

QR Algorithm

Katja Grasic

May 15, 2008

Introduction

Definitions

The Power Method

QR algorithm without shifts

Hessenberg form

QR algorithm with shifts

Implicit shifted QR algorithm

Introduction

- ▶ QR algorithm most widely used method for computing eigenvalues.

Introduction

- ▶ QR algorithm most widely used method for computing eigenvalues.
- ▶ The name QR is derived from the use of the letter Q to denote orthogonal matrices and the letter R to denote right triangular matrix.

Triangular matrix

Lower triangular matrix:

$$\begin{bmatrix} l_{1,1} & & & & & 0 \\ l_{2,1} & l_{2,2} & & & & \\ l_{3,1} & l_{3,2} & \ddots & & & \\ \vdots & \vdots & \ddots & \ddots & & \\ l_{n,1} & l_{n,2} & \dots & l_{n,n-1} & l_{n,n} & \end{bmatrix}$$

Triangular matrix

Upper triangular matrix:

$$\begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \dots & u_{2,n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & u_{n-1,n} \\ 0 & & & & u_{n,n} \end{bmatrix}$$

Unitary matrix

- ▶ Satisfying condition:

$$T^T T = T T^T = I_n$$

Unitary matrix

- ▶ Satisfying condition:

$$T^T T = T T^T = I_n$$

- ▶ $T^{-1} = T^T$

Schur decomposition

The Schur decomposition reads as follows: if A is an $n \times n$ square matrix with complex entries, then A can be expressed as

$$A = QTQ^*$$

where Q is a unitary matrix, Q^* is the conjugate transpose of Q , and T is an upper triangular matrix called the **Schur form**. The diagonal entries of T are exactly the eigenvalues of A .

Similar matrix

Two $n \times n$ matrices A and B are called similar if there exists invertible $n \times n$ matrix P , so that:

$$P^{-1}AP = B$$

Similar matrices share many properties:

- ▶ rank

Similar matrix

Two $n \times n$ matrices A and B are called similar if there exists invertible $n \times n$ matrix P , so that:

$$P^{-1}AP = B$$

Similar matrices share many properties:

- ▶ rank
- ▶ determinant

Similar matrix

Two $n \times n$ matrices A and B are called similar if there exists invertible $n \times n$ matrix P , so that:

$$P^{-1}AP = B$$

Similar matrices share many properties:

- ▶ rank
- ▶ determinant
- ▶ trace

Similar matrix

Two $n \times n$ matrices A and B are called similar if there exists invertible $n \times n$ matrix P , so that:

$$P^{-1}AP = B$$

Similar matrices share many properties:

- ▶ rank
- ▶ determinant
- ▶ trace
- ▶ eigenvalues (though the eigenvectors will in general be different)

Similar matrix

Two $n \times n$ matrices A and B are called similar if there exists invertible $n \times n$ matrix P , so that:

$$P^{-1}AP = B$$

Similar matrices share many properties:

- ▶ rank
- ▶ determinant
- ▶ trace
- ▶ eigenvalues (though the eigenvectors will in general be different)
- ▶ characteristic polynomial

The Power Method

- ▶ Let A be a matrix with a complete set of eigenvalues and eigenvectors pairs (λ_i, q_i)

The Power Method

- ▶ Let A be a matrix with a complete set of eigenvalues and eigenvectors pairs (λ_i, q_i)
- ▶ Assume the eigenvalues can be ordered:
 $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$

The Power Method

- ▶ Let A be a matrix with a complete set of eigenvalues and eigenvectors pairs (λ_i, q_i)
- ▶ Assume the eigenvalues can be ordered:
 $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$
- ▶ Then for any vector u , $A^k u = \gamma_1 \lambda_1^k q_1 + \sum_i \gamma_i \lambda_i^k q_i$

The Power Method

- ▶ Let A be a matrix with a complete set of eigenvalues and eigenvectors pairs (λ_i, q_i)
- ▶ Assume the eigenvalues can be ordered:
 $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$
- ▶ Then for any vector u , $A^k u = \gamma_1 \lambda_1^k q_1 + \sum_i \gamma_i \lambda_i^k q_i$
- ▶ As $k \rightarrow \infty$, the term containing λ_1 will dominate and $A^k u$ approaches a multiple of the dominant eigenvector q_1 .

