Homework 10B Eigenvectors

answers

Spectral Decomposition: A Proof

1. Prove that a matrix that is upper- or lower-triangular is normal $\Leftrightarrow$ it is diagonal.

First of all, every diagonal matrix is normal.
We need to show that diagonal matrices are the only triangular matrices that are normal. Equivalently,

Claim: If A is a triangular matrix that is not diagonal, then A is not normal.

Proof:

For simplicity, let's just consider real matrices. (For complex matrices, replace transposes with adjoints below.)

Note that A is normal if and only if A^T is normal.

Therefore it is enough to consider the case that A is upper triangular. (If A is lower triangular, take its transpose.)

Now

$$(A^TA)_{ij} = \sum_k A_{ki}A_{kj} = \text{inner product between column } i \text{ and column } j$$

$$(AA^T)_{ij} = \sum_k A_{ik}A_{jk} = \text{inner product between row } i \text{ and row } j.$$

Assume that A is normal.

In particular,

$$(A^TA)_{11} = \text{the first column's squared length} = A_{1,1}^2 \quad \text{since } A \text{ is upper triangular}$$

$$\| $$

$$(AA^T)_{1,1} = \text{the first row's squared length} = A_{1,1}^2 + A_{1,2}^2 + \cdots + A_{1,n}^2$$

Hence $A_{1,2} = A_{1,3} = \cdots = A_{1,n} = 0$.

Now just repeat that argument. Considering the 2,2 entries of A^TA and of AA^T gives

$$\underbrace{\cancel{A_{1,2}^2}}_{\substack{0 \\ \text{we just showed}}} + A_{2,2}^2 = A_{2,2}^2 + A_{2,3}^2 + \cdots + A_{2,n}^2$$

$$\Rightarrow A_{2,3} = A_{2,4} = \cdots = A_{2,n} = 0$$

And so on. Repeating the argument for the 3,3 entry, 4,4 entry, etc., gives that A is diagonal, as desired. $\square$

Spectral Decomposition: Experiments

2 a) Experiment. Generate 1×1, 2×2 and 3×3 random symmetric matrices A, and compare the eigenvalues and eigenvectors of A to those of $\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix}$.

What pattern do you find? Explain why.

(**Notice**: $\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix}$ and $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$ have different eigenvalues, in general!)

(**Hint**: It is easy to enter block matrices into Matlab, for example as follows:

```
>> m = 1;
>> n = 3;
>> A = randn(m, n)

A =

    0.8886   -0.7648   -1.4023

>> B = [zeros(m,m) A; A' zeros(n,n)]

B =

         0    0.8886   -0.7648   -1.4023
    0.8886         0         0         0
   -0.7648         0         0         0
   -1.4023         0         0         0
```
)

Answer:

```
>> n = 3; A = randn(n,n); A = diag(diag(A)) + triu(A,1) + triu(A,1)';
>> [UA, DA] = eig(A)

UA =

    0.6444    0.0436   -0.7634
    0.5277   -0.7479    0.4027
   -0.5533   -0.6624   -0.5050


DA =

    0.1992         0         0
         0    1.5135         0
         0         0    2.7905
```

```
>> B = [zeros(3,3) A; A zeros(3,3)]

B =

         0         0         0    1.7119   -0.8396    0.9610
         0         0         0   -0.8396    1.3546    0.1240
         0         0         0    0.9610    0.1240    1.4367
    1.7119   -0.8396    0.9610         0         0         0
   -0.8396    1.3546    0.1240         0         0         0
    0.9610    0.1240    1.4367         0         0         0

>> [UB, DB] = eig(B)

UB =

    0.5398   -0.0309   -0.4557   -0.4557    0.0309   -0.5398
   -0.2848    0.5288   -0.3732   -0.3732   -0.5288    0.2848
    0.3571    0.4684    0.3913    0.3913   -0.4684   -0.3571
   -0.5398    0.0309    0.4557   -0.4557    0.0309   -0.5398
    0.2848   -0.5288    0.3732   -0.3732   -0.5288    0.2848
   -0.3571   -0.4684   -0.3913    0.3913   -0.4684   -0.3571


DB =

   -2.7905         0         0         0         0         0
         0   -1.5135         0         0         0         0
         0         0   -0.1992         0         0         0
         0         0         0    0.1992         0         0
```

