CS 450 – Numerical Analysis

Chapter 4: Eigenvalue Problems [†]

Prof. Michael T. Heath

Department of Computer Science
University of Illinois at Urbana-Champaign
heath@illinois.edu

January 28, 2019

[†] Lecture slides based on the textbook *Scientific Computing: An Introductory Survey* by Michael T. Heath, copyright © 2018 by the Society for Industrial and Applied Mathematics. http://www.siam.org/books/cl80

Eigenvalue Problems

Eigenvalues and Eigenvectors

- Standard *eigenvalue problem*: Given $n \times n$ matrix $\boldsymbol{A}$, find scalar λ and nonzero vector $\boldsymbol{x}$ such that

$$\boldsymbol{A}\,\boldsymbol{x} = \lambda\,\boldsymbol{x}$$

- λ is *eigenvalue*, and $\boldsymbol{x}$ is corresponding *eigenvector*
- λ may be complex even if $\boldsymbol{A}$ is real
- *Spectrum* $= \lambda(\boldsymbol{A}) =$ set of all eigenvalues of $\boldsymbol{A}$
- *Spectral radius* $= \rho(\boldsymbol{A}) = \max\{|\lambda| : \lambda \in \lambda(\boldsymbol{A})\}$

Geometric Interpretation

- Matrix expands or shrinks any vector lying in direction of eigenvector by scalar factor
- Scalar expansion or contraction factor is given by corresponding eigenvalue λ
- Eigenvalues and eigenvectors decompose complicated behavior of general linear transformation into simpler actions

Examples: Eigenvalues and Eigenvectors

- $\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$: $\lambda_1 = 1,\ \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_2 = 2,\ \boldsymbol{x}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- $\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$: $\lambda_1 = 1,\ \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_2 = 2,\ \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
- $\boldsymbol{A} = \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$: $\lambda_1 = 2,\ \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \lambda_2 = 4,\ \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$
- $\boldsymbol{A} = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$: $\lambda_1 = 2,\ \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \lambda_2 = 1,\ \boldsymbol{x}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$
- $\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$: $\lambda_1 = i,\ \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad \lambda_2 = -i,\ \boldsymbol{x}_2 = \begin{bmatrix} i \\ 1 \end{bmatrix}$

  where $i = \sqrt{-1}$

Characteristic Polynomial and Multiplicity

Characteristic Polynomial

- Equation $\boldsymbol{Ax} = \lambda \boldsymbol{x}$ is equivalent to

  $$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$$

  which has nonzero solution $\boldsymbol{x}$ if, and only if, its matrix is singular

- Eigenvalues of $\boldsymbol{A}$ are roots λ_i of *characteristic polynomial*

  $$\boxed{\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0}$$

  in λ of degree n

- *Fundamental Theorem of Algebra* implies that $n \times n$ matrix $\boldsymbol{A}$ always has n eigenvalues, but they may not be real nor distinct

- Complex eigenvalues of real matrix occur in complex conjugate pairs: if $\alpha + i\beta$ is eigenvalue of real matrix, then so is $\alpha - i\beta$, where $i = \sqrt{-1}$

Example: Characteristic Polynomial

- Characteristic polynomial of previous example matrix is

$$\det\left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) =$$

$$\det\left(\begin{bmatrix} 3-\lambda & -1 \\ -1 & 3-\lambda \end{bmatrix}\right) =$$

$$(3-\lambda)(3-\lambda) - (-1)(-1) = \lambda^2 - 6\lambda + 8 = 0$$

so eigenvalues are given by

$$\lambda = \frac{6 \pm \sqrt{36-32}}{2}, \quad \text{or} \quad \lambda_1 = 2, \quad \lambda_2 = 4$$

Characteristic Polynomial, continued

- Computing eigenvalues using characteristic polynomial is *not* recommended because of
  - work in computing coefficients of characteristic polynomial
  - sensitivity of coefficients of characteristic polynomial
  - work in solving for roots of characteristic polynomial
- Characteristic polynomial is powerful theoretical tool but usually not useful computationally

