

# Numerical Methods for Computing Eigenvectors

Eric Zheng

21-241 Final Project

December 6, 2019

## Abstract

In this document, I present some background on numerical methods for computing the eigenvectors and singular vectors of matrices. A Julia implementation of the power and QR methods is given, and the two algorithms are compared

## 1 Mathematical Background

It is assumed that the reader is familiar with linear algebra at the college introductory level, but a brief background of the relevant concepts is presented in this section.

### 1.1 Eigenvectors and Eigenvalues

#### Definition 1.1: eigenvectors and eigenvalues

Consider an arbitrary  $n \times n$  matrix  $A$ . For some  $\mathbf{x} \in \mathbb{R}^n$  (with  $\mathbf{x} \neq \mathbf{0}$ ), we say that  $\mathbf{x}$  is an *eigenvector* of  $A$  if, for some  $\lambda \in \mathbb{R}$ ,  $A\mathbf{x} = \lambda\mathbf{x}$ . We denote  $\lambda$  as the corresponding *eigenvalue* of  $A$ .

From definition 1.1 follows immediately a way to compute the eigenvalues of a given matrix. Note that

$$A\mathbf{x} = \lambda\mathbf{x} \implies A\mathbf{x} - \lambda\mathbf{x} = \mathbf{0}$$

and  $\lambda\mathbf{x} = \lambda I\mathbf{x}$ , so we have

$$A\mathbf{x} - \lambda I\mathbf{x} = (A - \lambda I)\mathbf{x} = \mathbf{0}.$$

That is to say,  $\lambda \in \mathbb{R}^n$  is an eigenvalue of  $A$  if and only if  $A - \lambda I$  has a non-trivial null space. A matrix has a non-trivial null space if and only if it is singular, so we require that  $\det(A - \lambda I) = 0$ . This result is stated in theorem 1.1.

### Theorem 1.1: computing eigenvalues as polynomial roots

Some  $\lambda \in \mathbb{R}^n$  is an eigenvalue of the  $n \times n$  matrix  $A$  if and only if  $\det(A - \lambda I) = 0$ . We call  $\det(A - \lambda I)$  the *characteristic polynomial* for  $A$ . The problem then becomes identifying the roots of this polynomial.

Computing eigenvectors is useful for solving [blah blah]

### Theorem 1.2: spectral theorem for real symmetric matrices

If  $A$  is a real symmetric  $n \times n$  matrix, then  $A$  has  $n$  orthonormal eigenvectors.

## 1.2 Singular Vectors and Values

We have seen that eigenvectors and eigenvalues are useful for understanding the behavior of square matrices, but what about the more general case of  $m \times n$  matrices? To handle this, we define the notion of singular vectors and values in definition 1.2.

### Definition 1.2: singular vectors and values

Given an  $m \times n$  matrix  $A$ , we denote the eigenvectors of  $A^T A$  as the *singular vectors* of  $A$ . The corresponding eigenvalues of  $A^T A$  are called the *singular values* of  $A$ .

### Theorem 1.3: existence of singular vectors and values

Any  $m \times n$  matrix  $A$  admits a decomposition  $A = U\Sigma V^T$ , where

- $U$  is an  $m \times m$  orthogonal matrix
- $\Sigma$  is an  $m \times n$  matrix that is “diagonal” in that  $\Sigma_{ij} = 0$  if  $i \neq j$
- $V$  is an  $n \times n$  orthogonal matrix.

## 2 Numerical Methods

In general, we have seen that it is possible to compute eigenvectors of an  $n \times n$  matrix as the roots of an  $n$ -degree polynomial. Unfortunately, it is a well-known result that polynomials of degree 5 and higher do not in general admit a solution by radicals [cite]...

The most obvious way to compute the eigenvectors of  $A$  is to first find the eigenvalues by  $LU$  decomposition and then use elimination to find the null space of  $A - \lambda I$ .

## 2.1 The Power Algorithm

Present algorithm Explain why it works Possibly bounds on accuracy?

Conditions on convergence: if there are multiple eigenvalues with the same magnitude, this will not converge. (Since any symmetric matrix has real eigenvalues, this only occurs when  $\lambda$  and  $-\lambda$  are both eigenvalues of  $A$ .) [Is there some other condition on null spaces and stuff?] Also, the rate at which this converges depends greatly on the ratio between the two eigenvalues!