-2.7905	0	0	0	0	0
0	-1.5135	0	0	0	0
0	0	-0.1992	0	0	0
0	0	0	0.1992	0	0
0	0	0	0	1.5135	0
0	0	0	0	0	2.7905

If $A\vec{v} = \lambda\vec{v}$, then

$$\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix}\begin{pmatrix} \vec{v} \\ \vec{v} \end{pmatrix} = \begin{pmatrix} A\vec{v} \\ A\vec{v} \end{pmatrix} = \lambda \begin{pmatrix} \vec{v} \\ \vec{v} \end{pmatrix}$$

$$\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix}\begin{pmatrix} \vec{v} \\ -\vec{v} \end{pmatrix} = \begin{pmatrix} -A\vec{v} \\ A\vec{v} \end{pmatrix} = -\lambda \begin{pmatrix} \vec{v} \\ -\vec{v} \end{pmatrix}$$

Thus if A has an eigenvalue λ eigenvector $\vec{v}$, then $\begin{pmatrix} 0 & A \\ A & 0 \end{pmatrix}$ has two corresponding e-vectors, with e-values λ and $-\lambda$.

b) What are the eigenvalues and eigenvectors of

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}?$$

Now, as in part a, experiment with different matrices A to try to understand how the eigenvalues and eigenvectors of A relate to those of

$$\begin{pmatrix} 3A & A \\ 0 & 2A \end{pmatrix}.$$

What pattern do you find? Why?

Answer:

$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Eigenvalues	Eigenvectors
1	(1, 1)
−1	(1, −1)

$\begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}$

Eigenvalues	Eigenvectors
3	(1, 0)
2	(1, −1)

```
>> n = 2; A = randn(n,n); A = diag(diag(A)) + triu(A,1) + triu(A,1)';
>> [UA, DA] = eig(A)

UA =

  -0.8069   -0.5907
  -0.5907    0.8069


DA =

  -0.6943         0
        0    2.1038
```

λ_1 λ_2

```
>> B = [3*A 1*A; 0*A 2*A]

B =

    0.8460   -4.0010    0.2820   -1.3337
   -4.0010    3.3825   -1.3337    1.1275
         0         0    0.5640   -2.6674
         0         0   -2.6674    2.2550

>> [UB, DB] = eig(B)

UB =

   -0.8069    0.5907    0.5706   -0.4177
   -0.5907   -0.8069    0.4177    0.5706
         0         0   -0.5706    0.4177
         0         0   -0.4177   -0.5706

DB =
       3λ1       3λ2       2λ1
   -2.0830         0         0         0
         0    6.3114         0         0
         0         0   -1.3887         0
         0         0         0    4.2076
                                     2λ2

>> 3*diag(DA)

ans =
              3λ1
   -2.0830
    6.3114    3λ2

>> 2*diag(DA)

ans =
              2λ1
   -1.3887
    4.2076    2λ2
```

In general, if $A\vec{v} = \lambda\vec{v}$, then

$$\begin{pmatrix} 3A & A \\ 0 & 2A \end{pmatrix}\begin{pmatrix} \vec{v} \\ \vec{0} \end{pmatrix} = \begin{pmatrix} 3A\vec{v} + A\vec{0} \\ 2A\vec{0} \end{pmatrix} = 3\lambda\begin{pmatrix} \vec{v} \\ \vec{0} \end{pmatrix}$$

$$\begin{pmatrix} 3A & A \\ 0 & 2A \end{pmatrix}\begin{pmatrix} \vec{v} \\ -\vec{v} \end{pmatrix} = \begin{pmatrix} 3A\vec{v} - A\vec{v} \\ -2A\vec{v} \end{pmatrix} = 2\lambda\begin{pmatrix} \vec{v} \\ \vec{0} \end{pmatrix}.$$

Thus if A has an eigenvalue λ eigenvector $\vec{v}$, then $\begin{pmatrix} 3A & A \\ 0 & 2A \end{pmatrix}$ has two corresponding e-vectors, with e-values 3λ and 2λ.

SPECTRAL DECOMPOSITION: POWER METHOD

3

(a) What are the eigenvalues and corresponding eigenvectors of

$$A = \begin{pmatrix} 0 & -3 & -2 \\ 2 & 5 & 2 \\ -2 & -3 & 0 \end{pmatrix} ?$$

What is its determinant?

Find a nonsingular matrix $\mathcal{U}$ such that $\mathcal{U}^{-1}A\mathcal{U}$ is a diagonal matrix.

