July 9, 2014

Simultaneous iteration/Block power iteration

- Apply power iteration to several vectors at once and maintain linearly independence among the vectors.

- Start with: $v_1^{(0)}, v_2^{(0)}, \ldots, v_p^{(0)}$
  Then $A^k v_1^{(0)}$ converges to q_1 where $|\lambda_1|$ is largest.
  Thus span $\{ A^k v_1^{(0)}, \ldots, A^k v_p^{(0)} \}$ should converge to $\{ q_1, \ldots, q_p \}$ where $\lambda_1, \ldots, \lambda_p$ are the p largest eigenvalues.

- Write $V^{(0)} = [\, v_1^{(0)} \;\; v_2^{(0)} \;\; \ldots \;\; v_p^{(0)} \,]$.

  Define $V^{(k)} = A^{(k)} V^{(0)} = [\, v_1^{(k)} \;\; v_2^{(k)} \;\; \ldots \;\; v_p^{(k)} \,]$.

- As $k \to \infty$, the vectors $v_1^{(k)}, \ldots, v_p^{(k)}$ all converge to multiples of the same dominant eigenvector q_1.

- Orthogonalize the vectors at each step.

Algorithm

Pick $\hat{Q}^{(0)} \in R^{nxp}$ with orthonormal columns

for $k = 1, 2, \ldots$

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

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

end

Note: The column space of $\hat{Q}^{(k)}$ and $Z^{(k)}$ are the same. They are both equal to that of $A^{(k)} \hat{Q}^{(0)}$.

- Assumption 1: The leading p+1 e.v. are distinct in absolute values:

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_p| > |\lambda_{p+1}| \geq |\lambda_{p+2}| \ldots \geq |\lambda_n|$$

- Assumption 2: All the leading principal minors of $\hat{Q}^T V^{(0)}$ are nonsingular.

<u>Theorem</u>: Suppose the block power iteration is carried out and assumptions 1 & 2 hold. Then as $k \to \infty$,

$$\left\| q_j^{(k)} - (\pm q_j) \right\| = O(c^k) \qquad j = 1, 2, \ldots, p$$

where $c = \max\limits_{1 \leq k \leq p} \left| \frac{\lambda_{k+1}}{\lambda_k} \right| < 1$

Simultaneous iteration vs QR iteration

- QR iteration can be viewed as simultaneous iteration with $\hat{Q}^{(0)} = I$ and p = n.
- We can drop the hats on $\hat{Q}^{(k)}, \hat{R}^{(k)}$.
- $\underline{Q}^{(k)}$ = Q's from simultaneous iteration, $Q^{(k)}$ = Q's from QR iteration.

Simultaneous iteration can be written as:

$$\begin{aligned}
&\underline{Q}^{(0)} = I \\
&\text{for } k = 1, 2, \ldots \\
&\quad Z^{(k)} = A\,\underline{Q}^{(k-1)} \\
&\quad Z^{(k)} = \underline{Q}^{(k)} R^{(k)} \\
&\quad \left.\begin{aligned} A^{(k)} &= (\underline{Q}^{(k)})^T A\, \underline{Q}^{(k)} \\ \underline{R}^{(k)} &= R^{(k)} R^{(k-1)} \ldots R^{(1)} \end{aligned}\right\}\text{new matrices for proof purpose} \\
&\text{end}
\end{aligned}$$

QR iteration can be written as:

$$\begin{aligned}
&A^{(0)} = A \\
&\text{for } k = 1, 2, \ldots \\
&\quad A^{(k-1)} = Q^{(k)} R^{(k)} \\
&\quad A^{(k)} = R^{(k)} Q^{(k)} \\
&\quad \left.\begin{aligned} \underline{Q}^{(k)} &= Q^{(1)} Q^{(2)} \ldots Q^{(k)} \\ \underline{R}^{(k)} &= R^{(k)} R^{(k-1)} \ldots R^{(1)} \end{aligned}\right\}\text{new matrices for proof purpose} \\
&\text{end}
\end{aligned}$$

Theorem: The two algorithms generate identical sequences of matrices $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$, and $A^{(k)}$:

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

and

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)} \qquad (2)$$

Pf: By induction. The case $k = 0$ is trivial since

$$A^0 = I = \underline{Q}^{(0)} = \underline{R}^{(0)} \text{ and } A^{(0)} = A$$

Suppose it is true for k-1.

Simultaneous iteration:

(1): $A^k = A A^{k-1} = A \underline{Q}^{(k-1)} \underline{R}^{(k-1)}$ (induction hypo. (1))

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

$= \underline{Q}^{(k)} \underline{R}^{(k)}$ (by def of $R^{(k)}$)

(2): By algorithm

QR iteration:

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

$= \underline{Q}^{(k-1)} A^{(k-1)} \underline{R}^{(k-1)}$ (induction hypo. (2))

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

$= \underline{Q}^{(k)} \underline{R}^{(k)}$ (by def of $Q^{(k)}$ and $R^{(k)}$)

(2): $A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$ (by algorithm)

$= (Q^{(k)})^T (\underline{Q}^{(k-1)})^T A \underline{Q}^{(k-1)} Q^{(k)}$ (induction hypo. (2))

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

Convergence of the QR iteration

- (1) $\Rightarrow$ QR iteration effectively computes Q, R factors of A^k ; i.e. orthogonal basis for A^k.

