Honors Numerical Analysis

Lecture 13

Eigenvalue Algorithms

How do we compute eigenvalues in the middle?

I.e., if $\lambda_1 > \lambda_2 > \lambda_3 > \ldots\ldots > \lambda_{n-1} > \lambda_n$

then Power Method w/ Shift can compute either λ_1 or λ_n.

To compute $\lambda_2 \ldots \lambda_{n-1}$, we need a different idea.

<u>Idea Two</u> Apply the power method to find the eigenvalues of $(A-sI)^{-1}$. This is called the <u>Inverse Power Method with Shift.</u>

If A has eigenvalue λ, then A^{-1} has eigenvalue $\frac{1}{\lambda}$.

$$A\vec{v} = \lambda\vec{v} \implies \frac{1}{\lambda}\vec{v} = A^{-1}\vec{v}$$

Furthermore: $(A-sI)^{-1}$ has eigenvalue $\dfrac{1}{\lambda - s}$.

If we choose s properly to make $\frac{1}{\lambda-s}$ large, then the Inverse Power Method with shift can converge <u>very</u> rapidly.

Choosing s close to λ_ℓ causes $\dfrac{1}{\lambda_\ell - s}$ to become <u>very</u> large in absolute value, while $\dfrac{1}{\lambda_j - s}$ for $j \neq \ell$ remains bounded.

This scheme is of course much more expensive since "applying" A^{-1} requires solving a linear system ($O(n^3)$ vs. $O(n^2)$ flops).

<u>The algorithm</u>

① Set $\vec{w}_0$ to be random.

② Solve $(A-sI)\vec{y}_1 = \vec{w}_0$.

$$\iff \vec{y}_1 = (A-sI)^{-1}\vec{w}_0$$

③ Set $\vec{w}_1 = \vec{y}_1 / \|\vec{y}_1\|$

④ Proceed as in the Power Method.

The hard part is knowing what to choose for s. You need estimates for the eigenvalues.

Both schemes only compute <u>one</u> eigenvalue/vector at a time.

Jacobi's Method

Can we compute all eigenvalues and vectors at the same time.

If A were diagonal, then we immediately know the eigenvalues. Can we make A diagonal?

Recall Similarity transform: $B = M^{-1} A M$ then B is similar to A, i.e. they have the same eigenvalues.

Proof: Look at their characteristic polynomials:

$$p_A(\lambda) = \det(A - \lambda I) \quad \text{degree } n \text{ polynomial}$$

$$\begin{aligned}
p_B(\lambda) &= \det(B - \lambda I) \\
&= \det(M^{-1} A M - \lambda I) \\
&= \det(M^{-1} A M - \lambda M^{-1} M) \\
&= \det(M^{-1}(A - \lambda I) M) \\
&= \det(M^{-1}) \det(A - \lambda I) \det(M) \\
&= p_A(\lambda).
\end{aligned}$$

If M were chosen to be the matrix of eigenvectors of A, then

$$A = M \underset{\text{eigenvalues}}{D} M^{-1} \quad \Rightarrow \quad \underbrace{D = M^{-1} A M}_{\text{Diagonalization of } A.}$$

Ex: Let A be a real symmetric 2×2 matrix.

$$A = \begin{pmatrix} a & b \\ b & d \end{pmatrix} \Rightarrow$$ Eigenvalues are real, and it is diagonalized by an orthogonal matrix V.

All 2x2 orthogonal matrices can be parameterized as:

$$V = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \qquad (\text{2x2 rotation matrix}).$$

We want

$$\underbrace{V^T}_{=V^{-1}} A V = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Write out components for 12, 21 entries of $V^T A V$:

$$\begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \underbrace{\begin{pmatrix} a & b \\ b & d \end{pmatrix} \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix}} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

$$\begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} a\cos\varphi - b\sin\varphi & a\sin\varphi + b\cos\varphi \\ b\cos\varphi - d\sin\varphi & b\sin\varphi + d\cos\varphi \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

$$\Rightarrow \quad a\cos\varphi\sin\varphi + b\cos^2\varphi - b\sin^2\varphi - d\cos\varphi\sin\varphi = 0$$

$$a\cos\varphi\sin\varphi - b\sin^2\varphi + b\cos^2\varphi - d\cos\varphi\sin\varphi = 0$$

Next: Find φ.

