NLA 2021-2022

Martin Sombra

Linear equation solving

We want to solve

$$A\,x = b$$

for a nonsingular $n \times n$ matrix A and an n-vector b

Gaussian elimination consists in computing a factorization

$$A = P\,L\,U$$

with P a permutation, L a unit lower triangular, and U an upper triangular

Allows to solve $A\,x = b$ by solving the simpler equations

1. $P\,z = b$ (permute the entries of b)
2. $L\,y = z$ (forward substitution)
3. $U\,x = y$ (backwards substitution)

Constructing the PLU factorization

Set

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix}.$$

For $n = 1$

$$P = L = \begin{bmatrix} 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} a_{1,1} \end{bmatrix}$$

Constructing the PLU factorization (cont.)

Let $n \geqslant 2$ and choose k such that $a_{k,1} \neq 0$

Gaussian elimination with partial pivoting (GEPP)

k such that $|a_{k,1}|$ is maximal

Swap rows 1 and k premultiplying by the permutation matrix

$$P_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & & & 0 & & & \\ \vdots & & \ddots & & \vdots & & & \\ 0 & & & 1 & 0 & & & \\ 1 & 0 & \cdots & 0 & 0 & & & \\ 0 & & & & & 1 & & \\ \vdots & & & & & & \ddots & \\ 0 & & & & & & & 1 \end{bmatrix}.$$

Consider the 2×2-block

$$P_1^T A = \begin{bmatrix} a_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$$

Constructing the PLU factorization (cont.)

Then

$$\begin{bmatrix} a_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & \mathbb{0} \\ L_{2,1} & \mathbb{1}_{n-1} \end{bmatrix} \begin{bmatrix} u_{1,1} & U_{1,2} \\ \mathbb{0} & \widetilde{A}_{2,2} \end{bmatrix}$$

with

$$u_1 = a_{1,1}, \quad L_{2,1} = a_{1,1}^{-1} A_{2,1}, \quad U_{1,2} = A_{1,2} \quad \text{and} \quad \widetilde{A}_{2,2} = A_{2,2} - L_{2,1}\, U_{1,2}$$

The $(n-1) \times (n-1)$ matrix $\widetilde{A}_{2,2}$ is the *Schur complement*

By the inductive hypothesis (case $n-1$):

$$\widetilde{A}_{2,2} = \widetilde{P}\, \widetilde{L}\, \widetilde{U}$$

Constructing the PLU factorization (cont.)

Then

$$\begin{aligned} P_1^T A &= \begin{bmatrix} 1 & \mathbb{0} \\ L_{2,1} & \mathbb{1}_{n-1} \end{bmatrix} \begin{bmatrix} u_{1,1} & U_{1,2} \\ \mathbb{0} & \widetilde{P}\, \widetilde{L}\, \widetilde{U} \end{bmatrix} \\ &= \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & \widetilde{P} \end{bmatrix} \begin{bmatrix} 1 & \mathbb{0} \\ \widetilde{P}^T L_{2,1} & \widetilde{L} \end{bmatrix} \begin{bmatrix} u_{1,1} & U_{1,2} \\ \mathbb{0} & \widetilde{U} \end{bmatrix} \end{aligned}$$

Hence $A = P\, L\, U$ with

$$P = P_1 \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & \widetilde{P} \end{bmatrix}, \quad L = \begin{bmatrix} 1 & \mathbb{0} \\ \widetilde{P}^T L_{2,1} & \widetilde{L} \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} u_{1,1} & U_{1,2} \\ \mathbb{0} & \widetilde{U} \end{bmatrix}.$$

GEPP $\rightsquigarrow$ $L = [l_{i,j}]_{i,j}$ with $|l_{i,j}| \leqslant 1$ for all i, j

Complete pivoting

Gaussian elimination with complete pivoting (GECP): takes $|a_{k,l}|$ maximal among all the entries
$\rightsquigarrow$ swaps the rows 1 and k, and the columns 1 and l

Gives a factorization

$$A = P_1\, L\, U\, P_2$$

with P_1 and P_2 permutations

Can be more *numerically stable* but is also *more expensive* (in terms of speed)

The GEPP algorithm

for $i = 1, \ldots, n-1$

swap row k and row i of A and L for k such that $|a_{k,i}| = \max_{i \leqslant p \leqslant n} |a_{p,i}|$

for $j = i+1, \ldots, n$ (compute column i of L)
$l_{j,i} \leftarrow \frac{a_{j,i}}{a_{i,i}}$

for $j = 1, \ldots, n$ (compute row i of U)
$u_{i,j} \leftarrow a_{i,j}$

for $j, k = i+1, \ldots, n$ (update $A_{2,2}$)
$u_{j,k} \leftarrow a_{j,k} - l_{j,i}\, u_{i,k}$

Modified GEPP

Useful remark: the i-th column of A only used to compute the i-th column of L, and the i-th row of A only used to compute the i-th row of U

for $i = 1, \dots, n-1$

swap row k and row i of A and L for k such that $|a_{k,i}| = \max_{i \leqslant p \leqslant n} |a_{p,i}|$

for $j = i+1, \dots, n$

$l_{j,i} \leftarrow \frac{a_{j,i}}{a_{i,i}}$, (replace by $a_{j,i}$)

~~for $j = 1, \dots, n$~~

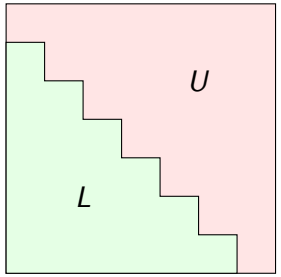
~~$u_{i,j} \leftarrow a_{i,j}$~~

for $j, k = i+1, \dots, n$

$u_{j,k} \leftarrow a_{j,k} - l_{j,i}\, u_{i,k}$ (replace by $a_{j,k}$, $a_{j,i}$ and $a_{i,k}$)

Modified GEPP (cont.)

We need no extra space to store L and U!



Complexity

The complexity (# flops) of GEPP on $n \times n$ matrices is

$$\sum_{i=1}^{n-1} \left(\sum_{j=i+1}^{n} 1 + \sum_{j=i+1}^{n} \sum_{k=i+1}^{n} 2 \right) = \sum_{i=1}^{n-1} ((n-i) + 2(n-i)^2)$$

$$= \left(\sum_{i=1}^{n-1} i \right) + 2 \left(\sum_{i=1}^{n-1} i^2 \right)$$

At this point, recall the asymptotic formulae for $k \geqslant 1$:

$$\sum_{i=1}^{n} i^k = \frac{n^{k+1}}{k+1} + O(n^k),$$

Hence this complexity is $\frac{2}{3} n^3 + O(n^2)$. Forward and backward substitutions have each a complexity of

$$n^2 + O(n)$$

Hence GEPP solves the equation $Ax = b$ with

$$\frac{2}{3} n^3 + O(n^2) \text{ flops}$$

Floating point arithmetic

A *floating point number* is

$$f = \pm 0.d_1 d_2 \dots d_t \times \beta^e$$

$$\beta \geqslant 2, \quad 0 \leqslant d_i < \beta \text{ with } d_1 \neq 0 \quad \text{and} \quad L \leqslant e \leqslant U$$

$\beta \geqslant 2$ the *base*, L the *underflow* and U the *overflow*

The *range* is

$$\beta^{L-1} \leqslant |f| \leqslant \beta^U (1 - \beta^{-t}).$$

Floating point operations are defined by picking the closest floating point number:

$$a \odot b \coloneqq \mathrm{fl}(a \cdot b)$$

Machine epsilon: a bound for the relative error of roundoffs:

$$\frac{|a - \mathrm{fl}(a)|}{|a|} \leqslant \varepsilon = \frac{\beta^{1-t}}{2}$$

Single precision IEEE standard

s	e	q
1	8	23
sign	exponent	coefficient/mantissa

The *coded number* is

$$f = (-1)^s(1+q)\,2^{e-127}$$

Corresponds to $t = 24$, $\beta = 2$, $L = -126$ and $U = 127$

The maximal *relative error* is

$$2^{-24} \approx 6 \cdot 10^{-8}$$

and the *range* is

$$2^{-127} \approx 6 \cdot 10^{-38} \leqslant f \leqslant 2^{129}(1 - 2^{-24}) \approx 7 \cdot 10^{39}$$

IEEE double precision standard

s	e	q
1	11	52
sign	exponent	coefficient/mantissa

The coded number is

$$f = (-1)^s(1+q)\,2^{e-1023}.$$

Corresponds to $t = 53$, $\beta = 2$, $L = -1022$ and $U = 1026$

The maximal error is

$$2^{-53} \approx 6 \cdot 10^{-16}$$

and the range is

$$2^{-1022} \approx 2 \cdot 10^{-308} \leqslant f \leqslant 2^{1029}(1 - 2^{-24}) \approx 7 \cdot 10^{309}$$

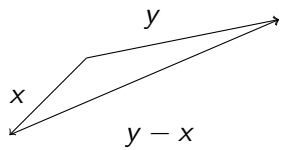
Vector and matrix norms

A *norm* on $\mathbb{R}^n$ is a function

$$\|\cdot\| \colon \mathbb{R}^n \longrightarrow \mathbb{R}$$

such that

1. for every vector x we have that $\|x\| \geqslant 0$, and $\|x\| = 0$ if and only if $x = 0$
2. for every vector x and scalar α we have that $\|\alpha\, x\| = |\alpha|\,\|x\|$
3. for every vectors x, y we have that $\|x + y\| \leqslant \|x\| + \|y\|$ *(triangle inequality)*.



Vector and matrix norms (cont.)

Example: for $1 \leqslant p \leqslant +\infty$ the *p-norm* is defined as

$$\|x\|_p = \begin{cases} \left(\sum_{i=1}^n |x_i|^p\right)^{1/p} & \text{if } 1 \leqslant p < +\infty \\ \max_i |x_i| & \text{if } p = +\infty \end{cases}$$

Any two norms can be compared: there are $c_1, c_2 > 0$ such that

$$\|\cdot\|_1 \leqslant c_1\,\|\cdot\|_2 \quad \text{and} \quad \|\cdot\|_2 \leqslant c_2\,\|\cdot\|_1$$

Example:

$$\|\cdot\|_2 \leqslant \|\cdot\|_1 \leqslant n^{1/2}\|\cdot\|_2 \quad \text{and} \quad \|\cdot\|_\infty \leqslant \|\cdot\|_1 \leqslant n\,\|\cdot\|_\infty$$

Vector and matrix norms (cont.)

A *matrix norm* is a norm on $\mathbb{R}^{n\times n}$ s.t. for all A, B we have that

$$\|A\,B\| \leqslant \|A\|\,\|B\|$$

Example: the *Frobenius norm*

$$\|A\|_{\mathrm{F}} = \left(\sum_{i,j} |a_{i,j}|^2\right)^{1/2}$$

Vector and matrix norms (cont.)

Definition: Given a norm on $\mathbb{R}^n$, the associated *operator norm* is

$$\|A\| = \sup_{x\neq 0} \frac{\|A\,x\|}{\|x\|}$$

It is a matrix norm

Properties:

1. $\|A\|_\infty = \max_i \sum_j |a_{i,j}|$ *(maximum row sum)*
2. $\|A\|_1 = \max_j \sum_i |a_{i,j}|$ *(maximum column sum)*
3. $\|A\|_2 = \lambda_{\max}(A^*A)^{1/2}$ with $A^* = \overline{A}^T$ and $\lambda_{\max}$ the maximal eigenvalue
4. if Q and Q' are *orthogonal* then

$$\|Q\,A\,Q'\|_2 = \|A\|_2 \quad \text{and} \quad \|Q\,A\,Q'\|_{\mathrm{F}} = \|A\|_{\mathrm{F}}$$

In particular $\|Q\|_2 = 1$ and $\|Q\|_{\mathrm{F}} = n^{1/2}$

Perturbation theory

A matrix A is *well/badly (or ill) conditionned* if small changes in A can cause small/large changes in the solution of

$$A\,x = b$$

Let x and $\widehat{x} = x + \delta x$ solutions to

$$A\,x = b \quad \text{and} \quad (A+\delta A)\,\widehat{x} = b + \delta b$$

We have that

$$\begin{array}{rcl} (A+\delta A)\,(x+\delta x) & = & b - \delta b \\ -\quad A\,x & = & b \\ \hline \delta A\,x + (A+\delta A)\,\delta x = & & \delta b \end{array}$$

Then

$$\delta x = A^{-1}(-\delta A\,\widehat{x} + \delta b)$$

The condition number

Fix a norm $\|\cdot\|$. Then

$$\|\delta x\| \leqslant \|A^{-1}\|\,(\|\delta A\|\,\|\widehat{x}\| + \|\delta x\|)$$

or equivalently

$$\frac{\|\delta x\|}{\|\widehat{x}\|} \leqslant \kappa_{\|\cdot\|}\left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\|\,\|\widehat{x}\|}\right) \tag{1}$$

with

$$\kappa_{\|\cdot\|} := \|A\|\,\|A^{-1}\|$$

the *condition number* of A with respect to $\|\cdot\|$

Precision of approximations

Let $\lambda \in \mathbb{R}$ and $\widehat{\lambda} = \lambda + \delta\lambda$ an approximation of λ with k correct digits in base $\beta \geqslant 2$. Then

$$\lambda = \beta^e \times d_1 \cdots d_k d_{k+1} \cdots \quad \text{and} \quad \widehat{\lambda} = \beta^e \times d_1 \cdots d_k \widetilde{d}_{k+1} \cdots$$

and so

$$\frac{|\delta\lambda|}{|\lambda|} \leqslant \beta^{-k}$$

or equivalently

$$-\log_\beta \left(\frac{|\delta\lambda|}{|\lambda|} \right) \geqslant k.$$

Roundoff with IEEE single precision and double precision give approximations with 24 and 53 correct bits, respectively:

$$-\log_2 \left(\frac{|\delta_{\text{single}}\lambda|}{|\lambda|} \right) \geqslant 24 \quad \text{and} \quad -\log_2 \left(\frac{|\delta_{\text{double}}\lambda|}{|\lambda|} \right) \geqslant 53.$$

Lost of precision

The inequality (1) translates into

$$-\log_\beta \frac{\|\delta x\|}{\|x\|} \geqslant -\log_\beta \kappa(A) - \log_\beta \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\|\,\|\widehat{x}\|} \right)$$

Warning: ill-conditionned matrices matrices destroy the quality of your approximations!

For instance, for *exact* data truncated with IEEE single or double precision, the computed result of $Ax = b$ will be meaningless as as soon

$$\kappa(A) > 2^{24} \approx 6 \cdot 10^8 \text{ (single precision)}$$

and

$$\kappa(A) > 2^{53} \approx 10^{16} \text{ (double precision)}$$

Distance to the ill-posed problems

The condition number of the 2-norm has a geometrical interpretation as the inverse if its distance to the set of ill-posed problems:

$$\kappa_2(A) = \frac{1}{\text{distance}(A, \Sigma)} \tag{2}$$

with $\Sigma = \{A \mid \det(A) = 0\}$

Error analysis in GEPP

We want to apply the two steps:

1. analyse roundoff errors to show that the matrix

   $$\widehat{A}_{\text{GEPP}} \coloneqq P_{\text{GEPP}} L_{\text{GEPP}} U_{\text{GEPP}}$$

   has a small relative error *(backward analysis)*
2. apply perturbation theory to bound the error in the computed solution x_{GEPP} of the equation

   $$A_{\text{GEPP}}\, x = b$$

Error analysis in GEPP (cont.)

Rounding off the entries of A gives $\widehat{A} = A + \delta A$ with

$$\frac{\|\delta A\|}{\|A\|} < \varepsilon \quad \text{(machine epsilon)}$$

By perturbation theory, this error will be amplified to

$$\frac{\|\delta x\|}{\|x\|} < \kappa_{\|\cdot\|}(A)\,\varepsilon.$$

To keep this bound, for $\delta_{\mathrm{GEPP}} A \coloneqq A_{\mathrm{GEPP}} - A$ we want

$$\frac{\|\delta_{\mathrm{GEPP}} A\|}{\|A\|} \leqslant C\,\varepsilon$$

with C as small as possible

The need of pivoting

Apply LU factorization without pivoting to the matrix

$$A = \begin{bmatrix} \eta & 1 \\ 1 & 1 \end{bmatrix}$$

with η a power of the base β that is smaller than ε, so that

$$1 \oplus \eta = \mathrm{fl}(1+\eta) = 1$$

For instance

$$\beta = 10, \quad \varepsilon = 0.5 \cdot 10^{-3} \quad \text{and} \quad \eta = 10^{-4}$$

The need of pivoting (cont.)

