Lecture 15
The QR Algorithm I

MIT 18.335J / 6.337J

Introduction to Numerical Methods

Per-Olof Persson

October 31, 2006

Real Symmetric Matrices

- We will only consider eigenvalue problems for real symmetric matrices
- Then $A = A^T \in \mathbb{R}^{m \times m}$, $x \in \mathbb{R}^m$, $x^* = x^T$, and $\|x\| = \sqrt{x^T x}$
- A then also has

$$\text{real eigenvalues: } \lambda_1, \ldots, \lambda_m$$

$$\text{orthonormal eigenvectors: } q_1, \ldots, q_m$$

- Eigenvectors are normalized $\|q_j\| = 1$, and sometimes the eigenvalues are ordered in a particular way
- Initial reduction to tridiagonal form assumed
  - Brings cost for typical steps down from $O(m^3)$ to $O(m)$

Rayleigh Quotient

- The *Rayleigh quotient* of $x \in \mathbb{R}^m$:

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

- For an eigenvector x, the corresponding eigenvalue is $r(x) = \lambda$
- For general x, $r(x) = \alpha$ that minimizes $\|Ax - \alpha x\|_2$
- x eigenvector of $A \Longleftrightarrow \nabla r(x) = 0$ with $x \neq 0$
- $r(x)$ is smooth and $\nabla r(q_j) = 0$, therefore quadratically accurate:

$$r(x) - r(q_J) = O(\|x - q_J\|^2) \text{ as } x \rightarrow q_J$$

Power Iteration

- Simple power iteration for largest eigenvalue:

Algorithm: Power Iteration

$v^{(0)} =$ some vector with $\|v^{(0)}\| = 1$

for $k = 1, 2, \ldots$

$w = Av^{(k-1)}$	apply A
$v^{(k)} = w/\|w\|$	normalize
$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$	Rayleigh quotient

- Termination conditions usually omitted

Convergence of Power Iteration

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

$$\begin{aligned} v^{(0)} &= a_1 q_1 + a_2 q_2 + \cdots + a_m q_m \\ v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k (a_1 \lambda_1^k q_1 + a_2 \lambda_2^k q_2 + \cdots + a_m \lambda_m^k q_m) \\ &= c_k \lambda_1^k (a_1 q_1 + a_2 (\lambda_2/\lambda_1)^k q_2 + \cdots + a_m (\lambda_m/\lambda_1)^k q_m) \end{aligned}$$

- If $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$ and $q_1^T v^{(0)} \neq 0$, this gives:

$$\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

- Finds the largest eigenvalue (unless eigenvector orthogonal to $v^{(0)}$)
- Linear convergence, factor $\approx \lambda_2/\lambda_1$ at each iteration

Inverse Iteration

- Apply power iteration on $(A - \mu I)^{-1}$, with eigenvalues $(\lambda_j - \mu)^{-1}$

Algorithm: Inverse Iteration

$v^{(0)} =$ some vector with $\|v^{(0)}\| = 1$

for $k = 1, 2, \ldots$

  Solve $(A - \mu I)w = v^{(k-1)}$ for w — apply $(A - \mu I)^{-1}$

  $v^{(k)} = w / \|w\|$ — normalize

  $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleigh quotient

- Converges to eigenvector q_J if the parameter μ is close to λ_J:

$$\|v^{(k)} - (\pm q_j)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \qquad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right)$$

Rayleigh Quotient Iteration

- Parameter μ is constant in inverse iteration, but convergence is better for μ close to the eigenvalue

- Improvement: At each iteration, set μ to last computed Rayleigh quotient

Algorithm: Rayleigh Quotient Iteration

$v^{(0)} =$ some vector with $\|v^{(0)}\| = 1$

$\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$ = corresponding Rayleigh quotient

for $k = 1, 2, \ldots$

Solve $(A - \lambda^{(k-1)} I)w = v^{(k-1)}$ for w — apply matrix

$v^{(k)} = w / \|w\|$ — normalize

$\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ — Rayleigh quotient

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

$$|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3)$$

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

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

The QR Algorithm

- Remarkably simple algorithm: QR factorize and multiply in reverse order:

