Bi7740: Scientific computing

Eigenvalues and eigenvectors

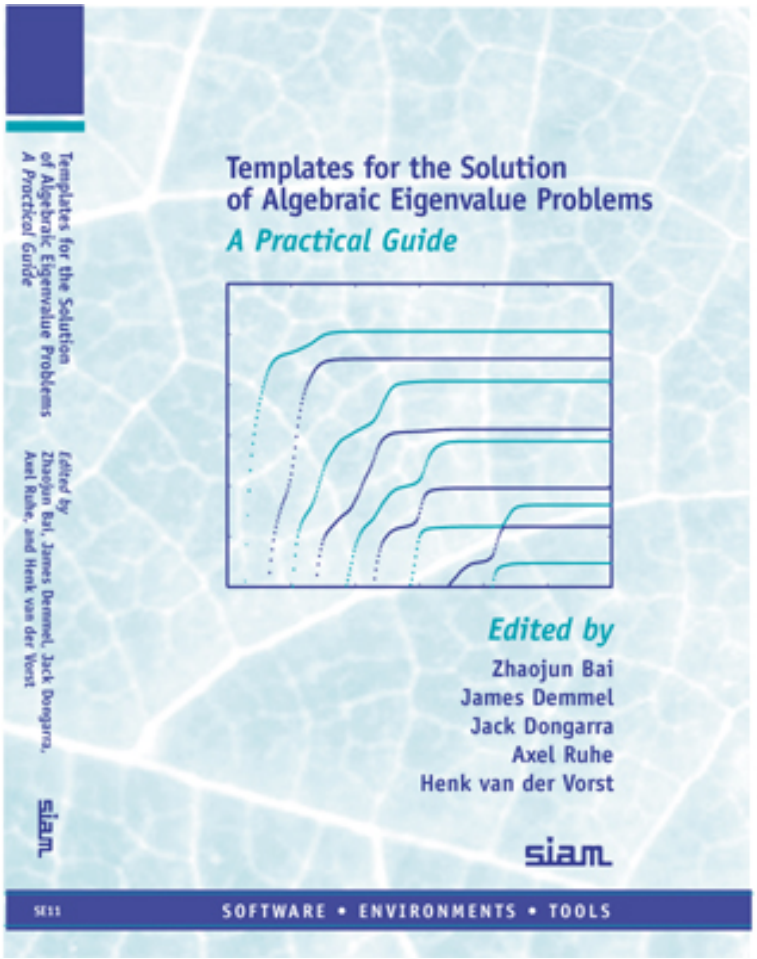
Vlad Popovici
popovici@iba.muni.cz

Institute of Biostatistics and Analyses
Masaryk University, Brno

Outline

Supplemental bibliography - online

http://web.eecs.utk.edu/~dongarra/etemplates/book.html



Outline

Eigenvalue problems

Standard eigenvalue problem

Given a square matrix $\mathbf{A} \in \mathcal{M}_{n\times n}(\mathbb{R})$, find a scalar λ and a vector $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}$, such that

$$\mathbf{Ax} = \lambda\mathbf{x}.$$

- λ is called eigenvalue and $\mathbf{x}$ is called eigenvector
- a similar "left" eigenvector can be defined as $\mathbf{y}^T\mathbf{A} = \lambda\mathbf{y}^T$, but this would be equivalent to a "right" eigenvalue problem (as above) with $\mathbf{A}^T$ as matrix
- the definition can be extended to complex-valued matrices
- λ can be complex, even if $\mathbf{A} \in \mathcal{M}_{n\times n}(\mathbb{R})$

Characteristic polynomial

- previous eq. is equivalent to $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0$ which admits nonzero solutions if and only if $(\mathbf{A} - \lambda\mathbf{I})$ is singular, i.e.