Set

$$A = L\,U = \begin{bmatrix} 1 & 0 \\ \eta^{-1} & 1 \end{bmatrix} \begin{bmatrix} \eta & 1 \\ 0 & 1-\eta^{-1} \end{bmatrix}$$

Then

$$L_{\mathrm{GEWP}} = \begin{bmatrix} 1 & 0 \\ \eta^{-1} & 1 \end{bmatrix} \quad \text{and} \quad U_{\mathrm{GEWP}} = \begin{bmatrix} \eta & 1 \\ 0 & -\eta^{-1} \end{bmatrix}$$

and so

$$A_{\mathrm{GEWP}} = L_{\mathrm{GEWP}}\,U_{\mathrm{GEWP}} = \begin{bmatrix} \eta & 1 \\ 1 & 0 \end{bmatrix},$$

is *not* close to A!

$$\frac{\|\delta A_{\mathrm{GEWP}}\|_\infty}{\|A\|_\infty} = \frac{\left\| \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \right\|_\infty}{\left\| \begin{bmatrix} \eta & 1 \\ 1 & 1 \end{bmatrix} \right\|_\infty} = \frac{1}{2}$$

$\rightsquigarrow$ GEWP is not backward stable

The need of pivoting (cont.)

The solution of $A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ is $x \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Solving

$$L_{\mathrm{GEWP}} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

gives $y_1 = 1$ and $y_2 = 2 \ominus \eta^{-1} = -\eta^{-1}$. Then

$$U_{\mathrm{GEWP}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\eta^{-1} \end{bmatrix}$$

gives $x_2 = \frac{-\eta^{-1}}{-\eta^{-1}} = 1$ and $x_1 = \frac{1 \ominus 1}{1 \ominus \eta} = 0$. Hence

$$x_{\mathrm{GEWP}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

is *not* close to x

The need of pivoting (cont.)

The instability is also reflected in the conditions numbers:

$$\|A\|_\infty \approx 4 \quad \text{well-conditioned}$$

whereas

$$\|L\|_\infty, \|U\|_\infty \approx \eta^{-2} \quad \text{ill-conditioned}$$

Formal error analysis of GEPP

When the intermediate quantities are too large, the information in A can be easily lost

Suppose that A is already pivoted. Then

$$A = L_{\text{GEPP}}\, U_{\text{GEPP}} + E \quad \text{with } |E| \leqslant n\,\varepsilon\,|L|\,|U|$$

where

- $|E|$ the $n \times n$ matrix whose entries are the absolute values of those of E (and similarly for $|L|$ and $|U|$)
- ε the machine epsilon

Formal error analysis of GEPP (cont.)

Hence

$$\|A - A_{\text{GEPP}}\|_\infty \leqslant n\,\varepsilon\,\||L|\|_\infty\||U|\|_\infty \leqslant n^3\,\varepsilon\,g_{\text{GEPP}}\,\|A\|_\infty$$

where

$$g_{\text{GEPP}} = \frac{\max_{i,j}|u_{i,j}|}{\max_{i,j}|a_{i,j}|} \quad \text{the pivot growth}$$

because

- $|l_{i,j}| \leqslant 1$ and so $\|L\|_\infty \leqslant n$
- $|u_{i,j}| \leqslant g_{\text{GEPP}}\|A\|_\infty$ and so $\|U\|_\infty \leqslant n\,g_{\text{GEPP}}\,\|A\|_\infty$

Thus

$$\frac{\|\delta_{\text{GEPP}}A\|_\infty}{\|A\|_\infty} \leqslant n^3\varepsilon\,g_{\text{GEPP}} \qquad (3)$$

Formal error analysis of GEPP (cont.)

In general $g_{\text{GEPP}} \leqslant 2^{n-1}$, and this bound can be attained:

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 8 \end{bmatrix}$$

Formal error analysis of GEPP (cont.)

This bound in (3) is too pesimistic in practice, since typically

$$\||L|\|_\infty \, \||U|\|_\infty \approx \|A\|_\infty$$

If this is the case, then

$$\frac{\|\delta_{\mathrm{GEPP}} A\|}{\|A\|} \lesssim n\,\varepsilon$$

and GEPP would be stable

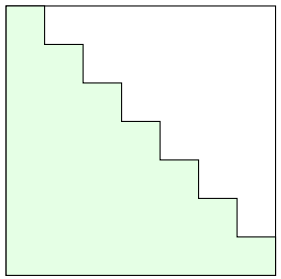
We thus say that GEPP is "backward stable in practice" (!?)

Symmetric matrices

An $n \times n$ matrix A is *symmetric* if

$$A^T = A$$

A symmetric matrix needs *half the space* to store its entries



We should be able to solve a symmetric problem

$$A\,x = b$$

with $\approx \frac{n^3}{3}$ flops instead of $\approx \frac{2\,n^3}{3}$

Pivoting destroys symmetry

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} = \begin{bmatrix} c & e & f \\ b & d & e \\ a & b & c \end{bmatrix}$$

$\rightsquigarrow$ for a symmetric A, we just aim to compute the *LU factorization*

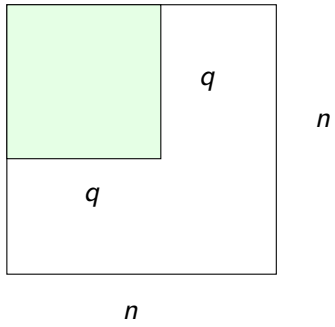
Matlab notation

For $p \leqslant q$ and $r \leqslant s$ set

$$A(p:q, r:s) = \left[a_{i,j}\right]_{p \leqslant i \leqslant q, r \leqslant j \leqslant s} \in \mathbb{R}^{(q-p+1)\times(s-r+1)}$$

For instance, the *leading* $q \times q$*-principal submatrix* of A is

$A(1:q, 1:q)$



When does the LU factorization exist?

Let A be an arbitrary $n \times n$ matrix (not necessarily symmetric)

The following are equivalent (TFAE):

1. there are unique L unit lower triangular and U upper triangular such that $A = L\,U$
2. all leading principal submatrices of A are nonsingular

The LDLT factorization

When A is symmetric and has an LU factorization, the factors are connected:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{c}{a} & 1 \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & d - \frac{bc}{a} \end{bmatrix} \begin{bmatrix} 1 & \frac{c}{a} \\ 0 & 1 \end{bmatrix}$$

In general symmetric case, set $d_i = u_{i,i}$ and write

$$U = D\,M$$

with $D = \mathrm{diag}(d_1, \ldots, d_n)$ and M *unit* upper triangular. Then

$$M = L^T$$

and so the LU factorization can be rewritten as

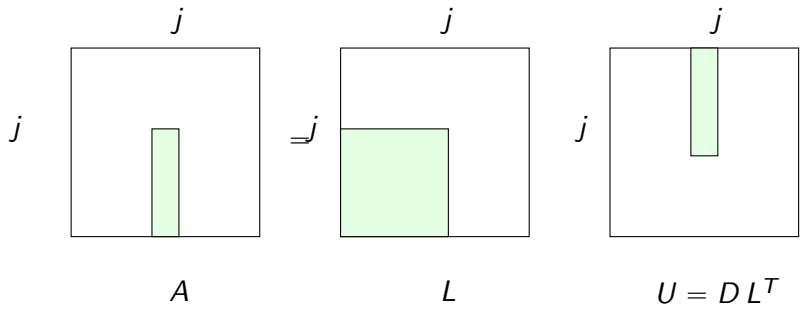
$$A = L\,D\,L^T$$

Computing the LDLT factorization

For $j = 1, \ldots, n$ set $A(j:n, j) \leftarrow L(j:n, 1:j)\, v(1:j)$ with

$$v = \begin{bmatrix} d_1\, l_{j,1} \\ \vdots \\ d_{j-1}\, l_{j,j-1} \\ d_j \end{bmatrix}$$

as in the figure



Computing the LDLT factorization (cont.)

Hence the equations

$$d_j = a_{j,j} - \sum_{k=1}^{j-1} d_k\, \ell_{j,k}^2$$

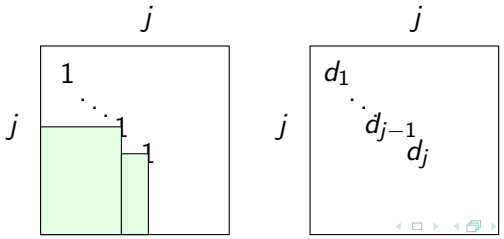
and

$$L(j+1:n, j) = \frac{1}{d_j}\left(A(j+1:n, j) - L(j+1:n, 1:j-1)\, v(1:j-1)\right)$$

gives the jth diagonal entry and the jth column

$$d_j \quad \text{and} \quad L(j+1:n, j)$$

from the previous diagonal entries and the $(j-1)$th column of L



The LDLT algorithm

for $j = 1, \ldots, n$
  for $i = 1, \ldots, j-1$
    $v_i \leftarrow \ell_{j,i}\, d_i$
  end
  $d_j \leftarrow a_{j,j} - L(j, 1:j-1)\, v(1:j-1)$

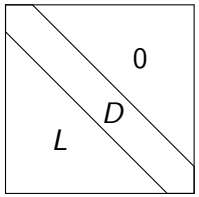
$$L(j+1:n, j) \leftarrow \frac{1}{d_j}\big(A(j+1:n, j) - L(j+1, n:1:j-1)\, v(1:j-1)\big)$$

The modified LDLT algorithm

for $j = 1, \ldots, n$
  for $i = 1, \ldots, j-1$
    $v_i \leftarrow a_{j,i}\, a_{i,i}$
  end
  $a_{j,j} \leftarrow a_{j,j} - A(j, 1:j-1)\, v(1:j-1)$

$$A(j+1:n, j) \leftarrow \frac{1}{a_{i,i}}\big(A(j+1:n, j) - A(j+1, n:1:j-1)\, v(1:j-1)\big)$$

$\rightsquigarrow$ over-writing scheme



Example

Let

$$A = \begin{bmatrix} 1 & -1 & 2 \\ -1 & 5 & 2 \\ 2 & 2 & 17 \end{bmatrix}$$

For $j = 1$:

$$L = \begin{bmatrix} 1 & & \\ -1 & & \\ 2 & & \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} 1 & & \\ & & \\ & & \end{bmatrix}$$

For $j = 2$:

$$[v] = -1, \quad L = \begin{bmatrix} & & \\ & 1 & \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} & & \\ & 4 & \\ & & \end{bmatrix}$$

coming from the operations

$$v_1 = (-1)\cdot 1 = -1, \quad d_2 = 5 - (-1)\cdot(-1) = 4, \quad \ell_{3,2} = \frac{1}{4}(2 - 2\cdot(-1))$$

Example (cont.)

For $j = 3$:

$$v = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} & & \\ & & \\ & & 9 \end{bmatrix}$$

coming from

$$v_1 = 2 \cdot 1 = 1, \quad v_2 = 1 \cdot 4 = 4, \quad d_3 = 17 - (2 \cdot 2 + 1 \cdot 4) = 9$$

Hence

$$\begin{bmatrix} 1 & -1 & 2 \\ -1 & 5 & 2 \\ 2 & 2 & 17 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix} \begin{bmatrix} 1 & -1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

In the machine:

$$A = \begin{bmatrix} 1 & -1 & 2 \\ -1 & 5 & 2 \\ 2 & 2 & 17 \end{bmatrix} \rightsquigarrow L \,\&\, D = \begin{bmatrix} 1 & & \\ -1 & 4 & \\ 2 & 1 & 9 \end{bmatrix}$$

Numerical (un)stability

The LU factorization of a symmetric matrix can be numerically unstable:

$$A = \begin{bmatrix} \eta & 1 \\ 1 & 1 \end{bmatrix}$$

with $0 < \eta < \varepsilon$ (machine epsilon)

Important fact from LA:

$$A \in \mathbb{R}^{n \times n} \text{ symmetric } \iff A = Q^T \Lambda Q$$

with Q orthogonal and Λ diagonal: A is diagonalizable over the reals through an orthogonal similarity, and

$$A \text{ is SPD } \iff \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n) \text{ with } \lambda_i > 0$$

Symmetric positive definite systems

A symmetric A is *positive definite (SPD)* if for all $x \in \mathbb{R}^n \backslash \{0\}$

$$x^T A x > 0$$

The Cholesky factorization

An SPD $n \times n$ matrix is nonsingular, and moreover all its leading principal submatrices are nonsingular

$\rightsquigarrow$ there is unit lower triangular L and a diagonal D such that

$$A = L D L^T \tag{4}$$

We have that $d_i > 0$ for all i and so we can write (4) as

$$A = G G^T$$

with $G = L \cdot \text{diag}(\lambda_1^{1/2}, \ldots, \lambda_n^{1/2})$ (*Cholesky factorization*)

Computing the Cholesky factorization

Note that

$$a_{i,i} = \sum_{k=0}^{j} g_{j,k}\, g_{i,k}$$

and so

$$g_{j,j}\, g_{i,j} = a_{i,j} - \sum_{k=1}^{j-1} g_{j,k}\, g_{i,k}$$

$\rightsquigarrow$ we can compute the j-th column of G from the previous ones

The Cholesky algorithm

for $j = 1, \ldots, n$
  $g_{j,j} \leftarrow (a_{j,j} - \sum_{k=1}^{j-1} g_{j,k}^2)^{1/2}$
  for $i = j+1, \ldots, n$

$$g_{i,j} \leftarrow \frac{1}{g_{j,j}} \left(a_{i,j} - \sum_{k=1}^{j-1} g_{i,k}\, g_{j,k} \right)$$

or alternatively:

for $j = 1, \ldots, n$
  $a_{j,j} \leftarrow (a_{j,j} - \sum_{k=1}^{j-1} a_{j,k}^2)^{1/2}$
  for $i = j+1, \ldots, n$

$$a_{i,j} \leftarrow \frac{1}{g_{j,j}} \left(a_{i,j} - \sum_{k=1}^{j-1} a_{i,k}\, a_{j,k} \right)$$

Can be overwritten over A and does not need the auxiliary vector v

The example revisited

Consider again

$$A = \begin{bmatrix} 1 & -1 & 2 \\ -1 & 5 & 2 \\ 2 & 2 & 17 \end{bmatrix}$$

For $j = 1$:

$$G = \begin{bmatrix} 1 & & \\ -1 & & \\ 2 & & \end{bmatrix}$$

and for $j = 2$:

$$G = \begin{bmatrix} & & \\ & 2 & \\ & 2 & \end{bmatrix}$$

coming from the operations

$$g_{2,2} = (5 - (-1)^2)^{1/2} = 2 \quad \text{and} \quad g_{3,2} = \frac{1}{2}(2 - (-1) \cdot 2) = 2$$

The example revisited (cont.)

For $j = 3$:

$$G = \begin{bmatrix} & & \\ & & \\ & & 3 \end{bmatrix}$$

coming from $g_{3,3} = (17 - (2^2 + 2^2))^{1/2} = 3$

Hence

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

In the machine

$$A = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ 2 & 1 & 1 \end{bmatrix} \rightsquigarrow G = \begin{bmatrix} 1 & & \\ -1 & 2 & \\ 2 & 2 & 3 \end{bmatrix}$$

Complexity

The complexity of this algorithm is

$$\begin{aligned}\sum_{j=1}^{n}\Big(2j-1+\sum_{i=j+1}^{n}(2j-1)\Big) &= \sum_{j=1}^{n}(2j-1)\,(n-j+1)\\ &= 2\,n\Big(\sum_{j=1}^{n}j\Big)-\sum_{j=1}^{n}j^2+O(n^2)\\ &= 2\,n\left(\frac{n^2}{2}+O(n)\right)-\frac{n^3}{3}+O(n^2)\\ &= \frac{1}{3}\,n^3+O(n^2)\end{aligned}$$

$\rightsquigarrow$ half the complexity of the LU factorization

Numerical stability

Pivoting is not necessary for the Cholesky algorithm to be numerically stable: the same analysis of GEPP shows that the Cholesky solution $\widehat{x}$ satisfies $(A+\delta A)\,\widehat{x}=b$ with

$$|\delta A| \leqslant 3\,n\,\varepsilon\,|G|\,|G^T|$$

By the *Cauchy-Schwartz inequality*, for each i,j we have that

$$\begin{aligned}(|G|\,|G^T|)_{i,j} &\leqslant \sum_{k=1}^{n}|g_{i,k}|\,|g_{j,k}|\\ &\leqslant \Big(\sum_{k=1}^{n}g_{i,k}^2\Big)^{1/2}\Big(\sum_{k=1}^{n}g_{j,k}^2\Big)^{1/2} = a_{i,i}^{1/2}\,a_{j,j}^{1/2} \leqslant \max_{i,j}|a_{i,j}|\end{aligned}$$

Hence $\||G|\,|G^T|\|_\infty \leqslant n\,\|A\|_\infty$ and so

$$\|\delta A\|_\infty \leqslant 3\,n^2\,\varepsilon\,\|A\|_\infty$$

$\rightsquigarrow$ Cholesky algorithm is backward stable

Is my A an SPD matrix?

