# NumPy: difference between linalg.eig() and linalg.eigh()

Asked 2 years, 8 months ago Active 2 years, 8 months ago Viewed 9k times



13

4

In a Python 3 application I'm using NumPy to calculate eigenvalues and eigenvectors of a symmetric real matrix.

Here's my demo code:

```
import numpy as np
a = np.random.rand(3,3)  # generate a random array shaped (3,3)

a = (a + a.T)/2  # a becomes a random simmetric matrix

evalues1, evectors1 = np.linalg.eig(a)

evalues2, evectors2 = np.linalg.eigh(a)
```

Except for the signs, I got the same eigenvectors and eigenvalues using `np.linalg.eig` and `np.linalg.eigh` . So, what's the difference between the two methods?

Thanks

**EDIT:** I've read the docs here https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.eig.html and here https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.eigh.html but still I can not understand why I should use `eigh()` when I have a symmetric array.

python-3.x numpy

edited Aug 1 '17 at 13:58 asked Aug 1 '17 at 10:10

2 Read the docs? docs.scipy.org/doc/numpy-1.10.1/reference/generated/… – MaxNoe Aug 1 '17 at 10:14

if you look at `evectors1` and `evectors2` - it's not just the sign that got flipped for some elements, the position of some elements have changed too. I would go with the read-the-doc option! (if you on in ipython notebook, just run `??np.linalg.eig` and `??np.linalg.eigh` to see the docs! – Atlas7 Aug 1 '17 at 10:18

Here is a hint: remove/comment line 4 and try again ;) – Feodoran Aug 1 '17 at 10:27

## 1 Answer

Active | Oldest | Votes

24

`eigh` guarantees you that the eigenvalues are sorted and uses a faster algorithm that takes advantage of the fact that the matrix is symmetric. **If you know that your matrix is symmetric, use this function.**
**Attention**, `eigh` doesn't check if your matrix is indeed symmetric, it by default just takes the lower triangular part of the matrix and assumes that the upper triangular part is defined by the symmetry of the matrix.

`eig` works for general matrices and therefore uses a slower algorithm, you can check that for example with IPythons magic command `%timeit` . If you test with larger matrices, you will also see that in general the eigenvalues are not sorted here.

answered Aug 1 '17 at 10:37

Michael H.
**2,034** 1 11 22

1 Excellent answer! – Kaushal28 May 4 '19 at 9:11

Warning: Numpy's documentation of `eig` states that `The eigenvalues are not necessarily ordered.` – kilojoules Nov 25 '19 at 19:29