Example: Characteristic Polynomial

- Consider

  $$\boldsymbol{A} = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}$$ — 0.1

  where ϵ is positive number slightly smaller than $\sqrt{\epsilon_{\text{mach}}}$

- Exact eigenvalues of $\boldsymbol{A}$ are $1 + \epsilon$ and $1 - \epsilon$

  1.1 0.9

- Computing characteristic polynomial in floating-point arithmetic, we obtain

  $$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = \lambda^2 - 2\lambda + (1 - \epsilon^2) = \lambda^2 - 2\lambda + 1$$

  which has 1 as double root

- Thus, eigenvalues cannot be resolved by this method even though they are distinct in working precision

Computing Eigenvalues and Eigenvectors

Problem Transformations

- *Shift*: If $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ and σ is any scalar, then $(\boldsymbol{A} - \sigma\boldsymbol{I})\boldsymbol{x} = (\lambda - \sigma)\boldsymbol{x}$, so eigenvalues of shifted matrix are shifted eigenvalues of original matrix

- *Inversion*: If $\boldsymbol{A}$ is nonsingular and $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ with $\boldsymbol{x} \neq \boldsymbol{0}$, then $\lambda \neq 0$ and $\boldsymbol{A}^{-1}\boldsymbol{x} = (1/\lambda)\boldsymbol{x}$, so eigenvalues of inverse are reciprocals of eigenvalues of original matrix

- *Powers*: If $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, then $\boldsymbol{A}^k\boldsymbol{x} = \lambda^k\boldsymbol{x}$, so eigenvalues of power of matrix are same power of eigenvalues of original matrix

- *Polynomial*: If $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ and $p(t)$ is polynomial, then $p(\boldsymbol{A})\boldsymbol{x} = p(\lambda)\boldsymbol{x}$, so eigenvalues of polynomial in matrix are values of polynomial evaluated at eigenvalues of original matrix

Similarity Transformation

- $\boldsymbol{B}$ is *similar* to $\boldsymbol{A}$ if there is nonsingular matrix $\boldsymbol{T}$ such that

$$\boxed{\boldsymbol{B} = \boldsymbol{T}^{-1}\boldsymbol{A}\,\boldsymbol{T}}$$

- Then

$$\boldsymbol{B}\boldsymbol{y} = \lambda\boldsymbol{y} \;\Rightarrow\; \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\boldsymbol{y} = \lambda\boldsymbol{y} \;\Rightarrow\; \boldsymbol{A}(\boldsymbol{T}\boldsymbol{y}) = \lambda(\boldsymbol{T}\boldsymbol{y})$$

  so $\boldsymbol{A}$ and $\boldsymbol{B}$ have same eigenvalues, and if $\boldsymbol{y}$ is eigenvector of $\boldsymbol{B}$, then $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{y}$ is eigenvector of $\boldsymbol{A}$

- Similarity transformations preserve eigenvalues, and eigenvectors are easily recovered

Example: Similarity Transformation

- From eigenvalues and eigenvectors for previous example,

$$\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

  and hence

$$\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

- So original matrix is similar to diagonal matrix, and eigenvectors form columns of similarity transformation matrix

Diagonal Form

- Eigenvalues of diagonal matrix are diagonal entries, and eigenvectors are columns of identity matrix
- Diagonal form is desirable in simplifying eigenvalue problems for general matrices by similarity transformations
- But not all matrices are diagonalizable by similarity transformation
- Closest one can get, in general, is *Jordan form*, which is nearly diagonal but may have some nonzero entries on first superdiagonal, corresponding to one or more multiple eigenvalues

Triangular Form

- Any matrix can be transformed into triangular (*Schur*) form by similarity, and eigenvalues of triangular matrix are diagonal entries

- Eigenvectors of triangular matrix less obvious, but still straightforward to compute

