AMS526: Numerical Analysis I
(Numerical Linear Algebra)
Lecture 16: Rayleigh Quotient Iteration

Xiangmin Jiao

SUNY Stony Brook
Solving Eigenvalue Problems

- All eigenvalue solvers must be iterative
- Iterative algorithms have multiple facets:
  1. Basic idea behind the algorithms
  2. Convergence and techniques to speed-up convergence
  3. Efficiency of implementation
  4. Termination criteria
- We will focus on first two aspects
We will consider eigenvalue problems for real symmetric matrices, i.e. \( A = A^T \in \mathbb{R}^{m \times m} \), and \( Ax = \lambda x \) for \( x \in \mathbb{R}^m \)

- Note: \( x^* = x^T \), and \( \|x\| = \sqrt{x^T x} \)

- \( A \) has real eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_m \) and orthonormal eigenvectors \( q_1, q_2, \ldots, q_m \), where \( \|q_j\| = 1 \)

- Eigenvalues are often also ordered in a particular way (e.g., ordered from large to small in magnitude)

- In addition, we focus on symmetric tridiagonal form

  - Why? Because phase 1 of two-phase algorithm reduces matrix into tridiagonal form
The Rayleigh quotient of $x \in \mathbb{R}^m$ is the scalar

$$r(x) = \frac{x^T A x}{x^T x}$$

For an eigenvector $x$, its Rayleigh quotient is $r(x) = x^T \lambda x / x^T x = \lambda$, the corresponding eigenvalue of $x$.

For general $x$, $r(x) = \alpha$ that minimizes $\|A x - \alpha x\|_2$.

$x$ is eigenvector of $A \iff \nabla r(x) = \frac{2}{x^T x} (A x - r(x)x) = 0$ with $x \neq 0$.

$r(x)$ is smooth and $\nabla r(q_J) = 0$ for any $j$, and therefore is quadratically accurate:

$$r(x) - r(q_J) = O(\|x - q_J\|^2) \text{ as } x \to q_J \text{ for some } J$$
Power Iteration

- Simple power iteration for largest eigenvalue

Algorithm: Power Iteration

\[
\begin{align*}
\mathbf{v}^{(0)} &= \text{some unit-length vector} \\
\text{for } k &= 1, 2, \ldots \\
\mathbf{w} &= A\mathbf{v}^{(k-1)} \\
\mathbf{v}^{(k)} &= \frac{\mathbf{w}}{\|\mathbf{w}\|} \\
\lambda^{(k)} &= r(\mathbf{v}^{(k)}) = (\mathbf{v}^{(k)})^T A \mathbf{v}^{(k)}
\end{align*}
\]

- Termination condition is omitted for simplicity
Convergence of Power Iteration

- Expand initial $\mathbf{v}^{(0)}$ in orthonormal eigenvectors $\mathbf{q}_i$, and apply $A^k$:

\[
\mathbf{v}^{(0)} = a_1 \mathbf{q}_1 + a_2 \mathbf{q}_2 + \cdots + a_m \mathbf{q}_m
\]

\[
\mathbf{v}^{(k)} = c_k A^k \mathbf{v}^{(0)}
\]

\[
= c_k (a_1 \lambda_1^k \mathbf{q}_1 + a_2 \lambda_2^k \mathbf{q}_2 + \cdots + a_m \lambda_m^k \mathbf{q}_m)
\]

\[
= c_k \lambda_1^k (a_1 \mathbf{q}_1 + a_2 (\lambda_2/\lambda_1)^k \mathbf{q}_2 + \cdots + a_m (\lambda_m/\lambda_1)^k \mathbf{q}_m)
\]

- If $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$ and $\mathbf{q}_1^T \mathbf{v}^{(0)} \neq 0$, this gives

\[
\|\mathbf{v}^{(k)} - (\pm \mathbf{q}_1)\| = O\left(|\lambda_2/\lambda_1|^k\right), \quad |\lambda^{(k)} - \lambda_1| = O\left(|\lambda_2/\lambda_1|^{2k}\right)
\]

where $\pm$ sign is chosen to be sign of $\mathbf{q}_1^T \mathbf{v}^{(k)}$

- It finds the largest eigenvalue (unless eigenvector is orthogonal to $\mathbf{v}^{(0)}$)

