"      "Examination"
## [4] "Education"        "Catholic"         "Infant.Mortality"
```

Explore data set

```
`?`(swiss)
```

[,1] Fertility Ig, 'common standardized fertility measure'

[,2] Agriculture % of males involved in agriculture as occupation

[,3] Examination % draftees receiving highest mark on army examination

[,4] Education % education beyond primary school for draftees.

[,5] Catholic % 'catholic' (as opposed to 'protestant').

[,6] Infant.Mortality live births who live less than 1 year.

```r
summary(swiss)
```

```
##    Fertility      Agriculture     Examination      Education    
##  Min.   :35.00   Min.   : 1.20   Min.   : 3.00   Min.   : 1.00  
##  1st Qu.:64.70   1st Qu.:35.90   1st Qu.:12.00   1st Qu.: 6.00  
##  Median :70.40   Median :54.10   Median :16.00   Median : 8.00  
##  Mean   :70.14   Mean   :50.66   Mean   :16.49   Mean   :10.98  
##  3rd Qu.:78.45   3rd Qu.:67.65   3rd Qu.:22.00   3rd Qu.:12.00  
##  Max.   :92.50   Max.   :89.70   Max.   :37.00   Max.   :53.00  
##     Catholic       Infant.Mortality
##  Min.   :  2.150   Min.   :10.80   
##  1st Qu.:  5.195   1st Qu.:18.15   
##  Median : 15.140   Median :20.00   
##  Mean   : 41.144   Mean   :19.94   
##  3rd Qu.: 93.125   3rd Qu.:21.70   
##  Max.   :100.000   Max.   :26.60   
```

Linear Regression with Swiss data set

```
linRegModel <- lm(Fertility ~ ., data = swiss)
summary(linRegModel)
```

```
##
## Call:
## lm(formula = Fertility ~ ., data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2743  -5.2617   0.5032   4.1198  15.3213
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
## Agriculture      -0.17211    0.07030  -2.448  0.01873 *
## Examination      -0.25801    0.25388  -1.016  0.31546
## Education        -0.87094    0.18303  -4.758 2.43e-05 ***
## Catholic          0.10412    0.03526   2.953  0.00519 **
## Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Estimate regression parameters using matrix notation

```r
X = cbind(1, as.matrix(swiss[, 2:6]))
Y = as.matrix(swiss[, 1])

(beta = solve(t(X) %*% X) %*% t(X) %*% Y)
```

```
##                        [,1]
##                  66.9151817
## Agriculture      -0.1721140
## Examination      -0.2580082
## Education        -0.8709401
## Catholic          0.1041153
## Infant.Mortality  1.0770481
```

Compare Estimates

```r
cbind(beta, linRegModel$coef)
```

```
##                        [,1]       [,2]
##                  66.9151817 66.9151817
## Agriculture      -0.1721140 -0.1721140
## Examination      -0.2580082 -0.2580082
## Education        -0.8709401 -0.8709401
## Catholic          0.1041153  0.1041153
## Infant.Mortality  1.0770481  1.0770481
```

Linear dependence

Let $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_p$ be $n \times 1$ vectors. Then they are linearly dependent if there exits a set of scalars $a_1, \cdots a_p$ that are not all zero such that:

$$\sum_{i=1}^{p} a_i \boldsymbol{x}_i = \mathbf{0}$$

If no set of a_is exists, then the set of $\boldsymbol{x_i}$s are linearly independent

Rank

The rank of a set of vectors is the number of linearly independent vectors there are in the set. Rank ranges from 0 to p where p would be "full rank."

For a matrix, rank is from the matrix columns. Let $\boldsymbol{A}$ be a $n \times p$ matrix.

Full rank: $rank(\boldsymbol{A}) = \min(n, p)$ $rank(\boldsymbol{A}) \leq \min(n, p)$

Full rank square matrix is also called singular.

Why is rank important in statistics?

Many tests rely on the assumption that the covariates are independent.

This is the same as the assumption that the covariate matrix, $\boldsymbol{X}$, is full rank.

Rank in R

```r
dim(X)
```

```
## [1] 47  6
```

```r
qr(X)$rank
```

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

Rank in R

```
X <- cbind(X[, 1] * 2, X)
dim(X)
```

```
## [1] 47  7
```

```
qr(X)$rank
```

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

Rank and invertible matrices

Square matrices that are less than full rank are not invertible.

$$\begin{aligned} rank(\boldsymbol{A}) < n &\Leftrightarrow \boldsymbol{A} \text{ is singular} \\ &\Leftrightarrow det(\boldsymbol{A}) = 0 \\ &\Leftrightarrow \boldsymbol{A}^{-1} \text{ does not exist} \end{aligned}$$

Revisit non-invertible matrix example

```r
A <- matrix(runif(6, min = -100, max = 100), nrow = 3)
(A <- cbind(A[, 1] * 2, A))  #Make A not full rank
```

```
##            [,1]      [,2]      [,3]
## [1,] 199.26216  99.63108 -49.30335
## [2,]  93.62911  46.81456  64.96728
## [3,] -88.11996 -44.05998 -95.25373
```

```r
round(det(A), digits = 8)
```

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

```r
solve(A)
```

```
## Error in solve.default(A): Lapack routine dgesv: system is exactly singu
```