- If

  $$\boldsymbol{A} - \lambda \boldsymbol{I} = \begin{bmatrix} \boldsymbol{U}_{11} & \boldsymbol{u} & \boldsymbol{U}_{13} \\ \boldsymbol{0} & 0 & \boldsymbol{v}^T \\ \boldsymbol{O} & \boldsymbol{0} & \boldsymbol{U}_{33} \end{bmatrix}$$

  is triangular, then $\boldsymbol{U}_{11}\boldsymbol{y} = \boldsymbol{u}$ can be solved for $\boldsymbol{y}$, so that

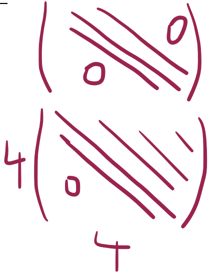
  $$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ -1 \\ \boldsymbol{0} \end{bmatrix}$$

  is corresponding eigenvector

Relevant Properties of Matrices

- Properties of matrix $\boldsymbol{A}$ relevant to eigenvalue problems

Property	Definition
diagonal	$a_{ij} = 0$ for $i \neq j$
tridiagonal	$a_{ij} = 0$ for $\lvert i - j \rvert > 1$
triangular	$a_{ij} = 0$ for $i > j$ (upper)
	$a_{ij} = 0$ for $i < j$ (lower)
Hessenberg	$a_{ij} = 0$ for $i > j + 1$ (upper)
	$a_{ij} = 0$ for $i < j - 1$ (lower)
orthogonal	$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I}$
unitary	$\boldsymbol{A}^H\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^H = \boldsymbol{I}$
symmetric	$\boldsymbol{A} = \boldsymbol{A}^T$
Hermitian	$\boldsymbol{A} = \boldsymbol{A}^H$
normal	$\boldsymbol{A}^H\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^H$



Eigenspaces and Invariant Subspaces

- Eigenvectors can be scaled arbitrarily: if $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, then $\boldsymbol{A}(\gamma\boldsymbol{x}) = \lambda(\gamma\boldsymbol{x})$ for any scalar γ, so $\gamma\boldsymbol{x}$ is also eigenvector corresponding to λ

- Eigenvectors are usually *normalized* by requiring some norm of eigenvector to be 1

- *Eigenspace* $= \mathcal{S}_\lambda = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}\}$

- Subspace $\mathcal{S}$ of $\mathbb{R}^n$ (or $\mathbb{C}^n$) is *invariant* if $\boldsymbol{A}\mathcal{S} \subseteq \mathcal{S}$

- For eigenvectors $\boldsymbol{x}_1 \cdots \boldsymbol{x}_p$, span($[\boldsymbol{x}_1 \cdots \boldsymbol{x}_p]$) is invariant subspace

Power Iteration

Power Iteration

- Simplest method for computing one eigenvalue-eigenvector pair is *power iteration*, which repeatedly multiplies matrix times initial starting vector

- Assume $\boldsymbol{A}$ has unique eigenvalue of maximum modulus, say λ_1, with corresponding eigenvector $\boldsymbol{v}_1$

- Then, starting from nonzero vector $\boldsymbol{x}_0$, iteration scheme

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1}$$

converges to multiple of eigenvector $\boldsymbol{v}_1$ corresponding to *dominant* eigenvalue λ_1

Convergence of Power Iteration

- To see why power iteration converges to dominant eigenvector, express starting vector $\boldsymbol{x}_0$ as linear combination

$$\boldsymbol{x}_0 = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i$$

  where $\boldsymbol{v}_i$ are eigenvectors of $\boldsymbol{A}$

- Then

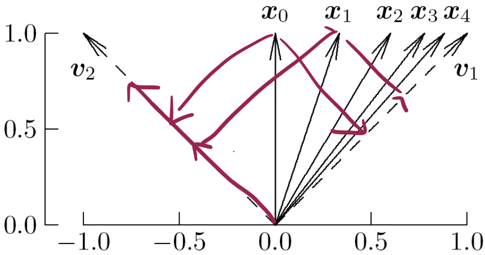
$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} = \boldsymbol{A}^2\boldsymbol{x}_{k-2} = \cdots = \boldsymbol{A}^k\boldsymbol{x}_0 =$$

$$\sum_{i=1}^{n} \lambda_i^k \alpha_i \boldsymbol{v}_i = \lambda_1^k \left(\alpha_1 \boldsymbol{v}_1 + \sum_{i=2}^{n} (\lambda_i/\lambda_1)^k \alpha_i \boldsymbol{v}_i \right)$$

