APPM 2460
Eigenstuff

1 Introduction

Given a square matrix A, we define the eigenvalues of A to be the scalar values λ that satisfy the relation:

$$A\vec{v} = \lambda\vec{v}$$

The vectors $\vec{v} \neq \vec{0}$ in the relation are known as the eigenvectors of A corresponding to the λ.

In words, we need A times the eigenvector to return the eigenvector multiplied by its associated eigenvalue. Note that $\vec{v}$ is not unique. That is, we can multiply it by any constant and it is still an eigenvector.

2 Finding Eigenvalues and Eigenvectors

2.1 The Clumsy Way

One way to find the eigenvalues and vectors of a matrix A is this:

- To find the eigenvalues of A, consider

  $$Av = \lambda v \iff (A - \lambda I)v = 0$$

  Recall that this homogeneous system has a unique solution if and only if $|A - \lambda I| \neq 0$, where the vertical bars $|\cdot|$ denote the matrix determinant. In this case, the only solution is the trivial solution $v = 0$. To get something interesting we seek

  $$|A - \lambda I| = 0$$

  This gives a polynomial in terms of λ, which we call the characteristic polynomial. The roots of this polynomial are the eigenvalues of A.

- As an example, let's find the eigenvalues of

  $$A = \begin{bmatrix} 1 & 2 & -1 \\ 1 & 0 & 1 \\ 4 & -4 & 5 \end{bmatrix}$$

  - The characteristic equation for A is

    $$p(\lambda) = |A - \lambda I| = \begin{vmatrix} 1-\lambda & 2 & -1 \\ 1 & -\lambda & 1 \\ 4 & -4 & 5-\lambda \end{vmatrix} = \lambda^3 - 6\lambda^2 + 11\lambda - 6$$

  - We would like to find the roots of $p(\lambda)$. To do this in Matlab, the first step is to set up a function for this characteristic polynomial. We can write it as an anonymous function:

```
p = @(x) x^3 - 6*x^2 + 11*x - 6;
```

- To find the roots of this equation, we use the `fzero` command. (Type `help fzero` for more information about this function.)

  The inputs to `fzero` are (i) the variable that is used to define the function, (ii) the name of the function whose root(s) we seek, and (iii) a guess for the value of the root (respectively):

  ```
  r = fzero( @(variable) function_name(x), initial guess )
  ```

  In our case, we can type:

  ```
  eval_1 = fzero( @(x) p(x), 4)
  ```

- To get the other eigenvalues, use the following snippets:

  ```
  eval_2 = fzero( @(x) p(x), 2.5)
  eval_3 = fzero( @(x) p(x), 0.5)
  ```

(Notice that we named the outputs of each `fzero` solve so that we can easily use these values of the roots – the eigenvalues – later on.)

- Each one of our eigenvalues has an eigenvector associated with it.
  - To find the eigenvector, we solve the system $(A - \lambda I)\vec{v} = \vec{0}$ for $\vec{v}$. Since the right-hand side is zero, it will not be affected by row operations and we can focus on finding the `rref` of the matrix $(A - \lambda I)$ for each value of λ.

    For example, type

    ```
    A = [1,2,-1;1,0,1;4,-4,5];
    B = A-eval_1*eye(3);
    rref(B)
    ```

  - We see from the rref of B that the eigenvector associated with eigenvalue, `eval_1`, is $\mathbf{v_1} = c[-1\ 1\ 4]^T$, where c is an arbitrary constant. We can choose any value for c, but often we choose c to either ensure that v_1 has length 1 or to eliminates fractions from the entries of v_1.

2.2 Finding Eigenvalues and Eigenvectors Quickly

Of course, the method just described was laborious and required quite a bit of hand-calculation.

- A much easier way to find the eigenvalues of a matrix is the `eig` command. Try typing

  ```
  eigenvalues = eig(A)
  ```

  We see that the eigenvalues of A are produced with just one command.

- To find the eigenvalues and eigenvectors all at once, type

  ```
  [V D] = eig(A)
  ```

V is a matrix whose the columns are the eigenvectors corresponding to each eigenvalue. The associated eigenvalues of A are located along the diagonal of the matrix D. *The eigenvector in column i of V is associated with the eigenvalue in column i of D.* The eigenvectors have been normalized to have length 1, potentially making them ugly. If we want Matlab to produce un-normalized eigenvectors, we adjust the command as follows:

```
[V D] = eig(A,'nobalance')
```

APPM 2460 Homework Week 12

Submit a published pdf of your script and any other supporting code needed to solve the following problem to Canvas by Monday, April 15 at 11:59 p.m. See the 2460 webpage for formatting guidelines.

If a matrix A has dimension $n \times n$ and has n linearly independent eigenvectors, it is diagonalizable. This means there exists a matrix P such that $P^{-1}AP = D$, where D is a diagonal matrix whose diagonal entries are made up of the eigenvalues of A. P is constructed by taking the eigenvectors of A and using them as the columns of P. Your task is to write a program (function) that does the following

(a) Write a code that performs the following (you *may* use built-in functions):

- Finds the eigenvectors of an input matrix A
- Checks if the eigenvectors are linearly independent. (Hint: Think determinants. You might want to use the `floor` function.)
  - if they are not linearly independent, exit the program & display error
- Displays P, P^{-1} and D (if possible)
- Shows that $PDP^{-1} = A$

(b) Run the program for a *diagonalizable* 3×3 matrix A. (That is, there are matrices P and D such that $A = PDP^{-1}$. Your program *should not* generate the error message in part (a).)

(c) Run the program for a *non-diagonalizable* 3×3 matrix B. (That is, there are *not* matrices P and D such that $B = PDP^{-1}$. Your program *should* generate the error message in part (a).)

Interesting Fact: Even if a matrix is not diagonalizable, we can get pretty close. Every matrix has something called a Jordan canonical form. For a diagonalizable matrix, this is just the diagonal form, but if we have insufficient eigenvectors, there will be the number 1 in the upper diagonal above the deficient eigenvalues on the diagonal. We construct P with something called the generalized eigenvectors. For more information, type `help jordan`. This is an extremely important theorem of linear algebra!