The Power Iteration

Choose $v^{(0)}$ such that $\|v^{(0)}\| = 1$, then for $k = 1, 2, \dots$ we do the following steps:

▶ $w = Av^{k-1}$

The Power Iteration

Choose $v^{(0)}$ such that $\|v^{(0)}\| = 1$, then for $k = 1, 2, \dots$ we do the following steps:

- ▶ $w = Av^{k-1}$
- ▶ $v^{(k)} = \frac{w}{\|w\|}$

The Power Iteration

Choose $v^{(0)}$ such that $\|v^{(0)}\| = 1$, then for $k = 1, 2, \dots$ we do the following steps:

- ▶ $w = Av^{k-1}$
- ▶ $v^{(k)} = \frac{w}{\|w\|}$
- ▶ $\lambda^k = (v^{(k)})^T Av^{(k)}$.

QR decomposition

A QR decomposition of a real square matrix A is a decomposition of A as

$$A = QR$$

- ▶ Q is an orthogonal matrix (meaning that $Q^T Q = I$)

QR decomposition

A QR decomposition of a real square matrix A is a decomposition of A as

$$A = QR$$

- ▶ Q is an orthogonal matrix (meaning that $Q^T Q = I$)
- ▶ R is an upper triangular matrix
- ▶ If A nonsingular, then this factorization is unique if we require that the diagonal elements of R are positive.

QR algorithm

▶ $A^{(0)} = A$

QR algorithm

- ▶ $A^{(0)} = A$
- ▶ for $k = 1, 2, 3, \dots$:

QR algorithm

- ▶ $A^{(0)} = A$
- ▶ for $k = 1, 2, 3, \dots$:
- ▶ $Q^{(k)}R^{(k)} = A^{(k-1)}$

QR algorithm

- ▶ $A^{(0)} = A$
- ▶ for $k = 1, 2, 3, \dots$:
- ▶ $Q^{(k)}R^{(k)} = A^{(k-1)}$
- ▶ $A^{(k)} = R^{(k)}Q^{(k)}$

QR algorithm

- ▶ $A^{(0)} = A$
- ▶ for $k = 1, 2, 3, \dots$:
- ▶ $Q^{(k)}R^{(k)} = A^{(k-1)}$
- ▶ $A^{(k)} = R^{(k)}Q^{(k)}$
- ▶ With some assumptions, $A^{(k)}$ converge to a Shur form of A (diagonal if A is symmetric)

QR algorithm

- ▶ To understand the QR algorithm, first consider a simpler algorithm

QR algorithm

- ▶ To understand the QR algorithm, first consider a simpler algorithm
- ▶ Simultaneous Iteration is power iteration applied to several vectors

QR algorithm

- ▶ To understand the QR algorithm, first consider a simpler algorithm
- ▶ Simultaneous Iteration is power iteration applied to several vectors
- ▶ Start with linearly independent $v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)}$

QR algorithm

- ▶ To understand the QR algorithm, first consider a simpler algorithm
- ▶ Simultaneous Iteration is power iteration applied to several vectors
- ▶ Start with linearly independent $v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)}$
- ▶ We know from power iteration that $A^k v_1^{(0)}$ converges to q_1

QR algorithm

- ▶ To understand the QR algorithm, first consider a simpler algorithm
- ▶ Simultaneous Iteration is power iteration applied to several vectors
- ▶ Start with linearly independent $v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)}$
- ▶ We know from power iteration that $A^k v_1^{(0)}$ converges to q_1
- ▶ With some assumptions, the space $\langle A^k v_1^{(0)}, A^k v_2^{(0)}, \dots, A^k v_n^{(0)} \rangle$ should converge to q_1, \dots, q_n

QR algorithm

- ▶ To understand the QR algorithm, first consider a simpler algorithm
- ▶ Simultaneous Iteration is power iteration applied to several vectors
- ▶ Start with linearly independent $v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)}$
- ▶ We know from power iteration that $A^k v_1^{(0)}$ converges to q_1
- ▶ With some assumptions, the space $\langle A^k v_1^{(0)}, A^k v_2^{(0)}, \dots, A^k v_n^{(0)} \rangle$ should converge to q_1, \dots, q_n
- ▶ Notation: Define initial matrix $V^{(0)}$ and $V^{(k)}$ at step k :

$$V^{(0)} = \left[v_1^{(0)} \mid v_2^{(0)} \mid \dots \mid v_n^{(0)} \right], \quad V^{(k)} = A^{(k)} V^{(0)} = \left[v_1^{(k)} \mid v_2^{(k)} \mid \dots \mid v_n^{(k)} \right]$$