Cholesky is the cheapest way of testing is a given symmetric $n\times n$ matrix is definite positive:

it will be the case if and only if the algorithm concludes!

Band matrices

A is a *band matrix* with *lower bandwith* b_L and *upper bandwith* b_U if

$$a_{i,j}=0$$

whenever $i>j-b_L$ or $i<j+b_U$:

$$A=\begin{bmatrix} a_{1,1} & \cdots & a_{1,b_U+1} & & \\ \vdots & & & \ddots & \\ a_{b_L+1,1} & & & & a_{n-b_U,n}\\ & \ddots & & & \vdots\\ & & a_{n,n-b_L} & \cdots & a_{n,n}\end{bmatrix}.$$

GE on band matrices

Key observation: the Schur complement of a band matrix is also banded, with the same upper and lower bandwiths

$\rightsquigarrow$ GEWP preserves the band structure:

$$A = L\,U$$

where L is unit lower triangular with lower bandwith b_L and U is upper triangular with upper bandwith b_U

This factorization can be computed with

$$2\,n\,b_L\,b_U + O(n\,(b_L + b_U)) \text{ flops}$$

Example

Let

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 4 & -1 & 3 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & 3 & 4 \end{bmatrix} \qquad (b_L = b_U = 1)$$

Then

$$L(1:4,1) = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 2 & -1 & 0 & 0 \end{bmatrix}$$

The *Schur complement* is given by $a_{j,k} \leftarrow a_{j,k} - l_{j,1}\,u_{1,k}$ for $j,k = 2,3,4$ and so

$$S_1 = \begin{bmatrix} 1 & 3 & 0 \\ -1 & -2 & 1 \\ 0 & 3 & 4 \end{bmatrix}$$

Example (cont.)

Next

$$L(2:4,2) = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}, \quad U(2,2:4) = \begin{bmatrix} 1 & 3 & 0 \end{bmatrix} \quad \text{and} \quad S_2 = \begin{bmatrix} 1 & 1 \\ 3 & 4 \end{bmatrix}$$

and then

$$L(3:4,3) = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad U(3,3:4) = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad \text{and} \quad S_3 = \begin{bmatrix} 1 \end{bmatrix}$$

Finally

$$L(4:4,4) = \begin{bmatrix} 1 \end{bmatrix} \quad \text{and} \quad U(4,4:4) = \begin{bmatrix} 1 \end{bmatrix}$$

Example (cont.)

Thus

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 4 & -1 & 3 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & 3 & 4 \end{bmatrix} = L\,U = \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 0 & -1 & 1 & \\ 0 & 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 0 \\ & 1 & 3 & 0 \\ & & 1 & 1 \\ & & & 1 \end{bmatrix}$$

GEPP on band matrices

GEPP can exploit band structure, but in a more involved way:

$$A = P\,L\,U$$

where U is banded with upper bandwith $b_L + b_U$ and L has at most $b_L + 1$ nonzero entries per column

Why?

At each step pivoting can only be done within the first b_L rows, and later permutations can reorder the earlier columns of L

Example

Let again

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

and swap rows 1 and 2:

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

Then

$$L(1:4,1) = \begin{bmatrix} 1 \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix}, \quad U(1,1:4) = \begin{bmatrix} 4 & -1 & 3 & 0 \end{bmatrix}, \quad S_1 = \begin{bmatrix} \frac{-1}{2} & \frac{-3}{2} & 0 \\ -1 & -2 & 1 \\ 0 & 3 & 4 \end{bmatrix}$$

Example (cont.)

We then swap the rows 2 and 3 of A to obtain

$$L(2:4,2) = \begin{bmatrix} 1 \\ \frac{1}{2} \\ 0 \end{bmatrix}, \quad U(2,2:4) = \begin{bmatrix} -1 & -2 & 1 \end{bmatrix}, \quad S_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 3 & 4 \end{bmatrix}$$

Swap the rows 3 and 4 of A:

$$L(3:4,3) = \begin{bmatrix} 1 \\ \frac{-1}{6} \end{bmatrix}, \quad U(3,3:4) = \begin{bmatrix} 3 & 4 \end{bmatrix}, \quad S_3 = \begin{bmatrix} \frac{1}{6} \end{bmatrix}$$

Finally

$$U(4,4:4) = \begin{bmatrix} \frac{1}{6} \end{bmatrix}$$

Example (cont.)

Hence

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ 4 & -1 & 3 & 0 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ \frac{1}{2} & \frac{1}{2} & \frac{-1}{6} & 1 \end{bmatrix} \begin{bmatrix} 4 & -1 & 3 & 0 \\ & -1 & -2 & 1 \\ & & 3 & 4 \\ & & & \frac{1}{6} \end{bmatrix}$$

L has at most 2 nonzero elements per column, and U has upper bandwith 2

Sparse matrices

GEPP does not preserve the sparse structure:

$$\begin{bmatrix} 1 & 0.1 & 0.1 & \dots & 0.1 \\ 0.1 & 1 & 0 & \dots & 0 \\ 0.1 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0.1 & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & & & \\ 0.1 & 1 & & & \\ 0.1 & -0.01 & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0.1 & -0.01 & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0.1 & 0.1 & \dots & 0.1 \\ & 0.99 & -0.01 & \dots & -0.01 \\ & & 0.99 & & -0.01 \\ & & & \ddots & \vdots \\ & & & & 0.99 \end{bmatrix}$$

The least square problem (LSP)

LSP: for an $m \times n$ matrix A and an m-vector b, find the n-vector $x_{\min}$ minimizing the quantity

$$\|Ax - b\|_2$$

$\rightsquigarrow$ gives the linear combination of the columns

$$Ax = \sum_{j=1}^{n} x_j \, \mathrm{col}_j(A) \in \mathbb{R}^m$$

that best approaches (in the 2-norm) the m-vector b

- If $m = n$ and A is nonsingular then $Ax_{\min} = b$
- If $m > n$ then typically $Ax = b$ has no solution, and the minimum is positive

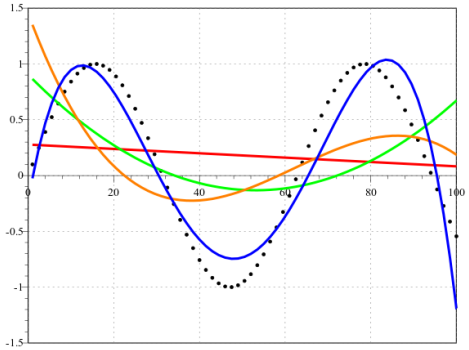
Central case: $m \geqslant n$ and $\mathrm{rank}(A) = n$

Example

Suppose we have pairs

$$(y_i, b_i) \in \mathbb{R}^2, \quad i = 1, \dots, m,$$

and we want to find the polynomial of degree d that best fits the b_i's as a function of the y_i's



Example (cont.)

Boils down to computing the polynomial

$$p(y) = \sum_{j=0}^{d} x_j \, y^j$$

minimizing the 2-norm of the *residual* $p(y_i) - b_i$, $i = 1, \dots, m$:

$$\sum_{i=1} (p(y_i) - b_i)^2$$

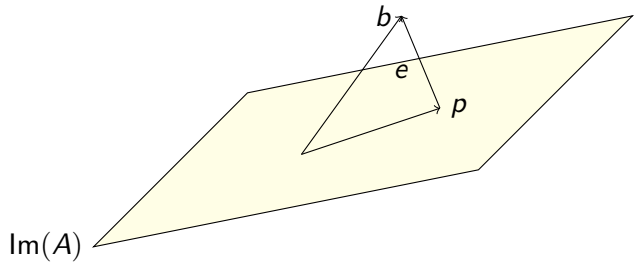
LSP for the data

$$A = \begin{bmatrix} 1 & y_1 & \cdots & y_1^d \\ \vdots & \vdots & & \vdots \\ 1 & y_m & \cdots & y_m^d \end{bmatrix} \in \mathbb{R}^{m \times (d+1)} \quad \text{and} \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$$

Three ways

- normal equations
- QR factorization
- SVD

LSP and its solution



- $p = A\,x_{\min}$ the orthogonal projection of b into $\mathrm{Im}(A)$
- $e = b - p$ the error vector

Normal equations

The m-vector e is perpendicular to $\mathrm{Im}(A)$: for all $x \in \mathbb{R}^n$

$$0 = \langle A\,x, e\rangle = (A\,x)^T(b - A\,x_{\min}) = x^T A^T(b - A\,x_{\min})$$

Hence

$$\boxed{A^T A\,x_{\min} = A^T b} \qquad (5)$$

The $n \times n$ matrix $A^T A$ is symmetric and positive definite

$$\Rightarrow \text{ it is nonsingular and } x_{\min} \text{ is the only solution of (5)}$$

By Pythagoras theorem, the 2-norm of the error is

$$\|e\|_2 = (\|b\|_2 - \|A\,x\|_2^2)^{1/2}$$

Normal equations algorithm

We can solve the normal equations with Cholesky factorization:

Compute the lower triangular part of $C \leftarrow A^T A$
Compute $d \leftarrow A^T b$
Compute the Cholesky factorization $C = G\,G^T$
Solve $G\,y = d$ and $G^T\,x_{\min} = y$

Its *complexity* is

$$\left(m + \frac{n}{3}\right) n^2 + O(n^2) \text{ flops}$$

For $m \gg n$ the complexity $m\,n^2$ of computing the product $A^T A$ dominates

- uses standard algorithms
- reliable when A is far from rank deficient (but unstable otherwise)

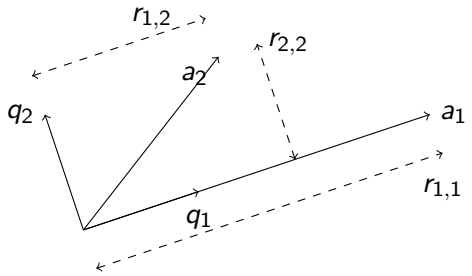
LSP and orthogonalization

The columns of A are independent ($\text{rank}(A) = n$) but not orthogonal

If this were the case, $x_{\min}$ would be easy to find!

Gram-Schmidt orthogonalization

Set $a_j = \text{col}_j(A) \in \mathbb{R}^m$, $j = 1, \ldots, n$



GS produces orthonormal m-vectors q_j, $j = 1, \ldots, n$, such that

$$\text{span}(q_1, \ldots, q_j) = \text{span}(a_1, \ldots, a_j), \quad j = 1, \ldots, n$$

Gram-Schmidt orthogonalization (cont.)

First step:

$$q_1 \leftarrow \frac{a_1}{\|a_1\|_2} \quad \text{(normalize)}$$

GO step:

$$\widetilde{a}_2 \leftarrow a_2 - \langle a_2, q_1 \rangle q_1 \quad \text{(orthogonalize)}$$
$$q_2 \leftarrow \frac{\widetilde{a}_2}{\|\widetilde{a}_2\|_2} \quad \text{(normalize)}$$

GO step:

$$\widetilde{a}_3 \leftarrow a_3 - \langle a_3, q_1 \rangle q_1 - \langle a_3, q_2 \rangle q_2$$
$$q_3 \leftarrow \frac{\widetilde{a}_3}{\|\widetilde{a}_3\|_2}$$

The QR factorization

We can write the a_j's in terms of the q_j's:

$$\begin{aligned} a_1 &= \|\widetilde{a}_1\|_2 \, q_1 \\ a_2 &= \langle a_2, q_1 \rangle \, q_1 + \|\widetilde{a}_2\|_2 \, q_2 \\ a_3 &= \langle a_3, q_1 \rangle \, q_1 + \langle a_3, q_2 \rangle \, q_2 + \|\widetilde{a}_3\|_2 \, q_3 \\ &\ldots \end{aligned}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 & \cdots \end{bmatrix} \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots \\ 0 & r_{2,2} & r_{2,3} & \cdots \\ 0 & 0 & r_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

that is

$$\boxed{A = Q\,R}$$

with

- Q orthogonal $m \times n$
- R upper triangular $n \times n$ matrix with positive diagonal entries

This factorization is unique!

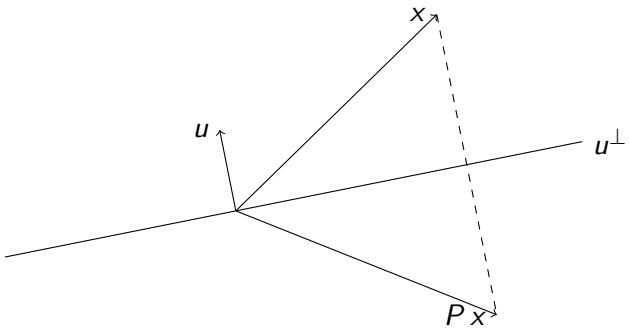
Solving the LSP with the QR factorization

The QR factorization solves the normal equations, and so the LSP:

$$\begin{aligned} x_{\min} &= (A^T A)^{-1} A^T b \\ &= ((QR)^T Q R)^{-1} (Q R)^T b \\ &= (R^T R)^{-1} R^T Q b \\ &= R^{-1} Q^T b \end{aligned}$$

- The GS algorithm is not stable when the columns of A are close to rank deficient

Householder reflections



A *Householder reflection* is

$$P = \mathbb{1}_m - 2\, u\, u^T \in \mathbb{R}^{m \times m}$$

for a unit vector $u \in \mathbb{R}^m$

It is symmetric ($P^T = P$) and orthogonal ($P^T P = \mathbb{1}_m$)

Householder reflections (cont.)

Given $y \in \mathbb{R}^m$ there is a reflection that zeroes all but the first entry:

$$P\, y = \begin{bmatrix} c \\ 0 \\ \vdots \\ 0 \end{bmatrix} = c\, e_1 \in \mathbb{R}^m$$

Since P is orthogonal

$$|c| = \|P\, y\|_2 = \|y\|_2$$

Householder reflections (cont.)

To compute u:

$$P\, y = (\mathbb{1}_m - 2\, u\, u^T)\, y = y - 2\langle u, y\rangle\, u = \pm \|y\|_2\, e_1$$

thus

$$\boxed{2\langle u, y\rangle\, u = y \pm \|y\|_2\, e_1}$$

Choose the sign so to avoid cancellations: u is a scalar multiple of

$$\widetilde{u} = y \pm \|y\|_2\, e_1 = \begin{bmatrix} y_1 + \operatorname{sign}(y_1)\, \|y\|_2 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

and so we set

$$u = \text{House}(y) := \frac{\widetilde{u}}{\|\widetilde{u}\|_2}$$

QR factorization with Householder reflections

Set $m = 4$ and $n = 3$:

$$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}$$

- Choose P_1 such that

$$A_1 \leftarrow P_1 A = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$

QR factorization with Householder reflections (cont.)

- Choose $P_2 = \begin{matrix} \\ 1 \\ 3 \end{matrix} \overset{\begin{matrix} 1 & 2 \end{matrix}}{\begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & P_2' \end{bmatrix}}$ such that

$$A_2 \leftarrow P_2 A_1 = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}$$

- Choose $P_3 = \begin{matrix} 1 \\ 1 \\ 2 \end{matrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbb{0} & \mathbb{0} & P_3' \end{bmatrix}$ such that

$$A_3 \leftarrow P_3 A_2 = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix}$$

QR factorization with Householder reflections (cont.)

$P_3 P_2 P_1 A = \widetilde{R} (= A_4)$ is *upper triangular*. Hence

$$\boxed{A = P_1^T P_2^T P_3^T \widetilde{R} = Q R}$$

with

- Q the first three columns of $P_1^T P_2^T P_3^T$
- R the first three rows of $\widetilde{R}$

Example

Let

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

Set

$$\widetilde{u}_1 = \begin{bmatrix} 1 + 2^{1/2} \\ 0 \\ -1 \end{bmatrix} \quad \text{and} \quad u_1 = \text{House} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \frac{\widetilde{u}_1}{\|\widetilde{u}_1\|_2} = \begin{bmatrix} 0.92 \\ 0 \\ -0.38 \end{bmatrix}$$

Then

$$P_1 = \mathbb{1}_3 - 2\, u_1\, u_1^T = \begin{bmatrix} -0.71 & 0 & 0.71 \\ 0 & 1 & 0 \\ 0.71 & 0 & 0.71 \end{bmatrix}, \quad A_1 = P_1 A = \begin{bmatrix} -1.41 & 1.41 \\ 0 & 2 \\ 0 & -2.83 \end{bmatrix}$$

Example (cont.)

