1 April 12th, 2019

1.1 QR Algorithm for Non-Symmetric Matrix

- Phase 1: We can not reduce A to a tridiagonal matrix. Instead we reduce A to an upper Hessenberg Matrix

  Definition 1.1. A **Hessenberg matrix** is a special kind of square matrix, one that is "almost" triangular. To be exact, an upper Hessenberg matrix has zero entries below the first subdiagonal, and a lower Hessenberg matrix has zero entries above the first superdiagonal.

$$\begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix} A = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \cdots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix} A \begin{bmatrix} 1 & 0 \\ 0 & H_1 \end{bmatrix}^T = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \cdots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix}$$

$$\vdots$$

$$Q^{(0)} A \left(Q^{(0)}\right)^T = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \times \end{bmatrix}$$

  which is a Hessenberg Matrix.

- Phase 2:

  > **Algorithm 1.2**
  > For $k = 1, 2, \ldots$
  > - Computer the QR Decomposition of $A^{(k)} = Q^{(k)} R^{(k)}$
  > - $A^{(k)} = R^{(k)} Q^{(k)}$.

- Note that $A^{(k)} = R^{(k)} Q^{(k)}$ is also a hessenberg Matrix:
  - For $A^{(k)} = R^{(k)} Q^{(k)} = R^{(k)} \left(G_k^{(1)}\right)^T \left(G_k^{(2)}\right)^T \cdots \left(G_k^{(n-1)}\right)^T$:

$$R^{(k)} \left(G_k^{(1)}\right)^T = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ & \times & \times & \cdots & \times \\ & & \times & \cdots & \times \\ & & & \ddots & \vdots \\ & & & & \times \end{bmatrix} \begin{bmatrix} c_1 & -s_1 & \\ s_1 & c_1 & \\ & & I \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ + & \times & \times & \cdots & \times \\ & & \times & \cdots & \times \\ & & & \ddots & \vdots \\ & & & & \times \end{bmatrix}.$$

$$R^{(k)}\left(G_k^{(1)}\right)^T\left(G_k^{(2)}\right)^T = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ + & \times & \times & \cdots & \times \\ & + & \times & \cdots & \times \\ & & & \ddots & \vdots \\ & & & & \times \end{bmatrix}.$$

- The idea is to first make it a Hessenberg Matrix, and then using Givens Rotation to remove the subdiagonal, giving us a upper triangular matrix.

The computational cost for the full QR algorithm is $O(n^3) + O(n^2 \cdot k)$, where k is the number of iterations (usually $O(n)$).

The algorithm generates $A^{(k)}$ satisfies:

$$A^{(k)} = \left(Q^{(k)}\right)^T \cdots \left(Q^{(0)}\right)^T AQ^{(0)} \cdots Q^{(k)}$$

which is similar to A, meaning that $A^{(k)}$ has the same eigenvalues of A.

However, we don't expect that this gives the eigenvalue decomposition of A. This is because:

1. the eigenvectors A are not necessarily orthogonal. (but $Q^{(i)}$ are orthogonal matrices), meaning that $X = Q^{(0)} \cdots Q^{(k)}$ is orthogonal.
2. the eigenvalue decomposition of A may not exist.

This QR algorithm converges to the Schur decomposition

Definition 1.3. For any matrix $A \in R^{n\times n}$, there exists $Q, S \in \mathbb{R}^{n\times n}$ such that

$$A = QSQ^T,$$

where

- $Q \in \mathbb{R}^{n\times n}$ is orthogonal, i.e. $Q^TQ = \mathbb{Q}^T = I$.
- $S \in \mathbb{R}^{n\times n}$ is a block upper triangular matrix, with 1×1 or 2×2 diagonal blocks. (i.e., there exists a partition of S:

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1p} \\ & S_{22} & \cdots & S_{2p} \\ & & \ddots & \\ & & & S_{pp} \end{bmatrix} \quad S_{ii} \text{ is either } 1\times 1 \text{ or } 2\times 2.$$

  Furthermore:

  - if $S_{ii} \in \mathbb{R}^{1\times 1}$, then it is an eigenvalue of A
  - if $S_{ii} \in \mathbb{R}^{2\times 2}$, then $S_{ii} = \begin{bmatrix} q & -b \\ b & a \end{bmatrix}$, with $a \pm bi$ being eigenvalues of A.
  - The blocks S_{ii} can be sorted s.t.

$$|\text{eig}(S_{11})| \geq |\text{eig}(S_{22})| \geq \ldots \geq |\text{eig}(S_{pp})|.$$

Theorem 1.4

Under mild assumption, the QR algorithm converges to the Schur decomposition, more precisely,

$$A^{(k)} \to S$$

and

$$Q^{(0)}Q^{(1)}\dots Q^{(k)} \to Q.$$

We can get eigenvalues of A from S and the eigenvectors of A from Q.

Remark 1.5 — Note that when A is symmetric, the Schur decomposition is the same as the eigenvalue decomposition, as S would by symmetric and upper triangular, i.e. S is diagonal. Because of this, all eigenvalues of A are real, since $S_{ii} \in \mathbb{R}^{1\times 1}$.

Remark 1.6 — There is no direct formula for the Schur decomposition, since it is related to the eigenvalues of the matrix, which are the roots of a polynomial, which doesn't have a general closed form solution.

Example 1.7

Consider:

$$A = \begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Since A is non-symmetric, then we would not have a unitary eivenvalue decomposition.

The Schur decomposition of A is:

$$A = QSQ^T = \begin{bmatrix} 1 & & \\ & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & -1 & \\ 1 & 1 & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

So the eigenvalues of A are:

- $1 = \det(S_{22})$
- $1 \pm i = \det(S_{11})$

1.2 Case Studies C - Applications of Eigenvalue Decomposition

Case Study I: Find the roots of a polynomial $p(x)$. Let

$$p(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1}x^{n-1} + x^n.$$

There are n solutions (roots) of $p(x) = 0$ in $\mathbb{C}$. We want to find all the roots of p.

Instead of using a root finding algorithm, we will use the eigenvalue decomposition to do so:

- Construct
$$A_p = \begin{bmatrix} 0 & & & & a_0 \\ 1 & \ddots & & & a_1 \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots \; 0 & a_{n-2} \\ & & & 1 & a_{n-1} \end{bmatrix}.$$
Then:
$$\det(\lambda I - A_p = a_0 + a_1\lambda + a_2\lambda^2 + \ldots + a_{n-1}\lambda^{n-1} + \lambda^n.$$
This means that the eigenvalues of A_p are the roots of $p(x)$, meaning we can use the QR algorithm to find eigenvalues of A_p This is widely used in available software packages.

Remark 1.8 — The reason why we might want to use the Schur decomposition to do so is because it is numerically stable due to the unitary transformation.

Case Study II: Ranking webpages. Once agian we have
$$\Pi = \frac{1-p}{n}\mathbb{1} + pA\Pi$$
$$(I - pA)\Pi = \frac{1-p}{n}\mathbb{1}$$
$$(I - pA)\Pi = \frac{1-p}{n}\mathbb{1}\mathbb{1}^T\Pi$$
$$(pA + \frac{1-p}{n}\mathbb{1}\mathbb{1}^T)\Pi = \Pi.$$
This is in the form of an eigenvalue problem