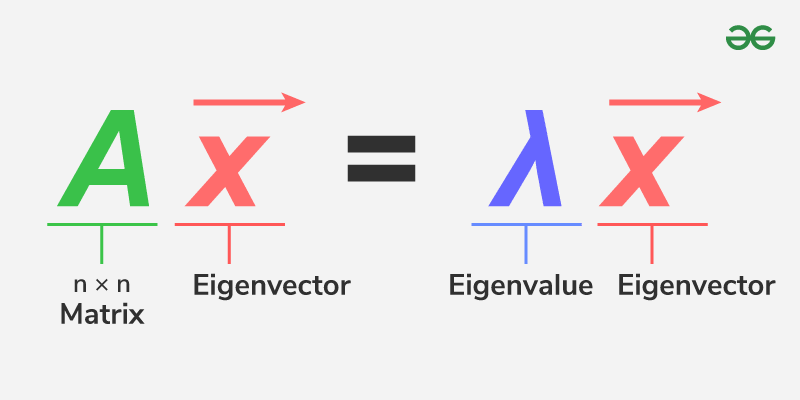
**Getting EigenValues**

By Ekow Yalley



Abstract: In this project, I hope to explore a way to solve the problem of getting a eigenvalue and eigenvector through parralization. Eigenvalues and vectors are a big part of PCA which is a big part of machine learning, so finding a way to make it easier to get eigenvalues and vectors is important to making it easier to train models to parse through large data sets, and perform Pca analysis. There are a variety of large data sets that this optimized eigen calculator could be used to parse through, for example a data set on whats most important to win an Nba championship. I want to optimize the eigenvalue finding algorithm with Numba, it’s the software I would like to learn more about. I plan on using my personal laptop as the hardware I work on, it’s a dell with 16 GB and Intel R core. I hope to benchmark the eigenvalue and apply it to multiple pca analyses on different data sets.

Schedule:

Sunday February 9 - Project Proposal Milestone Due

Week 2 of February- Research eigen values

Week 3 of February- Research PCA

Week 4 of February- Research numba/ work on part on of project

Sunday February 26 - Project Part 1 Due

Week 1 of March-Reasearch datasets to explore

Week 2 of March-Explore datasets

Week 3 of March- Begin bench marking both versions of calculator

Week 4 of March- Test calculators with PCA

Week 1 of April- Test PCA on datasets

Week 2 of April- Finish up project

April 15 - Final Project due

Software exploration:

The software that I plan to use for exploration is Numba through my research I found that it’s a popular software for optimizing Python code especially when it has to do with computations and making calculations. Numba is often used to optimize Python code for machine learning, so it would only be natural to use it. Another Popular optimization software is Cuda. I plan to use the Numba documentation to explore more and figure out how to optimize my code. I plan to make a list of how to use Numba to optimize when I gain a better understanding of Numba.

Benchmark:

I plan to benchmark a eigenvalue calculation algorithm in Python, with time on the terminal. There are versions of the eigenvalue calculation algorithm that can run in parallel so mine will also be able to run in parallel. What I think will be a successful outcome is if my benchmarking shows at least double the speed between the optimized and non-optimized versions of code. I am going to benchmark with OpenMp and prove the at least 2x speed increase. I would say the project is a success if the optimized eigen finder shows any sort of more than 50 percent increase in the speed it takes to do a Pca analysis.