# The Matrix Pt. 3

John Hohman and Micah Hanvich

# Table of Contents

# Matrix Inverse

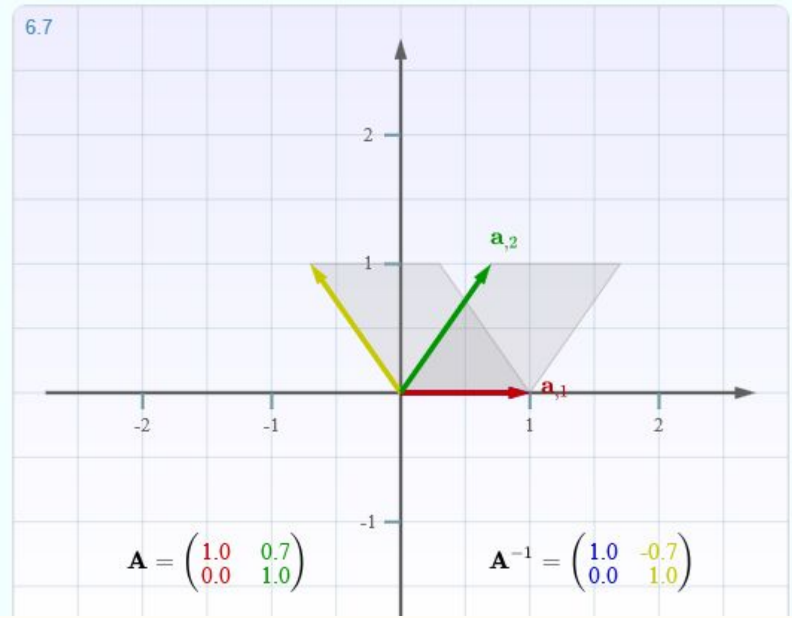
Requirement for if a matrix has an inverse

1. Is the Matrix square
   a. if yes continue
2. Is the determinant of the matrix = 0
   a. If yes then no
   b. If no then yes

https://www.geeksforgeeks.org/check-if-a-matrix-is-invertible/

$$\mathbf{A}\mathbf{x} = \mathbf{y} \iff \mathbf{A}\mathbf{x} = \mathbf{I}\mathbf{y}$$

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{I}\mathbf{y} \iff \mathbf{I}\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$




https://immersivemath.com/ila/ch06_matrices/ch06.html

# Inverse Properties

| |
|---|
| **If $A$ and $B$ are invertible, then $A^T$, $A^{-1}$, $B^{-1}$, and $AB$ are also invertible.** |
| $(A^{-1})^{-1} = A$ |
| $(AB)^{-1} = B^{-1}A^{-1}$ |
| $(A^{-1})^T = (A^T)^{-1}$ |

# Let solve a few!

1. Invert the following Matrix → $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{bmatrix}$

2. Invert the following Matrix → $\begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}$

# Change of Base

Given the following relationship between the two $\mathbb{R}^n$ bases $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ and $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \dots, \hat{\mathbf{e}}_n\}$,

$$
\begin{aligned}
\hat{\mathbf{e}}_1 &= b_{11}\mathbf{e}_1 + b_{21}\mathbf{e}_2 + \dots + b_{n,1}\mathbf{e}_n, \\
\hat{\mathbf{e}}_2 &= b_{12}\mathbf{e}_1 + b_{22}\mathbf{e}_2 + \dots + b_{n,2}\mathbf{e}_n, \\
&\dots \\
\hat{\mathbf{e}}_n &= b_{1,n}\mathbf{e}_1 + b_{2,n}\mathbf{e}_2 + \dots + b_{n,n}\mathbf{e}_n,
\end{aligned}
$$

where a particular vector $\mathbf{v}$ has the following two representations

$$
\begin{aligned}
\mathbf{v} &= v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + v_3\mathbf{e}_3 + \dots + v_n\mathbf{e}_n = \\
&= \hat{v}_1\hat{\mathbf{e}}_1 + \hat{v}_2\hat{\mathbf{e}}_2 + \hat{v}_3\hat{\mathbf{e}}_3 + \dots + \hat{v}_n\hat{\mathbf{e}}_n,
\end{aligned}
$$

and let $\mathbf{B}$ be the matrix with $\hat{\mathbf{e}}_i$ as column vectors, then it holds that

$$
\mathbf{v} = \mathbf{B}\hat{\mathbf{v}} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1,n} \\ b_{21} & b_{22} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{pmatrix} \hat{\mathbf{v}},
$$

where $\mathbf{v} = (v_1, v_2, v_3, \dots, v_n)$ and $\hat{\mathbf{v}} = (\hat{v}_1, \hat{v}_2, \hat{v}_3, \dots, \hat{v}_n)$.

