AM 205: lecture 23

- Last time: power method, Rayleigh quotient
- Today: QR algorithm, iterative methods for linear systems

QR Algorithm

The QR Algorithm

The QR algorithm for computing eigenvalues is one of the best known algorithms in Numerical Analysis[1]

It was developed independently in the late 1950s by John G.F. Francis (England) and Vera N. Kublanovskaya (USSR)

The QR algorithm efficiently provides approximations for all eigenvalues/eigenvectors of a matrix

We will consider what happens when we apply the power method to a set of vectors — this will then motivate the QR algorithm

[1] Recall that here we focus on the case in which $A \in \mathbb{R}^{n \times n}$ is symmetric

The QR Algorithm

Let $x_1^{(0)}, \ldots, x_p^{(0)}$ denote p linearly independent starting vectors, and suppose we store these vectors in the columns of X_0

We can apply the power method to these vectors to obtain the following algorithm:

1: choose an $n \times p$ matrix X_0 arbitrarily
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad X_k = AX_{k-1}$
4: **end for**

The QR Algorithm

From our analysis of the power method, we see that for each $i = 1, 2, \dots, p$:

$$\begin{aligned} x_i^{(k)} &= \left(\lambda_n^k \alpha_{i,n} v_n + \lambda_{n-1}^k \alpha_{i,n-1} v_{n-1} + \cdots + \lambda_1^k \alpha_{i,1} v_1\right) \\ &= \lambda_{n-p}^k \left(\sum_{j=n-p+1}^{n} \left(\frac{\lambda_j}{\lambda_{n-p}}\right)^k \alpha_{i,j} v_j + \sum_{j=1}^{n-p} \left(\frac{\lambda_j}{\lambda_{n-p}}\right)^k \alpha_{i,j} v_j \right) \end{aligned}$$

Then, if $|\lambda_{n-p+1}| > |\lambda_{n-p}|$, the sum in green will decay compared to the sum in blue as $k \to \infty$

Hence the columns of X_k will converge to a basis for $\text{span}\{v_{n-p+1}, \dots, v_n\}$

The QR Algorithm

However, this method doesn't provide a good basis: each column of X_k will be very close to v_n

Therefore the columns of X_k become very close to being linearly dependent

We can resolve this issue by enforcing linear independence at each step

The QR Algorithm

We orthonormalize the vectors after each iteration via a (reduced) QR factorization, to obtain the simultaneous iteration:

1: choose $n \times p$ matrix Q_0 with orthonormal columns
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad X_k = A\hat{Q}_{k-1}$
4: $\quad \hat{Q}_k \hat{R}_k = X_k$
5: **end for**

The column spaces of $\hat{Q}_k$ and X_k in line 4 are the same

Hence columns of $\hat{Q}_k$ converge to orthonormal basis for $\mathrm{span}\{v_{n-p+1}, \ldots, v_n\}$

The QR Algorithm

In fact, we don't just get a basis for $\mathrm{span}\{v_{n-p+1}, \ldots, v_n\}$, we get the eigenvectors themselves!

Theorem: The columns of $\hat{Q}_k$ converge to the p dominant eigenvectors of A

We will not discuss the full proof, but we note that this result is not surprising since:

- the eigenvectors of a symmetric matrix are orthogonal
- columns of $\hat{Q}_k$ converge to an orthogonal basis for $\mathrm{span}\{v_{n-p+1}, \ldots, v_n\}$

Simultaneous iteration approximates eigenvectors, we obtain eigenvalues from the Rayleigh quotient $\hat{Q}^T A \hat{Q} \approx \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$

The QR Algorithm

With $p = n$, the simultaneous iteration will approximate all eigenpairs of A

We now show a more convenient reorganization of the simultaneous iteration algorithm

We shall require some extra notation: the Q and R matrices arising in the simultaneous iteration will be underlined $\underline{Q}_k$, $\underline{R}_k$

(As we will see shortly, this is to distinguish between the matrices arising in the two different formulations...)

The QR Algorithm

Define[2] the k^{th} Rayleigh quotient matrix: $A_k \equiv \underline{Q}_k^T A \underline{Q}_k$, and the QR factors Q_k, R_k as: $Q_k R_k = A_{k-1}$

Our goal is to show that $A_k = R_k Q_k$, $k = 1, 2, \ldots$

Initialize $\underline{Q}_0 = \mathrm{I} \in \mathbb{R}^{n \times n}$, then in the first simultaneous iteration we obtain $X_1 = A$ and $\underline{Q}_1 \underline{R}_1 = A$

It follows that $A_1 = \underline{Q}_1^T A \underline{Q}_1 = \underline{Q}_1^T (\underline{Q}_1 \underline{R}_1) \underline{Q}_1 = \underline{R}_1 \underline{Q}_1$

Also $Q_1 R_1 = A_0 = \underline{Q}_0^T A \underline{Q}_0 = A$, so that $Q_1 = \underline{Q}_1$, $R_1 = \underline{R}_1$, and $A_1 = R_1 Q_1$

---

[2]We now we use the full, rather than the reduced, QR factorization hence we omit ˆ notation

The QR Algorithm

In the second simultaneous iteration, we have $X_2 = A\underline{Q}_1$, and we compute the QR factorization $\underline{Q}_2\underline{R}_2 = X_2$

Also, using our QR factorization of A_1 gives

$$X_2 = A\underline{Q}_1 = (\underline{Q}_1\underline{Q}_1^T)A\underline{Q}_1 = \underline{Q}_1 A_1 = \underline{Q}_1(Q_2 R_2),$$

which implies that $\underline{Q}_2 = \underline{Q}_1 Q_2 = Q_1 Q_2$ and $\underline{R}_2 = R_2$

Hence

$$A_2 = \underline{Q}_2^T A\underline{Q}_2 = Q_2^T\underline{Q}_1^T A\underline{Q}_1 Q_2 = Q_2^T A_1 Q_2 = Q_2^T Q_2 R_2 Q_2 = R_2 Q_2$$

The QR Algorithm

The same pattern continues for $k = 3, 4, \ldots$: we QR factorize A_k to get Q_k and R_k, then we compute $A_{k+1} = R_k Q_k$

The columns of the matrix $\underline{Q}_k = Q_1 Q_2 \cdots Q_k$ approximates the eigenvectors of A

The diagonal entries of the Rayleigh quotient matrix $A_k = \underline{Q}_k^T A \underline{Q}_k$ approximate the eigenvalues of A

(Also, due to eigenvector orthogonality for symmetric A, A_k converges to a diagonal matrix as $k \to \infty$)

The QR Algorithm

This discussion motivates the famous QR algorithm:

$A_0 = A$
for $k = 1, 2, \ldots$ **do**
    $Q_k R_k = A_{k-1}$
    $A_k = R_k Q_k$
end for