- Since $|\lambda_i/\lambda_1| < 1$ for $i > 1$, successively higher powers go to zero, leaving only component corresponding to $\boldsymbol{v}_1$

Geometric Interpretation

- Behavior of power iteration depicted geometrically



- Initial vector $x_0 = v_1 + v_2$ contains equal components in eigenvectors v_1 and v_2 (dashed arrows)

- Repeated multiplication by A causes component in v_1 (corresponding to larger eigenvalue, 2) to dominate, so sequence of vectors x_k converges to v_1

Example: Power Iteration

- Ratio of values of given component of $\boldsymbol{x}_k$ from one iteration to next converges to dominant eigenvalue λ_1

- For example, if $\boldsymbol{A} = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$ and $\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, we obtain

k	$\boldsymbol{x}_k^T$		ratio
0	0.0	1.0	
1	0.5	1.5	1.500
2	1.5	2.5	1.667
3	3.5	4.5	1.800
4	7.5	8.5	1.889
5	15.5	16.5	1.941
6	31.5	32.5	1.970
7	63.5	64.5	1.985
8	127.5	128.5	1.992

- Ratio is converging to dominant eigenvalue, which is 2

Normalized Power Iteration

- Geometric growth of components at each iteration risks eventual overflow (or underflow if $\lambda_1 < 1$)

- Approximate eigenvector should be normalized at each iteration, say, by requiring its largest component to be 1 in modulus, giving iteration scheme

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} \\ \boldsymbol{x}_k &= \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_\infty \end{aligned}$$

- With normalization, $\|\boldsymbol{y}_k\|_\infty \to |\lambda_1|$, and $\boldsymbol{x}_k \to \boldsymbol{v}_1 / \|\boldsymbol{v}_1\|_\infty$

Example: Normalized Power Iteration

- Repeating previous example with normalized scheme,

k	$\boldsymbol{x}_k^T$		$\|\boldsymbol{y}_k\|_\infty$
0	0.000	1.0	
1	0.333	1.0	1.500
2	0.600	1.0	1.667
3	0.778	1.0	1.800
4	0.882	1.0	1.889
5	0.939	1.0	1.941
6	0.969	1.0	1.970
7	0.984	1.0	1.985
8	0.992	1.0	1.992

$\langle$ interactive example $\rangle$

Power Iteration with Shift

- Convergence rate of power iteration depends on ratio $|\lambda_2/\lambda_1|$, where λ_2 is eigenvalue having second largest modulus

- May be possible to choose shift, $\boldsymbol{A} - \sigma \boldsymbol{I}$, such that

  $$\left| \frac{\lambda_2 - \sigma}{\lambda_1 - \sigma} \right| < \left| \frac{\lambda_2}{\lambda_1} \right|$$

  so convergence is accelerated

- Shift must then be added to result to obtain eigenvalue of original matrix

- In earlier example, for instance, if we pick shift of $\sigma = 1$, (which is equal to other eigenvalue) then ratio becomes zero and method converges in one iteration

- In general, we would not be able to make such fortuitous choice, but shifts can still be extremely useful in some contexts, as we will see later

Limitations of Power Iteration

Power iteration can fail for various reasons

- Starting vector may have *no* component in dominant eigenvector $\boldsymbol{v}_1$ (i.e., $\alpha_1 = 0$) — not problem in practice because rounding error usually introduces such component in any case
- There may be more than one eigenvalue having same (maximum) modulus, in which case iteration may converge to linear combination of corresponding eigenvectors
- For real matrix and starting vector, iteration can never converge to complex vector

