## 06.11.2019 - Wednesday

8:39 AM

So, I finished with the collaborative filtering algorithms that I wanted to add. Now it seems like it was a piece of cake.

Next thing that I want to do is to try using the Model-based techniques. First chapter of the book talks about the Decision Trees. Turns out that these trees are based on using the reduced matrices using the PCA or SVD technique. Well, penis :-D I avoided for so long to understand what SVD is and it turns out that it bit my ass now.

So, my next target was to understand what SVD is. I start with a couple of articles, the coolest one being <https://towardsdatascience.com/singular-value-decomposition-example-in-python-dab2507d85a0>

Well, that article says that SVD uses the PCA to find the best axes for projecting … Well, great, I have no clue how PCA works ☺ I switch to this article <https://towardsdatascience.com/https-medium-com-abdullatif-h-dimensionality-reduction-for-dummies-part-1-a8c9ec7b7e79> and it honestly just seems like there’s a full road ahead of me …

I also came back to videos of this guy <https://www.youtube.com/watch?v=LyGKycYT2v0>

At this point, I believe I’ll have to recall what the dot product is, what the eigenvectors are, etc. I do vaguely remember of all of that, but I honestly have no faintest idea how it works.

So, next tasks 🡪 recall the Dot product, try to understand the PCA and hopefully try to understand the SVD. Yeah, it can be done using the numpy library but I actually want to see what it is really …

And yeah, I’m a bit pissed because I have no idea if I’m just wasting my time now or not … Who knows …

Finally, to be honest, I believe that my Master thesis is already good enough as it is. But I just want to teach myself how the Model-based techniques work and, frankly, WTF SVD is …

First important thing that I learn is *“The main objective of dimensionality reduction is this:****find a low-dimensional representation of the data that retains as much information as possible.****”* ([source](https://towardsdatascience.com/https-medium-com-abdullatif-h-dimensionality-reduction-for-dummies-part-1-a8c9ec7b7e79))

*This is essentially****Dimensionality Reduction:****finding the best low dimensional representation of the data.* ([source](https://towardsdatascience.com/https-medium-com-abdullatif-h-dimensionality-reduction-for-dummies-part-1-a8c9ec7b7e79))

Seems like I’ll have to remind myself of axis rotation, etc. I do remember that 3 Blue 1 Brown guy was talking about it.

So, from what I understood so far, PCA is about finding the best axes where to project the data, then finding the axis that has highest variance and dropping the ones with lower variance. But, how to find those? That seems to be part of SVD or whatever? No idea yet …

9:00 AM

I was watching this <https://www.youtube.com/watch?v=LyGKycYT2v0&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&index=10&t=0s> but I don’t have time to finish it … hopefully tonight

6:00 PM

Ok so I’m pretty much tired and I’ll take just a quick glance over everything. So, to recap, this is what I want to understand:

1. What is PCA? What are the basic principles of it?
   1. I don’t have to be able to implement it myself but I should at least be familiar with its inner workings, what Eigenvectors are, etc
2. The god-damn dot product …. I guess I need to get more familiar with it
3. What is SVD and how it works? No need to know the inner details but I should at least be familiar with products of it

Once I’m familiar with those stuff, I’d like to be able to do following:

1. Make some sample implementations for PCA and SVD in Python
2. Go back to book and understand how are these techniques used in Recommender systems

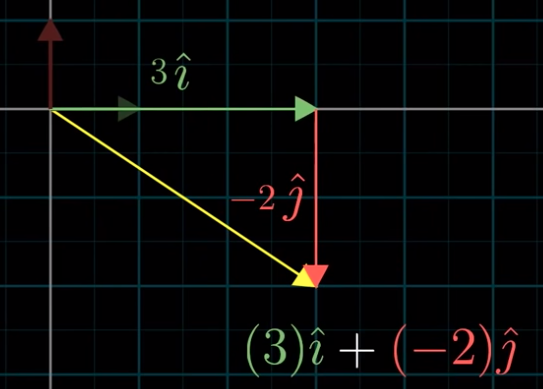
So, let’s start with PCA …

Perfect quote that I’ve seen in Video: “The introduction of numbers as *coordinates* is an act of violence!” ([source](https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&index=1)).

So vectors are, from “mathematican’s point of view” ordered sequences of numbers that can be represented graphically (using a coordinate system). E.g. is a vector that can be represented by moving by 3 to the left and 1 to the top.

