Finding Eigenvalues: Arnoldi Iteration and the QR Algorithm

Will Wheeler

Algorithms Interest Group

Jul 13, 2016

Outline

Finding the largest eigenvalue

- Largest eigenvalue $\leftarrow$ power method
- So much work for one eigenvector. What about others?
- More eigenvectors $\leftarrow$ orthogonalize Krylov matrix
- build orthogonal list $\leftarrow$ Arnoldi iteration

Solving the eigenvalue problem

- Eigenvalues $\leftarrow$ diagonal of triangular matrix
- Triangular matrix $\leftarrow$ QR algorithm
- QR algorithm $\leftarrow$ QR decomposition
- Better QR decomposition $\leftarrow$ Hessenberg matrix
- Hessenberg matrix $\leftarrow$ Arnoldi algorithm

Power Method

How do we find the largest eigenvalue of an $m \times m$ matrix $\mathbf{A}$?

- Start with a vector $\mathbf{b}$ and make a power sequence:

$$\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \ldots$$

- Higher eigenvalues dominate:

$$\mathbf{A}^n(\mathbf{v}_1 + \mathbf{v}_2) = \lambda_1^n \mathbf{v}_1 + \lambda_2^n \mathbf{v}_2 \approx \lambda_1^n \mathbf{v}_1$$

Vector sequence (normalized) converged?

- Yes: Eigenvector
- No: Iterate some more

Krylov Matrix

Power method throws away information along the way.

- Put the sequence into a matrix

$$K = [\mathbf{b}, \mathbf{Ab}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{n-1}\mathbf{b}] \tag{1}$$
$$= [\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots, \mathbf{x}_1] \tag{2}$$

- Gram-Schmidt approximates first n eigenvectors

$$\mathbf{v}_1 = \mathbf{x}_1 \tag{3}$$
$$\mathbf{v}_2 = \mathbf{x}_2 - \frac{\mathbf{x}_2 \cdot \mathbf{x}_1}{\mathbf{x}_1 \cdot \mathbf{x}_1}\mathbf{x}_1 \tag{4}$$
$$\mathbf{v}_3 = \mathbf{x}_3 - \frac{\mathbf{x}_3 \cdot \mathbf{x}_1}{\mathbf{x}_1 \cdot \mathbf{x}_1}\mathbf{x}_1 - \frac{\mathbf{x}_3 \cdot \mathbf{x}_2}{\mathbf{x}_2 \cdot \mathbf{x}_2}\mathbf{x}_2 \tag{5}$$
$$\ldots \tag{6}$$

- Why not orthogonalize as we build the list?

Arnoldi Iteration

- Start with a normalized vector $\mathbf{q}_1$

$$\mathbf{x}_k = \mathbf{A}\mathbf{q}_{k-1} \qquad \text{(next power)} \tag{7}$$

$$\mathbf{y}_k = \mathbf{x}_k - \sum_{j=1}^{k-1} (\mathbf{q}_j \cdot \mathbf{x}_k)\, \mathbf{q}_j \qquad \text{(Gram-Schmidt)} \tag{8}$$

$$\mathbf{q}_k = \mathbf{y}_k / \left|\mathbf{y}_k\right| \qquad \text{(normalize)} \tag{9}$$

- Orthonormal vectors $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ span Krylov subspace ($\mathcal{K}_n = \text{span}\{\mathbf{q}_1, \mathbf{A}\mathbf{q}_1, \ldots, \mathbf{A}^{n-1}\mathbf{q}_1\}$)
- These are not the eigenvectors of $\mathbf{A}$, but make a similarity (unitary) transformation $\mathbf{H}_n = \mathbf{Q}_n^\dagger \mathbf{A} \mathbf{Q}_n$

Arnoldi Iteration

- Start with a normalized vector $\mathbf{q}_1$

$$\mathbf{x}_k = \mathbf{A}\mathbf{q}_{k-1} \qquad \text{(next power)} \tag{10}$$

$$\mathbf{y}_k = \mathbf{x}_k - \sum_{j=1}^{k-1} (\mathbf{q}_j \cdot \mathbf{x}_k)\, \mathbf{q}_j \qquad \text{(Gram-Schmidt)} \tag{11}$$

$$\mathbf{q}_k = \mathbf{y}_k / |\mathbf{y}_k| \qquad \text{(normalize)} \tag{12}$$