Set then

$$\widetilde{u}_2 = \begin{bmatrix} 2 + (2^2 + (-2.83)^2)^{1/2} \\ -2.83 \end{bmatrix} \quad \text{and} \quad u_2 = \text{House} \begin{bmatrix} 2 \\ -2.83 \end{bmatrix} = \frac{\widetilde{u}_2}{\|\widetilde{u}_2\|_2} = \begin{bmatrix} 0.89 \\ -0.46 \end{bmatrix}$$

Then

$$P_2 = \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & \mathbb{1}_2 - 2\, u_2\, u_2^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.58 & 0.82 \\ 0 & 0.82 & 0.58 \end{bmatrix}$$

$$P_2\, A_1 = \begin{bmatrix} -1.41 & 1.41 \\ 0 & -3.49 \\ 0 & 0 \end{bmatrix} = \widetilde{R}$$

Example (cont.)

Finally

$$A = P_1^T P_2^T \widetilde{R} = \widetilde{Q}\, \widetilde{R} = \begin{bmatrix} -0.71 & 0.58 & 0.41 \\ 0 & -0.58 & 0.82 \\ 0.71 & 0.58 & 0.41 \end{bmatrix} \begin{bmatrix} -1.41 & 1.41 \\ 0 & -3.49 \\ 0 & 0 \end{bmatrix}$$

$$= Q\, R = \begin{bmatrix} -0.71 & 0.58 \\ 0 & -0.58 \\ 0.71 & 0.58 \end{bmatrix} \begin{bmatrix} -1.41 & 1.41 \\ 0 & -3.49 \end{bmatrix}$$

The algorithm

for $i = 1$ to $\min(m-1, n)$

$u_i \leftarrow \text{House}(A(i:m, i))$

$P_i \leftarrow \mathbb{1}_{m-i+1} - 2\, u_i\, u_i^T$

$A_i(i:m, i:n) \leftarrow P_i' A(i:m, i:n)$

- we do not really need P_i' but just the multiplication

  $(\mathbb{1}_{m-i+1} - 2\, u_i\, u_i^T)\, A(i:m, i:n) = A(i:m, i:n) - 2\, u_i\, (u_i^T A(i:m, i:n))$

- each P_i can be "stored" as u_i
- Q can be stored as $P_1 \cdots P_{n-1}$

The algorithm (cont.)

The complexity of this algorithm is

$$2\, n^2\, m - \frac{2}{3}\, n^2 \text{ flops}$$

Compared with solving the normal equations via Cholesky's algorithm:

- twice the complexity (for $m \gg n$)
- more numerically stable

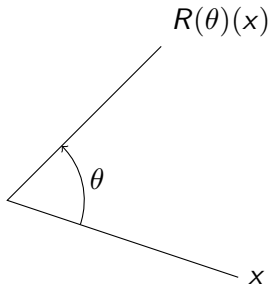
Plane rotations

A *rotation* on the plane with angle θ is the linear map

$$R(\theta)\colon \mathbb{R}^2 \to \mathbb{R}^2$$

given by the orthogonal matrix

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



Plane rotations

A *Givens rotation*: matrix of a rotation on the (i,j)-plane of $\mathbb{R}^m$

$$R(i,j,\theta) = \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} i & & & j & & \end{array} \\ \begin{bmatrix} \mathbb{1}_{i-1} & & & & & \\ & \cos(\theta) & & -\sin(\theta) & & \\ & & \mathbb{1}_{j-i-1} & & & \\ & \sin(\theta) & & \cos(\theta) & & \\ & & & & \mathbb{1}_{n-j-1} \end{bmatrix} \end{array}$$

QR factorization with Givens rotations

The QR factorization can be computed with Givens rotations similarly as with Householder reflections:

Given $x \in \mathbb{R}^m$ and $i > j$, we can zero x_j by choosing θ such that

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} (x_i^2 + x_j^2)^{1/2} \\ 0 \end{bmatrix}$$

or equivalently

$$\cos(\theta) = \frac{x_i}{(x_i^2 + x_j^2)^{1/2}} \quad \text{and} \quad \sin(\theta) = \frac{-x_j}{(x_i^2 + x_j^2)^{1/2}}$$

inverse trigonometric functions are not needed!

Example

Let

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

Setting

$$R_1 = \begin{bmatrix} 0.71 & 0 & -0.71 \\ 0 & 1 & 0 \\ 0.71 & 0 & 0.71 \end{bmatrix}$$

then

$$A_1 = R_1 A = \begin{bmatrix} 1.41 & -1.41 \\ 0 & 2 \\ 0 & -2.82 \end{bmatrix}$$

Example (cont.)

Then set

$$R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.58 & -0.82 \\ 0 & 0.82 & 0.58 \end{bmatrix}$$

so that

$$A_2 = R_2 A_1 = \begin{bmatrix} 1.41 & -1.41 \\ 0 & 3.47 \\ 0 & 0 \end{bmatrix} = \widetilde{R}$$

We conclude that $A = R_1^T R_2^T \widetilde{R} = Q R$ with

$$Q = \begin{bmatrix} 0.71 & -0.58 \\ 0 & 0.58 \\ 0.71 & -0.58 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 1.41 & -1.41 \\ 0 & 2 \end{bmatrix}$$

Complexity

The complexity of the QR factorization using Givens rotations is

$$3\,m\,n^2 + O(m\,n)$$

It is useful in special situations, e.g. for Hessenberg matrices

$$A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

The condition number of a rectangular matrix

For a $n \times n$ matrix A we have that

$$\|A\|_2 = \lambda_{\max}(A^T A)^{1/2} \qquad (\lambda_{\max} \text{ the largest eigenvalue})$$

and so

$$\kappa_2(A) = \left(\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)} \right)^{1/2}$$

When A is $m \times n$, we define its *condition number* as

$$\kappa_2(A) := \kappa_2(A^T A)^{1/2} = \left(\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)} \right)^{1/2}$$

It measures how far is A from being *rank deficient*

Numerical aspects

The *forward error analysis* of the LSP is controlled by $\kappa_2(A)$

QR factorization via Householder reflections or Givens rotations is *backwards stable*: if

$$A = Q\,R$$

and $Q + \delta\,Q$ and $R + \delta\,R$ are the computed factors, then

$$A + \delta\,A = (Q + \delta\,Q)\,(R + \delta\,R)$$

where the relative error is bounded by

$$\frac{\|\delta\,A\|_2}{\|A\|_2} \leqslant O(n\,\varepsilon)$$

with ε the machine epsilon

QR factorization *versus* normal equations

Hence when $m \gg n$, the QR factorization via Householder or Givens solves the LSP with $\approx 2\,n^2\,m$ flops or $\approx 3\,n^2\,m$ flops respectively, and a loss of precision of

$$\approx \log_b \kappa_2(A) \text{ digits}$$

On the other hand, solving the normal equations via Cholesky solves the LSP with $\approx n^2 m$ flops and a loss of precision of

$$\approx \log_b \kappa_2(A^T A) = 2 \log_b \kappa_2(A) \text{ digits}$$

Normal equations is the method of choice to solve the LSP when A is well-conditionned. If A is badly conditionned, then we might prefer applying applying the QR factorization via Householder or Givens, or the SVD *(to be discussed later)*

Rank deficient LSP

Let A be an $m \times n$ matrix and set $r = \text{rank}(A)$

It is *rank deficient* when

$$\boxed{r < n}$$

Typical in data analysis, when the variables have not been chosen wisely

If $r < n$ then the solution of the LSP is not unique: for all $x \in \text{Ker}(A) \simeq \mathbb{R}^{n-r}$ we have that

$$A\,(x_{\min} + x) = A\,x_{\min} + A\,x = A\,x_{\min}$$

also solves the LSP

Example

For some medical research on the effect of a drug on sugar levels in blood, we take the following data from patients:

- initial blood level (sugar)
- amount of drug
- weight on day i, $i = 1, \ldots, 7$
- final blood level

To correlate the final sugar level in terms of the rest of the data, we consider the LSP

$$\begin{matrix} \text{patient}_1 \\ \vdots \\ \text{patient}_m \end{matrix} \begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_9 \end{bmatrix}$$

Example (cont.)

The solution $x_{\min}$ would be the used as the predictor, since

$$b_i \approx \text{ patient}_i \cdot x_{\min}$$

The matrix A is probably close to rank 3: the experiment is not well-designed, and columns $3, \ldots, 9$ should be identified from our knowledge of the problem

Otherwise, we might take $x_{\min}$ very large, and a patient changing weight during the week would receive an irrealistic prediction

Rank deficient LSP via QR factorization

Suppose that $r < n$ and set

$$A = Q\,R = Q \begin{bmatrix} \overset{r}{R_{1,1}} & \overset{n-r}{R_{1,2}} \\ 0 & 0 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}$$

With round off, we hope to compute

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} \\ 0 & R_{2,2} \end{bmatrix}$$

with $R_{2,2}$ small (e.g. of size $\approx \varepsilon\,\|A\|_2$)

$\rightsquigarrow$ we set $R_{2,2} = \mathbb{0}$ and then minimize $\|A\,x - b\|_2$ as follows:

complete Q to an orthogonal $m \times m$ matrix $[Q\ \widetilde{Q}]$, so that

$$\|A\,x - b\|_2^2 = \left\| \begin{bmatrix} Q^T \\ \widetilde{Q}^T \end{bmatrix} (A\,x - b) \right\|_2^2 = \|R\,x - Q^T\,b\|_2^2 + \|\widetilde{Q}^T\,b\|_2^2$$

(multiplying by an orthogonal $m \times m$ matrix does not change the 2-norm)

Rank deficient LSP via QR factorization (cont.)

In general, this method is *not completely reliable* because R might be close to rank deficient even if $R_{2,2}$ is not small

Instead we apply QR factorization with pivoting

$$A\,P = Q\,R$$

with P is a permutation $m \times m$ matrix:

at step i, choose the largest column j of A with $i \leqslant j \leqslant n$ and compute the Householder reflection that zeroes the entries $i+1, \ldots, m$ in the i-th column

$\rightsquigarrow$ attempts to keep $R_{1,1}$ well-conditionned and $R_{2,2}$ small

Rank deficient LSP via QR factorization (cont.)

Write $Q = \begin{bmatrix} \overset{r}{Q_1} & \overset{n-r}{Q_2} \end{bmatrix}$ and $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{matrix} r \\ n-r \end{matrix}$. Then

$$\|A\,x - b\|_2^2 = \|R_{1,1}\,x_1 + R_{1,2}\,x_2 - Q_1^T\,b\|_2^2 + \|Q_2^T\,b\|_2^2 + \|\widetilde{Q}^T\,b\|_2^2$$

This expression is minimized by

$$x = \begin{bmatrix} R_{1,1}^{-1}(Q_1^T\,b - R_{1,2}\,x_2) \\ x_2 \end{bmatrix}$$

for any $(n-r)$-vector x_2

The typical choice is

$$x_2 = \mathbb{0}$$

Singular values and singular vectors

The *singular value decomposition (SVD)* extends this factorization to any matrix, even outside the square case.

The *key ingredient* is

$$\boxed{\text{decoupling } Q \text{ and } Q^T}$$

Let A be an $m \times n$ matrix with $m \geqslant n$. There are two sets of *singular vectors*

$$u_1, \ldots, u_m \in \mathbb{R}^m \ (\textit{left}) \quad \text{and} \quad v_1, \ldots, v_n \in \mathbb{R}^n \ (\textit{right})$$

Both sets form orthogonal bases, and are connected by the relation

$$A\,v_i = \sigma_i\,u_i, \quad i = 1, \ldots, n,$$

for the *singular values* $\sigma_1, \ldots, \sigma_n \geqslant 0$

The SVD

In matrix form: both

$$U = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \in \mathbb{R}^{m \times m} \quad \text{and} \quad V = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$$

are *orthogonal* square matrices:

$$U^{-1} = U^T \quad \text{and} \quad V^{-1} = V^T.$$

Set

$$\Sigma = \begin{bmatrix} \sigma_1 & & \overset{n}{} \\ & \ddots & \\ & & \sigma \\ & \mathbb{0} & \end{bmatrix}_{m-n}^{n}$$

The *(full) SVD* is the factorization $A\,V = U\,\Sigma$ or equivalently

$$A = U\,\Sigma\,V^T$$

Example

$$A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$$
$$= U\,\Sigma\,V^T = \begin{bmatrix} 0.32 & -0.95 \\ 0.95 & 0.32 \end{bmatrix} \begin{bmatrix} 6.71 & 0 \\ 0 & 2.24 \end{bmatrix} \begin{bmatrix} 0.71 & 0.71 \\ -0.71 & 0.71 \end{bmatrix}$$

Decomposition into matrices of rank 1

Set $r = \text{rank}(A)$. Then

- $u_1, \ldots, u_r$ is a basis of $\text{Im}(A)$
- $v_{r+1}, \ldots, v_n$ is a basis of $\text{Ker}(A)$

The column-row multiplication of $U\,\Sigma$ and V^T separates A into r pieces of rank 1:

$$A = \sum_{i=1}^{r} \sigma_i\, u_i\, v_i^T$$

In the example:

$$\begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = 6.71 \begin{bmatrix} 0.32 \\ 0.95 \end{bmatrix} \begin{bmatrix} 0.71 & 0.71 \end{bmatrix} + 2.24 \begin{bmatrix} -0.95 \\ 0.32 \end{bmatrix} \begin{bmatrix} -0.71 & 0.71 \end{bmatrix}$$

We have that

$$\sigma_1 = 6.71 > \sigma_2 = 2.24$$

$\rightsquigarrow$ the first piece is "more representative" of A

The thin and the reduced SVD's

The *thin SVD* avoids the 0's in the lower part of Σ and uses a diagonal matrix for the singular values:

$$A = U_n\,\Sigma_n\,V^T$$

with

$$U_n = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} \quad m \times n\text{-orthogonal}$$
$$\Sigma_n = \text{diag}(\sigma_1, \ldots, \sigma_n) \quad n \times n\text{-diagonal}$$

The *reduced SVD* keeps only the nonzero singular values to remove the parts that are going to produce zeros for sure:

$$A = U_r\,\Sigma_r\,V_r^T$$

with

$$U_r = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \quad m \times r\text{-orthogonal}$$
$$\Sigma_r = \text{diag}(\sigma_1, \ldots, \sigma_r) \quad r \times r\text{-diagonal}$$
$$V_r^T = \begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix} \quad n \times r\text{-orthogonal}$$

Reducing to SPD matrices

To identify the singular values and vectors, we can consider the SPD matrices

$$A^T A = (V \Sigma^T U^T)(U \Sigma V^T) = V \Sigma^T \Sigma V^T \in \mathbb{R}^{n \times n}$$
$$A A^T = (U \Sigma V^T)(V \Sigma^T U^T) = U \Sigma \Sigma^T U^T \in \mathbb{R}^{m \times m}$$

Then

- V contains the orthonormal eigenvectors of $A^T A$
- U contains the orthonormal eigenvectors of $A A^T$
- $\sigma_1, \ldots, \sigma_r$ are the nonzero eigenvalues of both $A^T A$ and $A A^T$

Existence and computation of the SVD (cont.)

For the full SVD, take

$$v_{r+1}, \ldots, v_n \in \mathbb{R}^n \quad \text{and} \quad u_{r+1}, \ldots, u_m \in \mathbb{R}^m$$

completing $v_1, \ldots, v_r$ and $u_1, \ldots, u_r$ to orthonormal bases. Then

$$A = U \Sigma V^T$$

with $U = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix}$, $V = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$ and

$$\Sigma = \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \\ & & & \mathbb{0} & & \end{bmatrix}$$

Existence and computation of the SVD

Consider the diagonalization

$$A^T A = Q \Lambda Q^T$$

and let $v_1, \ldots, v_n$ be the columns of Q, ordered so that $v_1, \ldots, v_r$ correspond to the nonzero eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_r > 0$

Set

$$\sigma_k = \lambda_k^{1/2} \quad \text{and} \quad u_k = \sigma_k^{-1} A v_k, \quad k = 1, \ldots, r$$

The u_k's are orthonormal:

$$\langle u_j, u_k \rangle = u_j^T u_k = (\sigma_j^{-1} A v_j)^T (\sigma_k^{-1} A v_k)$$
$$= \sigma_j^{-1} \sigma_k^{-1} v_j^T A^T A v_k = v_j^T v_k = \langle v_j, v_k \rangle = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}$$

$\rightsquigarrow$ the reduced SVD

$$A = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}$$

Example (cont.)

