Chandler Justice - A02313187

November 2, 2023
Suppose we want to define the smallest Eigenvalue:
Input: $A, v_0, \text{tol}, matIter$

while(error $<$ tol && iter $<$ matIter)
{ v1 $=$ 1 / magnitude(y) * y
w $=$ A * v1
$\lambda_1 = v_1^t * w$

error $=$ abs$(\lambda_0 - \lambda_1)$
iter++;
$\lambda_0 = \lambda_1$

$y = v_1$

}
return λ_1, error;
Finding the smallest Eigenvalue: If λ_1 is the largest eigenvector of A, then $\frac{1}{\lambda_1}$ is the smallest eigenvalue of A^{-1}. If λ_n is the smallest eigenvalue of A. the $\frac{1}{n}$ is the largest eigenvalue of A^{-1}

We have the means to compute approximations of λ_1 and λ_n

- λ_1: Use the power method
- λ_n: Use the inverse power method

properties of eigenvalues:

if (λ, v) an eigen pair, $v \neq 0$. this means $Av = \lambda v$. Then for $\mu \in \mathbb{R}$, $(Av - \mu I v) = (A - \mu I)v = \lambda v - \mu I v = (\lambda - \mu)v$
$\rightarrow (\lambda - \mu)$ is an eigenvalue of $(A - \mu I)$.
So, $(A - \mu I)v = (\lambda - \mu)v$

The idea is choose μ close our eigenvalue. We can use inverse iteration to derive an approximation of $\lambda - \mu$ and then add μ back to get λ.

Iteratively finding eigenvalues:

$v_0 = a_1 v_1 + a_2 v_2 + ... + a_n v_n$

$$Av_0 = a_1(\lambda_1 v_1) + a_2(\lambda_2 v_2) + ... + a_n(\lambda_n v_n)$$