- Orthonormal vectors $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ span Krylov subspace ($\mathcal{K}_n = \text{span}\{\mathbf{q}_1, \mathbf{A}\mathbf{q}_1, \ldots, \mathbf{A}^{n-1}\mathbf{q}_1\}$)
- These are not the eigenvectors of $\mathbf{A}$, but make a similarity (unitary) transformation $\mathbf{H}_n = \mathbf{Q}_n^\dagger \mathbf{A} \mathbf{Q}_n$

Arnoldi Iteration - Construct $\mathbf{H}_n$

We can construct the matrix $\mathbf{H}_n$ along the way.

- For $k \geq j$: $h_{j,k-1} = \mathbf{q}_j \cdot \mathbf{x}_k = \mathbf{q}_j^\dagger A \mathbf{q}_{k-1}$
- For $k = j$: $h_{j,k-1} = |\mathbf{y}_k| \quad$ (equivalent)
- For $k < j$: $h_{j,k-1} = 0$
- $\Rightarrow \mathbf{H}_n$ is upper Hessenberg

$$\mathbf{H}_n = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} & h_{1,6} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} & h_{2,6} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} & h_{3,6} \\ 0 & 0 & h_{4,3} & h_{4,4} & h_{4,5} & h_{4,6} \\ 0 & 0 & 0 & h_{5,4} & h_{5,5} & h_{5,6} \\ 0 & 0 & 0 & 0 & h_{6,5} & h_{6,6} \end{pmatrix} \tag{13}$$

- Why is this useful? $\mathbf{H}_n = \mathbf{Q}_n^\dagger \mathbf{A} \mathbf{Q}_n \rightarrow$ same eigenvalues as $\mathbf{A}$

Finding Eigenvalues

How do we find eigenvalues of a matrix?

- Start with a triangular matrix **A**
- The diagonal elements are the eigenvalues

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ 0 & 0 & a_{33} & a_{34} & a_{35} & a_{36} \\ 0 & 0 & 0 & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & 0 & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{pmatrix} \qquad (14)$$

What if **A** isn't triangular?

QR algorithm

Factorize $\mathbf{A} = \mathbf{QR}$

- $\mathbf{Q}$ is orthonormal (sorry this is a different $\mathbf{Q}$)
- $\mathbf{R}$ is upper triangular (aka right triangular)

Algorithm:

1. Start $\mathbf{A}_1 = \mathbf{A}$
2. Factorize $\mathbf{A}_k = \mathbf{Q}_k\mathbf{R}_k$
3. Construct $\mathbf{A}_{k+1} = \mathbf{R}_k\mathbf{Q}_k$
   $= \mathbf{Q}_k^{-1}\mathbf{Q}_k\mathbf{R}_k\mathbf{Q}_k = \mathbf{Q}_k^{-1}\mathbf{A}_k\mathbf{Q}_k \quad \leftarrow$ similarity transform
4. $\mathbf{A}_k$ converges to upper triangular

QR Algorithm Example (numpy.linalg.qr)

$$\mathbf{A}_1 = \begin{pmatrix} 0.313 & 0.106 & 0.899 \\ 0.381 & 0.979 & 0.375 \\ 0.399 & 0.488 & 0.876 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 1.649 & 0.05 & -0.256 \\ 0.035 & 0.502 & 0.534 \\ -0.014 & 0.004 & 0.017 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} 1.653e+00 & 2.290e-02 & 2.305e-01 \\ 5.966e-03 & 5.063e-01 & -5.342e-01 \\ 8.325e-05 & -9.429e-05 & 9.666e-03 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} 1.653e+00 & 1.872e-02 & -2.285e-01 \\ 1.800e-03 & 5.063e-01 & 5.349e-01 \\ -4.812e-07 & 1.803e-06 & 9.554e-03 \end{pmatrix}$$