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

- $\det(\dots)$ is the characteristic polynomial of matrix $\mathbf{A}$ and its roots λ_i are the eigenvalues of $\mathbf{A}$
- (from Fundamental Theorem of Algebra) for an $n \times n$ matrix there are n eigenvalues (may not all be real or distinct)

- reciprocal: a polynomial $p(\lambda) = c_0 + c_1\lambda + c_{n-1}\lambda^{n-1} + \lambda^n$ has a companion matrix

$$\begin{bmatrix} 0 & 0 & \dots & 0 & -c_0 \\ 1 & 0 & \dots & 0 & -c_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -c_{n-1} \end{bmatrix}$$

- the characteristic polynomial is not used in numerical computation, because:
  - finding its roots may imply an infinite number of steps
  - of the sensitivity of the coefficients
  - too much work to compute the coefficients and find the roots

Example

Let $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$. The characteristic equation is

$$\det(\mathbf{A} - \lambda \mathbf{I}) = 0 \Leftrightarrow$$
$$\lambda^2 + \lambda = 0$$

with solutions $\lambda_1 = 0$ and $\lambda_2 = -1$. For eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ (non-null!):

$$(\mathbf{A} - \lambda_1 \mathbf{I})\mathbf{v}_1 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} v_{21} \\ -v_{21} \end{bmatrix} := \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

so $v_{21} = 0$. We choose v_{11} such that $\|\mathbf{v}_1\| = 1$, so $v_{11} = 1$.

Similarly, for $\lambda_2 = -1$ we get $\mathbf{v}_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$.

Example - in MATLAB

```
>> A = [0 1; 0 -1];
>> [V, L] = eig(A)  % V: eigenvectors, L: eigenvalues
V =
    1.0000   -0.7071
         0    0.7071
L =
     0     0
     0    -1
>> eig(A)           % only eigenvalues

>> roots(poly(A))   % not the way to go normally
```

Sensitivity of the characteristic polynomial

- let $\mathbf{A} = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}$ with $\epsilon > 0$ and slightly smaller than ϵ_{mach}
- the exact eigenvalues are $1 + \epsilon$ and $1 - \epsilon$
- in floating-point arithmetic,

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

with the solution 1 (double root)

- a simple eigenvalue is a simple solution of the characteristic polynomial (multiplicity of the root is 1)
- a defective matrix has eigenvalues with multiplicity larger than 1, meaning less than n independent eigenvectors
- a nondefective matrix has exactly n linearly independent eigenvectors and can be diagonalized

$$\mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \Lambda$$

where $\mathbf{Q}$ is a nonsingular matrix of eigenvectors

MATLAB: `Adiag = inv(Q)*A*Q`

Eigen-decomposition

- it follows that if $\mathbf{A}$ admits n independent eigenvectors, it can be decomposed (factorized) as

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^{-1}$$

  with $\mathbf{Q}$ having the eigenvectors of $\mathbf{A}$ as columns, and Λ a diagonal matrix with eigenvalues on the diagonal
- theoretically, $\mathbf{A}^{-1} = \mathbf{Q}\Lambda^{-1}\mathbf{Q}^{-1}$ (if $\lambda_i \neq 0$ and all eigenvalues are distinct)
- if $\mathbf{A}$ is normal ($\mathbf{A}^H\mathbf{A} = \mathbf{A}^H\mathbf{A}$) then $\mathbf{Q}$ becomes unitary
- if $\mathbf{A}$ is real symmetric, then $\mathbf{Q}$ is orthogonal

Eigenvectors

- the eigenvectors can be arbitrarily scaled
- usually, the eigenvectors are normalized, $\|\mathbf{x}\| = 1$
- the eigenspace is $\mathcal{S}_\lambda = \{\mathbf{x} | \mathbf{A}\mathbf{x} = \lambda\mathbf{x}\}$
- a subspace $\mathcal{S} \subset \mathbb{R}^n$ is invariant if $\mathbf{A}\mathcal{S} \subseteq \mathcal{S}$
- for $\mathbf{x}_i$ eigenvectors, span$(\{\mathbf{x}_i\})$ is an invariant subspace

Some useful properties

- $\det(\mathbf{A}) = \prod_{i=1}^{N} \lambda_i^{n_i}$, where n_i is the multiplicity of eigenvalue λ_i
- $\text{tr}(\mathbf{A}) = \sum_{i=1}^{N} n_i \lambda_i$
- the eigenvalues of $\mathbf{A}^{-1}$ are λ_i^{-1} (for $\lambda_i \neq 0$)
- the eigenvectors of $\mathbf{A}^{-1}$ are the same as those of $\mathbf{A}$
- $\mathbf{A}$ admits an eigen-decomposition if all eigenvalues are distinct
- if $\mathbf{A}$ is invertible it does not imply that it can be eigen-decomposed; reciprocally, if $\mathbf{A}$ admits an eigen-decomposition, it does not imply it can be inverted
- $\mathbf{A}$ can be inverted if and only if $\lambda_i \neq 0, \forall i$

Before solving an eigenvalue problem...