Inverse and Rayleigh Quotient Iterations

Inverse Iteration

- To compute smallest eigenvalue of matrix rather than largest, can make use of fact that eigenvalues of $\boldsymbol{A}^{-1}$ are reciprocals of those of $\boldsymbol{A}$, so smallest eigenvalue of $\boldsymbol{A}$ is reciprocal of largest eigenvalue of $\boldsymbol{A}^{-1}$

- This leads to *inverse iteration* scheme

$$\begin{aligned} \boldsymbol{A}\boldsymbol{y}_k &= \boldsymbol{x}_{k-1} \\ \boldsymbol{x}_k &= \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_\infty \end{aligned}$$

  which is equivalent to power iteration applied to $\boldsymbol{A}^{-1}$

- Inverse of $\boldsymbol{A}$ not computed explicitly, but factorization of $\boldsymbol{A}$ used to solve system of linear equations at each iteration

- Inverse iteration converges to eigenvector corresponding to *smallest* eigenvalue of $\boldsymbol{A}$

- Eigenvalue obtained is dominant eigenvalue of $\boldsymbol{A}^{-1}$, and hence its reciprocal is smallest eigenvalue of $\boldsymbol{A}$ in modulus

Example: Inverse Iteration

- Applying inverse iteration to previous example to compute smallest eigenvalue yields sequence

k	$\boldsymbol{x}_k^T$		$\|\boldsymbol{y}_k\|_\infty$
0	0.000	1.0	
1	-0.333	1.0	0.750
2	-0.600	1.0	0.833
3	-0.778	1.0	0.900
4	-0.882	1.0	0.944
5	-0.939	1.0	0.971
6	-0.969	1.0	0.985

which is indeed converging to 1 (which is its own reciprocal in this case)

⟨ interactive example ⟩

Inverse Iteration with Shift

- As before, shifting strategy, working with $\boldsymbol{A} - \sigma \boldsymbol{I}$ for some scalar σ, can greatly improve convergence

- Inverse iteration is particularly useful for computing eigenvector corresponding to approximate eigenvalue, since it converges rapidly when applied to shifted matrix $\boldsymbol{A} - \lambda \boldsymbol{I}$, where λ is approximate eigenvalue

- Inverse iteration is also useful for computing eigenvalue closest to given value β, since if β is used as shift, then desired eigenvalue corresponds to smallest eigenvalue of shifted matrix

Rayleigh Quotient

- Given approximate eigenvector $\boldsymbol{x}$ for real matrix $\boldsymbol{A}$, determining best estimate for corresponding eigenvalue λ can be considered as $n \times 1$ linear least squares approximation problem

$$\boldsymbol{x}\lambda \cong \boldsymbol{A}\boldsymbol{x}$$

- From normal equation $\boldsymbol{x}^T\boldsymbol{x}\lambda = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$, least squares solution is given by

$$\lambda = \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}$$

- This quantity, known as *Rayleigh quotient*, has many useful properties

Example: Rayleigh Quotient

- Rayleigh quotient can accelerate convergence of iterative methods such as power iteration, since Rayleigh quotient $\boldsymbol{x}_k^T \boldsymbol{A} \boldsymbol{x}_k / \boldsymbol{x}_k^T \boldsymbol{x}_k$ gives better approximation to eigenvalue at iteration k than does basic method alone

- For previous example using power iteration, value of Rayleigh quotient at each iteration is shown below

k	$\boldsymbol{x}_k^T$		$\|\boldsymbol{y}_k\|_\infty$	$\boldsymbol{x}_k^T \boldsymbol{A} \boldsymbol{x}_k / \boldsymbol{x}_k^T \boldsymbol{x}_k$
0	0.000	1.0		
1	0.333	1.0	1.500	1.500
2	0.600	1.0	1.667	1.800
3	0.778	1.0	1.800	1.941
4	0.882	1.0	1.889	1.985
5	0.939	1.0	1.941	1.996
6	0.969	1.0	1.970	1.999