Set as before $A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$. Then

$$A^T A = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \quad \text{and} \quad A A^T = \begin{bmatrix} 9 & 12 \\ 12 & 41 \end{bmatrix}$$

The eigenvalues and eigenvectors of $A^T A$ are

$$\begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 45 \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 5 \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Hence the right singular vectors are

$$v_1 = \frac{1}{2^{1/2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix} \quad \text{and} \quad v_2 = \frac{1}{2^{1/2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix}$$

and the singular values are

$$\sigma_1 = 45^{1/2} = 6.71 \quad \text{and} \quad \sigma_2 = 5^{1/2} = 2.24$$

Example (cont.)

Moreover, the left singular vectors are

$$u_1 = \sigma_1^{-1} A\, v_1 = \frac{1}{10^{1/2}} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 0.32 \\ 0.95 \end{bmatrix}$$
$$u_2 = \sigma_2^{-1} A\, v_2 = \frac{1}{10^{1/2}} \begin{bmatrix} -3 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.95 \\ 0.32 \end{bmatrix}$$

Hence $A = U\,\Sigma\,V^T$ with

$$U = \begin{bmatrix} 0.32 & -0.95 \\ 0.95 & 0.32 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 6.71 & \\ & 2.24 \end{bmatrix}, \quad V = \begin{bmatrix} 0.71 & -0.71 \\ 0.71 & 0.71 \end{bmatrix}$$

Some particular cases

- Let $S = Q\,\Lambda\,Q^T$ be the diagonalization of a SPD matrix. Then
$$U = V = Q \quad \text{and} \quad \Sigma = \Lambda$$
- The singular values of an orthonormal $n \times n$ matrix Q are all equal to 1.
- Let $A = x\,y^T$ be an $m \times n$ matrix of rank 1, with $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$
Its reduced SVD is $A = U_1\,\Sigma_1\,V_1^T$
$$U_1 = \frac{x}{\|x\|_2}, \quad \Sigma_1 = \left[\|x\|_2\,\|y\|_2\right], \quad V_1 = \frac{y}{\|y\|_2}$$

The geometry of the SVD

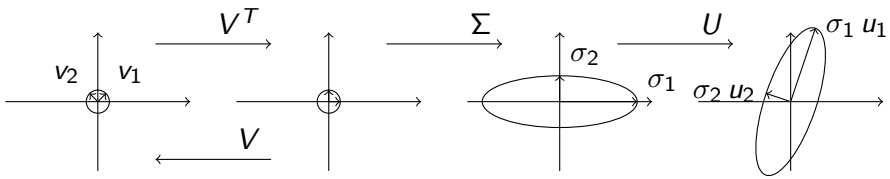
The SVD decomposes the matrix as

$$\text{orthogonal} \times \text{diagonal} \times \text{orthogonal}\,.$$

The unit sphere $\mathbb{S}_n$ of $\mathbb{R}^n$ is send to the ellipsoid $A\,\mathbb{S}_n$ of $\mathbb{R}^m$ centered at the origin and with axes

$$\sigma_i\,u_i, \quad i = 1, \ldots, r.$$

In dimension 2 we can draw the process: for $A = \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix}$ it gives



Computing the 2-norm and the Frobenius norm

The 2-norm of A can be computed in terms of its SVD: this norm is invariant with respect to multiplication by orthogonal matrices and so

$$\|A\|_2 = \|U^T A\,V\|_2 = \|\Sigma\|_2 = \sigma_1$$

The Frobenius norm is also invariant with respect to multiplication by orthogonal matrices, and so

$$\|A\|_{\mathrm{F}} = \|U^T A\,V\|_{\mathrm{F}} = \|\Sigma\|_{\mathrm{F}} = \Big(\sum_{i=1}^{r} \sigma_i^2\Big)^{1/2}$$

The best low rank approximation

Eckart-Young theorem: for $k = 1, \dots, r$, the matrix

$$A_k = \sum_{i=1}^{k} \sigma_i \, u_i \, v_i^T = U_k \, \Sigma_k \, V_k^T$$

is the best rank k approximation of A with respect to both the 2-norm and the Frobenius norm

Hence for $\|\cdot\| = \|\cdot\|_2$ or $\|\cdot\| = \|\cdot\|_{\mathrm{F}}$ we have that

$$\|A - B\| \geqslant \|A - A_k\|$$

for any other $m \times n$ matrix of rank $\leqslant k$

The best low rank approximation (cont.)

For both norms, by the orthogonal invariance we have that

$$\|A - A_k\| = \|\Sigma - \Sigma_k\| = \left\| \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & & \sigma_{k+1} & \\ & & & & \ddots \\ & & & & & \sigma_r \\ & & & \mathbb{0} & \end{bmatrix} \right\|$$

Hence

$$\|A - A_k\|_2 = \sigma_{k+1} \quad \text{and} \quad \|A - A_k\|_{\mathrm{F}} = \Big(\sum_{i=k+1}^{r} \sigma_i \Big)^{1/2}$$

Image compression

A B/W image of $m \times n$ pixels can be coded by an $m \times n$ matrix A with entries $a_{i,j} \in [0, 1]$, indicating the brightness of the (i, j)-pixel:

$$0 \text{ (black)} \cdots \text{ gray } \cdots 1 \text{ (white)}$$

Instead of transmitting/storing A, we can replace it by its k-th rank approximation

$$A_k = \sum_{i=1}^{k} \sigma_i \, u_i \, v_i^T,$$

which has size $k\,(m + n + 1)$ (or $k\,(m + n)$ if we store $\sigma_i \, u_i$)

The *relative error of the approximation* is

$$\frac{\|A - A_k\|_2}{\|A\|_2} = \frac{\sigma_{k+1}}{\sigma_1}$$

and its *compression ratio* is

$$\frac{k\,(m + n)}{m\,n}.$$

Image compression (cont.)

Here is a 320×200-picture of a clown and its approximations:

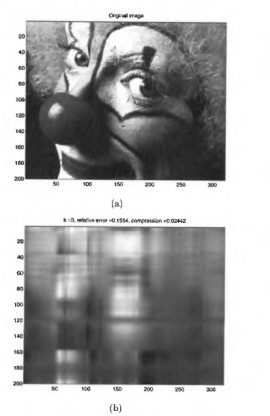


Fig. 3.3. *Image compression using the SVD.* (a) *Original image.* (b) *Rank $k = 3$ approximation.*

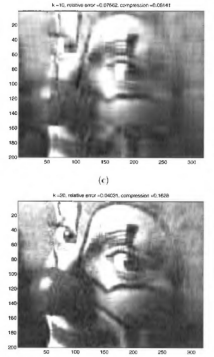


Fig. 3.3. *Continued.* (c) *Rank $k = 10$ approximation.* (d) *Rank $k = 20$ approximation.*

Principal component analysis

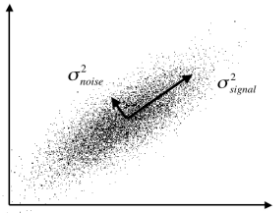
Let M be a data $m \times n$ matrix, for instance:

$$M = \begin{array}{c} \begin{matrix} \text{age} & \text{height} \end{matrix} \\ \begin{bmatrix} & \\ & \\ & \end{bmatrix} \end{array} \text{kids}$$

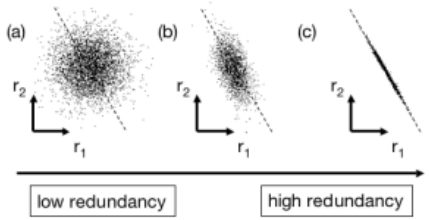
How can we find a simpler description?

Correlated *vs* uncorrelated data

Plotting the samples in $\mathbb{R}^n$ and centering them, we might find that they are *correlated*:



This correlation might be higher *(more significative)* or lower *(less significative)*.



Centering the data

The key parameters in probability and statistics are

mean and variance

The *mean* of the variables is the row n-vector

$$\mu = \frac{1}{n} \sum_{i=1}^{m} \text{row}_i(M)$$

The *centered data* is the $m \times n$ matrix

$$A = \begin{bmatrix} \text{row}_1(M) - \mu \\ \vdots \\ \text{row}_m(M) - \mu \end{bmatrix}$$

$\rightsquigarrow$ the mean of each variable *(=column)* in A is zero

The covariance matrix

The *covariance matrix* of M is the SPD $n \times n$ matrix

$$S = \frac{1}{m-1} A^T A$$

Its diagonal and off-diagonal entries are the *variances* and *covariances* of the variables in A:

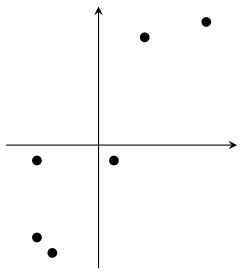
setting $a_j = \text{col}_j(A)$, $j = 1, \ldots, n$, we have that

$$s_{k,k} = \frac{\langle a_k, a_k \rangle}{m-1} = \text{Var}(a_k) \quad \text{and} \quad s_{k,l} = \frac{\langle a_k, a_l \rangle}{m-1} = \text{Cov}(a_k, a_l)$$

Example

Consider the centered data matrix of ages and weights

$$A = \begin{bmatrix} 3 & 7 \\ -4 & -6 \\ 7 & 8 \\ 1 & -1 \\ -4 & -1 \\ -3 & -7 \end{bmatrix}$$



Its covariance matrix is

$$S = \frac{1}{6-1} A^T A = \begin{bmatrix} 20 & 25 \\ 25 & 40 \end{bmatrix}$$

The total variance

The *total variance* of the variables in A is

$$\mathrm{Var}(A) = \sum_{j=1}^{n} \mathrm{var}(a_j) = \frac{1}{m-1} \sum_{i,j} a_{i,j}^2 = \frac{1}{m-1} \|A\|_{\mathrm{F}}$$

Consider the full SVD

$$A = U \Sigma V^T$$

By the orthogonal invariance of the Frobenius norm:

$$\mathrm{Var}(A) = \frac{1}{m-1} \|U \Sigma V^T\|_{\mathrm{F}} = \frac{1}{m-1} \|\Sigma\|_{\mathrm{F}} = \frac{1}{m-1} \sum_{j=1}^{n} \sigma_j^2$$

Orthonormal changes of variables

Let q_i, $i = 1, \ldots, n$, be an orthonormal basis of $\mathbb{R}^n$

For an n-vector x, its representation with respect to this basis is

$$x = \sum_{j=1}^{n} \langle q_j, x \rangle \, q_j = \sum_{j=1}^{n} (x^T q_j) \, q_j$$

Consider the orthogonal $n \times n$ matrix $Q = [q_1 \cdots q_n]$ and set

$$B = A \, Q = \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}$$

so that $b_{i,j} = \mathrm{row}_i(A) \, q_j$

We have that

- the j-variable in B is the linear combination $\langle q_j, x \rangle$ of the variables in A
- for each k, the $m \times k$ matrix $B_k = \begin{bmatrix} b_1 & \cdots & b_k \end{bmatrix}$ gives the projection of the centered data matrix A into the k-th linear subspace

$$\mathrm{span}(b_1, \ldots, b_k)$$

Orthonormal changes of variables (cont.)

By the orthogonal invariance of the Frobenius norm

$$\mathrm{Var}(B) = \frac{1}{m-1} \|B\|_{\mathrm{F}} = \frac{1}{m-1} \|A\|_{\mathrm{F}} = \mathrm{Var}(A)$$

the total variance is invariant by orthogonal changes of variables

Consider again the full SVD

$$A = U \Sigma V^T$$

For $Q = V$ we have that

$$\mathrm{Var}(B_k) = \mathrm{Var}(A \, V_k) = \frac{1}{m-1} \|U \Sigma_k\|_{\mathrm{F}} = \frac{1}{m-1} \sum_{j=1}^{k} \sigma_j^2$$

This is the *maximal* total variance among all possible orthogonal projections of the data into a k-th linear subspace of $\mathbb{R}^n$

The smallest distance

Also, the sum of the squares of the distances between the samples and its projections is *minimal* for this k-th linear subspace:

$$\begin{aligned}\sum_{i=1}^{m} \left\| \operatorname{row}_i(A) - \sum_{j=1}^{k} b_{i,j}\, v_j \right\|_2^2 &= \sum_{i=1}^{m} \left\| \sum_{j=k+1}^{n} b_{i,j}\, v_j \right\|_2^2 \\ &= \|A \cdot [0 \cdots 0\, v_{k+1} \cdots v_n]\|_{\mathrm{F}} \\ &= \sum_{j=k+1}^{n} \sigma_j^2\end{aligned}$$

This is a consequence of the Eckart-Young theorem

Principal components and principal directions

The *j-th principal direction* and the j-th *principal component* of the data matrix M are

$$q_j \in \mathbb{R}^n \quad \text{and} \quad b_j = \begin{bmatrix} \operatorname{row}_1(A)\, q_j \\ \vdots \\ \operatorname{row}_m(A)\, q_j \end{bmatrix} \in \mathbb{R}^m$$

These variables are not correlated, and are ordered according to their variances:

$$\langle b_i, b_j \rangle = 0 \text{ for all } i \neq j \quad \text{and} \quad \langle b_i, b_i \rangle = \sigma_i^2 \text{ for each } i$$

Principal components and principal directions (cont.)

The principal components $b_i = \sigma_i\, u_i$, $i = 1, \ldots, k$, account for

$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2}$$

of the total variance of the data

A good choice of k keeps the true *signal* and discards the *noise*

Example

The full SVD of A is given by

$$\begin{bmatrix} 3 & 7 \\ -4 & -6 \\ 7 & 8 \\ 1 & -1 \\ -4 & -1 \\ -3 & -7 \end{bmatrix} = \begin{bmatrix} -0.44 & -0.36 \\ 0.43 & 0.01 \\ -0.63 & 0.33 \\ 0.02 & 0.35 \\ 0.18 & -0.70 \\ 0.44 & 0.37 \end{bmatrix} \begin{bmatrix} 16.87 & 0 \\ 0 & 3.92 \end{bmatrix} \begin{bmatrix} -0.56 & -0.83 \\ 0.83 & -0.56 \end{bmatrix}$$

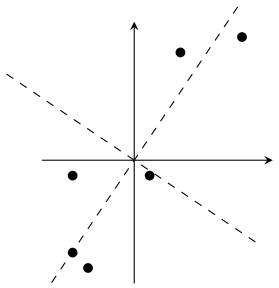
- The columns of $U\Sigma$ are the first and second principal components, and the columns of V give the first and second principal directions
- The variances and covariances of the principal components are given by

$$\frac{1}{6-1}(A\,V)^T(A\,V) = \begin{bmatrix} 56.92 & 0 \\ 0 & 3.07 \end{bmatrix},$$

Example (cont.)

Graphically

$$\begin{bmatrix} 3 & 7 \\ -4 & -6 \\ 7 & 8 \\ 1 & -1 \\ -4 & -1 \\ -3 & -7 \end{bmatrix} = \begin{bmatrix} -0.44 & -0.36 \\ 0.43 & 0.01 \\ -0.63 & 0.33 \\ 0.02 & 0.35 \\ 0.18 & -0.70 \\ 0.44 & 0.37 \end{bmatrix} \begin{bmatrix} 16.87 & 0 \\ 0 & 3.92 \end{bmatrix} \begin{bmatrix} -0.56 & -0.83 \\ 0.83 & -0.56 \end{bmatrix}$$



The full and the reduced SVD's

$$\overset{n}{\underset{m}{\left[A \right]}} = \overset{m}{\underset{m}{\left[U_r \right]}} \left[\Sigma\right]\left[V\right]^T = \overset{r}{\underset{m}{\left[U_r \right]}} \left[\Sigma_r\right]\left[V_r\right]^T,$$

that is, if $U = [u_1 \cdots u_m]$ and $V = [v_1 \cdots v_n]$ then $U_r = [u_1 \cdots u_r]$ and $V_r = [v_1 \cdots v_r]$, and if

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & \mathbb{0} & \end{bmatrix}$$

then $\Sigma_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$

The rank deficient LSP revisited

The SVD also applies to the LSP, and it is particularly appropriate in the rank deficient case

Let A be an $m \times n$ matrix of rank r. If $r < n$ then the solution $x_{\min}$ of the LSP

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$$

is not unique since

$$A\,x_{\min} = A\,(x_{\min} + y)$$

for any $y \in \mathrm{Ker}(A) \simeq \mathbb{R}^{n-r}$

A reasonable choice is the minimizer having the *smallest norm*

The Moore-Penrose inverse and the LSP

The *Moore-Penrose inverse* (or *pseudo-inverse*) of A is the $n \times m$ matrix

$$A^+ = V_r\,\Sigma_r^{-1}\,U_r^T$$

$\rightsquigarrow$ the solution of the LSP can be written as

$$x_{\min} = A^+ b$$

When A is rank deficient, it is the solution with the *smallest norm*

The Moore-Penrose inverse and the LSP (cont.)