- Error reduces by only a constant factor ($\approx |\lambda_2/\lambda_1|$) each step, and very slowly especially when $|\lambda_2| \approx |\lambda_1|$
Inverse Iteration

- Apply power iteration on \((A - \mu I)^{-1}\), with eigenvalues \(\{(\lambda_j - \mu)^{-1}\}\)
- If \(\mu \approx \lambda_J\) for some \(J\), then \((\lambda_J - \mu)^{-1}\) may be far larger than \((\lambda_j - \mu)^{-1}, j \neq J\), so power iteration may converge rapidly

Algorithm: Inverse Iteration

- \(v^{(0)} = \)some unit-length vector
- for \(k = 1, 2, \ldots\)
- Solve \((A - \mu I)w = v^{(k-1)}\) for \(w\)
- \(v^{(k)} = w / \|w\|\)
- \(\lambda^{(k)} = r(v^{(k)}) = (v^{(k)})^T Av^{(k)}\)

- Converges to eigenvector \(q_J\) if parameter \(\mu\) is close to \(\lambda_J\)

\[
\|v^{(k)} - (\pm q_J)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \quad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right)
\]

where \(\lambda_J\) and \(\lambda_K\) are closest and second closest eigenvalues to \(\mu\)

- Standard method for determining eigenvector given eigenvalue
Rayleigh Quotient Iteration

• Parameter $\mu$ is constant in inverse iteration, but convergence is better for $\mu$ close to the eigenvalue

• Improvement: At each iteration, set $\mu$ to last computed Rayleigh quotient

<table>
<thead>
<tr>
<th>Algorithm: Rayleigh Quotient Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v^{(0)} = \text{some unit-length vector}$</td>
</tr>
<tr>
<td>$\lambda^{(0)} = r(v^{(0)}) = (v^{(0)})^T Av^{(0)}$</td>
</tr>
<tr>
<td>for $k = 1, 2, \ldots$</td>
</tr>
<tr>
<td>Solve $(A - \lambda^{(k-1)} I)w = v^{(k-1)}$ for $w$</td>
</tr>
<tr>
<td>$v^{(k)} = w/|w|$</td>
</tr>
<tr>
<td>$\lambda^{(k)} = r(v^{(k)}) = (v^{(k)})^T Av^{(k)}$</td>
</tr>
</tbody>
</table>

• Cost per iteration is linear for tridiagonal matrix
Convergence of Rayleigh Quotient Iteration

- Cubic convergence in Rayleigh quotient iteration

\[ \| v^{(k+1)} - (\pm q_J) \| = O(\| v^{(k)} - (\pm q_J) \|^3) \]

and

\[ |\lambda^{(k+1)} - \lambda_J| = O\left(|\lambda^{(k)} - \lambda_J|^3\right) \]

- In other words, each iteration triples number of digits of accuracy

- Proof idea: If \( v^{(k)} \) is close to an eigenvector, \( \| v^{(k)} - (\pm q_J) \| \leq \epsilon \), then accuracy of Rayleigh quotient estimate \( \lambda^{(k)} \) is \( |\lambda^{(k)} - \lambda_J| = O(\epsilon^2) \). One step of inverse iteration then gives

\[ \| v^{(k+1)} - q_J \| = O( |\lambda^{(k)} - \lambda_J| \| v^{(k)} - q_J \| ) = O(\epsilon^3) \]

- Rayleigh quotient is great in finding largest (or smallest) eigenvalue and its corresponding eigenvector. What if we want to find all eigenvalues?
Operation Counts

In Rayleigh quotient iteration,

- if $A \in \mathbb{R}^{m \times m}$ is full matrix, then solving $(A - \mu I)w = v^{(k-1)}$ may take $O(m^3)$ flops per step
- if $A \in \mathbb{R}^{m \times m}$ is upper Hessenberg, then each step takes $O(m^2)$ flops
- if $A \in \mathbb{R}^{m \times m}$ is tridiagonal, then each step takes $O(m)$ flops