Do this 3 ways:

- By hand, using $\mathrm{Det}(A - \lambda I) = 0, \ldots$
  Show your work, and check your answers.
- In Matlab, using the power method
  In Matlab, using the built-in functions

(b) What are the eigenvalues and corresponding eigenvectors for

$$B = \begin{pmatrix} 1 & -3 & -2 \\ 2 & 6 & 2 \\ -2 & -3 & 1 \end{pmatrix} ?$$

Hint: Use your result from part (a). If this takes more than a line or two to solve, you're doing it wrong...

(a)
$$\det(A-\lambda I) = \det\begin{pmatrix} -\lambda & -3 & -2 \\ 2 & 5-\lambda & 2 \\ -2 & -3 & -\lambda \end{pmatrix}$$

$$= -\lambda\big(\lambda(\lambda-5)+6\big) + 3(-2\lambda+4) - 2\big(-6+2(5-\lambda)\big)$$

$$= -\lambda^3 + 5\lambda^2 - 8\lambda + 4$$

$$= -(\lambda-1)(\lambda-2)^2$$

$\Rightarrow$ Eigenvalues are 1 (multiplicity 1) and 2 (multiplicity 2).

$\Rightarrow \operatorname{Det} A = 1 \cdot 2^2 = 4$

- To find the e-value 1 e-vector, let's use Gaussian elimination (G.E.) to find $N(A-I)$:

$$A - I = \begin{pmatrix} -1 & -3 & -2 \\ 2 & 4 & 2 \\ -2 & -3 & -1 \end{pmatrix} \longrightarrow \begin{pmatrix} -1 & -3 & -2 \\ 0 & -2 & -2 \\ 0 & 3 & 3 \end{pmatrix}$$

$$\Rightarrow N(A-I) = N\left(\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix}\right) = \left\{ \begin{pmatrix} x \\ -x \\ x \end{pmatrix} \,\middle|\, x \in \mathbb{R} \right\}$$

$\Rightarrow$ An eigenvector is $(1,-1,1)$, and it is easy to check this by multiplying by A.

- Next the eigenvalue-2 eigenspace:

$$A - 2I = \begin{pmatrix} -2 & -3 & -2 \\ 2 & 3 & 2 \\ -2 & -3 & -2 \end{pmatrix} \longrightarrow \begin{pmatrix} 2 & 3 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\Rightarrow N(A-2I) = N\big((2\ 3\ 2)\big) = \left\{ x\begin{pmatrix} 3 \\ -2 \\ 0 \end{pmatrix} + y\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \,\middle|\, x,y \in \mathbb{R} \right\}$$

$\Rightarrow$ Two independent e-vectors are

$$\begin{pmatrix} 3 \\ -2 \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}.$$

You can check this by multiplying each by A.

You can check this by multiplying each by A.

For

$$U = \begin{pmatrix} 1 & 3 & 1 \\ -1 & -2 & 0 \\ 1 & 0 & -1 \end{pmatrix},$$

$$A = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} U^{-1}. \checkmark$$

• Next, the same calculation in Matlab:

```
A =

     0    -3    -2
     2     5     2
    -2    -3     0

>> [U,D] = eig(A)

U =

   -0.5774   -0.3333    0.1909
    0.5774    0.6667   -0.6294
   -0.5774   -0.6667    0.7533


D =

    1.0000         0         0
         0    2.0000         0
         0         0    2.0000
```

• Next, using the power method:

```
>> v = randn(3,1);
>> for j = 1:10000
       v = A * v;
       v = v / norm(v);
   end
>> v

v =

    0.4879
    0.2351
   -0.8406

>> A * v ./ v

ans =

     2
     2
     2
```

$\lambda = 2$ e-vector

```
>> w = randn(3,1);
>> w = w - dot(w,v) * v;
>> for j = 1:10000
       w = A * w;
       w = w - dot(w,v) * v;
       w = w / norm(w);
   end
>> w

w =

   -0.7257
    0.6444
   -0.2410

>> A * w ./ w

ans =

    2.0000
    2.0000
    2.0000

>> dot(w,v)

ans =

     0
```

another $\lambda = 2$ e-vector

```
>> x = randn(3,1);
>> B = A - 2*eye(3);
>> for j = 1:10000
       x = B * x;
       x = x / norm(x);
   end
>> A * x ./ x

ans =

    1.0000
    1.0000
    1.0000
```

this finds the $\lambda = 1$ e-vector, using the power method on $A - 2I$

(b) $B = A + I$. Therefore it has the same spectrum as A, just shifted by 1.