The solution given by the pseudo-inverse is well-conditionned when the smallest nonzero singular value of A is not too small:

Changing b to $b + \delta b$ changes x to $x + \delta x$ with

$$\|\delta x\|_2 \leqslant \frac{\|\delta b\|_2}{\sigma_r}$$

because

$$\delta x = V_r \, \Sigma_r^{-1} U_r^T \delta b$$

and so

$$\|\delta x\|_2 \leqslant \|\Sigma_r^{-1}\|_2 \, \|\delta b\|_2 = \frac{\|\delta b\|_2}{\sigma_r}$$

Example

Let

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Then

$$A^+ = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

and so

$$\hat{x} = A^+ b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

with condition number $\frac{1}{\sigma_1} = 1$

Setting $A_\varepsilon = \begin{bmatrix} 1 & 0 \\ 0 & \varepsilon \end{bmatrix}$ for $\varepsilon > 0$ gives

$$\hat{x}_\varepsilon = \begin{bmatrix} 1 \\ \frac{1}{\varepsilon} \end{bmatrix}$$

A practical strategy

The rank is not continuous, and so it might be affected by small perturbations: in the example, round off will make perturbations of size

$$O(\varepsilon) \, \|A\|_2$$

that might increase the condition number from $\frac{1}{\sigma_r}$ to $\frac{1}{\varepsilon}$

The SVD is *backward stable*: round off gives

$$(U + \delta U)(\Sigma + \delta\Sigma)(V + \delta V)^T = A + \delta A$$

with $\|\delta A\|_2 \leqslant O(\varepsilon) \, \|A\|_2$

$\rightsquigarrow$ the computed singular values $\sigma_i + \delta\sigma_i$ verify

$$|\delta\sigma_i| \leqslant O(\varepsilon) \, \|A\|_2$$

A practical strategy

Let $\mathrm{tol} > 0$ be a user supplied measure of uncertainty in A, e.g.

$$\mathrm{tol} = C \, \varepsilon \, \|A\|_2$$

for some small constant $C > 0$

Given the computed factors in the SVD of A

$$\widetilde{U}, \quad \widetilde{\Sigma}, \quad \widetilde{V}$$

set

$$\widehat{\sigma}_i = \begin{cases} \widetilde{\sigma}_i & \text{if } \widetilde{\sigma}_i \geqslant \mathrm{tol} \\ 0 & \text{else} \end{cases}$$

A practical strategy

Replace $\widetilde{\Sigma}$ by the *truncated SVD*

$$\widehat{\Sigma} = \begin{bmatrix} \widetilde{\sigma}_1 & & & & \\ & \ddots & & & \\ & & \widetilde{\sigma}_r & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \\ & & & \mathbb{0} & \end{bmatrix}$$

Then setting

$$\widehat{x} = \widetilde{V}_r \widehat{\Sigma}_r^{-1} \widetilde{U}_r^T$$

the error is bounded by $O(\mathrm{tol})$ and the condition number by $\frac{1}{\sigma_r}$

The PageRank algorithm and the eigenvalue problem

PageRank is the basic algorithm of the Google search machine. It had a huge influence on the development and structure of the internet, since it determines which kind of information and services are accessed more often

There are three main steps for ranking web pages:

1. Crawl the web and locate all public pages
2. Index the data from (1) to allow to search for keywords and phrases within it
3. Rate the importance of these pages

We assume that (1) and (2) are given, and focus on (3):

How can we define and quantify the "importance" of webpages?

The score vector

Suppose that our web of interest contains n pages, each indexed by an integer $1 \leqslant k \leqslant n$

We denote by x_k the *score* of the page k: nonnegative real numbers such $x_j > x_k$ indicates that the page j is more important than the page k

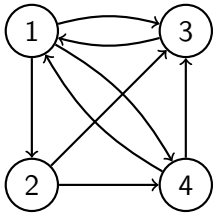
The *score vector*

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n_{\geqslant 0}$$

is normalized so that $\sum_{k=1}^n x_k = 1$

A directed graph

The web can be interpreted as a *directed graph*, where the pages correspond to the nodes and the links to the arrows, e.g.

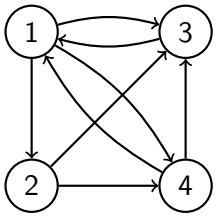


The basic idea

The web is understood as a "democracy" where pages "vote" for the importance of the other pages by linking to them

The simplest approach would consist in *counting links*: in the example this would give

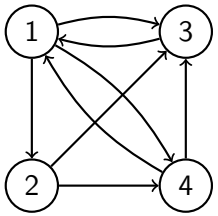
$$x_1 = \frac{2}{8}, \quad x_2 = \frac{1}{8}, \quad x_3 = \frac{3}{8}, \quad x_4 = \frac{2}{8}$$



The basic idea (cont.)

But this forgets an important aspect: receiving a link from an important page should count more!

For instance, pages 1 and 4 have the same number of backlinks (links pointing to them), but page 1 is linked by the important page 3, whereas 4 is linked by the less important page 1



The basic idea (cont.)

Another aspect to be taken into account is that a page should not increase its influence by increasing its number of outgoing links

If the page j has n_j links, then each should contribute with the score x_j/n_j to the page they link

$\rightsquigarrow$ the overall influence of the page j is its score x_j

The basic idea (cont.)

The score vector should satisfy the equations

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j}, \quad k = 1, \ldots, n,$$

with $L_k \subset \{1, \ldots, n\}$ the subset of the indices of the pages linking to the page k

The corresponding *link matrix* $A \in \mathbb{R}^{n \times n}$ is

$$A = \begin{cases} \frac{1}{n_j} & \text{if the page } j \text{ links to the page } k \\ 0 & \text{else} \end{cases}$$

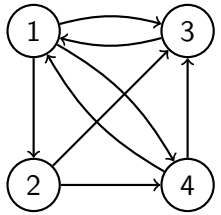
The system of linear equations above is equivalent to

$$Ax = x$$

The score vector is an eigenvector of A with eigenvalue 1

Example (cont.)

In the example



we have

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Example (cont.)

The normalized eigenvector of A is

$$x = \begin{bmatrix} \frac{12}{31} \\ \frac{4}{31} \\ \frac{9}{31} \\ \frac{6}{31} \end{bmatrix} = \begin{bmatrix} 0.387 \\ 0.129 \\ 0.290 \\ 0.194 \end{bmatrix}$$

The page 3 (linked by all the others) is less important than the page 1:

Indeed, the page 3 gives all its "vote" to the page 1 and together with the link from the page 2, gives the page 1 the highest score

Column stochastic matrices

For simplicity: assume that the web has no *dangling nodes*, that is, pages without outgoing links

$\rightsquigarrow$ the link matrix A is *column stochastic*: its entries are nonnegative real numbers and each column sums up to 1:

$$\sum_{i=1}^{n} a_{i,j} = 1, \quad j = 1, \ldots, n$$

Column stochastic matrices (cont.)

Column stochastic matrices have $\lambda = 1$ as one of its eigenvalues: indeed, A and its transposed A^T have the same *characteristic polynomial*:

$$\chi_{A^T} = \det(A^T - t\,\mathbb{1}_n) = \det(A - t\,\mathbb{1}_n)^T = \det(A - t\,\mathbb{1}_n) = \chi_A,$$

$\rightsquigarrow$ the eigenvalues of A and of A^T coincide

Since A is column stochastic then A^T is row stochastic, and so

$$A^T \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Hence $\lambda = 1$ is an eigenvalue of A^T, and so also an eigenvalue of A

The eigenspace

Denote by

$$V_1(A) = \{x \in \mathbb{R}^n \mid Ax = x\}.$$

the eigenspace of the eigenvalue $\lambda = 1$

Since this is an eigenvalue of A, we have that $V_1(A) \neq 0$

We would like to have

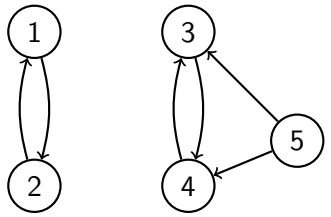
$$\dim(V_1(A)) = 1$$

so that it defines at most one normalized eigenvector

Not always true...

Example

The web



gives the link matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Example (cont.)

Both

$$x = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix}$$

are eigenvectors of A for the eigenvalue 1, and so $V_1(A)$ has dimension at least 2

Which of the vectors in this subspace should be used as a score?

Disconnected webs

In general, this situation arises when the web is disconnected:

Indeed, if the web consists of t subwebs, then the corresponding link matrix A splits into t blocks, and we have that

$$\dim V_1(A) \geqslant t$$

A uniform perturbation

To avoid this phenomenon, we modify the link matrix by adding to it a multiple of the *uniform matrix*

$$S = \left[\tfrac{1}{n}\right]_{i,j} = \begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix}.$$

Then for $0 \leqslant \alpha \leqslant 1$ we set

$$M = (1-\alpha)\, A + \alpha\, S$$

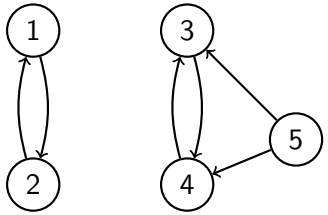
If $\alpha > 0$ then

$$\dim V_1(M) = 1$$

Google's value: $\alpha = 0.15$

Examples (cont.)

For the graph



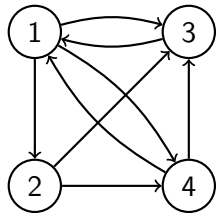
it gives

$$x_1 = 0.2, \quad x_2 = 0.2, \quad x_3 = 0.285, \quad x_4 = 0.285, \quad x_5 = 0.003,$$

allowing to compare the different pages

Examples (cont.)

For the graph



the modified link matrix M gives the scores

$$x_1 = 0.368, \quad x_2 = 0.142, \quad x_3 = 0.288, \quad x_4 = 0.202.$$

Slightly different to those for the link matrix A but giving the same order of importance to the pages

The eigenvalues of the modified link matrix

The fact that $\dim(V_1(M)) = 1$ follows from the following result:

Theorem: Let

$$\lambda_1 = 1, \lambda_2, \ldots, \lambda_n$$

be the eigenvalues of A, repeated according to their multiplicities

Then

$$|\lambda_i| \leqslant 1 \quad \text{for all } i$$

and the eigenvalues of M are

$$\lambda_1 = 1, (1-\alpha)\,\lambda_2, \ldots, (1-\alpha)\,\lambda_n$$

Proof

For each i choose $x = (x_1, \ldots, x_n) \neq 0$ such that $A\,x = \lambda_i\,x$

Then

$$\|A\,x\|_1 \geqslant |\lambda_i|\,\|x\|_1$$

and

$$\|A\,x\|_1 = \sum_{i=1}^{n} \Big| \sum_{j=1}^{n} a_{i,j}\,x_j \Big| \leqslant \sum_{j=1}^{n} \Big(\sum_{i=1}^{n} a_{i,j} \Big)\,|x_j| \leqslant \sum_{j=1}^{n} |x_j| = \|x\|_1$$

because A is column stochastic

Hence $|\lambda_i|\,\|x\|_1 \leqslant \|x\|_1$ and so $|\lambda_i| \leqslant 1$, as stated

Proof (cont.)

Since $U^T A\,U$ is similar to A, it has the same eigenvalues

$\rightsquigarrow$ the eigenvalues of T are $\lambda_2, \ldots, \lambda_n$

We have that

$$U^T e = \begin{matrix} {\scriptstyle 1} \\ {\scriptstyle n-1} \end{matrix} \begin{bmatrix} e^T \\ U_1^T \end{bmatrix} e = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n$$

and so

$$U^T M\,U = (1-\alpha)\,U^T A\,U + \alpha\,U^T e\,e^T U$$
$$= (1-\alpha) \begin{bmatrix} 1 & \mathbb{0} \\ w & T \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ \mathbb{0} \end{bmatrix} \begin{bmatrix} 1 & \mathbb{0} \end{bmatrix} = \begin{bmatrix} 1 & \mathbb{0} \\ (1-\alpha)\,w & (1-\alpha)\,T \end{bmatrix}$$

Hence the eigenvalues of M are $1, (1-\alpha)\,\lambda_2, \ldots, (1-\alpha)\,\lambda_n$, as stated.

Proof (cont.)

Next set

$$e = \begin{bmatrix} \frac{1}{\sqrt{n}} \\ \vdots \\ \frac{1}{\sqrt{n}} \end{bmatrix} \in \mathbb{R}^n$$

This is a unit n-vector such that $S = e\,e^T$

Let U_1 be an $n \times (n-1)$ matrix completing e to an orthonormal $n \times n$ matrix $U = [e\;U_1]$. Then

$$U^T A\,U = \begin{matrix} {\scriptstyle 1} \\ {\scriptstyle n-1} \end{matrix} \begin{bmatrix} e^T A \\ U_1^T A \end{bmatrix} \begin{matrix} [{\scriptstyle 1} & {\scriptstyle n-1} \\ e & U_1] \end{matrix} = \begin{bmatrix} e^T \\ U_1^T A \end{bmatrix} \begin{bmatrix} e & U_1 \end{bmatrix}$$

$$= \begin{matrix} {\scriptstyle 1} \\ {\scriptstyle n-1} \end{matrix} \begin{bmatrix} \overset{1}{1} & \overset{n-1}{e^T U_1} \\ U_1^T A\,e & U_1^T A\,U_1 \end{bmatrix} = \begin{bmatrix} 1 & \mathbb{0} \\ w & T \end{bmatrix}$$

where the second equality follows from the fact that A is column stochastic, and the fourth from the fact that e is orthogonal to U_1

Computing the score vector

To compute the score vector x we apply the *power method*, starting from a well chosen initial vector x_0 and iterating

$$x_k \leftarrow \frac{M\,x_{k-1}}{\|M\,x_{k-1}\|_1}, \quad k \geqslant 1.$$

This iterative method converges towards x, the normalized eigenvector of A corresponding to the largest eigenvalue $\lambda = 1$

The rate of convergence is *linear* and depends on the gap between the largest eigenvalue and the other ones:

the number of correct digits in base b of the k-th approximant x_k is bounded below by

$$-\log_b \|x - x_k\|_1 \geqslant k\,\log_b \Big(\frac{1}{|\lambda_2|} \Big) + \text{constant}$$

Computing the score vector (cont.)

The theorem implies that

$$|\lambda_2| \leqslant 1 - \alpha = 0.85$$

For the base $b = 10$ we have that

$$\log_b \left(\frac{1}{|\lambda_2|} \right) \geqslant 0.07$$

and so

$$-\log_b \|x - x_k\|_1 \geqslant 0.07\, k + \text{constant}$$

Computing the score vector (cont.)

Currently there are $\approx 4 \cdot 10^8$ active public webpages

Each iteration of the power method consists of a matrix-vector multiplication. In a practical implementation, it is computed for $v \in \mathbb{R}^n_{\geqslant 0}$ with $\|v\|_1 = 1$, as

$$M\, v = (1 - \alpha)\, A\, v + \alpha \begin{bmatrix} \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix}$$

because the multiplication of v by the uniform matrix S gives

$$S\, v = \begin{bmatrix} \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix}$$

Eigenvalues

Let A be an $n \times n$ matrix with complex coefficients

An element $\lambda \in \mathbb{C}$ is an *eigenvalue* of A if there is $x \in \mathbb{C}^n \backslash \{0\}$ such that

$$A\, x = \lambda\, x$$

that is, if A is a homothecy in the direction of this n-vector

We set

$$\lambda(A) = \{\text{eigenvalues of } A\} \quad \text{the } \textit{spectrum} \text{ of } A$$

The roots of the characteristic polynomial

The *characteristic polynomial* of the $n \times n$-matrix A is

$$\chi_A = \det(A - z\, \mathbb{1}_n) \in \mathbb{C}[z]_n,$$

where $\mathbb{C}[z]_n$ denotes the space of degree n polynomials with complex coefficients

Let $\lambda \in \mathbb{C}$. TFAE:

- $\lambda \in \lambda(A)$
- $A - \lambda\, \mathbb{1}_n$ is singular
- $\det(A - \lambda\, \mathbb{1}_n) = 0$
- λ is a root of χ_A

Hence

$$\lambda(A) = \{\lambda \in \mathbb{C} \mid \chi_A(\lambda) = 0\}.$$

Eigenvalues can be complex even if A is real!

