# H3: Eigenvalue problems

## 3.1 introduction, concept and useful properties

| | |
|---|---|
| def: eigenvalue, eigenvector | For a given nxn matrix A representing a linear trandsormation<br><br>> there is a certain vector **x**, the *eigenvector* for which:<br>$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$<br>λ is the *eigenvalue* for **x** |

### 3.1.1 characteristic polynomial

| | |
|---|---|
| characteristic polynomial p(λ) | For a square matrix A<br><br>> define: $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$<br><br>De roots of p(λ) give the eigenvalues |
| computing problems of p(λ) | Calculating the roots of p(λ) is not a good numerical way to find eigenvalues:<br>• Computing the coefficients of the characteristic polynomial for a large matrix is in itself already a substantial task<br>• The coefficients of the characteristic polynomial can be highly sensitive to small perturbations in $\mathbf{A}$ which can render their computation instable<br>• Rounding errors in finding the characteristic polynomial can destroy the accuracy of the roots<br>• Computing the roots of a polynomial of high degree is a nontrivial and substantial task |

### 3.1.2 properties and transformations

| | |
|---|---|
| λ for symmetric/Hermitian | If A is symmetric/Hermitian, all of its eigenvalues are real |
| transformations that preserve λ | • **Shift**: if $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ and $\sigma$ any scalar, then $(\mathbf{A} - \sigma\mathbf{I})\mathbf{x} = (\lambda - \sigma)\mathbf{x}$; The eigenvalues are shifted by $\sigma$, but the eigenvectors remain unchanged.<br>• **Inversion**: $\mathbf{A}^{-1}$ has the same eigenvectors as $\mathbf{A}$, and eigenvalues $1/\lambda$<br>• **Powers**: $\mathbf{A}^k$ has the same eigenvectors as $\mathbf{A}$, and eigenvalues $\lambda^k$<br>• **Polynomials**: for a general polynomial $p(t)$, $p(\mathbf{A})\mathbf{x} = p(\lambda)\mathbf{x}$. Thus the eigenvalues of a polynomial in a matrix $\mathbf{A}$ are given by the same polynomial, evaluated at the eigenvalues of $\mathbf{A}$ and the corresponding eigenvectors remain the same as those of $\mathbf{A}$.<br>• **Similarity**: A matrix $\mathbf{B}$ is *similar* to a matrix $\mathbf{A}$ if there exists an invertible matrix $\mathbf{T}$ such that<br>$\mathbf{B} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \quad (7)$<br>It follows that:<br>$\mathbf{B}\mathbf{y} = \lambda\mathbf{y} \Rightarrow \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{y} = \lambda\mathbf{y} \Rightarrow \mathbf{A}\mathbf{T}\mathbf{y} = \lambda\mathbf{T}\mathbf{y} \quad (8)$<br>In other words, $\mathbf{B} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ has the same eigenvalues as $\mathbf{A}$, but systematically transforms its eigenvectors. |

## 3.2 Calculating eigenvalues and eigenvectors

### 3.2.1 power iteration

| | |
|---|---|
| power iteration | = multiply an arbitrary nonzero vector repeatedly by the matrix<br><br>proof: Assume that we can express the starting vector $\mathbf{x}_0$ as a linear combination $\mathbf{x}_0 = \sum_{j=1}^{n} \alpha_j \mathbf{v}_j$, with $\mathbf{v}_j$ the eigenvectors of $\mathbf{A}$.<br>$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} = \mathbf{A}^2\mathbf{x}_{k-2} = \ldots = \mathbf{A}^k\mathbf{x}_0 \quad (9)$<br>$= \mathbf{A}^k \sum_{j=1}^{n} \alpha_j \mathbf{v}_j = \sum_{j=1}^{n} \alpha_j \mathbf{A}^k \mathbf{v}_j = \sum_{j=1}^{n} \lambda_j^k \alpha_j \mathbf{v}_j \quad (10)$<br>$= \lambda_1^k \left( \alpha_1 \mathbf{v}_1 + \sum_{j=2}^{n} (\lambda_j/\lambda_1)^k \alpha_j \mathbf{v}_j \right) \quad (11)$<br>Here is $\lvert\lambda_j/\lambda_1\rvert$ < 1 since $\lambda_1$ is of maximum modulus. As a result, this factor will converge to 0 when k becomes large. |