- (2) $\Rightarrow$ The diagonal of $A^{(k)}$ are Rayleigh quotient of column vectors of $\underline{Q}^{(k)}$.

- As columns of $\underline{Q}^{(k)} \longrightarrow$ eigenvectors, the Rayleigh quotients $\longrightarrow$ eigenvalues.

- $A_{ij}^{(k)} = (\underline{q}_i^{(k)})^T A\, (\underline{q}_j^{(k)})$ $\quad\quad$ $\underline{q}_i^{(k)}, \underline{q}_j^{(k)}$ col i and col j of $\underline{Q}^{(k)}$

$$\underline{q}_j^{(k)} \longrightarrow q_j, \quad \underline{q}_i^{(k)} \longrightarrow q_i, \quad A\, \underline{q}_j^{(k)} \sim \lambda_j\, q_j$$

$\Rightarrow$ $A_{ij}^{(k)} \sim \lambda_j\, q_i^T q_j = 0 \quad\quad (i \neq j)$

$\therefore$ $A^{(k)}$ converges to a diagonal matrix.

Theorem: Assume $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ and Q has all nonsingular leading principal minors. As $k \to \infty$, $A^{(k)}$ converges linearly to diag $(\lambda_1, \ldots, \lambda_n)$ and $\underline{Q}^{(k)}$ converges at the same rate to Q. The rate of convergence is

$$C = \max_k \left| \frac{\lambda_{k+1}}{\lambda_k} \right|$$

Practical QR

- It is expensive to compute the QR factorization of a square matrix ($4/3\ n^3$ flops).

- In practice, we first reduce A to a Hessenberg matrix if $A \neq A^T$ and to a tridiagonal matrix if $A = A^T$.

- The resulting QR factorization would be $O(n^2)$ if $A \neq A^T$ and $O(n)$ if $A = A^T$.

Reduction to Hessenberg or tridiagonal

- The matrix can be nonsymmetric in general.

- Why Hessenberg? Why not triangular?

e.g. Apply Householder Q_1 to A.

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{Q_1^T \times} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} = Q_1^T A$$

To compute the similarity transformation, multiply Q_1 on the right.

$$\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{\times Q_1} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} = Q_1^T A Q_1$$

The zeros are destroyed by right multiplication by Q_1.

- Less ambitious and choose Q_1^T that leaves 1st row unchanged.
- When Q_1 is multiplied on the right, it will leave the 1st col unchanged.

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^T \times} \underset{Q_1^T A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\times Q_1} \underset{Q_1^T A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}}$$

- Apply the same idea to other cols:

$$\underset{Q_1^T A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}} \longrightarrow \underset{Q_2^T Q_1^T A Q_1 Q_2}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}} \longrightarrow \underset{Q_3^T Q_2^T Q_1^T A Q_1 Q_2 Q_3}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}} \text{upper Hessenberg}$$

- $Q = Q_1\, Q_2 \, . \, . \, . \, Q_{n-2}$ and $Q^T A\, Q$ = upper Hessenberg
- Complexity:

  flops(Reduction to Hess) ~ $10/3\ n^3$

  flops(Reduction to tridiag) ~ $4/3\ n^3$

Algorithm

for $k = 1, 2, \ldots, n\text{-}2$

  $x = A(k+1:n, k)$

  $v_k = \text{sign}(x_1)||x||e_1 + x$

  $v_k = v_k / ||v_k||$

  $$\left.\begin{array}{l} \text{for } j = k, k+1, \ldots, n \\ \quad A(k+1:n, j) = A(k+1:n, j) - 2\, v_k\, (v_k^T A(k+1:n, j)) \\ \text{end} \end{array}\right\} Q_k^T \times$$

  $$\left.\begin{array}{l} \text{for } i = 1, 2, \ldots, n \\ \quad A(i, k+1:n) = A(i, k+1:n) - 2\, (A(i, k+1:n)\, v_k)\, v_k^T \\ \text{end} \end{array}\right\} \times Q_k$$

end

Symmetric case

- If $A = A^T$, then

$$(Q^T A Q)^T = Q^T A Q \qquad \text{is also symmetric}$$

- A symmetric Hessenberg matrix -> tridiagonal matrix.

Two-phase process (symmetric case)

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Phase 1}} \underset{T = Q^T A Q}{\begin{bmatrix} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}} \xrightarrow[\text{QR iteration}]{\text{Phase 2}} \underset{D}{\begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \end{bmatrix}}$$