Project Goal:

Implement PageRank

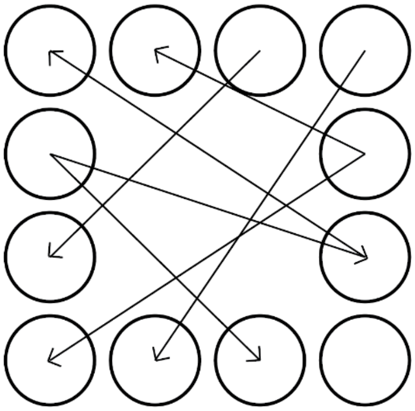
PageRank is an interesting algorithm that poses a technical challenge for this project.

Seek to reach reasonable efficiency in Markov Process

PageRank has a variety of implementations in both computer science and mathematics. There are algorithmic nuances that improve either efficiency or accuracy.

Apply PageRank to an interesting problem

- *To test and implement a legitimate algorithm, it is essential that it be tested on a dataset.*
- *I will write **generic** program files for any. All data will be pulled from common project **Wikipedia** pages.*



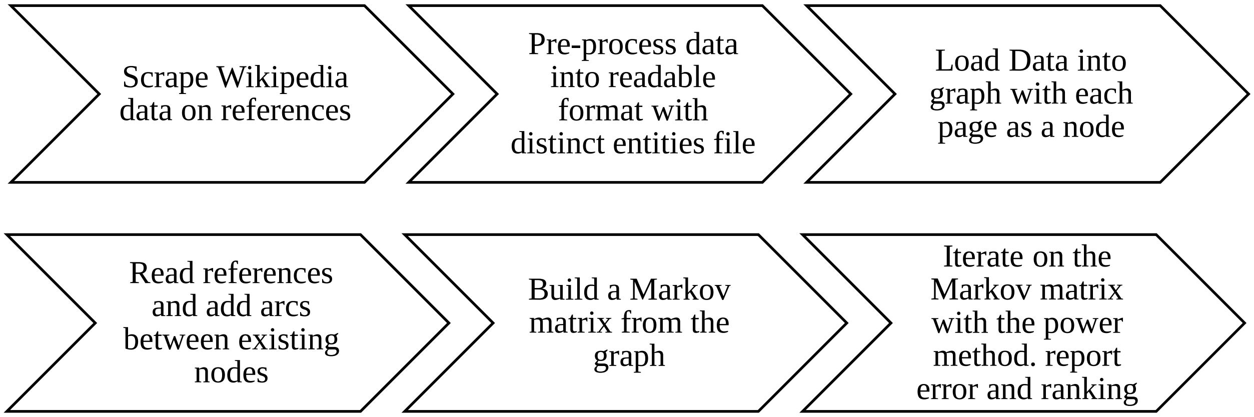
Background:

PageRank

- *Named after Larry Page, invented by the Google founders for sorting web pages for their search engine's results*
- *PageRank takes advantage of hyperlinked articles and websites to compute **probabilities** for users to navigate to other websites.*
- *Builds a **directional graph** of the internet.*
- *What differentiates PageRank is that it is not an indication of popularity (which websites have the most references to themselves) which can be easily manipulated, but rather which **websites confer the most ethos** by being referenced by other highly referenced websites.*

Engineering:

Program Structure



Scrape Wikipedia data on references

Pre-process data into readable format with distinct entities file

Load Data into graph with each page as a node

Read references and add arcs between existing nodes

Build a Markov matrix from the graph

Iterate on the Markov matrix with the power method. report error and ranking

Graph

- *Wikipedia pages are stored as nodes in a graph*
- *An arc is constructed from the page to each hyper link in the article*
- *Arcs have equal weight. Multiple links to a single article will be factored as a higher weight in the matrix*

