Day 5: Linear Algebra

Sarah Moore and Jason Seawright

Math Camp 2023

Linear Algebra

Up until now, we have mostly focused on uni-dimensional cases of algebra in that we only have a single variable, such as x. However, in most cases, we are concerned with more than just one variable x, and instead are interested in knowing about x, y, z, and so on, over various values.

Linear Algebra gives us a set of tools to be able to take on multiple values of multiple variables. These tools are more or less the same as those we have already learned for algebra, just applied to multidimensional data.

Vectors

- Ordered, finite set of numbers. Written two ways:

Option 1:

$$\mathbf{x} = (x_1, x_2, x_3)$$

Option 2:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Vectors

- Vectors are composed of *elements* and the length of the vector corresponds to the number of elements in it. A 3-vector has 3 elements, a 24-vector has 24 elements, and so on.
- We use subscripts based on the vectors name and the location of the element to refer to a specific element. For example, $\mathbf{x}_4$ refers to the fourth element of vector $\mathbf{x}$.
- The values of the elements in a vector are called **scalars**. This is not a new concept for us! We have essentially been working with scalars all this time. Most often, we are using real numbers $\mathbb{R}$ as scalars.
- Texts will refer to vectors in different ways, here we will differentiate vectors by **bolding** the vector's name.

Vector Operations

- Just like sets and variables, we can perform operations on vectors. However, there are some rules that apply for the different potential arithmetic operations that we'd like to perform.
- We can mostly expect the same algebraic properties hold in basic vector operations as in simple arithmetic operations (see Day 1). Consult this text for more information.

Vector Addition

- Two vectors can be added or subtracted if they equal in length. That is, 2-vectors can only be added to other 2-vectors, 8-vectors can only be differenced from other 8-vectors, etc.

$$\mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{bmatrix}$$

$$\mathbf{x} - \mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 - y_1 \\ x_2 - y_2 \\ x_3 - y_3 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 9 \end{bmatrix} - \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ 10 \end{bmatrix} = \begin{bmatrix} 14 \\ 13 \end{bmatrix}$$

$$\begin{bmatrix} 9 \\ 13 \end{bmatrix} + \begin{bmatrix} 7 \\ 9 \\ 3 \end{bmatrix} = \text{Not possible!}$$

Scalar-Vector Multiplication

$$(-3)\begin{bmatrix}7\\9\\3\end{bmatrix} = \begin{bmatrix}-21\\-27\\-9\end{bmatrix}$$

$$(0)\begin{bmatrix}4\\14\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

Inner Product

- You might also encounter this as the dot product.

$$\begin{bmatrix} 10 \\ 3 \end{bmatrix}^T \begin{bmatrix} 4 \\ 10 \end{bmatrix} = (10 \times 4) + (3 \times 10) = 27$$

Notice how the result does not just rely on the multiplication of the elements of the vector, but also adding each of the resulting elements together.

Example 1, Sentiment Analysis

Example (this text, p.22)

Let's say we have a vector of word occurrences in a document, and we weight each of the words by whether they are positive (1), negative (-1), or neutral (0) and then take the sum of the weighted values, the dot product is a measure of the overall sentiment of the document.

superb $=$ 41 obs.

fine $=$ 30 obs.

horrible $=$ 15 obs.

$$\begin{bmatrix} 41 \\ 30 \\ 15 \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$(41 \times 1) + (30 \times 0) + (15 \times -1) = 26$$

Example 2, Basic regression model

We previously reviewed linear functions, i.e. functions that map vectors to real numbers and have the properties of scalability (also referenced as homogeneity) and additivity. A special subset of linear functions are affine functions, these add a constant to a basic linear function. For example $y = mx + b$, where b is a constant.

- A common affine function is the regression model

$$\mathbf{y} = \mathbf{x}^T \beta + \mathbf{e}$$

Here the equation is written in matrix notation, but we can also expand these terms into vectors. Any idea how we would do that?