Vector addition really represents what you’d get if you just moved straight there. For example: + = . This is the same as saying – first move 3 to the left, then 1 up. Then go 4 left and 2 up. We end up at 7 left and 3 up, which is the same destination we’d get to if we moved 7 left and 3 up.

Basis vectors are and . They are vectors of length 1 and generally, any number in the vector (e.g. means that is three times and is 1 time):



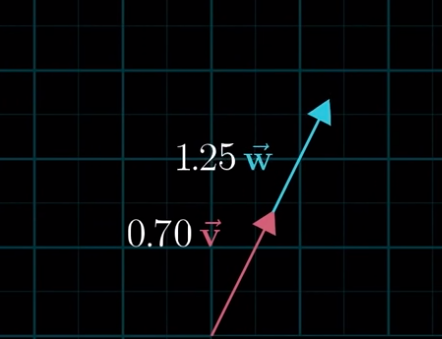
Technically, we can choose any pair of basis vectors. Depending on what we choose, most of the time we can really reach ANY other vector. And these vectors that we reach are actually called “Linear combinations”.

Also, The “span” of and is the set of all their linear combinations:

A close up of a screen

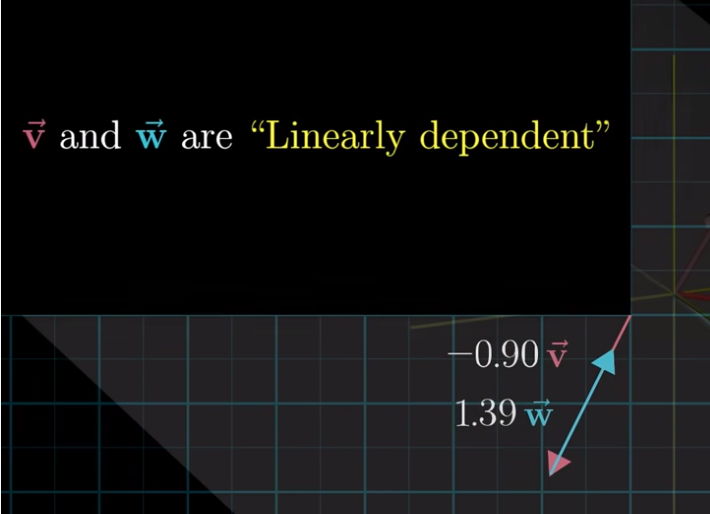
Description automatically generated

Generally, span of any two basis vectors is all vectors in that dimensional space. Exceptional scenario is when basis vectors are lined up:



Then, the span is only one dimension, as it can only move up or down …

The terminology when we have vectors that are sitting on the same plane and where one of them is redundant is called [“Linearly dependent vectors”](https://youtu.be/k7RM-ot2NWY?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&t=511)



If, however, new vector DOES add a new dimension to the span, they are said to be [“linearly independent”](https://youtu.be/k7RM-ot2NWY?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&t=532).

**The basis of a vector space is a set of linearly independent vectors that span the full space.**

According to [Wikipedia](https://en.wikipedia.org/wiki/Basis_(linear_algebra)): “In mathematics, a set B of elements (vectors) in a vector space V is called a basis, if every element of V may be written in a unique way as a (finite) linear combination of elements of B. The coefficients of this linear combination are referred to as components or coordinates on B of the vector. The elements of a basis are called basis vectors.”.

Vector transformations describe how vectors move from starting to ending position.

[Linear transformations](https://youtu.be/kYB8IZa5AuE?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab) are called linear if they satisfy the following properties:

1. Lines remain lines
2. Origin remains fixed

Funny enough, if we describe each transformation as “where the” and basis vectors land, then the matrix multiplications is really a transformation of a vector using the linear transformation. Specifically, it describes where the vector lands after applying the transformation.

Funny enough, seems like matrix multiplications is really just a transformation of space … LOL.

One way of capturing where vector lands after TWO linear transformations is:

A close up of a clock

Description automatically generated

Which is interesting.

A close up of a clock

Description automatically generated

Basically, we can think of matrix multiplication as applying one transformation, then another to get to the end result.

A screenshot of a cell phone

Description automatically generated

It’s really all about where the and go.

All in all, thinking about multiplication as a series of transformations is actually so cool that … hell, how do you even forget it, lol. Relevant video: <https://www.youtube.com/watch?v=XkY2DOUCWMU&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&index=4>

## 07.11.2019 – Thursday

08:00

Battle plan for today: watch two videos about Math from 3Blue 1Brown, then take one area that I didn’t finish in Master thesis and finish it (e.g. Implicit VS Explicit ratings stuff).