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

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

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, \dots, q_n

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

Simultaneous Iteration

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

- ▶ Pick $\hat{Q}^{(0)}$
- ▶ for $k = 1, 2, \dots$:
- ▶ $Z = A\hat{Q}^{(k-1)}$

Simultaneous Iteration

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

- ▶ Pick $\hat{Q}^{(0)}$
- ▶ for $k = 1, 2, \dots$:
- ▶ $Z = A\hat{Q}^{(k-1)}$

Simultaneous Iteration

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

- ▶ Pick $\hat{Q}^{(0)}$
- ▶ for $k = 1, 2, \dots$:
- ▶ $Z = A\hat{Q}^{(k-1)}$

Simultaneous Iteration

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

- ▶ Pick $\hat{Q}^{(0)}$
- ▶ for $k = 1, 2, \dots$:
- ▶ $Z = A\hat{Q}^{(k-1)}$
- ▶ $\hat{Q}^{(k)}\hat{R}^{(k)} = Z$

Simultaneous Iteration

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

- ▶ Pick $\hat{Q}^{(0)}$
- ▶ for $k = 1, 2, \dots$:
- ▶ $Z = A\hat{Q}^{(k-1)}$
- ▶ $\hat{Q}^{(k)}\hat{R}^{(k)} = Z$
- ▶ The QR algorithm is equivalent to simultaneous iteration with $Q^{(0)} = I$

Hessenberg form

- ▶ It is now to focus on the cost of the QR algorithm.

Hessenberg form

- ▶ It is now to focus on the cost of the QR algorithm.
- ▶ This is prohibitively expensive as usually a large number of iterations is required to attain convergence.

Hessenberg form

- ▶ It is now to focus on the cost of the QR algorithm.
- ▶ This is prohibitively expensive as usually a large number of iterations is required to attain convergence.
- ▶ The cure to this problem is the transformation of the matrix to upper-Hessenberg form H . When this is done the cost/iteration reduces to $O(n^2)$.

Householder reflections

We define a Householder reflection as:

$$H = I - uu^*,$$

where $\|u\| = \sqrt{2}$.

Hessenberg form via Householder reflection

$$A = \begin{pmatrix} \alpha_{11} & a_{12} \\ a_{21} & A_{22} \end{pmatrix},$$

choose the householder reflection H_1 such that $H_1 a_{21} = \nu_1 e_1$ then

$$H_1 A H_1 = \begin{pmatrix} \alpha_{11} & a_{12} H_1 \\ \nu_1 e_1 & H_1 A_{22} H_1 \end{pmatrix},$$

thus we have annihilated the elements below the first column via a similarity transformation. We can repeat this process for all steps to obtain an upper Hessenberg form.

QR algorithm with shifts

The explicit QR algorithm with shifts can be written as follows for any matrix A . Let $A_0 = A$, and let $k = 1, 2, 3, \dots$ then given shifts κ_j

$$\blacktriangleright A_k - \kappa_k I = Q_k R_k$$

Notice that the iterates satisfy

$$A_{k+1} = R_k Q_k + \kappa_k I = Q_k^* (A_k - \kappa_k I) Q_k + \kappa_k I = Q_k^* A Q_k,$$

and therefore are related through a similarity transformation.

QR algorithm with shifts

The explicit QR algorithm with shifts can be written as follows for any matrix A . Let $A_0 = A$, and let $k = 1, 2, 3, \dots$ then given shifts κ_j

- ▶ $A_k - \kappa_k I = Q_k R_k$
- ▶ $A_{k+1} = R_k Q_k + \kappa_k I$

Notice that the iterates satisfy

$$A_{k+1} = R_k Q_k + \kappa_k I = Q_k^* (A_k - \kappa_k I) Q_k + \kappa_k I = Q_k^* A Q_k,$$

and therefore are related through a similarity transformation.