Matrices

Our data might also be arranged as an array of rows and colums.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

Matrices are described based on their $n \times d$ dimensions, that is the number of rows $\times$ columns in the matrix.

Indexing Data in Matrices

Like variables, matrices are given names. Matrix names are typically capital letters like A, B, X, I, Φ. Matrix elements are referenced by the matrix name in lower case and the element's row and column number. Generally, this is shown by a_{ij}.

Suppose we have the following matrix. What are the dimensions of matrix B below, and what value is b_{23}?

$$\begin{bmatrix} 3 & 2 & 8 \\ 9 & 81 & 21 \\ 13 & 7 & 6 \end{bmatrix}$$

Dimensions are 3 x 3; b_{23} is 21.

Characterizing Matrices

Matrices may be characterized by their specific shape. For example, a matrix with an equal number of rows and columns is a square matrix. Otherwise, a matrix is rectangular, but may be either tall or wide.

Square Matrix:

$$\begin{bmatrix} a & b & c \\ 1 & 0 & 0 \\ 4 & 8 & 5 \end{bmatrix}$$

Rectangular Matrix:

$$\begin{bmatrix} a & b & c & d \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Types of Matrices

We won't get into the details of all sorts of different matrices but it might be helpful to know that there are "special" matrices:

- *Vector Matrices*: Only one row or column
- *Submatrix*: Subset of a matrix
- *Triangular Matrix*: Top or bottom part of matrix is zeros
- *Diagonal Matrix*: Only the diagonal is non-zero
- *Zero Matrix*: Everything is zeros
- *Identity Matrix* (I): All zeros except on diagonal AND diagonal is only ones

Identifying Matrices

Define the kinds of matrices below:

$$A = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix} \; B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \; C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix} \; D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Identifying Matrices

Define the kinds of matrices below:

$$A = \begin{bmatrix} 1 & 3 & 5 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix} D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A Vector, B Zero, C Diagonal, D Identity

Matrix Transposition

- Matrix transposition is an operation that ‘flips’ our matrix over the diagonal such that the rows and columns are switched. A transpose of matrix A is notated as A^T.

$$A = \begin{bmatrix} 1 & 4 & 5 \\ 2 & 7 & 6 \end{bmatrix}; A^T \begin{bmatrix} 1 & 2 \\ 4 & 7 \\ 5 & 6 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 7 \\ 9 & 10 \\ 8 & 2 \end{bmatrix}; B^T \begin{bmatrix} 0 & 9 & 8 \\ 7 & 10 & 2 \end{bmatrix}$$

Adding Matrices

- Like with vectors, we can only perform addition and subtraction on matrices with similar dimensions. Once you identify that matrices have the same number of rows and columns, you can perform addition or subtraction across corresponding elements.

$$\begin{bmatrix} 1 & 13 & 15 \\ 2 & 4 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 3 & 17 & 21 \\ 3 & 7 & 11 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 13 & 15 \\ 2 & 4 & 6 \end{bmatrix} - \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix} = \begin{bmatrix} -1 & 9 & 9 \\ 1 & 1 & 1 \end{bmatrix}$$

Scalar-Matrix Multiplication

We can extend scalar multiplication to the matrix case as well.

$$(3) \times \begin{bmatrix} -1 & 9 & 9 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -3 & 27 & 27 \\ 3 & 3 & 3 \end{bmatrix}$$

Matrix Multiplication

- The same principles of the inner product in vector multiplication can be extended to the matrix case as well.
- To multiply matrices, the columns in matrix A must match the number of rows in matrix B. So, if matrix A is 2 x 3 then matrix B must be 3 x p. Therefore, the commutative property, that is the order of multiplication, *does not hold* for matrix algebra.