- do I need all the eigenvalues?
- do I need the eigenvectors as well?
- is **A** real or complex?
- is **A** small, dense or large and sparse?
- is there anything special about **A**? e.g.: symmetric, diagonal, orthogonal, Hermitian, etc etc

- conditioning of EV problem is different than conditioning of linear systems for the same matrix
- sensitivity is "not uniform" among eigenvectors/eigenvalues
- for a simple eigenvalue λ, the condition is $1/|\mathbf{y}^H\mathbf{x}|$, where $\mathbf{x}$ and $\mathbf{y}$ are the corresponding right and left normalized eigenvectors (and $\mathbf{y}^H$ is the conjugate transpose)
- so the condition is $1/\cos(\widehat{\mathbf{x},\mathbf{y}})$
- a perturbation of order ϵ in $\mathbf{A}$ may perturb the eigenvalue λ by as much as $\epsilon/\cos(\widehat{\mathbf{x},\mathbf{y}})$
- for special cases of $\mathbf{A}$, special forms of conditioning can be derived

Outline

Computation - general ideas

- a matrix **B** is similar to **A** if there exists a nonsingular matrix **T** such that $\mathbf{B} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$
- if **y** is an eigenvector of **B** then $\mathbf{x} = \mathbf{T}\mathbf{y}$ is an eigenvector of **A** and

  HOMEWORK: prove that **A** and **B** have the same eigenvalues
- transformations:
  - *shift:* $\mathbf{A} \leftarrow \mathbf{A} - \sigma\mathbf{I}$
  - *inversion:* $\mathbf{A} \leftarrow \mathbf{A}^{-1}$ (if **A** is nonsingular)
  - *power:* $\mathbf{A} \leftarrow \mathbf{A}^{k}$
  - *polynomial:* let p be a polynomial, then $\mathbf{A} \leftarrow p(\mathbf{A})$

Forms attainable by similarity

For a matrix **A** with given property, the matrices **T** and **B** exist such that $\mathbf{B} = \mathbf{T}^{-1}\mathbf{AT}$ has the desired property:

A	**T**	**B**
distinct eigenvalues	nonsingular	diagonal
real symmetric	orthogonal	real diagonal
complex Hermitian	unitary	real diagonal
normal	unitary	diagonal
arbitrary real	orthogonal	real block triangular (Schur)
arbitrary	unitary	upper triangular (Schur)
arbitrary	nonsingular	almost diagonal

If **A** is diagonal...

- the eigenvalues are the diagonal entries
- the eigenvectors are the columns of the identity matrix

If a matrix is not diagonalizable, one can obtain a *Jordan form*:

$$\begin{bmatrix} \lambda_1 & 1 & & & & & & \\ & \lambda_1 & 1 & & & & & \\ & & \lambda_1 & & & & & \\ & & & \lambda_2 & 1 & & & \\ & & & & \lambda_2 & & & \\ & & & & & \lambda_3 & & \\ & & & & & & \ddots & \\ & & & & & & & \lambda_k & 1 \\ & & & & & & & & \lambda_k \end{bmatrix}$$

If $\mathbf{A}$ is triangular (Schur form, in general)...

