

Week 3: I can't believe it's not called coding!

Amath 301

TA Session

Today

- 0. Study HW 1,2,3 for exam
- 1. Homework 2 - wrap up questions?
- 2. Numpy eigenvalues/SVD
- 3. Matlab eigenvalues/SVD
- 4. Homework 3 - questions?

Homework 2 wrap up questions?

Numpy Spectral functions

- `np.linalg.eig` Compute eigenvalues & eigenvectors of a non-symmetric matrix
- `np.linalg.eigh` Compute eigenvalues/eigenvectors of a **real symmetric** or **Hermitian** matrix. Uses the matrix structure.
- `np.linalg.eigvals` Just compute eigenvalues. Faster than `eig` if you just don't need eigenvectors
- `np.linalg.eigvalsh` Just compute eigenvalues for a **real symmetric** or **hermitian** matrix. Faster than `eigh` if you just don't need eigenvectors

Demo code

Numpy computes all eigenvalues and eigenvectors unless you tell it otherwise.

```
A = np.asarray([[0,1],[1,1]])
eigenvalues, eigenvectors = np.linalg.eig(A) # Use standard algorithm
eigenvalues_h, eigenvectors_h = np.linalg.eigh(A) # Matrix is symmetric

# If we just want eigenvalues
just_eigvals = np.linalg.eigvals(A) # General algorithm
just_eigvals_h = np.linalg.eigvalsh(A) # Matrix is symmetric
```

If we want the first `k` eigenvalues, we need need to use the `scipy.linalg` versions of `eig` and `eigh` and specify the parameter `subset_by_index` .

Matlab Spectral functions

- `eig` Compute all eigenvalues and eigenvectors. Both returned as matrices.
- `eigs` Returns a subset of eigenvalues & eigenvectors. Add a second argument to specify how many eigenvalues to return (performance benefits for large matrices, `n>1000`)

No special function calls for symmetric matrix.

Demo code

Eig command - gets all eigenvalues/eigenvectors

```
A = [0, 1; 1, 1];  
eigenvalues = eig(A) % Compute a vector of eigenvalues  
[eigenvectors, eigenvaluematrix] = eig(A) % eigenvalues and eigenvectors  
eigenvalues_second_way = diag(eigenvaluematrix) % get values as vector
```

Eigs command - useful for large matrices - return subsets

```
% Eigs  
six_largest_eigenvalues = eigs(A) % up to 6 largest eigenvalues  
% add a second argument for more eigenvalues  
k_largest_eigenvalues = eigs(A, 2) % Get first 2 values  
[eigenvector_mat, eigenvectors_mat] = eigs(A)  
% 6 largest eigenvalues and eigenvectors
```

Numpy SVD

- `np.linalg.svd` Very flexible function to compute SVD. Can compute full SVD, or just singular values.
 - Optional flag for Hermitian/complex-symmetric `A` for performance
 - Singular values returned as a **vector**
 - Option `full_matrices=False` offers Econ mode behavior

Matlab SVD

- `svd` Very flexible function
 - `S = svd(A)` Singular values only
 - `[U, S, V] = svd(A)` Full decomposition
 - `[U, S, V] = svd(A, "econ")` Economy svd: U, V corresponding to nonzero
S

Homework 3 questions?