Below is a simple example:

$$\begin{bmatrix} -1 & 9 & 9 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} (-1 * -1) + (9 * 1) + (9 * 0) \\ (1 * -1) + (1 * 1) + (1 * 0) \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

Matrix Example

```
mat_a = matrix(c(3, 0,
                 7, 2,
                 2, 0), nrow = 3, ncol = 2, byrow = T)

mat_b = matrix(c(0, 2, 5,
                 9, 3, 0), nrow = 2, ncol = 3, byrow = T)

mat_a %*% mat_b
```

```
##      [,1] [,2] [,3]
## [1,]    0    6   15
## [2,]   18   20   35
## [3,]    0    4   10
```

Let's work that out

$$\begin{bmatrix} (3\times 0)+(0\times 0) & (3\times 2)+(0\times 3) & (3\times 5)+(0\times 0) \\ (7\times 0)+(2\times 9) & (7\times 2)+(2\times 3) & (7\times 5)+(2\times 0) \\ (2\times 0)+(0\times 9) & (2\times 2)+(0\times 3) & (2\times 5)+(0\times 0) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 6 & 15 \\ 18 & 20 & 35 \\ 0 & 4 & 10 \end{bmatrix}$$

The matrix product is a linear combination.

This animation shows what we mean by this.

Matrix Inverse

- Earlier we mentioned the identity matrix. The identity matrix is indicated by I_n for all n diagonal elements that are 1.

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

An inverse matrix A is invertible if there is an $n \times n$ matrix A^{-1} such that

$$AA^{-1} = A^{-1}A = I_n$$

Matrix Inverse

Solving for the inverse of matrix is therefore algebraic operation in solving for A^{-1} since both A and I_n are known. Note that inversion can only be completed on *square* matrices. Example from here.

$$\begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \times \begin{bmatrix} 3a+b & 3c+d \\ 2a+4b & 2c+4d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Solve for a and b: $3a + b = 1$ and $2a + 4b = 0$, then plug in these values. Therefore:

$$A^{-1} = \begin{bmatrix} \frac{2}{5} & -\frac{1}{10} \\ -\frac{1}{5} & \frac{3}{10} \end{bmatrix}$$

Finding the Inverse

But this is relatively easy in R:

```
mat_a = matrix(c(1, -1,
                 -1, -1), nrow = 2, ncol = 2, byrow = T)


inverse_a = solve(mat_a)

inverse_a
```

```
##       [,1] [,2]
## [1,]  0.5 -0.5
## [2,] -0.5 -0.5
```

Tying this back

```r
mat_a %*% inverse_a # What matrix is this?
```

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

What is this useful for?

- Among other things the matrix inverse can be used to solve a system of linear equations. An example of this is a OLS regression.

Working out linear regression with matrices

Let's first simulate some data:

```
econ_growth = rnorm(50)
num_corrupt_officials = rpois(50, lambda = 2)
```

Working out linear regression with matrices

Then run the regression:

```
reg = summary(lm(num_corrupt_officials ~ econ_growth))
```

Let's express the OLS estimate in matrix terms

$$b = (X^T X)^{-1} X^T Y$$

Let's transform our matrices and perform the operations we need to.

```
X = as.matrix(econ_growth)
X = cbind(rep(1, 50), X)
Xt = t(X)

Y = as.matrix(num_corrupt_officials)

XtXinv <- solve(Xt %*% X)
XtXinv
```

```
##                   [,1]        [,2]
## [1,] 0.020252803 0.001874949
## [2,] 0.001874949 0.013905815
```

Outcome of the regression via matrix operations

```
XtXinv %*% Xt %*% Y
```

```
##           [,1]
## [1,] 2.2569713
## [2,] 0.5708677
```

Check against the summary output

```
reg
```

```
##
## Call:
## lm(formula = num_corrupt_officials ~ econ_growth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4767 -1.0511 -0.0113  0.8781  2.9286
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2570     0.1904  11.852 7.31e-16 ***
## econ_growth   0.5709     0.1578   3.618 0.000712 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Residual standard error: 1.338 on 48 degrees of freedom
```