- eigenvalues are the elements on the diagonal
- eigenvectors are obtained as follows:
  If

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

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

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

  is the corresponding eigenvector

Symmetric matrices - Jacobi method

- idea: start with a symetric matrix $\mathbf{A}_0$ and iteratively form $\mathbf{A}_{k+1} = \mathbf{J}_k^T \mathbf{A}_k \mathbf{J}_k$, where $\mathbf{J}_k$ is a plane rotation chose to annihilate a *symmetric pair* of entries in $\mathbf{A}_k$ with the goal of diagonalizing $\mathbf{A}$
- a rotation matrix has the form

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

- the problem is to find θ
- for $\mathbf{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ and requiring that $\mathbf{J}^T\mathbf{A}\mathbf{J}$ is diagonal, we obtain

$$1 + \tan\theta \frac{a-c}{b} - \tan^2\theta = 0$$

from which we use the root with the smallest magnitude

Outline

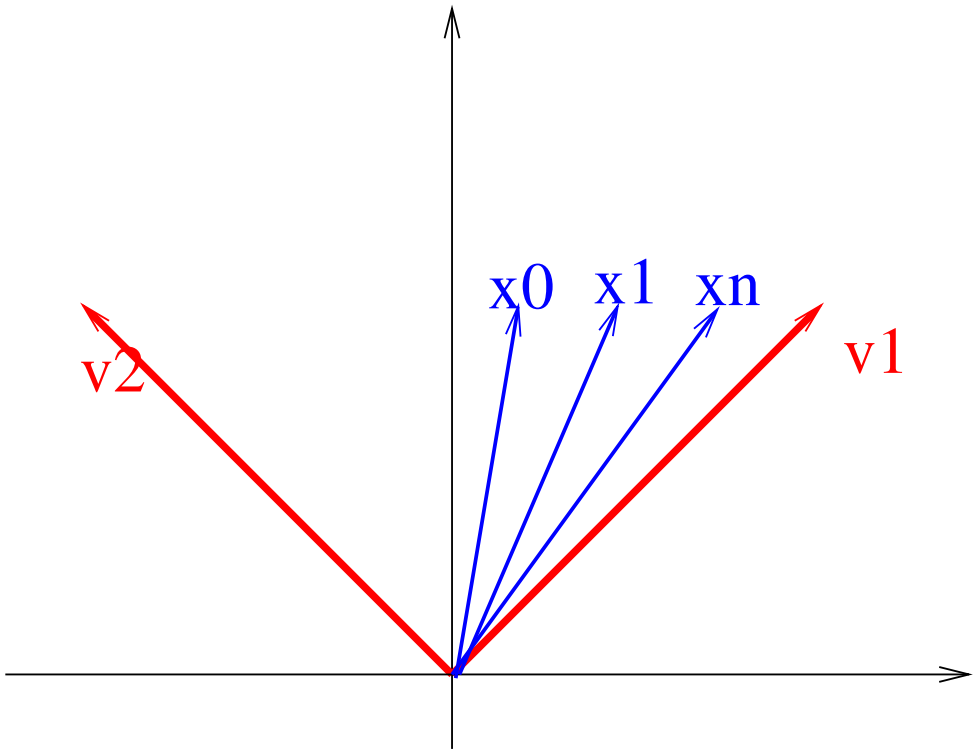
Power iterations

- is the simplest metod to compute one eigenvalue-eigenvector pair
- the matrix is repeatedly multiplied by an intial starting vector
- let λ_1 be the absolute largest eigenvalue of $\mathbf{A}$, with the corresponding eigenvector $\mathbf{v}_1$
- start with $\mathbf{x}_0 \neq \mathbf{0}$ and iterate:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} \qquad k = 1, 2, \ldots$$

- the process converges to a scaled version of $\mathbf{v}_1$ corresponding to the eigenvalue λ_1

Power iterations - geometrical interpretation



Power iterations - convergence

Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the eigenvectors of $\mathbf{A}$. Then, any vector $\mathbf{x}_0$ can be written as

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

Then

$$\begin{aligned}\mathbf{x}_k =& \mathbf{A}\mathbf{x}_{k-1} = \cdots = \mathbf{A}^k \mathbf{x}_0 = \sum_{i=1}^{n} \lambda_i^k \alpha_i \mathbf{v}_i \\ =& \lambda_1^k \left(\alpha_1 \mathbf{v}_1 + \sum_{i=2}^{n} (\lambda_i/\lambda_1)^k \alpha_i \mathbf{v}_i \right)\end{aligned}$$

and $\lim_{k\to\infty}(\lambda_i/\lambda_1)^k \to 0$ since $|\lambda_i/\lambda_1| < 1$.