Rayleigh Quotient Iteration

- Given approximate eigenvector, Rayleigh quotient yields good estimate for corresponding eigenvalue

- Conversely, inverse iteration converges rapidly to eigenvector if approximate eigenvalue is used as shift, with one iteration often sufficing

- These two ideas combined in *Rayleigh quotient iteration*

$$\sigma_k = \boldsymbol{x}_k^T \boldsymbol{A} \boldsymbol{x}_k / \boldsymbol{x}_k^T \boldsymbol{x}_k$$

$$(\boldsymbol{A} - \sigma_k \boldsymbol{I}) \boldsymbol{y}_{k+1} = \boldsymbol{x}_k$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{y}_{k+1} / \|\boldsymbol{y}_{k+1}\|_\infty$$

starting from given nonzero vector $\boldsymbol{x}_0$

Example: Rayleigh Quotient Iteration

- Using same matrix as previous examples and randomly chosen starting vector $\boldsymbol{x}_0$, Rayleigh quotient iteration converges in two iterations

k	$\boldsymbol{x}_k^T$		σ_k
0	0.807	0.397	1.896
1	0.924	1.000	1.998
2	1.000	1.000	2.000

Deflation

Deflation

- After eigenvalue λ_1 and corresponding eigenvector $\boldsymbol{x}_1$ have been computed, then additional eigenvalues $\lambda_2, \ldots, \lambda_n$ of $\boldsymbol{A}$ can be computed by *deflation*, which effectively removes known eigenvalue

- Let $\boldsymbol{H}$ be any nonsingular matrix such that $\boldsymbol{Hx}_1 = \alpha \boldsymbol{e}_1$, scalar multiple of first column of identity matrix (Householder transformation is good choice for $\boldsymbol{H}$)

- Then similarity transformation determined by $\boldsymbol{H}$ transforms $\boldsymbol{A}$ into form

$$\boldsymbol{HA}\overset{x_1}{\cancel{\boldsymbol{H}^{-1}}} = \underline{\begin{bmatrix} \lambda_1 & \boldsymbol{b}^T \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix} Hx_1} = \lambda_1 Hx_1 \qquad \begin{pmatrix} \alpha\lambda_1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

where $\boldsymbol{B}$ is matrix of order $n-1$ having eigenvalues $\lambda_2, \ldots, \lambda_n$

Deflation, continued

- Alternative approach lets $\boldsymbol{u}_1$ be any vector such that $\boldsymbol{u}_1^T \boldsymbol{x}_1 = \lambda_1$
- Then $\boldsymbol{A} - \boldsymbol{x}_1 \boldsymbol{u}_1^T$ has eigenvalues $0, \lambda_2, \ldots, \lambda_n$
- Possible choices for $\boldsymbol{u}_1$ include
  - $\boldsymbol{u}_1 = \lambda_1 \boldsymbol{x}_1$, if $\boldsymbol{A}$ is symmetric and $\boldsymbol{x}_1$ is normalized so that $\|\boldsymbol{x}_1\|_2 = 1$
  - $\boldsymbol{u}_1 = \lambda_1 \boldsymbol{y}_1$, where $\boldsymbol{y}_1$ is corresponding left eigenvector (i.e., $\boldsymbol{A}^T \boldsymbol{y}_1 = \lambda_1 \boldsymbol{y}_1$) normalized so that $\boldsymbol{y}_1^T \boldsymbol{x}_1 = 1$
  - $\boldsymbol{u}_1 = \boldsymbol{A}^T \boldsymbol{e}_k$, if $\boldsymbol{x}_1$ is normalized so that $\|\boldsymbol{x}_1\|_\infty = 1$ and kth component of $\boldsymbol{x}_1$ is 1

