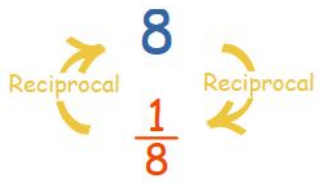
Matrix Inversion Capstone

Matrix Inversion

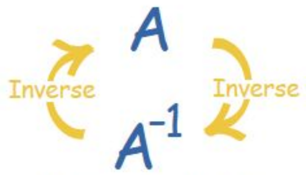
What is the Inverse of a Matrix?

Just like a **number** has a reciprocal ...



Reciprocal of a Number (note: $\frac{1}{8}$ can also be written 8^{-1})

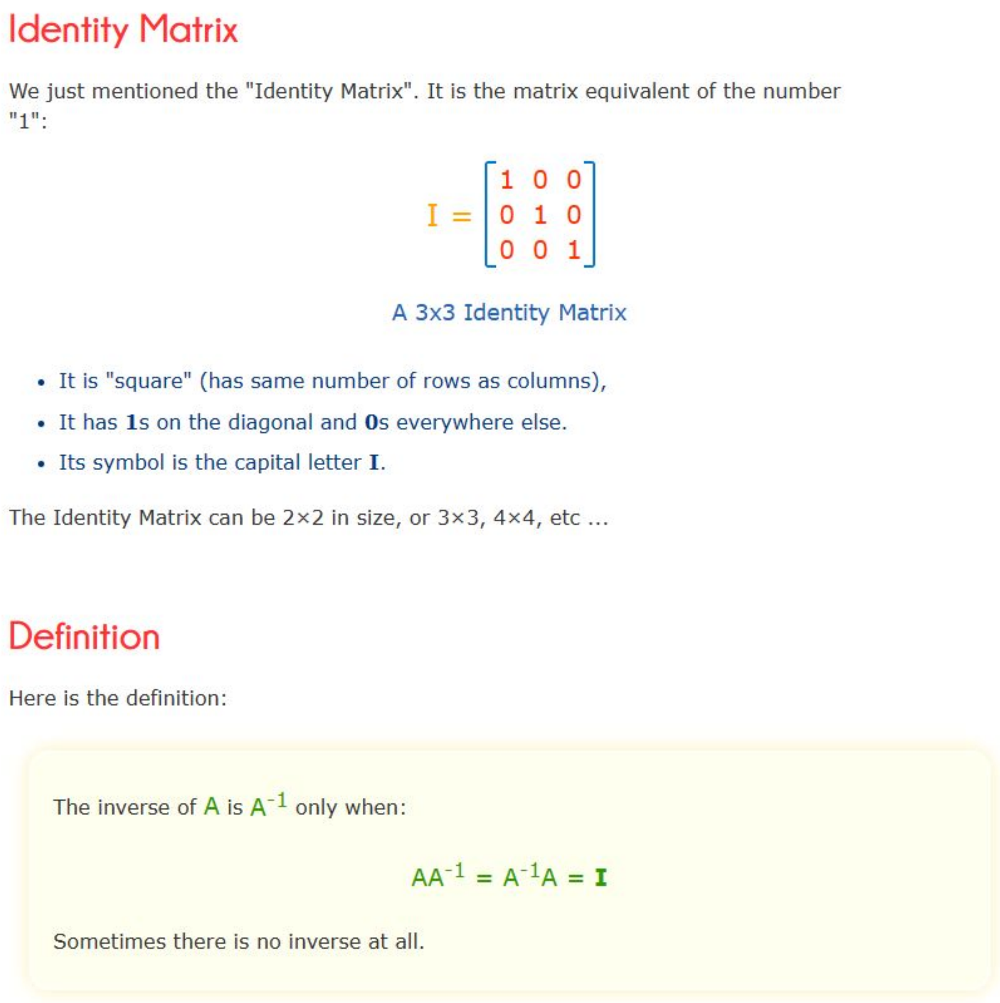
... a **matrix** has an **inverse :**



Inverse of a Matrix

We write A^{-1} instead of $\frac{1}{A}$ because we don't divide by a matrix!

Source: Inverse of a Matrix (mathsisfun.com)



Identity Matrix

We just mentioned the "Identity Matrix". It is the matrix equivalent of the number "1":

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

A 3x3 Identity Matrix

- It is "square" (has same number of rows as columns),
- It has **1**s on the diagonal and **0**s everywhere else.
- Its symbol is the capital letter **I**.

The Identity Matrix can be 2×2 in size, or 3×3, 4×4, etc ...