Power iterations, cont'd

- theoretically, it can happen that $\mathbf{x}_0$ has no component in $\mathbf{v}_1$ (i.e. $\alpha_1 = 0$)
- the iterations cannot converge to a complex solution
- there might be several equally large and maximal eigenvalues, so the iterations converge to a linear combination of the corresponding eigenvectors
- the values of $\mathbf{x}_k$ grow geometrically with k and this can lead to over-/under-flow $\rightarrow$ use normalization: at each step normalize $\mathbf{x}_k$ by $\|\mathbf{x}_k\|_\infty$
- the rate of convergence depends on $|\lambda_2/\lambda_1|$: smaller the ratio, faster the convergence $\rightarrow$ it might be possible to find a shift by σ such that $|(\lambda_2 - \sigma)/(\lambda_1 - \sigma)| < |\lambda_2/\lambda_1|$ which accelerates convergence

Power iterations: Exercise

Implement the following procedure in MATLAB, to find the largest eigenvalue and the corresponding eignvector for a matrix $\mathbf{A}$:

- start with an initial vector $\mathbf{x}_0 \neq \mathbf{0}$, $\lambda_0 = 0$
- for $k = 1, 2, \ldots$ compute the new approximation of the
  - eigenvector: $\boxed{\mathbf{x}_k = \frac{\mathbf{A}\mathbf{x}_{k-1}}{\|\mathbf{x}_{k-1}\|_\infty}}$
  - eigenvalue: $\boxed{\lambda_k = \max\{x_{1k}, \ldots, x_{nk}\}}$
- stop iterating if a maximum number of iterations has been attained or if the changes between two consecutive iterations is below a threshold: $\|x_k - x_{k-1}\| < \epsilon$ and $|\lambda_k - \lambda_{k-1}| < \epsilon$
- scale the final approximation such that $\|\mathbf{x}_K\| = 1$

Scaling at each iteration prevents over-/under-flow and ensures that the largest component of $\mathbf{x}_k$ is λ_k.

Inverse iteration

- if the smallest eigenvalue is needed: the eigenvalues of $\mathbf{A}$ are the reciprocals of the eigenvalues of $\mathbf{A}^{-1}$. Try:
  `[v,l] = eig_power(inv(A)); l = 1/l;`
- inverse iteration scheme:

$$\mathbf{A}\mathbf{y}_k = \mathbf{x}_{k-1}$$
$$\mathbf{x}_k = \mathbf{y}_k / \|\mathbf{y}_k\|_\infty$$

- this is equivalent to power iterations applied to $\mathbf{A}$
- $\mathbf{A}^{-1}$ is not computed explicitly
- factorization of $\mathbf{A}$ is used to solve the system of linear eqs.
- converges to the eigenvector corresponding to the smallest eigenvalue
- the shifting strategy can also be applied

Shifted inverse power iterations

- if one wants eigenvalues close to a certain value s (not equal to any eigenvalue): transform the problem:

$$\mathbf{Av} = \lambda\mathbf{v} \longrightarrow (\mathbf{A} - s\mathbf{I})\mathbf{v} = (\lambda - s)\mathbf{v}$$

- the eigenvalue sought is

$$\lambda_s = \frac{1}{\text{largest eigenvalue of } (\mathbf{A} - s\mathbf{I})^{-1}} + s$$

- this method works only if there is a single eigenvalue λ_s
- try: `[v,l] = eig_power(inv(A - ... s*eye(size(A)))); l = 1/l+s;`

Rayleigh quotient

- let **A** be a real matrix with **x** an approximate eigenvector
- to find the corresponding eigenvalue λ one can solve the system

$$\mathbf{A}\mathbf{x} \approxeq \lambda\mathbf{x}$$

  for λ unknown ($n \times 1$ least squares approx. problem)
- form normal eqs.: $\mathbf{x}^T\mathbf{A}\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x}$ and obtain the LS solution

