# Eigenvalues and eigenvectors

## A brief theory reminder

$$\boldsymbol{A}\vec{u}=\lambda\vec{u} \;\Rightarrow\; \left(\boldsymbol{A}-\lambda\boldsymbol{I}\right)\vec{u}=\vec{0}$$

For this to have non-trivial solutions, $\det\left(\boldsymbol{A}-\lambda\boldsymbol{I}\right)=0.$

For an *n*×*n* matrix, this will be a degree *n* polynomial. It will always have exactly *n* (in general, complex) roots, but some of these may be multiple, so the number of distinct eigenvalues may be less than *n*. The set of eigenvalues is called the spectrum of the matrix.

A triangular matrix has eigenvalues equal to its diagonal elements.

A single (not multiple) eigenvalue has one and only one associated eigenvector (defined up to a factor, so a vector multiplied by a constant is still considered the same vector). An eigenvalue $\lambda$ of multiplicity $m_a(\lambda)$ has at most $m_a(\lambda)$ linearly independent eigenvectors, but it can be as few as 1. The number of linearly independent eigenvectors associated with a particular eigenvalue, $m_g(\lambda)$, is called its geometric multiplicity and $m_a(\lambda)$ is called its algebraic multiplicity [$1\le m_g(\lambda)\le m_a(\lambda)$].

An eigenvector associated with a particular eigenvalue is always linearly independent of all the vectors associated with all other distinct eigenvalues.

If all the geometric multiplicities are the same as the algebraic multiplicities, then there are *n* linearly independent eigenvectors – as many as the dimensionality of the space in which they are defined, so they form a full basis. A sufficient (but not necessary) condition of this is that all the eigenvalues are distinct.

Denote these eigenvectors $\vec{v}^{(j)} = \left( v_1^{(j)}, v_2^{(j)}, \ldots, v_n^{(j)} \right), \quad j = 1, \ldots, n$

In this case, if we consider an $n \times n$ matrix $\boldsymbol{P}$ whose columns are the $n$ linearly independent eigenvectors, $\boldsymbol{P} = \left( \vec{v}^{(1)}, \vec{v}^{(2)}, \ldots, \vec{v}^{(n)} \right)$,

$$\boldsymbol{A}\,\boldsymbol{P} = \boldsymbol{A}\left( \vec{v}^{(1)}, \vec{v}^{(2)}, \ldots, \vec{v}^{(n)} \right) = \left( \lambda_1 \vec{v}^{(1)}, \lambda_2 \vec{v}^{(2)}, \ldots, \lambda_n \vec{v}^{(n)} \right)$$

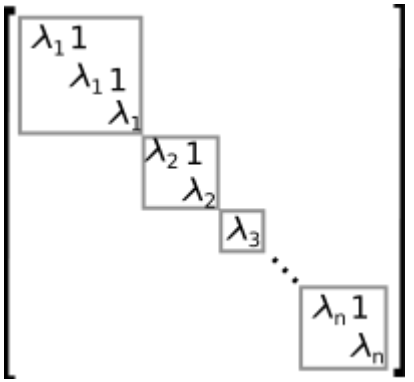
$$\boldsymbol{P}^{-1}\,\boldsymbol{A}\,\boldsymbol{P} = \boldsymbol{P}^{-1}\left( \lambda_1 \vec{v}^{(1)}, \lambda_2 \vec{v}^{(2)}, \ldots, \lambda_n \vec{v}^{(n)} \right) = \begin{pmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ 0 & 0 & \lambda_3 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 0 & \lambda_n \end{pmatrix}$$

For any invertible matrix ***X***, $X^{-1}AX$ has the same eigenvalues as ***A***.

Consider $X^{-1}\vec{v}^{(j)}$.

$$X^{-1}AX\left(X^{-1}\vec{v}^{(j)}\right)=X^{-1}A\vec{v}^{(j)}=\lambda_j X^{-1}\vec{v}^{(j)}.$$

This is called a similarity transformation and matrices $A$ and $B=X^{-1}AX$ are called similar. A matrix is called diagonalizable if it is similar to a diagonal matrix. As we have seen, *n*×*n* matrices with *n* linearly independent eigenvectors are diagonalizable, but if there are fewer than n eigenvectors (defective matrices), then the closest they get to diagonal in a similarity transformation is the Jordan normal form (note: Camille Jordan vs. Wilhelm Jordan in Gauss-Jordan):