Also, does this handle degenerate eigenvalues?

We first present algorithm 2.2 for symmetric matrices only because it is highly illustrative. Additionally, [all we need for singular vectors]

We square and then normalize by the first nonzero column of  $A^n$ . If such a column does not exist (i.e.  $A^n$  is the zero matrix), then  $A$  must have been a nilpotent matrix. In this case, the only eigenvalue can be zero, as proved in theorem 2.1. In this case, the problem reduces to finding the null space of  $A$ .

Now a matrix can have at most  $n$  independent eigenvectors, so we only have to iterate up to  $n$ .

### Theorem 2.1: eigenvalues of a nilpotent matrix

If  $A$  is a nilpotent matrix (i.e.  $A^n$  is the zero matrix for some  $n \in \mathbb{N}^+$ ), then the only eigenvalue of  $A$  is  $\lambda = 0$ .

*Proof.* Suppose  $\mathbf{x} \neq \mathbf{0}$  were an eigenvector of  $A$  with eigenvalue  $\lambda \neq 0$ . Then  $A^n \mathbf{x} = \lambda^n \mathbf{x}$ , but if  $\lambda \neq 0$ , then this is some nonzero vector. However,  $A^n = 0$ , so we also have  $A^n \mathbf{x} = \mathbf{0}$ , a contradiction.  $\square$

### Algorithm 2.1: DominantEigen

```

input      : real diagonalizable  $n \times n$  matrix  $A$ 
parameter: tolerance  $\varepsilon > 0$ , max iterations  $N > 0$ 
output     : dominant eigenpair  $(\mathbf{x}, \lambda)$ 

 $A' \leftarrow A$ ;
 $i \leftarrow 0$ ; // number of iterations so far
 $\mathbf{x} \leftarrow \text{FirstNonzeroCol}(A')$ ;
 $A' \leftarrow \text{SquareAndNormalize}(A')$ ;
while  $\|\mathbf{x} - \text{FirstNonzeroCol}(A')\| > \varepsilon$  and  $i < N$  do
     $\mathbf{x} \leftarrow \text{FirstNonzeroCol}(A')$ ;
     $A' \leftarrow \text{SquareAndNormalize}(A')$ ;
     $i \leftarrow i + 1$ ;
 $\mathbf{x} \leftarrow \text{FirstNonzeroCol}(A')$ ;
 $\lambda \leftarrow (A\mathbf{x} \cdot \mathbf{x}) / (\mathbf{x} \cdot \mathbf{x})$ ;

```

**Algorithm 2.2: EigenPowerSymmetric**

```

input      : real symmetric  $n \times n$  matrix  $S$ 
parameter: tolerance  $\varepsilon > 0$ , max iterations  $N > 0$ 
output    : list  $xs$  of eigenvectors and  $\lambda s$  of eigenvalues

 $xs \leftarrow$  empty list;
 $\lambda s \leftarrow$  empty list;
 $Z \leftarrow n \times n$  zero matrix;
for  $i \in [n]$  do
     $\mathbf{x}, \lambda \leftarrow \text{DominantEigen}(S - Z)$ ;
     $Z \leftarrow Z + \lambda \mathbf{x} \mathbf{x}^T$ ;
    append  $\mathbf{x}$  to  $xs$ ;
    append  $\lambda$  to  $\lambda s$ ;

```

Algorithm 2.2 also gives us an easy way to compute the singular values and singular vectors of any real matrix  $A$ : simply apply `EigenPowerSymmetric( $A^T A$ )`.

Finding the eigenvectors of a general square matrix (not necessarily symmetric) is somewhat trickier. We will

[probably put a disclaimer at the top that we will only concern ourselves with real matrices, although we will sometimes emphasize this point in our algorithms. should we permit complex eigenvalues? nah, that moves us from  $\mathbb{R}^n$  to  $\mathbb{C}^n$ ]

## 2.2 The QR Algorithm

**Algorithm 2.3: EigenQR**

```

input      : real  $m \times n$  matrix  $A$  with independent columns
output    : eigenvectors  $\mathbf{x}_i$  and corresponding eigenvalues  $\lambda_i$ 

```

## 3 Results

Compare the two methods