For instance, if

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

then $\lambda(A) = \{\pm i\}$

Eigenspaces and multiplicities

For $\lambda \in \lambda(A)$, the associate *eigenspace* is

$$V_\lambda(A) = \{x \in \mathbb{C}^n \mid A\,x = \lambda\,x\}$$

It is an *invariant subsapce*:

$$A\,V_\lambda(A) \subset V_\lambda(A)$$

- The *geometric multiplicity* of λ is the dimension $\dim(V_\lambda(A))$
- The *algebraic multiplicity* of λ, denoted by $e_\lambda(A)$, is the exponent of the factor $\lambda - z$ in the irreducible factorization of χ_A

These quantities are related by

$$1 \leqslant \dim(V_\lambda(A)) \leqslant e_\lambda(A)$$

Similarities

An $n \times n$ matrix B is *similar* to A if there is a nonsingular $n \times n$ matrix S such that

$$A = S\,B\,S^{-1}$$

The matrices A and B have the same eigenvalues, and the eigenvectors of A can be read from those of B:

$$\lambda(A) = \lambda(B),$$

and y is an eigenvector of B for λ if and only if $S\,y$ is an eigenvector of A for λ

Similarities (cont.)

If $B = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix, then

$$e_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_i$$

is an eigenvector of B for λ_i and so the i-th column in the matrix S

$$s_i = S\,e_i$$

is an eigenvector of A for λ_i

Decoupling

Many eigenvalue computations involve breaking down the problem into smaller ones *(decoupling)*: if

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ \mathbb{0} & A_{2,2} \end{bmatrix}$$

then $\lambda(A) = \lambda(A_{1,1}) \cup \lambda(A_{2,2})$

In particular, if A is upper triangular then its eigenvalues coincide with its diagonal entries.

The Schur decomposition

An $n \times n$ matrix Q is *unitary* if

$$Q^* = Q^{-1}$$

For reasons of numerical stability, we prefer to consider similarities given by unitary matrices

$\rightsquigarrow$ the *Schur decomposition*:

$$A = Q\, T\, Q^*$$

with Q unitary and T upper triangular

The Schur decomposition (cont.)

The existence of this factorization can be verified by induction:

$n = 1$:

$$Q = [1] \quad \text{and} \quad T = A$$

$n > 1$:

Take an eigenvalue $\lambda \in \lambda(A)$ and a unit eigenvector $u \in \mathbb{C}^n$ of it, and choose an $n \times (n-1)$ matrix $\widetilde{U}$ such that $U = [u\, \widetilde{U}]$ is unitary. Then

$$U^* A\, U = \begin{matrix} 1 \\ n-1 \end{matrix}\!\!\begin{bmatrix} u^* \\ \widetilde{U}^* \end{bmatrix} A \begin{bmatrix} \overset{1}{u} & \overset{n-1}{\widetilde{U}} \end{bmatrix} = \begin{matrix} 1 \\ n-1 \end{matrix}\!\!\begin{bmatrix} \overset{1}{u^* A\, u} & \overset{n-1}{u^* A\, \widetilde{U}} \\ \widetilde{U}^* A\, u & \widetilde{U}^* A\, \widetilde{U} \end{bmatrix}$$

We have that $u^* A\, u = u^* \lambda\, u = \lambda$ and $\widetilde{U}^* A\, u = \widetilde{U}^* \lambda\, u = \mathbb{0}$, and so

$$U^* A\, U = \begin{bmatrix} \lambda & \widetilde{a}_{1,2} \\ \mathbb{0} & \widetilde{A}_{2,2} \end{bmatrix}$$

The Schur decomposition (cont.)

By induction, there is an $(n-1) \times (n-1)$-unitary matrix P and an $(n-1) \times (n-1)$-upper triangular matrix $\widetilde{T}$ such that

$$\widetilde{A}_{2,2} = P\, \widetilde{T}\, P^*$$

Hence

$$A = U \begin{bmatrix} \lambda & \widetilde{a}_{1,2} \\ \mathbb{0} & \widetilde{A}_{2,2} \end{bmatrix} U^* = Q\, T\, Q^*$$

with

$$Q = U \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & P \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} \lambda & \widetilde{a}_{1,2} P \\ \mathbb{0} & \widetilde{T} \end{bmatrix}$$

Some remarks

- The Schur decomposition is *not unique*, e.g. the eigenvalues can appear in the diagonal of T in any possible order
- The columns
$$Q = [q_1 \cdots q_n]$$
are called the *Schur vectors* of A. Writing $T = [t_{i,j}]_{i,j}$, for $k = 1, \ldots, n$ we have that
$$A\, q_k = \sum_{i=1}^{k} t_{i,k}\, q_i$$
and so the linear subspace $\operatorname{span}(q_1, \ldots, q_k)$ is invariant
- When A is real symmetric, it admits a Schur decomposition $A = Q\, T\, Q^*$ with Q orthogonal and T diagonal

Diagonalization

A is *diagonalizable* if there exists a nonsingular matrix S and a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ such that
$$A = S\, \Lambda\, S^{-1}.$$

This is equivalent to the fact that the geometric and the algebraic multiplicities of the eigenvalues of A coincide, that is,
$$\dim V_\lambda(A) = e_\lambda(A) \quad \text{for all } \lambda \in \lambda(A)$$

The Jordan decomposition

It is the factorization
$$A = S\, J\, S^{-1}$$
with J a block diagonal matrix $J = [J_1 \cdots J_q]$ where each block is of the form
$$J_i = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix} \in \mathbb{C}^{n_i \times n_i}$$
and $n_1 + \cdots + n_q = n$

The Jordan decomposition (cont.)

This decomposition gives full information about the eigenvalues and the eigenvectors of A:

- for each i, the eigenvalue of J_i is λ_i with eigenvector e_i
- the eigenvalues of A are the λ_i's
- for each λ_i the basis of the eigenspace $V_\lambda(A)$ is given by the columns of S corresponding to the first columns of the Jordan blocks with eigenvalue λ_i

Drawbacks

- The Jordan decomposition is *not* continuous
- It is difficult to compute numerically for a non diagonalizable matrix
- It cannot be computed stably: after computing S and J, we cannot guarantee that

$$A + \delta A = S\, J\, S^{-1}$$

with δA small, because S might have a large condition number

Example 1

The Jordan form of $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ is

$$J = A$$

For $\varepsilon_1 \neq \varepsilon_2$ small, the perturbed matrix

$$\widetilde{A} = \begin{bmatrix} \varepsilon_1 & 1 \\ 0 & \varepsilon_2 \end{bmatrix}$$

has two different eigenvalues ε_1 and ε_2, and so its Jordan form is

$$\widetilde{J} = \begin{bmatrix} \varepsilon_1 & 0 \\ 0 & \varepsilon_2 \end{bmatrix}$$

Example 2

Let

$$A = \begin{bmatrix} 1+\varepsilon & 1 \\ 0 & 1-\varepsilon \end{bmatrix}$$

with ε small. Up to a scalar, the eigenvectors of A are

$$s_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad s_2 = \begin{bmatrix} 1 \\ -2\,\varepsilon \end{bmatrix}$$

The Jordan decomposition of A is

$$A = \begin{bmatrix} 1 & 1 \\ 0 & -2\,\varepsilon \end{bmatrix} \begin{bmatrix} 1+\varepsilon & 0 \\ 0 & 1-\varepsilon \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2\varepsilon} \\ 0 & \frac{-1}{2\varepsilon} \end{bmatrix}$$

and so $\kappa_\infty(S) \approx \varepsilon^{-1}$

Computing the eigenvectors from the Schur decomposition

Consider the Schur decomposition

$$A = Q\, T\, Q^*$$

and y an eigenvector of T for an scalar λ. Then

$$A\, Q\, y = Q\, T\, y = \lambda\, Q\, y$$

and so $Q\, y$ is an eigenvector of A with eigenvalue λ

$\rightsquigarrow$ to find the eigenvalues of A it is enough to find those of T

Computing the eigenvectors (cont.)

Suppose that $\lambda = t_{i,i}$ is *simple*, that is $e_\lambda(A) = 1$

Write the equation $(T - \lambda \mathbb{1}_n)\, y = 0$ as

$$0 = \begin{matrix} \\ i-1 \\ 1 \\ n-i \end{matrix} \begin{matrix} \begin{matrix} i-1 & 1 & n-i \end{matrix} \\ \begin{bmatrix} T_{1,1} - \lambda \mathbb{1}_{i-1} & T_{1,2} & T_{1,3} \\ \mathbb{0} & 0 & T_{2,3} \\ \mathbb{0} & 0 & T_{3,3} - \lambda \mathbb{1}_{n-i} \end{bmatrix} \end{matrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

$$= \begin{bmatrix} (T_{1,1} - \lambda \mathbb{1}_{i-1})\, y_1 + T_{1,2}\, y_2 + T_{1,3}\, y_3 \\ T_{2,3}\, y_3 \\ (T_{3,3} - \lambda \mathbb{1}_{n-i})\, y_3 \end{bmatrix}$$

Computing the eigenvectors (cont.)

Since λ is simple, both $T_{1,1} - \lambda \mathbb{1}_{i-1}$ and $T_{3,3} - \lambda \mathbb{1}_{n-i}$ are nonsingular, and so

$$y_3 = 0 \in \mathbb{R}^{i-1}$$

Set $y_2 = 1$ and compute

$$y_1 = -(T_{1,1} - \lambda \mathbb{1}_{i-1})^{-1} T_{1,2} \in \mathbb{R}^{n-i}$$

solving an upper triangular system

The resulting eigenvector is

$$y = \begin{bmatrix} -(T_{1,1} - \lambda \mathbb{1}_{i-1})^{-1} T_{1,2} \\ 1 \\ \mathbb{0} \end{bmatrix}$$

Example

For $T = \begin{bmatrix} 3 & -3 \\ 0 & 2 \end{bmatrix}$ the eigenvector for $\lambda = 2$ is computed as

$$(T - \lambda \mathbb{1}_2) \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} y_1 - 3\, y_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which gives

$$y = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Hence for a Schur decomposition $A = Q\, T\, Q^*$, the eigenvector for $\lambda = 2$ is

$$Q \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 3\, q_1 + q_2$$

Perturbation theory

Eigenvalues are continuous with respect to perturbations of the matrix, but they might have an *infinite* condition number, because in general their variation is not bounded by a linear function

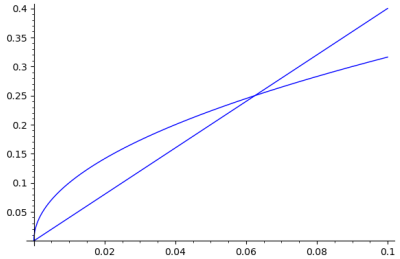
Example

Let

$$A_\varepsilon = \begin{bmatrix} 0 & 1 \\ \varepsilon & 0 \end{bmatrix}$$

with ε small

$\rightsquigarrow \lambda(A_\varepsilon) = \{\pm\varepsilon^{1/2}\}$, and $\varepsilon^{1/2}$ grows faster than $|\varepsilon|$ for $\varepsilon \to 0$



For $\varepsilon = 0$, the condition number of the eigenvalue of $\lambda = 0$ is $+\infty$

Example (cont.)

More formally, set

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad A + \delta A = \begin{bmatrix} 0 & 0 \\ \varepsilon & 0 \end{bmatrix}$$

and let $\lambda = 0$ and $\lambda + \delta\lambda$ be the respective eigenvalues. Hence

$$|\delta\lambda| > c\,\frac{\|\delta A\|}{\|A\|} \approx \varepsilon$$

for any fixed $c > 0$ and ε sufficiently small

The condition number

Let A be an $n \times n$ matrix and λ a simple eigenvalue of it. The *condition number* of λ is defined as

$$\kappa(A, \lambda) = \frac{1}{y^* x}$$

where x and y are unit eigenvalues of A for λ and of A^* for $\overline{\lambda}$, respectively:

$$A x = \lambda x \quad \text{and} \quad y^* A = \lambda y^*$$

The condition number (cont.)

For a small perturbation δA we have that

$$\begin{array}{rcc} (A + \delta A) \cdot (x + \delta x) & = & (\lambda + \delta\lambda) \cdot (x + \delta x) \\ - \quad A x & = & \lambda x \\ \hline A \cdot \delta x + \delta A \cdot x + \delta A \cdot \delta x = \lambda \cdot \delta x + \delta\lambda \cdot x + \delta\lambda \cdot \delta x & & \end{array}$$

If we ignore the second order terms and multiply by y^* we obtain

$$y^* \cdot A \cdot \delta x + y^* \cdot \delta A \cdot x = y^* \cdot \lambda \cdot \delta x + y^* \cdot \delta\lambda \cdot x$$

Hence up to second order

$$\delta\lambda = \frac{y^* \cdot \delta A \cdot x}{y^* \cdot x}$$

and so $|\delta\lambda| = \kappa(A, \lambda)\,\|\delta A\|$

Example (cont.)

Let

$$A_\varepsilon = \begin{bmatrix} 0 & 1 \\ \varepsilon & 0 \end{bmatrix}$$

with ε small

The eigenvalues of A and A^* for $\lambda_\varepsilon = \varepsilon^{1/2}$ are

$$x_\varepsilon = \frac{1}{(1+\varepsilon)^{1/2}} \begin{bmatrix} 1 \\ \varepsilon^{1/2} \end{bmatrix} \quad \text{and} \quad y_\varepsilon = \frac{1}{(1+\varepsilon)^{1/2}} \begin{bmatrix} \varepsilon^{1/2} \\ 1 \end{bmatrix}$$

Hence

$$\kappa(A_\varepsilon, \lambda_\varepsilon) = \frac{1}{y_\varepsilon^* \cdot x_\varepsilon} = \frac{1+\varepsilon}{2\,\varepsilon^{1/2}} \approx \frac{1}{2\,\varepsilon^{1/2}} \longrightarrow_{\varepsilon \to 0} +\infty$$

The symmetric case

When A is symmetric the condition number of a simple eigenvalue is 1: in this case $x = y$ and so

$$\kappa(A, \lambda) = \frac{1}{y^* \cdot x} = \frac{1}{\|x\|_2^2} = 1$$

Computing eigenvalues and eigenvectors

Let A be an $n \times n$ matrix and consider the *QR iteration*:

$H_0 \leftarrow Q_0^*\, A\, Q_0$
for $k = 1, 2, \ldots$
  $H_{k-1} = Q_k\, R_k$ *(QR factorization)*
  $H_k \leftarrow R_k\, Q_k$

Since $H_k = R_k\, Q_k = Q_k^*\, Q_k\, R_k\, Q_k = Q_k^*\, H_{k-1}\, Q_k$ we have that

$$A = (Q_0 \cdots Q_k)\, H_k (Q_0 \cdots Q_k)^*$$

and so H_k unitarily similar to A

In most situations

$$H_k \longrightarrow_{k \to \infty} H \quad \text{upper triangular } \textit{(Schur form)}$$

but this is less obvious!

The power method

Fix a norm $\|\cdot\|$. The *power method* is the iteration

Choose $x_0 \in \mathbb{C}^n$ with $\|x_0\| = 1$
for $k = 0, 1, 2, \ldots$
  $y_{k+1} \leftarrow A\, x_k$
  $x_{k+1} \leftarrow \frac{y_{k+1}}{\|y_{k+1}\|}$

When stopping at an integer l we set

$\widetilde{x} \leftarrow x_{l+1}$ approximate eigenvector
$\widetilde{\lambda} \leftarrow \widetilde{x}^* A\, \widetilde{x}$ approximate eigenvalue

Error analysis

Suppose A diagonalizable, that is

$$A = S \Lambda S^{-1}$$

with

$$S = [s_1 \cdots s_n] \quad \text{and} \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$$

and suppose that

$$|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$$

Error analysis (cont.)