Wikipedia

Symmetric real matrices (or Hermitian complex matrices) are always diagonalizable, their eigenvectors corresponding to different eigenvalues are orthogonal and those corresponding to a multiple eigenvalue can be made orthogonal.

$$\begin{aligned} \boldsymbol{A}\cdot\vec{u}=\lambda_1\vec{u} \;&\Rightarrow\; \vec{v}\cdot\boldsymbol{A}\cdot\vec{u}=\lambda_1\vec{u}\cdot\vec{v} \\ \boldsymbol{A}\cdot\vec{v}=\lambda_2\vec{v} \;&\Rightarrow\; \vec{u}\cdot\boldsymbol{A}\cdot\vec{v}=\lambda_2\vec{u}\cdot\vec{v} \end{aligned} \qquad \vec{v}\cdot\boldsymbol{A}\cdot\vec{u}=\vec{u}\cdot\boldsymbol{A}\cdot\vec{v}$$

If all the eigenvectors are normalized, the corresponding transformation matrix $\boldsymbol{P}=\left(\vec{v}^{(1)},\vec{v}^{(2)},\ldots,\vec{v}^{(n)}\right)$ is orthogonal: $\boldsymbol{P}^T\boldsymbol{P}=\boldsymbol{I}$. Orthogonal matrices are often denoted $\boldsymbol{Q}$.

## Estimating eigenvalues: Gershgorin circle theorem

Consider matrix $A=(a_{ij})$. The Gershgorin disk for row *i* is centred at $a_{ii}$ and has radius $R_i=\Sigma_{i\neq j}|a_{ij}|$. Every eigenvalue lies within one of Gershgorin disks.

Same for the columns.

Moreover, if one disk does not overlap with the rest, there is one and only one eigenvalue inside it. More generally, if the union of *k* disks does not overlap with the union of the rest *n*-*k* disks, then the first union contains exactly *k* eigenvalues and the second one exactly *n*-*k* eigenvalues.

Useful in two respects: 1) to estimate where the eigenvalues lie (useful for some methods); 2) many methods diagonalize the matrix approximately. Gives some idea how accurate these eigenvalues are.

# Back to the power method

$$|\lambda_1|>|\lambda_2|>|\lambda_3|\geq|\lambda_4|\geq\ldots$$

$$\vec{v}^{(k)}=A^k\vec{v}^{(0)}=A^k\sum_{i=1}^{N}c_i\vec{u}_i=\sum_{i=1}^{N}\lambda_i^k c_i\vec{u}_i=\lambda_1^k\left(c_1\vec{u}_1+\sum_{i=2}^{N}c_i\left(\lambda_i/\lambda_1\right)^k\vec{u}_i\right)$$

After a large number of iterations, will be parallel to $\vec{u}_1$.

Note it would be not quite correct to say that this **converges** to a particular vector, because, obviously, there is a factor $\lambda_1^k c_1$ that changes with every iteration. Even if we normalize, considering $\vec{v}^{(k)}/\|\vec{v}^{(k)}\|$, this will give the factor $\lambda_1^k/|\lambda_1^k|$ that changes if $\lambda_1$ is negative or complex.

To calculate the eigenvalue, consider the ratio (for an arbitrary *j*)

$$\frac{v_j^{(k+1)}}{v_j^{(k)}}=\frac{\lambda_1^{k+1}\left(c_1u_{1,j}+\sum_{i=2}^{N}c_i\left(\lambda_i/\lambda_1\right)^{k+1}u_{i,j}\right)}{\lambda_1^{k}\left(c_1u_{1,j}+\sum_{i=2}^{N}c_i\left(\lambda_i/\lambda_1\right)^{k}u_{i,j}\right)}\approx\lambda_1\frac{1+c_2\left(\lambda_2/\lambda_1\right)^{k+1}u_{i,j}/\left(c_1u_{1,j}\right)}{1+c_2\left(\lambda_2/\lambda_1\right)^{k}u_{i,j}/\left(c_1u_{1,j}\right)}$$