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

- this is the Rayleigh quotient

Rayleigh q., cont'd

- R.q. gives a good estimate of the eigenvalue corresponding to an eigenvector
- R.q. can be used as a shift to speed up convergence of the inverse iteration
- Rayleigh quotient iteration: for some $\mathbf{x}_0 \neq \mathbf{0}$,

$$\sigma_k = \frac{\mathbf{x}_k^T \mathbf{A} \mathbf{x}_k}{\mathbf{x}_k^T \mathbf{x}_k}$$

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

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

- usually 2-3 iterations are enough

Rayleigh q., cont'd

- R.q. iteration is very efficient for symmetric matrices
- solving a different system (different shift) at each iteration introduces some overhead, depending on the form of the matrix
- the method can be extended to complex matrices using the conjugate transpose
- the R.q. has values between the minimum and maximum eigenvalues of **A** - this is sometimes called *numerical range* (or field of values) of the matrix **A**

Deflation

- computes sequentially each of the (eigenvalue, eigenvector) pairs
- if λ_1 and $\mathbf{x}_1$ are already computed, then transform the matrix to remove them and proceed to compute λ_2 and $\mathbf{x}_2$; iterate
- this process is known as deflation
- let $\mathbf{H}$ be a nonsingular matrix such as $\mathbf{H}\mathbf{x}_1 = \alpha_1 \mathbf{e}_1$ (e.g. a Householder transformation)
- apply this transformation to $\mathbf{A}$:

$$\mathbf{H}\mathbf{A}\mathbf{H}^{-1} = \begin{bmatrix} \lambda_1 & \mathbf{b}^T \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

where $\mathbf{B}$ is a matrix of order $n - 1$ with eigenvalues $\lambda_2, \ldots, \lambda_n$

Deflation, cont'd

- then, use **B** to compute λ_2
- eigenvectors and eigenvalues of **B** are linked to those of **A** as follows:
  - if $\mathbf{y}_2$ is an eigenvector of **B** corresponding to λ_2, then

$$\mathbf{x}_2 = \mathbf{H}^{-1} \begin{bmatrix} \alpha \\ \mathbf{y}_2 \end{bmatrix}$$

    where

$$\alpha = \frac{\mathbf{b}^T \mathbf{y}_2}{\lambda_2 - \lambda_1},$$

    provided that $\lambda_1 \neq \lambda_2$.
- ...and repeat...

Alternative deflation

- let $\mathbf{u}_1$ be a vector such that $\mathbf{u}_1^T \mathbf{x}_1 = \lambda_1$
- it follows that $\mathbf{A} - \mathbf{x}_1 \mathbf{u}_1^T$ has the eigenvalues $0, \lambda_2, \ldots, \lambda_n$
- examples of $\mathbf{u}$ vectors:
  - $\mathbf{u}_1 = \lambda_1 \mathbf{x}_1$ if $\mathbf{A}$ is symmetric and $\|\mathbf{x}_1\|_2 = 1$
  - $\mathbf{u}_1 = \lambda_1 \mathbf{y}_1$ where $\mathbf{y}_1$ is the left eigenvector normalized such that $\mathbf{y}_1^T \mathbf{x}_1 = 1$
  - $\mathbf{u}_1 = \mathbf{A}^T \mathbf{e}_k$, is $\mathbf{x}_1$ is normalized such that $\|\mathbf{x}_1\|_\infty = 1$ and the $k-$th component of $\mathbf{x}_1$ is 1.

Simultaneous iteration

- in power iteration method, $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}$ converged to an eigevector
- why not using a matrix, such that $\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1}$ would converge *simultaneously* to several eigenvectors?
- start with a $n \times p$ matrix $\mathbf{X}_0$ of rank p and iterate

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

- span(bX_k) converges to an invariant space determined by the p largest eigenvalues of $\mathbf{A}$, provided that $|\lambda_p| > |\lambda_{p+1}|$
- the method is called simultaneous iteration of subspace iteration

Simultaneous iteration, cont'd

- to avoid over-/under-flow and bad conditioning with increasing k, the columns of $\mathbf{X}_k$ need to be normalized
- using the *reduced QR decomposition* avoids these problems:

$$\mathbf{Q}_k\mathbf{R}_k = \mathbf{X}_{k-1} \quad \text{QR decomposition of previous } \mathbf{X}$$
$$\mathbf{X}_k = \mathbf{A}\mathbf{Q}_k$$