Definition

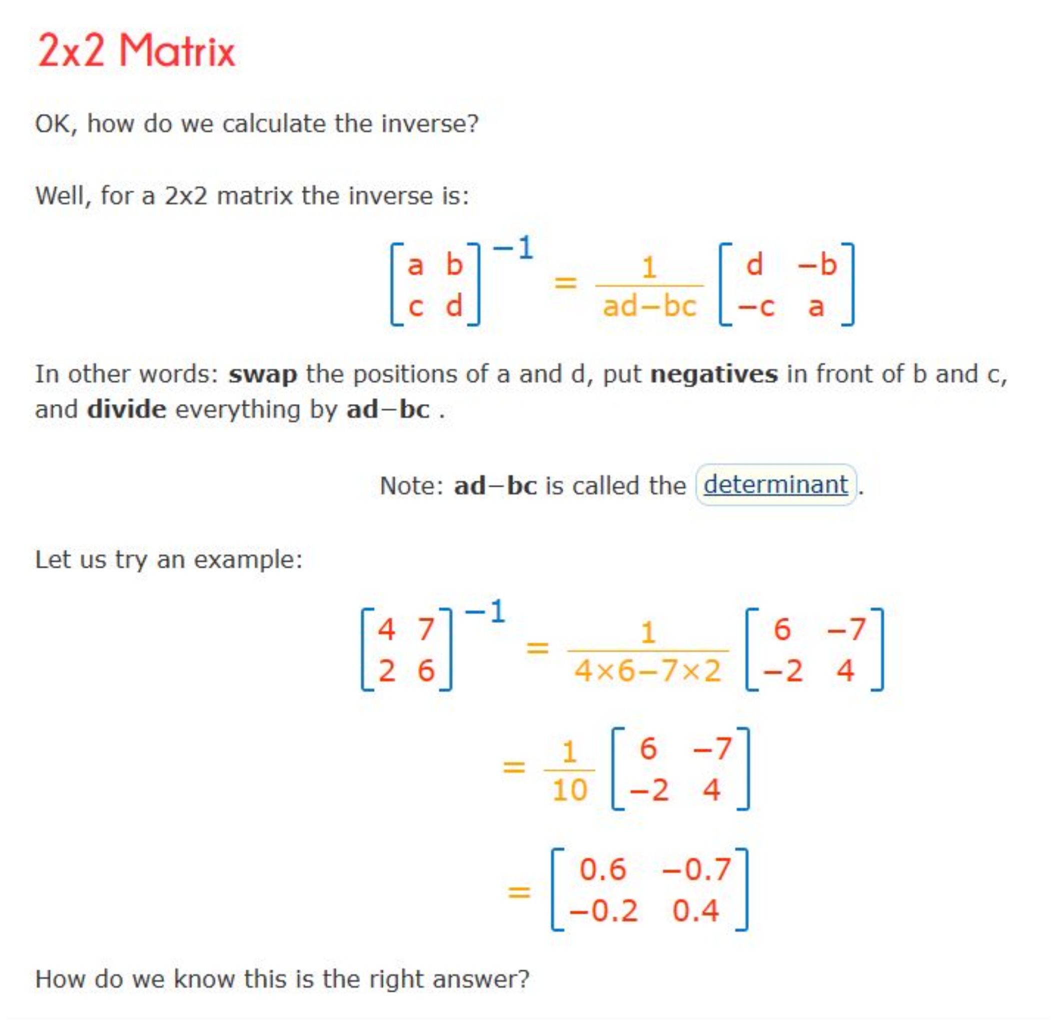
Here is the definition:

The inverse of A is A^{-1} only when:

$$AA^{-1} = A^{-1}A = \mathbf{I}$$

Sometimes there is no inverse at all.

Source: [Inverse of a Matrix (mathsisfun.com)]



2x2 Matrix

OK, how do we calculate the inverse?

Well, for a 2x2 matrix the inverse is:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

In other words: **swap** the positions of a and d, put **negatives** in front of b and c, and **divide** everything by **ad−bc** .

Note: **ad−bc** is called the determinant.

Let us try an example:

$$\begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}^{-1} = \frac{1}{4\times6-7\times2}\begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix}$$

$$= \frac{1}{10}\begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix}$$

$$= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.4 \end{bmatrix}$$

How do we know this is the right answer?

Source: Inverse of a Matrix (mathsisfun.com)

Sprint 1

Investigate linear regression, and decision tree classification between a matrix and its inverse for each matrix element

I will consider a 3x3 matrix with elements between 0 and 6 - not to simple but not overly complicated. This is a preliminary analysis - next sprint planning can be done to possibly test other matrices or go in a different direction entirely

Consider a linear regression with dependent variables being all elements of original matrix

Also consider all possible combinations of each element multiplied by another element in the matrix a00*a01, a00*a02, etc

By lucky mistake i also added the square of each element to the linear regression and found this as a way to increase value of a model

With further trial and error I found some better tests as well

Possible Impact of Sprint 1 results

Would be very interesting to see the r squared values for the various linear regressions - I will be considering the average r squared value of all inverse matrix entries

R-squared is a **statistical measure of how well the linear regression model fits the data**. It indicates **the percentage of the variance in the dependent variable that is explained by the independent variables**.

And see if there is an improvement when considering elements of the matrix multiplied by each other

Possibly find a linear regression to determine the inverse of a matrix

Applications of Matrix Inversion

There are a tremendous number of applications for matrix inversion. Back Propagation in neural nets is a major one. Other applications include:

Markov Chains: Matrix inversion is utilized in the study of Markov chains, a mathematical system that undergoes transitions between different states. The stationary distribution of a Markov chain can be found by solving a system of linear equations involving matrix inversion.