Outline of the implicit shifted QR algorithm

- ▶ We determine the first column c of $C = H^2 - 2\operatorname{Re}(\kappa)H + |\kappa|^2 I$.
- ▶ Now let Q_0 be a Householder transformation such that $Q_0^* c = \sigma e_1$.

Outline of the implicit shifted QR algorithm

- ▶ We determine the first column c of $C = H^2 - 2\operatorname{Re}(\kappa)H + |\kappa|^2 I$.
- ▶ Now let Q_0 be a Householder transformation such that $Q_0^*c = \sigma e_1$.
- ▶ Next let $H_1 = Q_0^*AQ_0$.

Outline of the implicit shifted QR algorithm

- ▶ We determine the first column c of $C = H^2 - 2\operatorname{Re}(\kappa)H + |\kappa|^2 I$.
- ▶ Now let Q_0 be a Householder transformation such that $Q_0^*c = \sigma e_1$.
- ▶ Next let $H_1 = Q_0^*AQ_0$.
- ▶ Use Householder transformations to reduce H_1 into upper Hessenberg form and call it \hat{H} . Call Q_1 the accumulated transformations.

Outline of the implicit shifted QR algorithm

- ▶ We determine the first column c of $C = H^2 - 2\operatorname{Re}(\kappa)H + |\kappa|^2 I$.
- ▶ Now let Q_0 be a Householder transformation such that $Q_0^*c = \sigma e_1$.
- ▶ Next let $H_1 = Q_0^*AQ_0$.
- ▶ Use Householder transformations to reduce H_1 into upper Hessenberg form and call it \hat{H} . Call Q_1 the accumulated transformations.
- ▶ Set $\hat{Q} = Q_0Q_1$

Outline of the implicit shifted QR algorithm

- ▶ To show that this algorithm works we need to show that \hat{Q} and \hat{R} are the matrices that one would have obtained using the explicit QR algorithm.

Outline of the implicit shifted QR algorithm

- ▶ To show that this algorithm works we need to show that \hat{Q} and \hat{R} are the matrices that one would have obtained using the explicit QR algorithm.
- ▶ We also need to show that c and the reduction of H_1 to upper Hessenberg can be computed rapidly

Implicit Q theorem

The implicit Q theorem states that for a matrix A of order n , let $H = Q^*AQ$ be a reduction of A to Hessenberg form. If the elements in the lower diagonal of H are non-zero then Q and H are uniquely determined by the first or last column of Q .

Comparison of explicit QR and implicit QR algorithm

- ▶ The computation of c takes only $O(n)$ operations because only the first three components of c are nonzero

Comparison of explicit QR and implicit QR algorithm

- ▶ The computation of c takes only $O(n)$ operations because only the first three components of c are nonzero
- ▶ The reduction of H_1 to upper Hessenberg can be done in $O(n^2)$

Comparison of explicit QR and implicit QR algorithm

- ▶ The computation of c takes only $O(n)$ operations because only the first three components of c are nonzero
- ▶ The reduction of H_1 to upper Hessenberg can be done in $O(n^2)$
- ▶ Total cost for implicit QR for k iterations is $2kn^3$ additions plus $2kn^3$ multiplications

Comparison of explicit QR and implicit QR algorithm

- ▶ The computation of c takes only $O(n)$ operations because only the first three components of c are nonzero
- ▶ The reduction of H_1 to upper Hessenberg can be done in $O(n^2)$
- ▶ Total cost for implicit QR for k iterations is $2kn^3$ additions plus $2kn^3$ multiplications
- ▶ Total cost for explicit QR for k iterations is $8kn^3$ additions plus $12kn^3$ multiplications

Comparison of explicit QR and implicit QR algorithm

- ▶ The computation of c takes only $O(n)$ operations because only the first three components of c are nonzero
- ▶ The reduction of H_1 to upper Hessenberg can be done in $O(n^2)$
- ▶ Total cost for implicit QR for k iterations is $2kn^3$ additions plus $2kn^3$ multiplications
- ▶ Total cost for explicit QR for k iterations is $8kn^3$ additions plus $12kn^3$ multiplications
- ▶ Further notice that one step of implicit = two steps of explicit