

1 Notation

1. $\|\cdot\|$ denotes the 2-norm of a vector, or the operator norm of a matrix.
2. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.
3. $\sigma_i(A)$ denotes the i -th singular value of a matrix A .
4. $\lambda_i(A)$ denotes the i -th largest eigenvalue of a matrix A (we consider A to be hermitian)
5. ϵ_{mach} denotes the machine epsilon.

2 Introduction

A naive approach to compute the SVD of a matrix A is to compute the eigenvalues and eigenvectors of the matrix $A^T A$. However, this is not efficient and can be unstable:

We know that singular values are stable under perturbations, i.e:

$$\|\sigma_i(A) - \sigma_i(A + \hat{A})\| \leq \|\hat{A}\| \quad \forall i$$

We also know that the eigenvalues of a hermitian matrix are stable under perturbations, i.e:

$$\|\lambda_i(A) - \lambda_i(A + \hat{A})\| \leq \|\hat{A}\| \quad \forall i$$

This means that if we have a backward-stable algorithm, i.e $\frac{\|\hat{A}\|}{\|A\|} \in O(\epsilon_{mach})$, we should get:

$$\|\sigma_i(A) - \sigma_i(A + \hat{A})\| \in O(\epsilon_{mach} \|A\|) \quad \forall i$$

However, if we use the $A^T A$ method, we get eigenvalues with error:

$$\|\lambda_i(A^T A) - \lambda_i(A^T A + \hat{A}^T A)\| \in O(\epsilon_{mach} \|A^T A\|) = O(\epsilon_{mach} \|A\|^2) \quad \forall i$$

Computing the square roots of these to get the singular values, we get:

$$\|\sigma_i(A) - \sigma_i(A + \hat{A})\| \in O(\epsilon_{mach} \frac{\|A\|^2}{\sigma_i(A)}) \quad \forall i$$

So we are accumulating an additional error of $O(\|A\|/\sigma_i(A))$ in each singular value. We know that $\|A\| = \sigma_1(A)$, thus the error is $O(\frac{\sigma_1(A)}{\sigma_i(A)})$. In the case when $\sigma_1(A) \gg \sigma_i(A)$, the problem for the i^{th} singular value is ill-conditioned.

In this project, I explore the Golub-Kahan bidiagonalization algorithm, followed by an implicit QR algorithm to diagonalize the bidiagonal matrix.

I discuss the algorithm in the presentation. I have uploaded the notes of the presentation (they are not exhaustive as compared to the presentation, but they give the basic idea of the algorithm).

The code is within this repository.