Comparison of Matrix Algebra Computational Performance Between RcppEigen and Base R

Costa Stavrianidis
GitHub Repository - https://github.com/Costa-Stavrianidis/BIOS823_Final

Abstract

In this study, we compare the performance of various matrix algebra computations across functions created using the RcppEigen package and functions in base R. The goal is to quantify the efficiency differences between the two for computations that complete the same goal.

Introduction

Compiled programming languages have their programs compiled into machine-readable instructions before execution. Examples of compiled languages include C, C++, Rust, and Fortran. Interpreted languages have their programs read and executed by an interpreter rather than translating the program into machine-readable instructions. Examples of interpreted languages include Python, R, and JavaScript.

Both types of languages contain their own advantages and disadvantages. The advantage of creating a compiled program is that it is faster than an interpreted program at execution.

Table 1: Time in Microseconds of First Set of RcppEigen and Base R Functions by Square Matrix Dimensions

Dimensions	Eigen_matrixmult	R_matrixmult	Eigen_transpose	R_transpose	Eigen_eigenvalues	R_eigenvalues
10	37,510.9	1,734.3	34,083.3	1,693.3	50,934.3	122,688.4
50	13,538.2	42,972.1	3,731.0	4,883.1	125,382.1	362,886.9
100	75,095.6	374,772.8	16,637.8	16,350.8	644,934.1	1,476,168.1
200	773,501.9	2,879,044.6	65,964.9	61,992.0	4,299,268.2	8,889,730.7
500	8,344,074.0	41,626,746.5	643,544.2	489,072.6	67,355,476.5	126,801,782.4

Table 1: Time to execute in microseconds for various matrix algebra functions using RcppEigen and base R. Dimensions column indicates dimensions of square matrix used for computation. Matrixmult computes product of matrix multiplication between two matrices, transpose transposes a matrix, and eigenvalues computes the eigenvalues of a matrix.

Table 2: Time in Microseconds of Second Set of RcppEigen and Base R Functions by Square Matrix Dimensions

Dimensions	Eigen_eigeninv	R_eigeninv	Eigen_cholinv	R_cholinv	Eigen_linearsolve	R_linearsolve
10	38,482.6	11,976.1	35,678.2	1,902.4	52,488.2	5,924.5
50	36,690.9	93,098.7	29,093.6	29,257.6	35,534.7	39,187.8
100	172,462.4	524,816.4	158,202.6	170,195.1	185,291.3	206,845.0
200	935,275.6	3,715,202.7	767,360.1	1,131,284.3	965,021.1	1,252,242.5
500	11,822,083.5	53,687,556.6	9,980,109.3	16,105,919.3	11,579,359.4	17,356,956.4

Table 2: Time to execute in microseconds for various matrix algebra functions using RcppEigen and base R. Dimensions column indicates dimensions of square matrix used for computation. Eigeninv inverts the matrix, cholinv inverts the matrix using the Cholesky decomposition, and linearsolve solves the linear equation $Ax = b$ when given A and b by taking the inverse of matrix A and multiplying it by vector b.

Figure 1: Time Efficiency of Matrix Algebra Functions