$$=\lambda_1\left[1+O\left(\left|\lambda_2/\lambda_1\right|^k\right)\right]$$

Converges slowly if the first and the second highest λ are close in abs. value

To speed up convergence somewhat: consider Rayleigh quotient

$$\frac{\vec{v}^{(k)}\cdot\vec{v}^{(k+1)}}{\vec{v}^{(k)}\cdot\vec{v}^{(k)}}=\frac{\lambda_1^k\lambda_1^{k+1}\left(c_1\vec{u}_1+\sum_{i=2}^N c_i\left(\lambda_i/\lambda_1\right)^k\vec{u}_i\right)\cdot\left(c_1\vec{u}_1+\sum_{i=2}^N c_i\left(\lambda_i/\lambda_1\right)^{k+1}\vec{u}_i\right)}{\lambda_1^k\lambda_1^k\left(c_1\vec{u}_1+\sum_{i=2}^N c_i\left(\lambda_i/\lambda_1\right)^k\vec{u}_i\right)\cdot\left(c_1\vec{u}_1+\sum_{i=2}^N c_i\left(\lambda_i/\lambda_1\right)^k\vec{u}_i\right)}$$

If the matrix is symmetric so different eigenvectors are orthogonal, then

$$\frac{\vec{v}^{(k)}\cdot\vec{v}^{(k+1)}}{\vec{v}^{(k)}\cdot\vec{v}^{(k)}}=\lambda_1\left[1+O\left(\left|\lambda_2/\lambda_1\right|^{2k}\right)\right]$$

Convergence rate is twice as high.

To avoid overflow or underflow, normalize, e.g., by the largest component of the vector.

Obviously, when calculating the eigenvalue, calculate the ratio before normalizing, i.e., given $\vec{v}^{(k)}$, first compute $\tilde{\vec{v}}^{(k+1)}=\boldsymbol{A}\,v^{(k)}$, then $\left(\tilde{v}_j^{(k+1)}/v_j^{(k)}\right)$ or $\dfrac{\vec{v}^{(k)}\cdot\tilde{\vec{v}}^{(k+1)}}{\vec{v}^{(k)}\cdot\vec{v}^{(k)}}$ and only then normalize $\vec{v}^{(k+1)}=\tilde{\vec{v}}^{(k+1)}/\max\ v_j^{(k+1)}$

# Modifications of the power method

## Inverse iteration

The smallest (by absolute value) eigenvalue is the largest eigenvalue of the inverse matrix.

$$\vec{\tilde{v}}^{(k+1)} = \boldsymbol{A}^{-1}\,\vec{v}^{(k)}$$

In practice, solve the equation $\boldsymbol{A}\,\vec{\tilde{v}}^{(k+1)} = \vec{v}^{(k)}$ or $\boldsymbol{L}\boldsymbol{U}\,\vec{\tilde{v}}^{(k+1)} = \vec{v}^{(k)}$

## Shifting

If the eigenvalues of ***A*** are $\lambda_i$, then the eigenvalues of ***A***-*s**I*** are $\lambda_i$-*s*.

Useful for several purposes:

1) in combination with inverse iteration, to find the eigenvalue closest to a given number *s*. After shifting, will be closest to zero, i.e., lowest by absolute value.

$$\left(\boldsymbol{A} - s\,\boldsymbol{I}\right)\vec{\tilde{v}}^{(k+1)} = \vec{v}^{(k)}$$

2) after finding $\lambda_1$, shift by $\lambda_1$ to find the eigenvalue at the opposite end of the spectrum

## Shifting (continued)

3) minimize the ratio $\lambda_2/\lambda_1$ to improve convergence. Best *s* is half-way between $\lambda_2$ and $\lambda_n$ assuming $\lambda_1 > \lambda_2 \geq \lambda_3 \geq ... \geq \lambda_n$ (or the other way around)

How do we find $\lambda_2$?

1) for a symmetric ***A***, start with a vector orthogonal to $\vec{u}_1$.

$$\vec{v}^{(0)} = \vec{z} - \frac{\vec{z}\cdot\vec{u}_1}{\vec{u}_1\cdot\vec{u}_1}\vec{u}_1 \text{ for arbitrary } z$$

Since, as mentioned, round-off errors introduce a component $\|\vec{u}_1$,

re-orthogonalize occasionally.

2) Deflation. Suppose we can find matrix ***H*** such that $\boldsymbol{H}\vec{u}_1 = \alpha\vec{e}_1$,

where $\vec{e}_1 = (1\,0\,0\,...\,0)^T$. Consider similarity transformation $\boldsymbol{HAH}^{-1}$.