| | |
|---|---|
| problems with power iteration | It might fail because of:<br>• The starting vector $\mathbf{x}_0$ may have *no* component in the dominant eigenvector $\mathbf{v}_1$. In practice this is very unlikely and is mitigated after a few iterations due to rounding errors that introduce such a component.<br>• There may be more than 1 eigenvalue with the same maximum modulus, in which case the algorithm might converge to a linear combination of the corresponding eigenvectors.<br>• For a real matrix and real starting vector, the iteration can never converge to a complex vector. |
| **3.2.2 inverse iteration** | |
| inverse iteration | = method to find the smallest eigenvalue<br><br>eigenvalues of $A^{-1}$ are $1/\lambda$<br>> we could do power iteration of $A^{-1}$<br><br>instead: the system of linear eq. is solved at each iteration using the triangular factors<br>> eg. from LU-factorization of A<br>> using **L** and **U** we can efficiently solve **Ay = x** |
| **3.2.3 Rayleigh quotient iteration** | |
| Rayleigh quotient | the eigenvalue problem can be considered a linear least squares problem:<br>$$\mathbf{x}\lambda \cong \mathbf{A}\mathbf{x}$$<br>It's solution, the **Rayleigh quotient** is given by<br>$$\lambda = \frac{\mathbf{x^T A x}}{\mathbf{x^T x}}$$<br>Given an eigenvector, this is a good estimate for the eigenvalue |
| Rayleigh quotient iteration | = Combination of Rayleigh quotient and inverse iteration |
| **3.2.4 deflation** | |
| deflation | = process that removes a known eigenvalue of a matrix<br>> further eigenvalues and eigenvectors can be determined<br>> similar remove a root $\lambda_1$ from $p(\lambda)$ by dividing it out: $p(\lambda)/(\lambda-\lambda_1)$<br><br>This can be achieved by letting $\mathbf{u_1}$ be any vector such that \mathbf{u_1^Tx_1}=\lambda_1. Then the matrix $\mathbf{A} - \mathbf{x_1}\mathbf{u_1^T}$ has eigenvalues $0, \lambda_2, \ldots, \lambda_n$. |
| **3.2.5 QR iteration** | |
| QR-iteration | = fastest method of finding eigenvalues<br><br>For a matrix **A** define the following sequence<br>$$\begin{aligned}\mathbf{A}_m &= \mathbf{Q}_m\mathbf{R}_m \\ \mathbf{A}_{m+1} &= \mathbf{R}_m\mathbf{Q}_m\end{aligned}$$<br>with **Q** an orthogonal matrix and an upper triangular matrix **R**<br>> sequence will converge to a triangular matrix with eigenvalues of **A** on its diagonal |

## 3.3 calculating the singular value decomposition

**single value decomposition**

For an mxn matrix A this has the form:

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

With **U** an mxm orthogonal matrix
**V** an nxn orthogonal matrix
**Σ** an mxn diagonal matrix with singular values:

$$\sigma_{ij} = \begin{cases} 0, & \text{for } i \neq j \\ \sigma_i \geq 0, & \text{for } i = j \end{cases}$$

the columns $\mathbf{u}_i$ of **U** and $\mathbf{v}_i$ of **V** are the corresponding left and right singular vectors

The singular values of **A** are the nonnegative square roots of the eigenvalues of $\mathbf{A^TA}$ and the columns of **U** and **V** are orthrogonal eigenvectors of $\mathbf{AA^T}$ and $\mathbf{A^TA}$ respectively

vb:

**Example**

The singular value decomposition of the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \tag{20}$$

is given by

$$\mathbf{U\Sigma V^T} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{21}$$

This statement can be verified by explicitly calculating $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$. We begin with

$$\mathbf{A^TA} = \mathbf{AA^T} = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix} \tag{22}$$

which are equal here because $\mathbf{A}$ is a symmetric matrix. We can employ one of the methods discussed above to calculate its eigenvalues and eigenvectors. These are $\lambda_1 = \sigma_1^2 = 16$ with eigenvector $\mathbf{v_1} = [1, 1]^\mathbf{T}$ and $\lambda_2 = \sigma_2^2 = 4$ with $\mathbf{v_2} = [-1, 1]^\mathbf{T}$. The eigenvectors are easily converted to their orthonormal form, which results in

$$\mathbf{U} = \mathbf{V} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{23}$$

Now we construct $\mathbf{\Sigma}$ as $\mathrm{diag}\left(\sqrt{\lambda_1}, \sqrt{\lambda_2}\right)$ and transpose $\mathbf{V}$ in order to find the proposed SVD.

## 3.4 Software

**eigenvectors calculating**

| Method | Description |
|---|---|
| `eig` | Solve an ordinary or generalized eigenvalue problem of a square matrix. |
| `eigvals` | Compute eigenvalues from an ordinary or generalized eigenvalue problem. |
| `eigh` | Solve a standard or generalized eigenvalue problem for a complex Hermitian or real symmetric matrix. |
| `eigvalsh` | Solves a standard or generalized eigenvalue problem for a complex Hermitian or real symmetric matrix. |
| `eig_banded` | Solve real symmetric or complex Hermitian band matrix eigenvalue problem. |
| `eigvals_banded` | Solve real symmetric or complex Hermitian band matrix eigenvalue problem. |
| `eigh_tridiagonal` | Solve eigenvalue problem for a real symmetric tridiagonal matrix. |
| `eigvalsh_tridiagonal` | Solve eigenvalue problem for a real symmetric tridiagonal matrix. |
| `svd` | Compute the single decomposition matrices. |
| `svdvals` | Compute singular values of a matrix. |