Linear Algebra: Matrix inversion and transpose

Glenn Bruns
CSUMB

Much of the material in these slides comes from Géron's notes:
https://github.com/ageron/handson-ml/blob/master/math_linear_algebra.ipynb

Learning outcomes

After this lecture you should be able to:

1. Name some algebraic properties of matrix multiplication
2. Define and perform:
   - matrix transpose
   - matrix inverse
3. Explain how a matrix can be used as a linear transformation
4. Define the problem of linear regression with matrices

Matrix multiplication review

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 4 & 6 & 8 \\ 2 & 7 & 5 \end{pmatrix} = \begin{pmatrix} 15 & 33 & 34 \\ 13 & 19 & 30 \end{pmatrix}$$

Two things to remember:

1. The size of the matrices must be compatible

   m x **n** **n** x p → m x p

2. Each element in the result comes from a dot product

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 4 & 6 & 8 \\ 2 & 7 & 5 \end{pmatrix} = \begin{pmatrix} 15 & \boxed{33} & 34 \\ 13 & 19 & 30 \end{pmatrix}$$

Properties of matrix multiplication

1. Commutative
   - how do you write the law?
   - do you think it holds?
2. Associative
   - how would you write the rule?
   - do you think it holds?
3. Distributes over addition
   - how would you write the rule?
   - do you think it holds?

Transpose

The transpose of a matrix A is written A^T

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

Think of "spinning" the matrix:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

The definition of transpose: $(A^T)_{i,j} = A_{j,i}$

In NumPy:

```
A = np.array([
        [10, 20, 30],
        [40, 50, 60]
    ])
# transpose A
A.T
```

If the size of A is $m \times n$, then the size of A^T is n x m.

Properties of transpose

$(A^T)^T = ?$

$(A + B)^T = ?$

$(AB)^T = B^T A^T$

Inverses and equation solving

Suppose we have a square matrix A. Can we find the inverse matrix A^{-1} such that

$$A\,A^{-1} = I \quad \text{and} \quad A^{-1}A = I$$

How could we use this?

These equations:

$$\begin{aligned} 2x_1 - x_2 &= 0 \\ -x_1 + 2x_2 &= 3 \end{aligned}$$

can be written as equation $Ax = b$

$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

If we could find the inverse matrix A^{-1}, then:

$$\begin{gathered} Ax = b \\ A^{-1}x = A^{-1}b \\ Ix = A^{-1}b \\ x = A^{-1}b \end{gathered}$$

In other words, we could solve $Ax = b$ just like we solve $3x = 12$

Matrix inverse

We know some equations don't have solutions

(just like $0x = 12$ has no solution)

So some matrices have no inverse.

A square matrix A is invertible if there exists a square matrix B such that

$$AB = I$$

If A is invertible, then its inverse is unique, and written A^{-1}

```
import numpy.linalg as LA
F_shear = np.array([
        [1, 1.5],
        [0, 1]
    ])
F_inv_shear = LA.inv(F_shear)
```

```
In [5]: F_inv_shear
Out[103]: array([[ 1. , -1.5],
                 [ 0. , 1. ]])
```

No all matrices have inverses

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Is there a square matrix B such that

$$AB = I \text{ ?}$$

A square matrix that is not invertible is called singular (or degenerate).

```
A = np.array([
        [1, 0],
        [0, 0]
    ])
LA.inv(A)
```

```
In [5]: LA.inv(A)
Out[103]: ...
LinAlgError: Singular matrix
```

Useful properties of inverse

- only square matrices have inverses
- $A\,A^{-1} = A^{-1}A = I$ (assuming A has inverse)
- $(AB)^{-1} = B^{-1}A^{-1}$ (socks and shoes)
- $(A^{-1})^{-1} = A$
- $(A^T)^{-1} = (A^{-1})^T$

Questions:

1. How would you describe the last fact: $(S^{-1})^T = S^{-1}$?
2. Can you prove that S^{-1} is symmetric?

Matrices in ML

We've discussed in linear regression, a prediction can be written as the dot product:

$$\hat{y} = \theta^T \cdot \mathbf{x}$$

Remember, we set x_0 to 1 to make this work out.

Training linear regression

Suppose we have this training data:

$$\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)}\right)$$

where each $x^{(i)}$ is a feature vector of length $n + 1$

We want to choose coefficient vector θ to minimize this:

$$\text{MSE}\ (\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(\theta^T \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

Writing RSS using matrices

Let $\mathbf{X}$ be the $m \times (n+1)$ matrix of training data, where each row is a feature vector.

Let $\mathbf{y}$ be the vector of training labels.

Now we can write RSS this way:

$$\mathrm{MSE}(\theta) = \frac{1}{m}(y - \mathbf{X}\theta)^T (y - \mathbf{X}\theta)$$

1 x n n x 1 → 1 x 1

It may help to see it this way:

$$\frac{1}{m}\begin{pmatrix} e^{(1)} & e^{(2)} & e^{(3)} \end{pmatrix}\begin{pmatrix} e^{(1)} \\ e^{(2)} \\ e^{(3)} \end{pmatrix} = \frac{1}{m}\left(\left(e^{(1)}\right)^2 + \left(e^{(2)}\right)^2 + \left(e^{(3)}\right)^2\right)$$

Writing RSS using matrices

Let's break down the last slide:

$$\text{MSE}(\theta) = \frac{1}{m}(y - \mathbf{X}\theta)^T (y - \mathbf{X}\theta)$$

What are the sizes of:

$\mathbf{X}$?

$\mathbf{X}\theta$?

$\mathbf{y}$?

$\mathbf{y} - \mathbf{X}\theta$?

$(\mathbf{y} - \mathbf{X}\theta)^T$?

Summary

1. Matrix multiplication is not commutative, but is associative, and distributes over addition
2. Transpose a matrix by “spinning it” on a diagonal axis
3. If matrix A has an inverse A^{-1}, then $A\,A^{-1} = I$
4. Linear algebra provides a compact way to describe the problem of linear regression

Linear transformations (optional)

Suppose has function f has this form:

$$f(u) = Au$$

Every such f is guaranteed to have these properties:

1. $f(u+v) = f(u) + f(v)$
2. $f(cu) = c\,f(u)$

Any function satisfying 1 and 2 is a linear transformation.

In other words:

- Functions of the form at the top can only express linear transformations.
- But also, functions of the form at the top can express every linear transformation.

(see Theorems MBLT and MLTCV in Beezer)