As a reminder – goal is to understand the PCA, SVD, eigenvectors, etc. so that I can implement at least one algorithm that uses them.

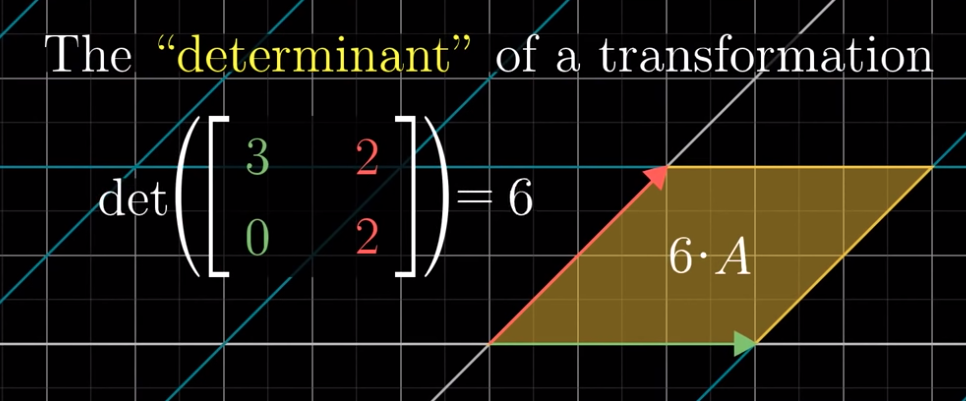
So, first video was about 3D transformations and basically describing that matrix multiplications is nothing more than vector transformations:

Screen of a cell phone

Description automatically generated

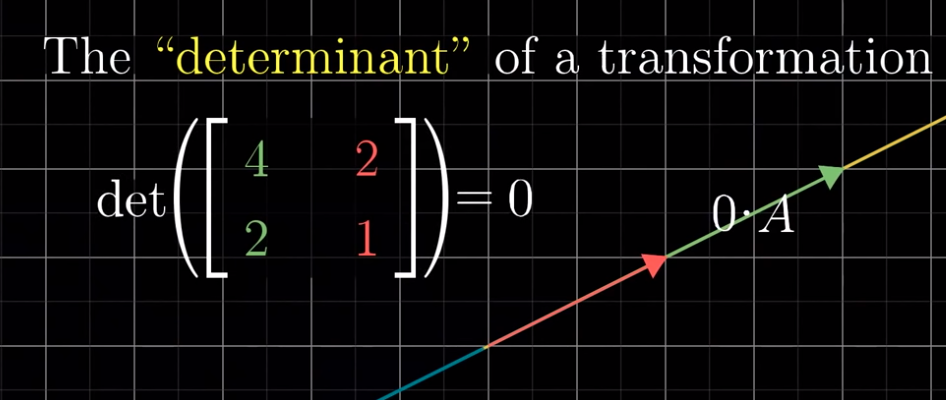
Next video is about determinant. That’s not bad to see …

[Next video](https://www.youtube.com/watch?v=Ip3X9LOh2dk&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&index=6) is about a determinant. Apparently, what a DETERMINANT is – it describes how much the space stretches or shrinks after applying the transformation:



In above case it increases the space by factor of 6.

Interesting scenario is when it squishes the space onto a single line:



Then, it’s a zero.

[Another interesting scenario](https://youtu.be/Ip3X9LOh2dk?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&t=227) is – whenever the orientation of matrix changes (i.e. invertes) then the determinant is negative:

A picture containing text

Description automatically generated

What is meant by intert is that, in the starting position is to the LEFT of . If after transformation it ends up to the RIGHT, then we say that the orientation has inverted.

In terms of 3D, it tells you how much the VOLUME scales.

Formula:

A close up of a sign

Description automatically generated

[Here’s intuition](https://youtu.be/Ip3X9LOh2dk?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&t=460).

Really cool stuff. I totally forgot about it.

[Next video](https://www.youtube.com/watch?v=uQhTuRlWMxw&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab&index=7) is about Inverse matrices. That’ll stay for tonight. Let’s see what is left in Thesis now.

09:00

Watched two videos, as planned. Wrote about Explicit and Implicit ratings and Similarity coefficients.

Started reading about Recommender Systems in Software Engineering in <https://www.researchgate.net/profile/Gerald_Ninaus/publication/261263565_recommender_systems_future/links/00463533bd544bf3c3000000/recommender-systems-future.pdf>

To be continued tonight …