QR Iteration

Simultaneous Iteration

- Simplest method for computing many eigenvalue-eigenvector pairs is *simultaneous iteration*, which repeatedly multiplies matrix times matrix of initial starting vectors

- Starting from $n \times p$ matrix $\boldsymbol{X}_0$ of rank p, iteration scheme is

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{X}_{k-1}$$

- $\text{span}(\boldsymbol{X}_k)$ converges to invariant subspace determined by p largest eigenvalues of $\boldsymbol{A}$, provided $|\lambda_p| > |\lambda_{p+1}|$

- Also called *subspace iteration*

Orthogonal Iteration

- As with power iteration, normalization is needed with simultaneous iteration

- Each column of $\boldsymbol{X}_k$ converges to dominant eigenvector, so columns of $\boldsymbol{X}_k$ become increasingly ill-conditioned basis for span$(\boldsymbol{X}_k)$

- Both issues can be addressed by computing QR factorization at each iteration

$$\begin{aligned} \hat{\boldsymbol{Q}}_k \boldsymbol{R}_k &= \boldsymbol{X}_{k-1} \\ \boldsymbol{X}_k &= \boldsymbol{A}\hat{\boldsymbol{Q}}_k \end{aligned}$$

  where $\hat{\boldsymbol{Q}}_k \boldsymbol{R}_k$ is *reduced* QR factorization of $\boldsymbol{X}_{k-1}$

- This *orthogonal iteration* converges to block triangular form, and leading block is triangular if moduli of consecutive eigenvalues are distinct

QR Iteration

- For $p = n$ and $\boldsymbol{X}_0 = \boldsymbol{I}$, matrices

  $$\boldsymbol{A}_k = \hat{\boldsymbol{Q}}_k^H \boldsymbol{A} \hat{\boldsymbol{Q}}_k$$

  generated by orthogonal iteration converge to triangular or block triangular form, yielding all eigenvalues of $\boldsymbol{A}$

- *QR iteration* computes successive matrices $\boldsymbol{A}_k$ without forming above product explicitly

- Starting with $\boldsymbol{A}_0 = \boldsymbol{A}$, at iteration k compute QR factorization

  $$\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{A}_{k-1}$$

  and form reverse product

  $$\boldsymbol{A}_k = \boldsymbol{R}_k \boldsymbol{Q}_k$$

QR Iteration, continued

- Successive matrices $\boldsymbol{A}_k$ are unitarily similar to each other

$$\boldsymbol{A}_k = \boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k^H \boldsymbol{A}_{k-1} \boldsymbol{Q}_k$$

- Diagonal entries (or eigenvalues of diagonal blocks) of $\boldsymbol{A}_k$ converge to eigenvalues of $\boldsymbol{A}$

- Product of orthogonal matrices $\boldsymbol{Q}_k$ converges to matrix of corresponding eigenvectors

- If $\boldsymbol{A}$ is symmetric, then symmetry is preserved by QR iteration, so $\boldsymbol{A}_k$ converge to matrix that is both triangular and symmetric, hence diagonal

Example: QR Iteration

- Let $\boldsymbol{A}_0 = \begin{bmatrix} 7 & 2 \\ 2 & 4 \end{bmatrix}$

- Compute QR factorization

$$\boldsymbol{A}_0 = \boldsymbol{Q}_1 \boldsymbol{R}_1 = \begin{bmatrix} .962 & -.275 \\ .275 & .962 \end{bmatrix} \begin{bmatrix} 7.28 & 3.02 \\ 0 & 3.30 \end{bmatrix}$$

and form reverse product

$$\boldsymbol{A}_1 = \boldsymbol{R}_1 \boldsymbol{Q}_1 = \begin{bmatrix} 7.83 & .906 \\ .906 & 3.17 \end{bmatrix}$$

- Off-diagonal entries are now smaller, and diagonal entries closer to eigenvalues, 8 and 3

- Process continues until matrix is within tolerance of being diagonal, and diagonal entries then closely approximate eigenvalues