QR Decomposition - Gram-Schmidt

How do we get the matrices $\mathbf{Q}$ and $\mathbf{R}$?

http://www.seas.ucla.edu/ vandenbe/103/lectures/qr.pdf

Recursively solve for the first column of $\mathbf{Q}$ and the first row of $\mathbf{R}$:

- $\mathbf{A} = \mathbf{QR}$
- $\begin{bmatrix} \mathbf{a}_1 & \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} r_{11} & \mathbf{R}_{12} \\ 0 & \mathbf{R}_{22} \end{bmatrix}$
- $\begin{bmatrix} \mathbf{a}_1 & \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} r_{11}\mathbf{q}_1 & \mathbf{q}_1\mathbf{R}_{12} + \mathbf{Q}_2\mathbf{R}_{22} \end{bmatrix}$
- $r_{11} = ||\mathbf{a}_1||$, $\mathbf{q}_1 = \mathbf{a}_1 / r_{11}$
- Note $\mathbf{q}_1^\intercal \mathbf{q}_1 = 1$ and $\mathbf{q}_1^\intercal \mathbf{Q}_2 = \mathbf{0}$
- $\mathbf{R}_{12} = \mathbf{q}_1^\intercal \mathbf{A}_2$
- Solve $\mathbf{A}_2 - \mathbf{q}_1\mathbf{R}_{12} = \mathbf{Q}_2\mathbf{R}_{22}$

QR Decomposition of Hessenberg Matrix

What if we have a matrix in upper Hessenberg form?

$$\mathbf{H}_n = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} & h_{1,6} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} & h_{2,6} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} & h_{3,6} \\ 0 & 0 & h_{4,3} & h_{4,4} & h_{4,5} & h_{4,6} \\ 0 & 0 & 0 & h_{5,4} & h_{5,5} & h_{5,6} \\ 0 & 0 & 0 & 0 & h_{6,5} & h_{6,6} \end{pmatrix} \tag{15}$$

We just need to remove the numbers below the diagonal by combining rows $\rightarrow$ Givens rotation.

Givens Rotation

A Givens rotation acts only on two rows and leaves the others unchanged.

$$\mathbf{G}_1 = \begin{pmatrix} \gamma & -\sigma & 0 & 0 & 0 \\ \sigma & \gamma & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{16}$$

Want $\mathbf{G} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}, \quad r = \sqrt{a^2 + b^2}$

- $\gamma = a/\sqrt{a^2 + b^2}$
- $\sigma = -b/\sqrt{a^2 + b^2}$

QR Decomposition of Hessenberg Matrix

$$\mathbf{G}_2\mathbf{G}_1\mathbf{H}_n = \begin{pmatrix} \widetilde{h}_{11} & \widetilde{h}_{12} & \widetilde{h}_{13} & \widetilde{h}_{14} & \widetilde{h}_{15} \\ 0 & \widetilde{h}_{22} & \widetilde{h}_{23} & \widetilde{h}_{24} & \widetilde{h}_{25} \\ 0 & 0 & \widetilde{h}_{33} & \widetilde{h}_{34} & \widetilde{h}_{35} \\ 0 & 0 & h_{43} & h_{44} & h_{45} \\ 0 & 0 & 0 & h_{54} & h_{55} \end{pmatrix} \tag{17}$$

So $\mathbf{H}_n = \mathbf{G}_1^\intercal \mathbf{G}_2^\intercal \ldots \mathbf{G}_{n-1}^\intercal \mathbf{R} = \mathbf{QR}$

QR Decomposition of Hessenberg Matrix

QR algorithm is iterative; is **RQ** also upper Hessenberg?