Optimization: In optimization problems, especially in linear programming, matrix inversion is employed to find optimal solutions. The inversion of matrices is often part of algorithms that iteratively improve solutions until an optimum is reached.

Control Systems: In control theory, matrix inversion is used to design controllers for dynamic systems. It plays a crucial role in determining the control law that stabilizes a system and achieves desired performance.

Computer Graphics: Matrix inversion is used in computer graphics to transform and manipulate 3D graphics. Techniques like affine transformations and perspective projection involve matrix operations, including inversion.

Image Processing: In image processing, matrix inversion is applied to techniques like image registration and geometric transformations. This is important in aligning and transforming images for further analysis.

Quantum Mechanics: Matrix inversion is frequently used in quantum mechanics, particularly in the representation of quantum states and the evolution of quantum systems.

Machine Learning: In machine learning, matrix inversion can be involved in certain algorithms, such as linear regression or certain optimization problems. However, due to computational considerations, alternative methods like gradient descent are often preferred.

Back Propagation

Neural Networks use matrix inversion as the calculation for the back propagation step.

There are many different algorithms for back propagation and there is a tremendous amount of research ongoing in this area.

I am investigating these algorithms and looking into data science investigations for improving algorithm efficiency.

Results of Sprint 1

Average R Squared values of the inverse matrix were very low. 0.06 was the highest result I could generate.

However I stumbled upon how the r squared value increased significantly when the square of each matrix element was added to the linear regression.

This got me thinking and i thought about squaring each element before multiplying it by every other element in the matrix, along with the cube of itself. This also significantly increased the r squared value.

This could have very well been an occurrence for the generated set of matrices, so I tried with 6 different sets and this pattern remained.

I really like this result and look forward to using it in the next sprints.

Overview of tests and results

Test 1: linear regression between matrix elements and inverse matrix elements

Test 2: linear regression with test 1 and all possible combinations of one matrix element multiplied by every other matrix element

Test 3: linear regression with test 2 and the square of each matrix element

Test 4: linear regression with test 1 and all possible combinations of one matrix element squared and multiplied by every other matrix element and the cube of each matrix element

Note how Test 4 does not even the parts of test 3 that included all possible combinations of one matrix element multiplied by every other matrix element - these are the tests I am reporting to demonstrate the importance of the aspect of test 4

Results

Average R Squared values for the Inverse Matrix entries

	Random Seed 1	Random Seed 2	Random Seed 3	Random Seed 4	Random Seed 5	Random Seed 6	Random Seed 7
Test 1	0.001	0.0035	0.0025	0.004	0.0022	0.0026	0.0034
Test 2	0.0035	0.024	0.012	0.029	0.0088	0.014	0.02
Test 3	0.0047	0.026	0.015	0.034	0.011	0.017	0.024
Test 4	0.0087	0.04	0.024	0.06	0.017	0.027	0.037

It is very clear that each following test always increases the r squared value and that test 4 is the best always bettering test three by at least a third of test 4. These results clearly show the importance of test 4 along with the increasing importance of each test

Results

To recap, the first result is that adding each matrix element multiplied by every other matrix element to simply a regression of a matrix and its inverse increases the r squared value.

Adding the square of each matrix element to the regression increases the r squared value.

Using the square of each element further multiplied by every other matrix element, along with the cube of each matrix element increases the r squared value to a level higher then other tests

Sprint 2

Consider multiple combination of different size matrices and larger or smaller integer values, and floating point values:

-the same linear regression test from Sprint 1 but with integers from 0-10 and 10000 matrices

-the same linear regression test from Sprint 1 but with integers from 0-20 and 10000 matrices

-the same linear regression test from Sprint 1 but with random floats between 0.5 and 1.9 from and 10000 matrices

-the same linear regression test from Sprint 1 but with a 3x3 hermitian matrix with values between 0.0 and 1.0

Sprint 2 results - r squared values

	Test 1	Test 2	Test 3	Test 4
Case 1	0.0005	0.0009	0.0005	0.0003
Case 2	0.002	0.0024	0.002	0.001
Case 3	0.003	0.007	0.003	0.002
Case 4	0.0055	0.011	0.005	0.003

No new results from sprint 1. We see an increase in r squared values throughout the cases as we did with cases from sprint 1. The Hermitian matrix had the same results as non hermitian matrices.

Next Steps

Consider a decision tree model and possibly linked with an linear regression in some way. More research required.

At this time I am considering non invertible matrices and creating a formula that generate a numerically close inversion of a matrix that is non invertible. Interestingly google and youtube do not have a solution for this, further investigation required.

Also neural nets and tensorflow for determining a formula for the inverse of a matrix, however i will need to consider larger matrices as the formula for invertible matrices of size 2 and 3 is very simple. Inverting larger matrices is more complicated, if an exact inverse exists.