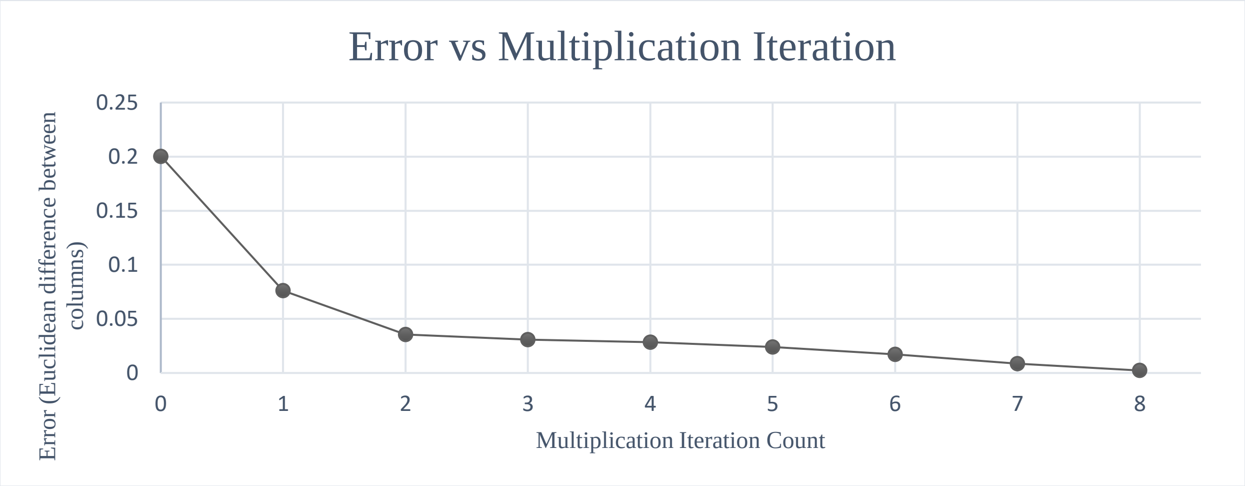
Algorithm

- *Markov matrix is built - columns sun to 1, column values represent probabilities of users clicking link in n steps*
- *Squaring a Markov matrix leads it to converge so each row is comprised of the same value.*
- *The first column contains the ranks. Higher value indicates a higher rank*
- *There are many iteration methods: brute force, iterative, eigenvalues, power method*
- *The power method was chosen for easy evaluation of error and its exponential improvement in accuracy with each iteration*
- *Matrices are copied and squared repeatedly*
- *A padding value of 0.15 is used to prevent a sparse matrix*

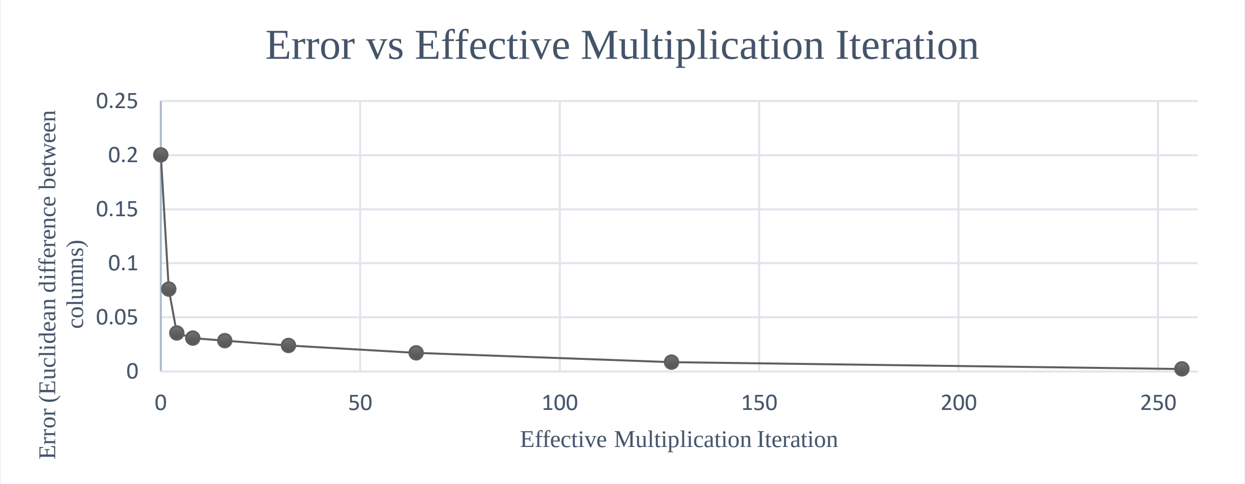
Findings:

PageRank Performance

- *Based on the error graphs, the PageRank algorithm was effective.*
- *In 8 iterations, the error ~0.00228. Error reduces, it's rate of reduction decreases with order as expected from a converging matrix.*



Error vs Multiplication Iteration



Error vs Effective Multiplication Iteration

- *Run time of ~5 minutes for a 1,700 by 1,700 matrix to be squared. Matrix multiplication is O(n^3)*
- *The effective multiplication grows by n^2, so the method is comparable to other methods when a high number of iterations (5 or more) need to be completely*
- *For a dataset larger than 5,000 elements, the Power method probably would not be used.*

Topic Specific Findings

I applied this algorithm to a variety of topics on Wikipedia, including the ranking of all philosophers listed. Here are the top 18 rankings!

1 Aristotle
2 Immanuel Kant
3 Plato
4 David Hume
5 Edward Zalta
6 Georg Hegel
7 Thomas Aquinas
8 Nietzsche
9 Karl Marx
10 John Locke
11 Augustine
12 Rene Descartes
13 Baruch Spinoza
14 John Stuart Mill
15 Kierkegaard
16 Averroes
17 Isaac Newton
18 Heidegger

Final Notes:

Challenging Components

- *Implementing and testing a matrix multiplication method was the most challenging. Initially, I wrote methods iterating as I would by hand*
- *The power method was more efficient, but it still had high error on large datasets. Perron's Theorem which introduces a 0.15 buffer to the sparse matrix, removing all the 0s which would prevent the matrix from converging in reasonable time*

Further Improvements

- *This power method would be more effective if it had parallel processing.*
- *For significantly larger datasets, it may be more viable to implement the iterative method in which an equal probability initial vector is multiplied by a Markov matrix. This would have $O(n^2)$ efficiency but would not have exponential multiplication.*
- *It may also be interesting to explore other datasets in which citations rather than somewhat arbitrary Wikipedia hyperlinks are used.*

Application of the PageRank Algorithm to Wikipedia Articles

Rayan Krishnan 106X 2019

Application of the PageRank Algorithm to Wikipedia Articles

Rayan Krishnan 106X 2019