- this is the orthogonal iteration scheme
- if the eigenvalues are distinct in modulus, the process converges to a block trinagular form
- if $p = n$ and $\mathbf{X}_0 = \mathbf{I}$, the series of matrices

$$\mathbf{A}_k = \mathbf{Q}_k^H\mathbf{A}\mathbf{Q}_k$$

converges to (block) triangular form yielding all the eigenvalues of $\mathbf{A}$

Simultaneous iteration, cont'd

- alternative: QR iteration: compute $\mathbf{A}_k$ without forming the product explicitly
- start with $\mathbf{A}_0 = \mathbf{A}$ and at step k:

$$\mathbf{Q}_k \mathbf{R}_k = \mathbf{A}_{k-1} \quad \text{compute the QR decomposition}$$
$$\mathbf{A}_k = \mathbf{R}_k \mathbf{Q}_k \quad \text{form the inverse product}$$

- diagonal entries (or eigenvalues of diagonal blocks) converge to eigenvalues of $\mathbf{A}$
- the product of orthogonal matrices $\mathbf{Q}_k$ converges to the matrix of corresponding eigenvectors
- if $\mathbf{A}$ is symmetric, $\mathbf{A}_k$ converges to a diagonal matrix
- special forms of $\mathbf{A}$ lead to faster convergence $\rightarrow$ use some pretransformations of the matrix to speed up

- cost of QR iteration method:
  - for symmetric matrices: $\sim \frac{4}{3}n^3$ for eigenvalues only and $\sim 9n^3$ for both eigenvalues and eigenvectors
  - for general matrices: $\sim 10n^3$ for eigenvalues only and $\sim 25n^3$ for both eigenvalues and eigenvectors
- other methods:
  - Krylov subspace methods: reduce **A** to a tridiagonal matrix and find eigen-values/-vectors by QR
  - Lanczos method for symmetric matrices
  - spectrum-slicing: for real symmetric matrices, it can find how many eigenvalues are below a given $\sigma \in \mathbb{R} \rightarrow$ by "slicing" the space, the eigenvalues can be isolated; see also the Sturm sequence

Outline

The generalized eigenvalue problem

- it has the form

$$\mathbf{Ax} = \lambda \mathbf{Bx}$$

  where **A** and **B** are $n \times n$ matrices
- if any of **A** or **B** is nonsingular, the problem can be transformed in a standard eigenvalue problem
- this is not recommended because loss of accuracy (roundoff errors) and loss of symmetry (if one of **A** or **B** is symmetric
- better: use the QZ algorithm

QZ algorithm

- if **A** and **B** are triangular, the eigenvalues are $\lambda_i = a_{ii}/b_{ii}$, for $b_{ii} \neq 0$
- the QZ algorithm reduces **A** and **B** *simultaneously* to upper triangular matrices by orthogonal transformations

Singular Value Decomposition - again

- we saw that SVD of a $m \times n$ matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

  where $\mathbf{U}$ is $m \times m$ orthogonal matrix and $\mathbf{V}$ is $n \times n$ orthogonal matrix and Σ is $m \times n$ diagonal matrix with non-negative elements on the diagonal
- this is a eigenvalue-*like* problem
- the columns of $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors, respectively and σ_{ii} are the singular values

The relation between SVD and the eigen-decomposition

- SVD can be applied to any $m \times n$ matrix, while the eigen-decomposition is applied only to square matrices
- the singular values are *non-negative* while the eigenvalues can be negative
- let $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ be SVD of $\mathbf{A} \Rightarrow$
  $\mathbf{A}^T\mathbf{A} = (\mathbf{V}\Sigma^T\mathbf{U}^T)(\mathbf{U}\Sigma\mathbf{V}^T) = \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T$
- also, $\mathbf{A}^T\mathbf{A}$ is symmetric real matrix, so it has a eigendecomposition $\mathbf{A}^T\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T$, with $\mathbf{Q}$ orthogonal. By unicity of decompositions, it follows that

$$\Sigma^T\Sigma = \Lambda$$
$$\mathbf{V} = \mathbf{Q}$$

- so $\sigma_i = \sqrt{\lambda_i}$