# Orthogonal Matrices

A special set of matrices are the so called *orthogonal* matrices, and they have the convenient property that the inverse can be obtained by just taking the transpose. They are important because, for example, they can describe change of basis between two orthonormal bases.

# Orthogonal Matrices

An orthogonal matrix is a square matrix where the column vectors constitute an orthonormal basis.

## What is an Orthonormal Basis?

For an $n$-dimensional orthonormal basis, consisting of the set of basis vectors, $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$, the following holds

$$\mathbf{e}_i \cdot \mathbf{e}_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

This simply means that the basis vectors are of unit length, i.e., they are normalized, and that they are pairwise orthogonal.

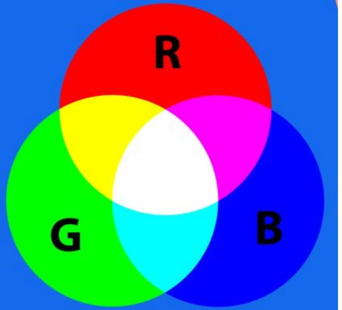
# Flashback!

Remember our image compression discussion?

One representation of Images on computers is made up of **3 values**:
Red, Green, and Blue color data.

Each pixel has this data, which can make reading a large image’s data and displaying it a monumental task.

The solution is compression! While there are a large number of ways to do image compression, we’re going to be focusing on is compression often used for **videos**, aka multiple images back to back.




In Section 6.1, we had one image to the left (original) and another image to the right. The right image is the left image manipulated in a certain way using a matrix. A TV or computer display contains of a number (often millions) of pixels (picture elements) and each pixel has a red, green, and a blue component. For each, pixel we can put these into a vector, i.e.,

$$\mathbf{p} = \begin{pmatrix} r \\ g \\ b \end{pmatrix}, \tag{6.100}$$

where $r$ is the red component of the pixel, $g$ is the green component, and $b$ is the blue component. In the example, we also had a $3 \times 3$ matrix $\mathbf{M}$ that was applied to each pixel. This was done as

$$\mathbf{p}' = \begin{pmatrix} r' \\ g' \\ b' \end{pmatrix} = \mathbf{M}\mathbf{p} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{23} \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}, \tag{6.101}$$

Note that we have used $r$, $g$, $b$ for the vector components in this example instead of $x$, $y$, $z$ or $p_x$, $p_y$, $p_z$. This is to make it clearer what we are manipulating. Using the rules for matrix-vector multiplication, we see, for example, that $r' = m_{11}r + m_{12}g + m_{13}b$, etc. Hence, if we use the identity matrix, $\mathbf{I}$, the original image is obtained and using $\mathbf{M} = \left(\begin{smallmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{smallmatrix}\right)$, we see that $r' = r + g + b$, while $g' = b' = 0$, which results in a red image. Finally, if all rows in $\mathbf{M}$ are identical, we will obtain $r' = g' = b'$, i.e., a gray image. There are many more ways to manipulate images with, but matrices can take you a long way.

# Questions?

# Let's do some practice!

https://images.pexels.com/photos/796603/pexels-photo-796603.jpeg?cs=srgb&dl=hand-desk-notebook-796603.jpg&fm=jpg

# Onward to practicality

https://github.com/fastai/numerical-linear-algebra/blob/master/nbs/2.%20Topic%20Modeling%20with%20NMF%20and%20SVD.ipynb

# Next Week - Determinants

## Reinforcement Learning

Goal: Start Notebook 3 in https://github.com/fastai/numerical-linear-algebra/blob/master/README.md