(b) B = A + I. Therefore it has the same spectrum as A, just shifted by 1.

Eigenvalues | Eigenvectors

$\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$

$\begin{pmatrix} 3 \\ -2 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$.

4 The following code creates a 100×100 symmetric matrix, each entry of which is uniformly random from $[0, 1]$:

```
rng(1)  % seed the random number generator
        % this way everyone will get the same answer!
n = 100;
A = rand(n,n)
A = diag(diag(A)) + triu(A,1) + triu(A,1)'
    % (triu(A,1) extracts the portion of A strictly above the diagonal,
    % so triu(A,1)' mirrors that below the diagonal)
```

```python
import numpy as np

np.random.seed(1)

n = 100
A = np.random.rand(n, n)
A = np.diag(np.diag(A)) + np.triu(A,1) + np.triu(A,1).T
```

Run this code in Matlab or Octave to initialize A.

(a) Use the power method to compute the eigenvector of A corresponding to the largest-magnitude eigenvalue. Verify that you have indeed computed an eigenvector. What is the eigenvalue? Show your work.

(After you are done, you can check your answer by calling eigs(A,1), but please solve this problem, and the parts (b), (c), (d) below, without using the eig() or eigs() functions.)

```
>> x = randn(n,1);
for j = 1:10000
  x = A * x;
  x = x / norm(x);
end
Ax = A * x;
lambda1 = Ax(1) / x(1)
norm(A*x - lambda1 * x)

lambda1 =

   49.3756

ans =

   7.8442e-15
```

probably more iterations than needed

largest-magnitude eigenvalue

a measure of the error

(b) A is a nonsingular matrix. Now use the power method to compute the smallest-magnitude eigenvalue and a corresponding eigenvector. Of course, show your work.

(Hint: If λ is an eigenvalue of A, then λ^{-1} is an eigenvalue of A^{-1} — so the smallest-magnitude eigenvalues of A correspond to the largest-magnitude eigenvalues of A^{-1}.)

Please do this without computing the inverse matrix A^{-1}. You don't need to compute the LU decomposition of A, but why might doing so speed up your calculations?

```
>> x = randn(n,1);
for j = 1:10000
  x = A \ x;
  x = x / norm(x);
end
Ax = A * x;
lambdasm1 = Ax(1) / x(1)
norm(A*x - lambdasm1 * x)

lambdasm1 =

  -0.0225

ans =

  7.0532e-15
```

instead of $x_j = A^{-1} x_{j-1}$, I just solve $A x_j = x_{j-1}$ for x_j

smallest-magnitude eigenvalue

a measure of the error

Since I am repeatedly solving the same set of equations $Ax=b$, but with different values for b, the LU decomposition would likely speed things up.

ⓒ Finally, use the power method to find the 2^{nd} & 3^{rd} smallest magnitude eigenvalues and corresponding eigenvectors.

(Note: You can check your answer by calling
eigs(A, 3, 'sm')
The 'sm' option tells Matlab to look for the smallest-magnitude eigenvalues. Once again, though, don't use this in your solution. I want you to use the power method!)

There are different ways of doing this, including doing it all in one loop. Here I use two separate loops to compute the eigenvectors (approximately) one at a time:

```
>> y = randn(n,1);
y = y - (x' * y) * x;        <- project ⊥ to first e-vector
for j = 1:10000
  y = A \ y;
  y = y - (x' * y) * x;      <- keep on projecting ⊥ to first e-vector
  y = y / norm(y);              (to control numerical errors)
end
Ay = A * y;
lambdasm2 = Ay(1) / y(1)
norm(Ay - lambdasm2 * y)

lambdasm2 =

   0.0671

ans =

   4.7634e-15    <- error measure

>> z = randn(n,1);
z = z - (x' * z) * x - (y' * z) * y;      <- the same as above, except projecting ⊥ to both the e-vectors already found
for j = 1:10000
  z = A \ z;
  z = z - (x' * z) * x - (y' * z) * y;    <- repeat in every step of the loop to prevent numerical errors from blowing up
  z = z / norm(z);
end
Az = A * z;
lambdasm3 = Az(1) / z(1)
norm(Az - lambdasm3 * z)

lambdasm3 =

  -0.0845

ans =

   2.9029e-15
```