Using the fastQR package

Mauro Bernardi

2025-01-28

Introduction

In numerical linear algebra, matrix factorization techniques are fundamental for solving systems of equations, least squares problems, and eigenvalue computations. One of the most widely used methods is the QR decomposition, which factors a given matrix $A \in \mathbb{R}^{m \times n}$ into the product of an orthogonal matrix Q and an upper triangular matrix R. The decomposition is particularly advantageous due to its numerical stability and efficiency when applied to various computational problems, including solving linear systems and computing the singular value decomposition (SVD) [1]. 1

QR decomposition can be computed through several algorithms, such as the Gram-Schmidt process, Householder reflections, and Givens rotations, each offering different trade-offs in terms of computational efficiency and numerical stability. The Gram-Schmidt process provides an intuitive geometrical approach but may suffer from numerical instability, especially in the presence of nearly linearly dependent columns. On the other hand, Householder reflections are more numerically stable and computationally efficient for large matrices. Givens rotations are particularly suited for sparse matrices due to their ability to zero out specific elements in the matrix.

Due to its versatility and importance in both theoretical and practical contexts, QR decomposition has been extensively studied and remains a fundamental tool in scientific computing and data analysis.

We introduce two innovative R packages, designed for high-performance statistical computing (`C++ Eigen`):

1. `fastQR`: a package for efficient QR decomposition, updating/downdating. It enhances QR decomposition processes, including updates and downdates of the QR, R and L decompositions;
2. `fastBVS`: a tool for Bayesian variable selection. It provides advanced techniques that enables flexible and efficient model selection in Bayesian frameworks. Key features are:

- RJ-type algorithms, with Dirac spike-and-slab (DSS) prior;
- consider both linear Gaussian and probit regressions;
- consider both univariate and multivariate responses;
- SSVS algorithms, with DSS and continuous spike-and-slab (CSS) prior;
- tools for fast processing and analyzing the output, monitoring