The first column of the resulting matrix is $\boldsymbol{HAH}^{-1}\vec{e}_1 = \boldsymbol{HA}\vec{u}_1/\alpha = \boldsymbol{H}\frac{\lambda_1}{\alpha}\vec{u}_1 = \lambda_1\vec{e}_1$

So $\boldsymbol{HAH}^{-1}=\begin{pmatrix}\lambda_1 & \vec{b}^T\\ \vec{0} & \boldsymbol{B}\end{pmatrix}$. Matrix $\boldsymbol{B}$ will now have eigenvalues $\lambda_2$ ,...., $\lambda_n$

If the eigenvector of B corresponding to $\lambda_2$ is $\vec{y}_2$ ,

$$\begin{pmatrix}\lambda_1 & \vec{b}^T\\ \vec{0} & \boldsymbol{B}\end{pmatrix}\begin{pmatrix}\beta\\ \vec{y}_2\end{pmatrix}=\begin{pmatrix}\lambda_1\beta+\vec{b}\cdot\vec{y}_2\\ \lambda_2\vec{y}_2\end{pmatrix}=\begin{pmatrix}\lambda_2\beta\\ \lambda_2\vec{y}_2\end{pmatrix}\Rightarrow\ \beta=\vec{b}\cdot\vec{y}_2/(\lambda_2-\lambda_1)$$

## Rayleigh quotient iteration

$$\sigma_k=\frac{\vec{v}^{(k)}\cdot\boldsymbol{A}\cdot\vec{v}^{(k)}}{\vec{v}^{(k)}\cdot\vec{v}^{(k)}}\qquad\qquad(\boldsymbol{A}-\sigma_k\boldsymbol{I})\vec{v}^{(k+1)}=\vec{v}^{(k)}$$

While the disadvantage is that every time a system with a new matrix needs to be solved, convergence is cubic!

---

While the derivation of the power method was done assuming that there are n linearly independent eigenvectors, it does work when this is not the case.

The power method is only really appropriate when one needs to find one or a small number of eigenvalues and eigenvectors. If all eigenvalues and/or eigenvectors are desired, different methods should be used.

Such methods are based on similarity transformations that are designed to bring the matrix closer to diagonal with every iteration.

I will consider only methods for symmetric matrices (generalizable to Hermitian). Much simpler and work better than for general matrices.

# Jacobi method

Starting with $\boldsymbol{A}_0 = \boldsymbol{A}$, do a series of transformations $\boldsymbol{A}_{k+1} = \boldsymbol{P}_k^T \boldsymbol{A}_k \boldsymbol{P}_k$,

each of which corresponds to a plane rotation.

Rotation in 3D by angle $\theta$ around the z axis is

$$\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Generalize:

$$\boldsymbol{P}_{pq} = \begin{pmatrix} 1 & \dots & 0 & \dots & \dots & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & c & \dots & 0 & \dots & s & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 1 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -s & \dots & \dots & \dots & c & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \dots & \dots & \dots & 0 & \dots & 1 \end{pmatrix} \begin{matrix} \\ \\ p \\ \\ \\ \\ q \\ \\ \\ \end{matrix}$$

$p$ $q$

Consider 2x2 matrices.

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix}\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}\begin{pmatrix} c & s \\ -s & c \end{pmatrix}=\begin{pmatrix} c & -s \\ s & c \end{pmatrix}\begin{pmatrix} a_{11}c-a_{12}s & a_{11}s+a_{12}c \\ a_{12}c-a_{22}s & a_{12}s+a_{22}c \end{pmatrix}$$

$$=\begin{pmatrix} a_{11}c^2-2a_{12}cs+a_{22}s^2 & a_{12}(c^2-s^2)+(a_{11}-a_{22})cs \\ a_{12}(c^2-s^2)+(a_{11}-a_{22})cs & a_{11}s^2+2a_{12}cs+a_{22}c^2 \end{pmatrix}$$

Make the off-diagonal elements zero:

$$a_{12}(c^2-s^2)+(a_{11}-a_{22})cs=0 \;\Rightarrow\; a_{12}(1-(s/c)^2)+(a_{11}-a_{22})(s/c)=0$$

$$s/c=t \qquad \frac{a_{22}-a_{11}}{2a_{12}}=\theta \qquad t^2+2t\theta-1=0$$

Choose the root smaller by absolute value:

$$t=\begin{cases} \dfrac{1}{\theta+\sqrt{\theta^2+1}}, & \theta>0, \\ \dfrac{1}{\theta-\sqrt{\theta^2+1}}, & \theta<0. \end{cases} \qquad t=\frac{\operatorname{sign}\,\theta}{|\theta|+\sqrt{\theta^2+1}}$$

$$c=\frac{1}{\sqrt{t^2+1}} \qquad s=tc$$

For n > 2, the expressions for *t*, *c*, *s* are still valid. $\theta = \frac{a_{qq} - a_{pp}}{2 a_{pq}}$

