21-241 Matrices and Linear Transformations **Fall 2019**
Final project **Due Friday, December 6**

In this final project you will explore one application of linear algebra in depth. You may choose one of the topics listed below, or choose your own application, subject to approval of the instructor. Your project must include some coding in Julia and a written paper. The following descriptions give an idea of the minimum requirements for a good project, but you are encouraged to go deeper in the directions you find interesting.

You may work alone or with a partner. If you work as a pair, you should make only one submission, and both partners will receive the same grade. You will have the opportunity to discuss project topics in recitation on Thursday, November 21. The deadline to choose a partner (if working in a pair) and a project topic is Friday, November 22.

You will submit a single pdf document on gradescope. You should write a paper summarizing the linear algebra background required for your application, the code you wrote and results you obtained for your application, and a summary of what you learned. Your document must contain your code: You can either typeset it in *LaTeX* using the `verbatim` command, or print a Julia notebook to pdf and include it as an appendix to your paper.

Your grade will be a combination of your mathematics and code, and your exposition. We will evaluate:

Mathematics and code

- Does your code produce the intended results on all valid inputs?
- Is your code well organized and documented?
- Are the mathematical ideas implemented correctly?
- Are the choices (e.g. of parameters) made in the code appropriate?

Paper

- Is the paper structured (in sections and paragraphs) to clearly explain your work?
- Is the writing clear, grammatical, and free of errors?
- Is the mathematical notation defined and consistent?
- Are the following things clearly and correctly explained?
  - Mathematical background
  - Algorithms and computations (Pseudocode and examples are critical)
  - Coding and algorithm design decisions
  - Results and interpretation of computations

- Does your paper have a complete bibliography that contains the background for your work and any other work you refer to?

Project 1: Iterative methods to compute eigenvectors and singular vectors

Eigenvalues, eigenvectors, singular values, and singular vectors are computed using iterative methods, such as the Power method and QR method. In this project you should learn what these methods are and why they work.

Why can they be counted on to be fast and accurate?

How are they computed? How can they be made more efficient for sparse matrices? What are the differences in performance for different implementations?

Write code to compute singular values and singular vectors using at least one of these methods.

References (will be linked from course website)

- Strang Chapter 11.3 (p. 528-529)
- *Foundations of Data Science* by Avrim Blum, John Hopcroft, and Ravindran Kannan, Chapter 3.7

Project 2: Markov chains, random walks, and PageRank

PageRank is the original algorithm developed to rank web pages based on the link structure of the internet. In this project you should learn how this algorithm works and implement a version of it. You should understand the modification made to change the algorithm from a naive random walk and why this modification is necessary.

You should learn how this algorithm can be run efficiently on a truly large graph.

You can also explore successors to PageRank, such as the HITS algorithm.

Your code should take as input a list of websites (indexed by natural numbers) and for each website a list of links. It should output a list of ranks, one for each site.

References (will be linked from course website)

- Strang Chapter 7.3 (p. 386-387)
- How Google works: Markov chains and eigenvalues, by Christiane Rousseau
- *Google's PageRank and Beyond*, by Langville and Meyer

Project 3: Spectral clustering

A clustering problem seeks to group the elements of a data set into a fixed number of clusters.

Create a similarity (weighted) graph for your data. Then from the adjacency matrix for your graph (or normalized Laplacian), project your data down to a lower dimensional space, where a standard clustering algorithm (e.g k-means) can be run efficiently.

What considerations go into choosing the number of clusters?

Why is this projection useful? (How would k-means perform on high-dimensional data?) Why is the space spanned by the eigenvectors/singular vectors with the largest eigenvalues/singular values be a good choice for projection?

You should write code to perform one or more spectral clustering algorithms (input: a dataset, output: data sorted into groups (clusters)). Apply your results to some real data and interpret your results.

References (will be linked from course website)

- Spectral clustering for beginners by Amine Aoullay
- *Foundations of Data Science* by Avrim Blum, John Hopcroft, and Ravindran Kannan, Chapter 7 (Specifically, Chapter 7.4)