Rayleigh Quotient Iteration

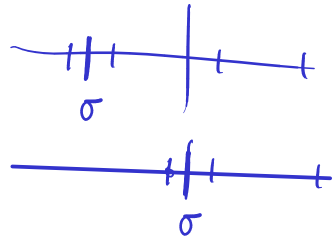
- Describe inverse iteration.

x_0

$AAAAx_0$

$A^k x_0$

$(A^{-1})^k x_0$



$\left((A-\sigma I)^{-1}\right)^k x_0$

- Describe Rayleigh Quotient Iteration.

$$\frac{x^T A x}{x^T x}$$

Demo: Power Iteration and its Variants

Computing Multiple Eigenvalues

- All Power Iteration Methods compute one eigenvalue at a time. What if I want *all* eigenvalues?

- Suppose I know λ so that $Ax = \lambda x$

$$A = Q \begin{pmatrix} \lambda & \\ 0 & \\ \vdots & \times \\ 0 & \end{pmatrix} Q^T$$

↳ Deflation

- Power it. w/ multiple vectors

Simultaneous Iteration

- What happens if we carry out power iteration on multiple vectors simultaneously?

$$X_0 \in \mathbb{R}^{n \times p}$$

$$X_{k+1} = A X_k$$

Drawbacks: - p times same answer (eigvec for largest ev.)
- expensive
- ill-conditioned

Orthogonal Iteration

- $X_0 \in \mathbb{R}^{n \times p}$

  $Q_k R_k = X_k$

  $X_{k+1} = A Q_k$

$Q_0 R_0 = X_0$

$X_1 = A Q_0 \Rightarrow A = Q_1 R_1 Q_0^T$

$Q_1 R_1 = X_1$

$X_2 = A Q_1$

$Q_k R_k = X_k \Rightarrow A = Q_{k+1} R_{k+1} Q_k^T$

$X_k = A Q_k$

$(p \le n)$ starting vectors

- expensive
- slow / linear convergence

$A = Q \left(\quad \right) Q^T$

$A \approx Q_k R_k Q_k^T$

$\hat{X}_2 = Q_k^T A Q_k \approx R_k$

Demo: Orthogonal Iteration

In-class activity: Eigenvalue Iterations

QR Iteration/QR Algorithm

Orthogonal iteration:

$X_0 = A$

$Q_k R_k = X_k$

$X_{k+1} = A\,Q_k$

QR iteration:

$\bar{X}_0 = A$

$\bar{Q}_k \bar{R}_k = \bar{X}_k$

$\bar{X}_{k+1} = \bar{R}_k \bar{Q}_k$

- slow-ish convergence (still same as PI)

- expensive ($O(n^3)$/it.)

Tracing through reveals:

- $\hat{X}_k = \bar{X}_{k+1}$
- $Q_0 = \bar{Q}_0$
  $Q_1 = \bar{Q}_0 \bar{Q}_1$
  $Q_k = \bar{Q}_0 \bar{Q}_1 \cdots \bar{Q}_k$

Orthogonal iteration showed: $\hat{X}_k = \bar{X}_{k+1}$ converge. Also:

$$\bar{X}_{k+1} = \bar{R}_k \bar{Q}_k = \bar{Q}_k^T \bar{X}_k \bar{Q}_k,$$

so the $\bar{X}_k$ are all similar $\rightarrow$ all have the same eigenvalues.
$\rightarrow$ QR iteration produces Schur form.

QR Iteration: Incorporating a Shift

- How can we accelerate convergence of QR iteration using shifts?

$$\bar{X}_0 = A$$

$$\bar{Q}_k \bar{R}_k = \bar{X}_k - \sigma_k I \rightsquigarrow \bar{R}_k = \bar{Q}_k^T[\bar{X}_k - \sigma_k I]$$

$$\bar{X}_{k+1} = \bar{R}_k \bar{Q}_k + \sigma_k I$$

$$\bar{X}_{k+1} = \bar{R}_k \bar{Q}_k + \sigma_k I = \bar{Q}_k^T[\bar{X}_k - \sigma_k I]\bar{Q}_k + \sigma_k I$$

$$= \bar{Q}_k^T \bar{X}_k \bar{Q}_k - \sigma_k \cancel{\bar{Q}_k^T} I \cancel{\bar{Q}_k} + \sigma_k I$$

$$= \bar{Q}_k^T \bar{X}_k \bar{Q}_k - \cancel{\sigma_k} I + \cancel{\sigma_k} I$$

(1) Picking $\sigma_k \approx (\bar{X}_k)_{nn}$

(2) Pick two eigenvalues $\begin{pmatrix} & \\ & \square \end{pmatrix}$ in the BR of $\bar{X}_k$

$$Ax = \lambda x$$

$$(A - \lambda I)x = 0$$

QR Iteration: Computational Expense

- A full QR factorization at each iteration costs $O(n^3)$–can we make that cheaper?

4.4 Krylov Space Methods