<u>Add equations:</u> $(a-d)\cos\varphi\sin\varphi + b(\cos^2\varphi - \sin^2\varphi) = 0$

$$\Rightarrow (a-d)\tfrac{1}{2}\sin 2\varphi + b\cos 2\varphi = 0$$

$$\Rightarrow \tan 2\varphi = \frac{2b}{d-a} \qquad \Rightarrow \qquad \varphi = \frac{1}{2}\operatorname{atan}\left(\frac{2b}{d-a}\right).$$

If $d-a = 0$, then $\frac{2b}{d-a} = \infty$,

$\Rightarrow$ in C or Fortran use

$$\varphi = \tfrac{1}{2}\operatorname{atan2}(d-a, 2b).$$

So we found φ such that $V^T A V = D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.

The Jacobi Method for an $n \times n$ matrix

Define $R^{pq}(\varphi) =$

$$
\begin{array}{cc}
\begin{pmatrix}
1 & & & & & & & \\
& 1 & & & & & & \\
& & 1 & & & & & \\
& & & \cos\varphi & & \sin\varphi & & \\
& & & & \ddots & & & \\
& & & -\sin\varphi & & \cos\varphi & & \\
& & & & & & \ddots & \\
& & & & & & & 1
\end{pmatrix}
& \begin{array}{l} \\ \\ \\ \text{row } p \\ \\ \text{row } q \\ \\ \\ \end{array} \\
\quad\quad\quad\quad \text{column } p \quad\quad \text{column } q. &
\end{array}
$$

$R^{pq}(\varphi)$ can be used to set the pq and qp elements of a real symmetric matrix A to zero.

$R^{pq}(\varphi)^T A \, R^{pq}(\varphi)$ leaves all rows and columns unchanged except for row/column p and row/column q.

The algorithm:

1. Set $A^{(0)} = A$
2. Find pq element in $A^{(k)}$ with maximum absolute value.
3. Compute $\varphi_k = \frac{1}{2} \operatorname{atan}\left(\dfrac{2 a_{pq}^{(k)}}{a_{qq}^{(k)} - a_{pp}^{(k)}} \right)$
4. Set $A^{(k+1)} = R^{pq}(\varphi_k)^T A^{(k)} R^{pq}(\varphi_k)$

Continue this algorithm until all $|a_{pq}^{(k)}| < \epsilon$, $p \neq q$.

Then $A^{(k)} \rightarrow \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$ as $k \rightarrow \infty$

and furthermore: $R^{(k)} = R(\varphi_1) R(\varphi_2) \cdots R(\varphi_k) \rightarrow \overbrace{\left(\vec{v}_1 \ldots \vec{v}_n \right)}^{\text{matrix of eigenvectors.}}$

To summarize:

Apply a sequence of $R(\varphi_k)$'s to A that zero out all off-diagonal elements.

Question: Do elements that were set to zero stay zero forever in the Jacobi method? <u>No</u>

<u>Idea behind convergence</u>: Every $\underbrace{\text{Jacobi rotation}}_{\text{apply } R^T \text{ and } R}$ moves some "mass" of the matrix from off-diagonal positions to diagonal positions.

What can we say about the convergence of Jacobi's Method?

First a Lemma:

<u>Lemma</u>: If R is an orthogonal transformation and $A^T = A$, then

$$\|A\|_F = \|R^T A R\|_F$$

← Frobenius Norm

$$\|A\|_F = \Big(\sum_{ij} |a_{ij}|^2 \Big)^{1/2}$$

<u>Proof</u>: Let $B = R^T A R$. Then A and B have the same eigenvalues, and

$$B^2 = (R^T A R)(R^T A R) = R^T A^2 R.$$

$\Rightarrow$ A^2 and B^2 have the same eigenvalues, and therefore $\operatorname{trace}(A^2) = \operatorname{trace}(B^2)$.

<u>But</u> $\|A\|_F^2 = \operatorname{trace}(A^T A) = \operatorname{trace}(\underbrace{AA}_{A^2}) = \operatorname{trace}(B^2) = \|B\|_F^2$.

(This was proven in Homework.) $\square$.