This will make $a'_{pq} = 0$, but, unfortunately, it also changes other off-diagonal elements, including those that may have been made zero before.

$$a'_{rp} = c a_{rp} - s a_{rq} \qquad a'_{rq} = c a_{rq} + s a_{rp} \qquad r \neq p,\ r \neq q$$

But still, if we consider the sum of squares of the affected off-diagonal elements,

$$S = \sum_{r \neq p,q} 2\left(|a_{rp}|^2 + |a_{rq}|^2\right) + 2|a_{pq}|^2$$

$$S' = \sum_{r \neq p,q} 2\left(|a'_{rp}|^2 + |a'_{rq}|^2\right) = \sum_{r \neq p,q} 2\left(|a_{rp}|^2 + |a_{rq}|^2\right) = S - 2|a_{pq}|^2$$

Once we are satisfied that the off-diagonal matrix elements are sufficiently low, the diagonal matrix elements are the eigenvalues of the original matrix. The eigenvectors are the columns of $\boldsymbol{V} = \boldsymbol{P}_1 \boldsymbol{P}_2 \boldsymbol{P}_3 \ldots$

Order of transformations $\boldsymbol{P}_{pq}$. Jacobi chose p,q corresponding to the largest $|a_{pq}|$ at each step. This is time-consuming [$O(N^2)$] and not necessary. Just choose transformations in strict order, e.g., $\boldsymbol{P}_{12}$, $\boldsymbol{P}_{13}$,..., $\boldsymbol{P}_{1N}$; $\boldsymbol{P}_{23}$, $\boldsymbol{P}_{24}$,...

It turns out it is much better to first reduce a matrix to a tridiagonal form and only then start iterating.

Givens method:

Do similar Jacobi rotations, but for a different purpose.

First, use $\boldsymbol{P}_{23}$ to make $a'_{13}=a'_{31}=0$; then $\boldsymbol{P}_{24}$ to make $a'_{14}=a'_{41}=0$; ...

When we then use $\boldsymbol{P}_{34}$ to make $a'_{24}=a'_{42}=0$, $a'_{13}$ and $a'_{14}$ do not reappear, because both of them are already zero.

$n^2/2$ rotations, $O(n^3)$ operations in total.

A somewhat faster (~ factor of 2) approach: Householder reduction

Householder matrix: $\boldsymbol{P} = \boldsymbol{I} - 2\,\vec{w}\otimes\vec{w} \qquad \|\vec{w}\|^2 = 1 \qquad \boldsymbol{P}^T = \boldsymbol{P}$

Check orthogonality:

$$\boldsymbol{P}^2 = (\boldsymbol{I} - 2\,\vec{w}\otimes\vec{w})(\boldsymbol{I} - 2\,\vec{w}\otimes\vec{w}) = \boldsymbol{I} - 4\,\vec{w}\otimes\vec{w} + 4(\vec{w}\otimes\vec{w})(\vec{w}\otimes\vec{w}) \rightarrow$$

$$\delta_{ij} - 4\,w_i\,w_j + 4\,w_i\,w_k\,w_k\,w_j = \delta_{ij}$$

Let's see what it does to a vector $\vec{x}$. If it is orthogonal to $\vec{w}$, does not get transformed. If it is parallel, $\boldsymbol{P}\,\vec{x} = \vec{x} - 2\,\vec{w}(\vec{w}\cdot\vec{x}) = \vec{x} - 2\,\vec{w}\|\vec{x}\| = -\vec{x}$.

Reflection in the plane perpendicular to $\vec{w}$.

$$\boldsymbol{P} = \boldsymbol{I} - \vec{u}\otimes\vec{u}/H,\ H = \frac{1}{2}\|u\|^2$$

$$\boldsymbol{P} = \boldsymbol{I} - \vec{u} \otimes \vec{u} / H\,,\ H = \frac{1}{2}\|u\|^2$$