Algorithm: "Pure" QR Algorithm

$A^{(0)} = A$

for $k = 1, 2, \ldots$

$Q^{(k)} R^{(k)} = A^{(k-1)}$ QR factorization of $A^{(k-1)}$

$A^{(k)} = R^{(k)} Q^{(k)}$ Recombine factors in reverse order

- With some assumptions, $A^{(k)}$ converge to a Schur form for A (diagonal if A symmetric)
- Similarity transformations of A:

$$A^{(k)} = R^{(k)} Q^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$$

Unnormalized Simultaneous Iteration

- To understand the QR algorithm, first consider a simpler algorithm
- *Simultaneous Iteration* is power iteration applied to several vectors
- Start with linearly independent $v_1^{(0)}, \ldots, v_n^{(0)}$
- We know from power iteration that $A^k v_1^{(0)}$ converges to q_1
- With some assumptions, the space $\langle A^k v_1^{(0)}, \ldots, A^k v_n^{(0)} \rangle$ should converge to $q_1, \ldots, q_n$
- Notation: Define initial matrix $V^{(0)}$ and matrix $V^{(k)}$ at step k:

$$V^{(0)} = \left[\begin{array}{c|c|c} & & \\ v_1^{(0)} & \cdots & v_n^{(0)} \\ & & \end{array} \right], \quad V^{(k)} = A^k V^{(0)} = \left[\begin{array}{c|c|c} & & \\ v_1^{(k)} & \cdots & v_n^{(k)} \\ & & \end{array} \right]$$

Unnormalized Simultaneous Iteration

- Define well-behaved basis for column space of $V^{(k)}$ by $\hat{Q}^{(k)}\hat{R}^{(k)} = V^{(k)}$
- Make the assumptions:
  - The leading $n+1$ eigenvalues are distinct
  - All principal leading principal submatrices of $\hat{Q}^T V^{(0)}$ are nonsingular, where columns of $\hat{Q}$ are $q_1, \ldots, q_n$

  We then have that the columns of $\hat{Q}^{(k)}$ converge to eigenvectors of A:

$$\|q_j^{(k)} - \pm q_j\| = O(C^k)$$

  where $C = \max_{1 \leq k \leq n} |\lambda_{k+1}|/|\lambda_k|$

- *Proof.* Textbook / Black board

Simultaneous Iteration

- The matrices $V^{(k)} = A^k V^{(0)}$ are highly ill-conditioned
- Orthonormalize at each step rather than at the end:

Algorithm: Simultaneous Iteration

Pick $\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$

for $k = 1, 2, \ldots$

$Z = A\hat{Q}^{(k-1)}$

$\hat{Q}^{(k)}\hat{R}^{(k)} = Z$ Reduced QR factorization of Z

- The column spaces of $\hat{Q}^{(k)}$ and $Z^{(k)}$ are both equal to the column space of $A^k\hat{Q}^{(0)}$, therefore same convergence as before

Simultaneous Iteration $\Longleftrightarrow$ QR Algorithm

- The QR algorithm is equivalent to simultaneous iteration with $\hat{Q}^{(0)} = I$
- Notation: Replace $\hat{R}^{(k)}$ by $R^{(k)}$, and $\hat{Q}^{(k)}$ by $\underline{Q}^{(k)}$

Simultaneous Iteration:

$$\underline{Q}^{(0)} = I$$
$$Z = A\underline{Q}^{(k-1)}$$
$$Z = \underline{Q}^{(k)} R^{(k)}$$
$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$$

Unshifted QR Algorithm:

$$A^{(0)} = A$$
$$A^{(k-1)} = Q^{(k)} R^{(k)}$$
$$A^{(k)} = R^{(k)} Q^{(k)}$$
$$\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \cdots Q^{(k)}$$

- Also define $\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \cdots R^{(1)}$
- Now show that the two processes generate same sequences of matrices

Simultaneous Iteration $\Longleftrightarrow$ QR Algorithm

- Both schemes generate the QR factorization $A^k = \underline{Q}^{(k)}\underline{R}^{(k)}$ and the projection $A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$
- *Proof.* $k = 0$ trivial for both algorithms.

  For $k \geq 1$ with simultaneous iteration, $A^{(k)}$ is given by definition, and

$$A^k = A\underline{Q}^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k)}R^{(k)}\underline{R}^{(k-1)} = \underline{Q}^{(k)}\underline{R}^{(k)}$$

  For $k \geq 1$ with unshifted QR, we have

$$A^k = A\underline{Q}^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k-1)}A^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k)}\underline{R}^{(k)}$$

  and

$$A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}$$

18.335J / 6.337J Introduction to Numerical Methods
Fall 2010