Yes: Acting on the right, the Givens rotations mix two columns instead of rows, but change the same zeros.

$$\mathbf{G}_2\mathbf{G}_1\mathbf{H}_n = \begin{pmatrix} \widetilde{r}_{11} & \widetilde{r}_{12} & \widetilde{r}_{13} & r_{14} & r_{15} \\ \widetilde{r}_{21} & \widetilde{r}_{22} & \widetilde{r}_{23} & r_{24} & r_{25} \\ 0 & \widetilde{r}_{32} & \widetilde{r}_{33} & r_{34} & r_{35} \\ 0 & 0 & 0 & r_{44} & r_{45} \\ 0 & 0 & 0 & 0 & r_{55} \end{pmatrix} \tag{18}$$

Householder Reflections

- Arnoldi finds a few eigenvalues of a large matrix
- In practice, QR uses Householder reflections
- www.math.usm.edu/lambers/mat610/sum10/lecture9.pdf

Reflection across the plane perpendicular to a unit vector $\mathbf{v}$:

$$\mathbf{P} = I - 2\mathbf{v}\mathbf{v}^{\mathsf{T}} \tag{19}$$

$$\mathbf{P}\mathbf{x} = \mathbf{x} - 2\mathbf{v}\mathbf{v}^{\mathsf{T}}\mathbf{x} \tag{20}$$

$$\mathbf{P}\mathbf{x} - \mathbf{x} = -\mathbf{v}(2\mathbf{v}^{\mathsf{T}}\mathbf{x}) \tag{21}$$

Want to arrange first column into zeros: find $\mathbf{v}$ so that $\mathbf{P}\mathbf{x} = \alpha\mathbf{e}_1$.

Householder Reflections

Want to arrange first column into zeros: find $\mathbf{v}$ so that $\mathbf{Px} = \alpha \mathbf{e}_1$.

$$||\mathbf{x}|| = \alpha \tag{22}$$

$$\mathbf{x} - 2\mathbf{v}\mathbf{v}^\mathsf{T}\mathbf{x} = \alpha \mathbf{e}_1 \tag{23}$$

$$\frac{1}{2}(\mathbf{x} - \alpha \mathbf{e}_1) = \mathbf{v}(\mathbf{v}^\mathsf{T}\mathbf{x}) \tag{24}$$

So we know

- $\mathbf{v} \propto \mathbf{x} - \alpha \mathbf{e}_1$
- $\mathbf{v}$ is a unit vector

$$\mathbf{v} = \frac{\mathbf{x} - \alpha \mathbf{e}_1}{||\mathbf{x} - \alpha \mathbf{e}_1||} \tag{25}$$

Lanczos

What if we apply the Arnoldi algorithm to a Hermitian matrix?

- Start with a normalized vector $\mathbf{q}_1$

$$\mathbf{x}_k = \mathbf{A}\mathbf{q}_{k-1} \qquad \text{(next power)} \tag{26}$$

$$\mathbf{y}_k = \mathbf{x}_k - \sum_{j=k-2}^{k-1} (\mathbf{q}_j \cdot \mathbf{x}_k)\, \mathbf{q}_j \qquad \text{(Gram-Schmidt)} \tag{27}$$

$$\mathbf{q}_k = \mathbf{y}_k / |\mathbf{y}_k| \qquad \text{(normalize)} \tag{28}$$

Since $\mathbf{H}_n = \mathbf{Q}_n^\dagger \mathbf{A} \mathbf{Q}_n$,

- $\mathbf{H}_n$ is also Hermitian
- Entries of $\mathbf{H}_n$ above the first diagonal are 0 (tridiagonal)
- We don't need to calculate them!

Lanczos

What if we apply the Arnoldi algorithm to a Hermitian matrix?
Since $\mathbf{H}_n = \mathbf{Q}_n^\dagger \mathbf{A} \mathbf{Q}_n$,

- $\mathbf{H}_n$ is also Hermitian
- Entries of $\mathbf{H}_n$ above the first diagonal are 0 (tridiagonal)
- We don't need to calculate them!

$$\mathbf{H}_n = \begin{pmatrix} h_{11} & h_{21}^* & 0 & 0 & 0 \\ h_{21} & h_{22} & h_{32}^* & 0 & 0 \\ 0 & h_{32} & h_{33} & h_{43}^* & 0 \\ 0 & 0 & h_{43} & h_{44} & h_{54}^* \\ 0 & 0 & 0 & h_{54} & h_{55} \end{pmatrix} \tag{29}$$