

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

I assume that the reader is familiar with linear algebra at the college introductory level. In this section, I present the mathematical concepts behind different methods for computing the eigenvectors and singular vectors of a matrix.

1.1 Polynomial Roots

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 iff, for some $\lambda \in \mathbb{R}$, $A\mathbf{x} = \lambda\mathbf{x}$. We denote λ as the corresponding *eigenvalue* of A .

From definition 1.1 follows immediately a somewhat naive 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, $\lambda \in \mathbb{R}$ 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.

Once the eigenvalues have been computed, we can find the corresponding eigenvectors by finding the null space of $A - \lambda I$, for example by using the reduced row echelon form. The key drawback of this method is that polynomial root-finding is sensitive to small numerical errors.

Example 1.1: sensitivity of polynomial root finding

Consider the matrix

However, the equivalence of eigenvalue-finding and polynomial root-finding gives some insight into how we could approach the problem with more advanced methods. It is known that polynomials of degree five and higher do not in general have a solution by radicals [cite], although we can use iterative methods to get arbitrarily good approximations of these roots. In the following sections, we will apply similar iterative methodologies to find the eigenvectors and eigenvalues of matrices.

1.2 The Power Method

Definition 1.2: dominant eigenvector

Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be the eigenvectors of the matrix A , and let $\lambda_1, \dots, \lambda_n$ be the corresponding eigenvalues. We call \mathbf{x}_i a *dominant eigenvector* if $|\lambda_i| > |\lambda_j|$ whenever $i \neq j$. The corresponding λ_i is called the *dominant eigenvalue*.

Remark. We say a *dominant eigenvector* instead of the *dominant eigenvector* because any nonzero multiple of an eigenvector is still an eigenvector with the same eigenvalue. There are thus infinitely many eigenvectors that correspond to the dominant eigenvalue. When we list out the eigenvectors as $\mathbf{x}_1, \dots, \mathbf{x}_n$, we generally mean independent eigenvectors.

The power method for computing eigenvectors takes successive powers of the matrix A^k until some stopping criterion is reached. As k grows large, the columns of A^k will approach the dominant eigenvector of A . This is stated in theorem 1.2.

Theorem 1.2: the power method

If A is an $n \times n$ diagonalizable matrix with a dominant eigenvector \mathbf{x} , then the columns of A^k approach a multiple of \mathbf{x} as k grows arbitrarily large.

Proof. Since A is diagonalizable, let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be a basis of eigenvectors for \mathbb{R}^n , where we order the eigenvectors so that \mathbf{x}_1 is a dominant eigenvector. Now since the \mathbf{x}_i 's form a basis, any $\mathbf{v} \in \mathbb{R}^n$ can be expressed as the linear combination

$$\mathbf{v} = c_1\mathbf{x}_1 + \dots + c_n\mathbf{x}_n.$$

Then by linearity, we have

$$\begin{aligned} A^k\mathbf{v} &= A^k(c_1\mathbf{x}_1 + \dots + c_n\mathbf{x}_n) \\ &= A^k c_1\mathbf{x}_1 + \dots + A^k c_n\mathbf{x}_n \\ &= \lambda_1^k c_1\mathbf{x}_1 + \dots + \lambda_n^k c_n\mathbf{x}_n. \end{aligned}$$

But since $|\lambda_1| > |\lambda_i|$ for all $i \geq 2$, we see that the first term $\lambda_1^k c_1\mathbf{x}_1$ dominates as k grows very large. Hence for large k , $A^k\mathbf{v} \approx \lambda_1^k c_1\mathbf{x}_1$. Taking \mathbf{v} to be the standard basis vectors then produces the desired result. \square

One issue with the power method is that it assumes that A is both diagonalizable and has a dominant eigenvector. These flaws are highlighted in examples 1.2 and 1.3.

Example 1.2: power method on a non-diagonalizable matrix

Consider the matrix

Example 1.3: power method without a dominant eigenvector

Consider the matrix

The diagonalizability problem can be avoided in most cases because the matrices we are typically interested in are diagonalizable.

[then the faster power method should be presented]

1.3 The QR Method

1.4 Connection to Singular Vectors

Up until now, we have been almost exclusively concerned with finding the eigenvectors and eigenvalues of a matrix. What of the singular vectors?

To make the connection even better, we state theorem 1.3 without proof. The reader is referred to [cite] for a full proof.

Theorem 1.3: spectral theorem for real symmetric matrices

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

But the matrix $A^T A$ is a symmetric matrix, so by theorem 1.3, it is diagonalizable

1.5 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 1.3.

Definition 1.3: 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 1.4: 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
- Σ 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

In this section, we present the *power method*, a simple method for computing the eigenvectors and eigenvalues of A . As the name suggests, this method involves

taking matrix powers A^k for increasingly large values of k . To use the power method, we first choose a random starting vector $\mathbf{x}_0 \in \mathbb{R}^n$.

Theorem 2.1: convergence of the power method

Consider an $n \times n$ matrix A . If $|\lambda_1| > |\lambda_i|$ whenever $i \neq 1$, then there exists some vector $\mathbf{x}_0 \in \mathbb{R}^n$ such that $A^k \mathbf{x}_0$ converges to a scalar multiple of \mathbf{x}_1 as k grows arbitrarily large.

Proof. Suppose A has $e > 1$ eigenvalues. Now let $\mathbf{x}_0 =$

□

This suggests the method behind algorithm 2.1.

There remain a few important questions to be addressed:

- How quickly does the power method converge?
- What about the case when $|\lambda_1| = |\lambda_2|$?
- What about zero eigenvalues?
- How do we choose a suitable initial \mathbf{x}_0 ?
- What if A does not have a full basis of eigenvectors? (i.e. our \mathbf{x}_0 might have a component orthogonal to the span of the eigenvectors—then convergence may not be guaranteed?)
- What about degenerate eigenvalues?

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 λ 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 2.2. 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 2.2: 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]

Example 2.1: failure of the power method

The power method can fail when the original matrix does not have a dominant eigenvalue. For example, consider the matrix

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

which has eigenvalues $\lambda = \pm 1$.

Example 2.2: another failure of the power method

The power method can also fail when the original matrix has complex eigenvalues. For example, consider the matrix

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which is intuitively a rotation around the z -axis. In this case, there is only one real eigenvalue, $\lambda = 1$. If we choose a good initial $\mathbf{x}_0 = (0, 0, 1)$, the algorithm does converge to the correct solution.

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 λ_i

Algorithm 2.4: Singular

```
input      : real  $m \times n$  matrix  $A$ 
parameter: tolerance  $\varepsilon > 0$ , max iterations  $N > 0$ 
output    : list of singular vectors  $vs$  and singular values  $\sigma s$ 
/* here we use the power method, but we could also use
   the QR method                                     */
 $xs, \lambda s \leftarrow \text{EigenPowerSymmetric}(A^T A);$ 
 $\sigma s \leftarrow$  filter positive  $\lambda s$ ;
 $l \leftarrow$  length of  $\sigma s$ ;
 $vs \leftarrow$  first  $l$  of  $xs$ ;
return  $vs, \sigma s$ 
```

3 Results

Compare the two methods