At stage *m* (*m*=1,2,...,*n*-2),

$$\vec{u}^T = (a_{i1}, a_{i2}, \ldots, a_{i,i-2}, a_{i,i-1} \pm \sqrt{\sigma}, 0, \ldots, 0),\ i = n - m + 1,\ \sigma = (a_{i1})^2 + \ldots + (a_{i,i-1})^2$$

$$\vec{u}^T \cdot (a_{i1}, \ldots, a_{in})^T = \sigma \pm \sqrt{\sigma}\, a_{i,i-1} \qquad H = \frac{1}{2}(2\sigma \pm 2\sqrt{\sigma}\, a_{i,i-1})$$

$$\boldsymbol{P}(a_{i1}, \ldots, a_{in})^T = (a_{i1}, \ldots, a_{i,n})^T - (a_{i1}, a_{i2}, \ldots, a_{i,i-2}, a_{i,i-1} \pm \sqrt{\sigma}, 0, \ldots, 0)^T$$

$$= (0, \ldots, \mp\sqrt{\sigma}, a_{i,i}, \ldots, a_{i,n})^T$$

$$\boldsymbol{P}(x_{i1}, \ldots, x_{in})^T = (\mathrm{s\ t\ u\ f\ f}, x_{i,i}, \ldots, x_{i,n})^T$$

Sign chosen identical to that of the $a_{i,i-1}$ element

Note the ith row is preserved, so it's still $(a_{i1}, \ldots, a_{in})$.

$$(a_{i1}, \ldots, a_{i,n})\boldsymbol{P} = (0, \ldots, \mp\sqrt{\sigma}, a_{i,i}, \ldots, a_{i,n}) \qquad (x_{i1}, \ldots, x_{in})\boldsymbol{P} = (\mathrm{s\ t\ u\ f\ f}, x_{i,i}, \ldots, x_{i,n})$$

So the effect of ***PAP*** is that it makes the first i-2 elements in the ith column and ith row =0 and at the same time preserves elements in the lower-right block with the corner at (i,i). Thus, we end up with a tridiagonal matrix in n-2 iterations, each involving $O(n^2)$ operations.

# Tridiagonal matrices

$$\begin{pmatrix} a_1 & c_1 & 0 & 0 & \dots & 0 & 0 \\ c_1 & a_2 & c_2 & 0 & \dots & 0 & 0 \\ 0 & c_2 & a_3 & c_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & c_{n-1} & a_n \end{pmatrix}$$

$$p_0(\lambda) = 1$$

$$p_1(\lambda) = \lambda - a_1$$

$$p_2(\lambda) = (\lambda - a_2)(\lambda - a_1) - c_1^2$$

$$p_i(\lambda) = (\lambda - a_i)\, p_{i-1}(\lambda) - c_{i-1}^2\, p_{i-2}(\lambda)$$

Zeros of $p_i$ are separated by those of $p_i$ -1 and if $p_n$ ($\gamma$) 0, the number of eigenvalues > $\gamma$ is equal to the number of sign variations in the sequence $p_n(\gamma)$, $p_n$-1($\gamma$),...,$p_1(\gamma)$, 1.

# QL iterations

Any matrix ***A*** can be decomposed as $\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{L} \Rightarrow \boldsymbol{L}=\boldsymbol{Q}^T\boldsymbol{A}$

Consider $\boldsymbol{A}'=\boldsymbol{L}\boldsymbol{Q}=\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q}$

Start with $\boldsymbol{A}_0=\boldsymbol{A}$. A sequence of transformations:

$$\boldsymbol{A}_s=\boldsymbol{Q}_s\boldsymbol{L}_s \quad \text{(decomposition)}$$

Form new matrix $\boldsymbol{A}_{s+1}=\boldsymbol{L}_s\boldsymbol{Q}_s$

This sequence actually converges to a lower triangular matrix.

Just like in power method, convergence can be slow if eigenvalues are close. Accelerate by shifting:

$$\boldsymbol{A}_s-k_s\boldsymbol{I}=\boldsymbol{Q}_s\boldsymbol{L}_s \qquad \boldsymbol{A}_{s+1}=\boldsymbol{L}_s\boldsymbol{Q}_s+k_s\boldsymbol{I}=\boldsymbol{Q}_s^T\boldsymbol{A}\boldsymbol{Q}_s$$

$$\boldsymbol{Q}_s^T=\boldsymbol{P}_1^{(s)}\boldsymbol{P}_2^{(s)}\ldots\boldsymbol{P}_{n-1}^{(s)} \qquad \boldsymbol{P}_i^{(s)} \text{ makes } a_{i,i+1} \text{ zero}$$

$k_s$ is chosen as the eigenvalue of matrix $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ closer to $a_{11}$.