MTH343 Final Project - Lanczos Algorithm

Will Wolff-Myren (willw@pdx.edu)

2018-03-19

Overview

In this programming assignment, I will create a library of functions for operating on matrices stored in a Compressed Sparse Row format, as well as a presentation layer (web application), using Angular and TypeScript.

More specifically, this library will be able to:

- Read a matrix A from a CSR-format file
- Produce $B = A^T$, and store B in CSR format.
- Compute $C = AB$, check C for symmetry, and store C in CSR format.

This application will also:

- Implement the Lanczos algorithm, to generate the tridiagonal matrix $T = Q^T AQ$, with the option to stop at any step $m \geq 1$, where $Q = [q_1, q_2, ..., q_m]$ will consist of only the first m orthonormal vectors copmuted by the Lanczos algorithm.
- Compute the eigenvalues of the tridiagonal matrix T, for selected values of m.

In the subsequent sections of this report, I will briefly explain the algorithms used, as well as technical implementation details, and an analysis of the behavior of the eigenvalues of T for increasing values of m.

Technical Implementation

I decided to base my implementation around the Angular application framework, as a chance to utilize some of the experience that I already have with Angular and with TypeScript in general. I wanted to implement this assignment in such a way that it could be easily demonstrated without needing to be compiled/installed, and I thought it would be easier to visualize within a web browser.

In addition to the benefit of the inherent portability of constructing an application in a format that can be statically hosted (in other words, it does not require a server to run; all of the processing happens on the client, which in this case is the browser, or node), JavaScript and TypeScript have somewhat unique properties that I wanted to utilize in my matrix construction.

There are a variety of well-established numerical libraries available, including BLAS, EISPACK, LAPACK, and SciPy, but relatively few (that I am aware of) at the moment which are available for JavaScript, and even fewer available for TypeScript.

- TODO: Review libraries that I chose for this implementation
  - Bluemath
  - Math.js
  - scijs
- TODO: Compare/contrast matrix implementations in the libraries above
  - Bluemath and scijs both use NDArray
  - Math.js uses DenseMatrix and SparseMatrix
- TODO: Describe performance of each of the implementation approaches
  - Discuss the performance analysis provided by https://github.com/mikolalysenko/ndarray-experiments

Analysis

...

Eigenvalue Comparison (for selected values of m)

...

Conclusion

...

References

Algorithms

Compressed Sparse Row (CSR)

Compressed Sparse Row

Related:

- Sparse matrix

Lanczos

Lanczos algorithm

Related:

- Generalized minimal residual method
- Matrix-free methods
- Arnoldi iteration
- Householder transformation
- Singular-value decomposition

Application Libraries

LAPACK and OpenBLAS

- LAPACK
- OpenBLAS

JavaScript Libraries

- math.js | an extensive math library for JavaScript and Node.js
- scijs/packages

Data Sets

- The Matrix Market Top Ten
- UF Sparse Matrix Collection - sorted by id