Write

$$x_0 = a_1 s_1 + \cdots + a_n s_n$$

and suppose furthemore that $a_1 \neq 0$. Then

$$A^k x_0 = a_1 \lambda_1^k s_1 + \cdots a_n \lambda_n^k s_n = a_1 \lambda_1^k \Big(s_1 + \sum_{j=2}^{n} \frac{a_j}{a_1} \Big(\frac{\lambda_j}{\lambda_1} \Big)^k s_j \Big)$$

and so for $x_k = \frac{A^k x_0}{\|A^k x_0\|}$ and $\widetilde{\lambda}_k = x_k^* A x_k$ we have that

$$\|x_k - s_1\|, \|\widetilde{\lambda}_k - \lambda_1\| \leqslant O\Big(\Big| \frac{\lambda_2}{\lambda_1} \Big|^k \Big)$$

The number of correct digits in base b of these approximations is

$$-\log_b \|x_k - s_1\|_2, -\log_b \|\widetilde{\lambda}_k - \lambda_1\|_2 \geqslant k \log_b \Big(\Big| \frac{\lambda_1}{\lambda_2} \Big| \Big) + \text{constant}$$

Pros and cons

- It is based on matrix-vector multiplications
- Its speed of convergence depends on the ratio $|\lambda_1|/|\lambda_2|$
- It assumes $a_1 \neq 0$

In PageRank:

- Matrix-vector multiplications reduce to multiplications by the link matrix of the web, which is sparse
- The ratio $|\lambda_1|/|\lambda_2|$ is uniformly bounded below:

$$\frac{|\lambda_1|}{|\lambda_2|} \geqslant 0.85^{-1} \approx 1.18$$

- x_0 is chosen as the score vector of the previous web

Pros and cons (cont.)

- It only converges to a pair eigenvalue/eigenvector only if there is a dominant one
- the convergence is only *linear*

The power method does not converge in many situations, including:

- orthogonal matrices (all eigenvalues have absolute value 1)
- real matrices with complex eigenvalues (the eigenvalues come in pairs of conjugate complex numbers)

Orthogonal iteration

It is generalization of the power method aimed to compute higher dimensional invariant subspaces

Fix $1 \leqslant r \leqslant n$. The *orthogonal iteration* writes down as

Choose a unitary $n \times r$ matrix Q_0
for $k = 0, 1, 2, \ldots$
$Y_{k+1} \leftarrow A\, Q_k$
$Y_{k+1} = Q_{k+1}\, R_{k+1}$ *(QR factorization)*

For $r = 1$ it coincides with the power method:

$$y_{k+1} = x_{k+1}\, \|A\, y_{k+1}\|_2$$

is the QR factorization of y_{k+1}

$$\rightsquigarrow Q_{k+1} = x_{k+1} \text{ for all } k$$

Funtoriality

The QR factorization is compatible with restricting columns:

$$\left[\begin{array}{c|c} A' & A \end{array}\right] = \left[\begin{array}{c|c} Q' & Q \end{array}\right] \left[\begin{array}{c|c} R' & R \\ \hline & \end{array}\right]$$

so that if $A = Q\, R$ is the QR factorization of A, then $A' = Q'\, R'$ is the QR factorization of A'

$\rightsquigarrow$ for $1 \leqslant s \leqslant r$ the first s columns of Q_k coincide with the orthogonal iteration of dimension s starting with the first s columns of Q_0

Setting $r = n$ runs the orthogonal iteration simultaneously for all intermediate dimensions

Error analysis

Suppose $A = S\, \Lambda\, S^{-1}$ diagonalizable with

$$S = [s_1 \cdots s_n] \quad \text{and} \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$$

and that

$$|\lambda_1| \geqslant \cdots \geqslant |\lambda_r| > |\lambda_{r-1}| \geqslant \cdots \geqslant |\lambda_n|$$

Then

$$\operatorname{span}(Q_k) = \operatorname{span}(Y_k) = \cdots = \operatorname{span}(A^k\, Q_0)$$

$\rightsquigarrow$ Q_k gives an orthonormal basis of the linear subspace $\operatorname{span}(A^k\, Q_0)$

For $k \to \infty$ it converges to a basis of the invariant subspace generated by the *dominant eigenvectors* $s_1, \ldots, s_r$:

$$\operatorname{span}(Q_k) = \operatorname{span}(A^k\, Q_0) \longrightarrow_{k \to \infty} \operatorname{span}(s_1, \ldots, s_r)$$

Convergence

Suppose for simplicity that

$$|\lambda_1| > \cdots > |\lambda_n|$$

Then for a "typical" unitary $n \times n$ matrix Q_0, for $k \to \infty$ we have that $Q_k \to Q$ unitary and $Q_k^*\, A\, Q_k \to T$ upper triangular, giving the Schur decomposition

$$A = Q\, T\, Q^*$$

Moreover

$$\|Q - Q_k\|, \|T - Q_k^*\, A\, Q_k\| \leqslant O\left(\max_i \frac{|\lambda_{i+1}|}{|\lambda_i|} \right)$$

The QR iteration

Set

$$T_k = Q_k^* A Q_k \quad \text{for } k = 0, 1, 2, \ldots.$$

The QR iteration arises when considering how to compute T_k directly from T_{k-1}

From the definition of Q_k and R_k and the relation $Q_{k-1} A = R_k Q_k$:

$$T_{k-1} = Q_{k-1}^* A Q_{k-1} = (Q_{k-1}^* Q_k) R_k \tag{6}$$
$$T_k = Q_k^* A Q_k = (Q_k^* A Q_{k-1}) Q_{k-1}^* Q_k = R_k (Q_{k-1}^* Q_k) \tag{7}$$

We have that $Q_{k-1}^* Q_k$ is unitary and (8) is the QR factorization of T_{k-1}

Drawbacks

- a single iteration costs $O(n^3)$ flops
- convergence (when it exists) is only linear

The real QR iteration

Matrices arising from applications have real entries! From now

$$\boxed{A \text{ real } n \times n \text{ matrix}}$$

The *real QR iteration* is defined by choosing an orthogonal $n \times n$ matrix Q_0 and setting

$H_0 \leftarrow Q_0^T A Q_0$
for $k = 1, 2, \ldots$
  $H_{k-1} = Q_k R_k$ *(QR factorization)*
  $H_k \leftarrow R_k Q_k$

The real Schur form

If the eigenvalues of A are not real, then T_k cannot converge to an upper triangular matrix

$\rightsquigarrow$ we content ourselves with convergence to the *real Schur form*:

$$A = Q T Q^T$$

with Q orthogonal and T block upper triangular with 1×1 and 2×2 diagonal blocks, that is

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} & \cdots & T_{1,n} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & T_{n-1,n} \\ 0 & \cdots & 0 & T_{n,n} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

where each $T_{i,i}$ is a 1×1 block or a 2×2 block with complex conjugate eigenvalues

Hessenberg reduction

To implement the QR iteration, we have to choose Q_0 carefully such that

$$T_0 = Q_0^T A Q_0$$

is an upper Hessenberg matrix, which would lower the complexity to $O(n^2)$ flops

Hessenberg reduction (cont.)

It is a variation of the QR factorization, and can be done with a sequence of Householder reflections

Let $n = 5$ and choose $P_1 = \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & \tilde{P}_1 \end{bmatrix}$ with $\tilde{P}_1$ a Householder 4×4 matrix such that

$$P_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, \quad A_1 = P_1 A P_1^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

- P_1 leaves the first row of $P_1 A$ unchanged
- P_1^T leaves the first column of $(P_1 A) P_1^T$ unchanged, including the zeros

Hessenberg reduction (cont.)

Choose $P_2 = \begin{bmatrix} \mathbb{1}_2 & \mathbb{0} \\ \mathbb{0} & \tilde{P}_2 \end{bmatrix}$ with $\tilde{P}_2$ a 3×3 Householder such that

$$P_2 A_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}, \quad A_2 = P_2 A_1 P_2^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

- P_1 leaves the first and second rows of $P_2 A_1$ unchanged
- P_2^T leaves the first and second columns of $(P_2 A_2) P_2^T$ unchanged

Hessenberg reduction (cont.)

Finally choose $P_3 = \begin{bmatrix} \mathbb{1}_3 & \mathbb{0} \\ \mathbb{0} & \tilde{P}_3 \end{bmatrix}$ with $\tilde{P}_2$ a 2×2 Householder s.t.

$$P_3 A_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}, \quad A_3 = P_3 A_2 P_3^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

This constructs an orthogonal $n \times n$ matrix

$$Q_0 = P_{n-2} \cdots P_1$$

such that $H_0 = Q_0^T A Q_0$ is Hessenberg

The complexity of this procedure is $5\,n^3 + O(n^2)$ flops

The Hessenberg form through the QR iteration

The Hessenberg form is preserved during the QR iteration: let

$$H = Q\,R \quad \text{and} \quad H_+ = R\,Q$$

with H Hessenberg

Since R is upper triangular, the j-th column of Q is a linear combination of the first columns of H: write

$$H = [h_1 \cdots h_n] \quad \text{and} \quad Q = [q_1 \cdots q_n]$$

Setting $S = R^{-1} = [s_{i,j}]_{i,j}$ we have that

$$q_j = \sum_{k=1}^{j} s_{k,j}\, h_k, \quad j = 1, \ldots, n$$

and so Q is Hessenberg

Similarly the j-th row of H_+ is a linear combination of the last j rows of Q, and so H_+ is Hessenberg

The QR iteration

Set

$$T_k = Q_k^*\, A\, Q_k \quad \text{for } k = 0, 1, 2, \ldots.$$

The QR iteration arises when considering how to compute T_k directly from T_{k-1}

From the definition of Q_k and R_k and the relation $Q_{k-1}\, A = R_k\, Q_k$:

$$T_{k-1} = Q_{k-1}^* A\, Q_{k-1} = (Q_{k-1}^* Q_k)\, R_k \tag{8}$$

$$T_k = Q_k^*\, A\, Q_k = (Q_k^*\, A\, Q_{k-1})\, Q_{k-1}^* Q_k = R_k\, (Q_{k-1}^* Q_k) \tag{9}$$

We have that $Q_{k-1}^* Q_k$ is unitary and (8) is the QR factorization of T_{k-1}

Drawbacks

- a single iteration costs $O(n^3)$ flops
- convergence (when it exists) is only linear

The real QR iteration

Matrices arising from applications have real entries! From now

$$\boxed{A \text{ real } n \times n \text{ matrix}}$$

The *real QR iteration* is defined by choosing an orthogonal $n \times n$ matrix Q_0 and setting

$H_0 \leftarrow Q_0^T\, A\, Q_0$
for $k = 1, 2, \ldots$
  $H_{k-1} = Q_k\, R_k$ *(QR factorization)*
  $H_k \leftarrow R_k\, Q_k$

The real Schur form

If the eigenvalues of A are not real, then T_k cannot converge to an upper triangular matrix

$\rightsquigarrow$ we content ourselves with convergence to the *real Schur form*:

$$A = Q\, T\, Q^T$$

with Q orthogonal and T block upper triangular with 1×1 and 2×2 diagonal blocks, that is

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} & \cdots & \cdots & T_{1,n} \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & T_{n-1,n} \\ 0 & \cdots & \cdots & 0 & T_{n,n} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

where each $T_{i,i}$ is a 1×1 block or a 2×2 block with complex conjugate eigenvalues

Hessenberg reduction

To implement the QR iteration, we have to choose Q_0 carefully such that

$$T_0 = Q_0^T A\, Q_0$$

is an upper Hessenberg matrix, which would lower the complexity to $O(n^2)$ flops

Hessenberg reduction (cont.)

It is a variation of the QR factorization, and can be done with a sequence of Householder reflections

Let $n = 5$ and choose $P_1 = \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & \widetilde{P}_1 \end{bmatrix}$ with $\widetilde{P}_1$ a Householder 4×4 matrix such that

$$P_1 A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}, \quad A_1 = P_1 A P_1^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

- P_1 leaves the first row of $P_1 A$ unchanged
- P_1^T leaves the first column of $(P_1 A)\, P_1^T$ unchanged, including the zeros

Hessenberg reduction (cont.)

Choose $P_2 = \begin{bmatrix} \mathbb{1}_2 & \mathbb{0} \\ \mathbb{0} & \widetilde{P}_2 \end{bmatrix}$ with $\widetilde{P}_2$ a 3×3 Householder such that

$$P_2 A_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}, \quad A_2 = P_2 A_1 P_2^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

- P_1 leaves the first and second rows of $P_2 A_1$ unchanged
- P_2^T leaves the first and second columns of $(P_2 A_2)\, P_2^T$ unchanged

Hessenberg reduction (cont.)

Finally choose $P_3 = \begin{bmatrix} \mathbb{1}_3 & \mathbb{0} \\ \mathbb{0} & \widetilde{P}_3 \end{bmatrix}$ with $\widetilde{P}_2$ a 2×2 Householder s.t.

$$P_3 A_2 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}, \quad A_3 = P_3 A_2 P_3^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

This constructs an orthogonal $n \times n$ matrix

$$Q_0 = P_{n-2} \cdots P_1$$

such that $H_0 = Q_0^T A Q_0$ is Hessenberg

The complexity of this procedure is $5\,n^3 + O(n^2)$ flops

The Hessenberg form through the QR iteration (cont.)

The QR factorization of H is computed with $n-1$ Givens rotations:

$$Q^T H = R$$

where R is upper triangular and $Q = G_1 \cdots G_{n-1}$ with

$$G_i = \text{Givens}(i, i+1, \theta_i)$$

Then

$$H_+ = R\,Q = R\,(G_1 \cdots G_{n-1})$$

This requires $6\,n^2 + O(n)$ flops

The Hessenberg form through the QR iteration

The Hessenberg form is preserved during the QR iteration: let

$$H = Q\,R \quad \text{and} \quad H_+ = R\,Q$$

with H Hessenberg

Since R is upper triangular, the j-th column of Q is a linear combination of the first columns of H: write

$$H = [h_1 \cdots h_n] \quad \text{and} \quad Q = [q_1 \cdots q_n]$$

Setting $S = R^{-1} = [s_{i,j}]_{i,j}$ we have that

$$q_j = \sum_{k=1}^{j} s_{k,j}\, h_k, \quad j = 1, \ldots, n$$

and so Q is Hessenberg

Similarly the j-th row of H_+ is a linear combination of the last j rows of Q, and so H_+ is Hessenberg

Deflation

A Hessenberg matrix $H \in \mathbb{R}^{n \times n}$ is *reduced* if it has a zero subdiagonal entry

In this case there is $0 < p < n$ such that

$$H = \begin{array}{c} \\ {\scriptstyle p} \\ {\scriptstyle n-p} \end{array} \begin{array}{c} \begin{array}{cc} {\scriptstyle p} & {\scriptstyle n-p} \end{array} \\ \begin{bmatrix} H_{1,1} & H_{2,2} \\ \mathbb{0} & H_{2,2} \end{bmatrix} \end{array}$$

and the eigenproblem problem for H decouples into the two smaller problems for $H_{1,1}$ and $H_{2,2}$ *(deflation)*

In practice, this happens when a subdiagonal entry is sufficiently small, for instance when

$$|h_{p+1,p}| \leqslant c\,\varepsilon\,(|h_{p,p}| + |h_{p+1,p+1}|)$$

for ε the machine epsilon and c a small constant

Typically for $p = n-1$ and $p = n-2$

The shifted QR iteration

For $\mu \in \mathbb{R}$ consider the single shift QR iteration

$H_0 \leftarrow Q_0^T A\, Q_0$ *(Hessenberg reduction)*
for $k = 1, 2, \ldots$
$\quad H_{k-1} - \mu\, \mathbb{1}_n = Q_k\, R_k$ *(QR factorization)*
$\quad H_k \leftarrow R_k\, Q_k + \mu\, \mathbb{1}_n$

Each H is Hessenberg and orthogonally similar to A:

$$\widetilde{H} = R\, Q + \mu\, \mathbb{1}_n = Q^T (Q\, R + \mu\, \mathbb{1}_n)\, Q = Q^T H\, Q$$

The shifted QR iteration (cont.)

If μ is an eigenvalue then deflation occurs in one step

$$\widetilde{h}_{n,n-1} = 0 \quad \text{and} \quad \widetilde{h}_{n,n} = \lambda$$

In practice, we recognize that the QR iteration is converging when $h_{n,n-1}$ is small

When this occurs, we use $\mu = h_{n,n}$ and we obtain a quadratically converging algorithm: if

$$|h_{n,n-1}| \leqslant \eta\, \|H\| \quad \text{with } 0 \leqslant \eta \ll 1$$

then for $\mu = h_{n,n}$ we have that

$$|\widetilde{h}_{n,n-1}| \leqslant O(\eta^2 \|H\|)$$

$\rightsquigarrow$ the number of correct digits of $h_{n,n} \approx \lambda$ *doubles* at each step

The double shift strategy

When the eigenvalue λ of A is not real, we might perform two complex single shifts QR steps in succession:

$$\begin{aligned} H - \lambda\, \mathbb{1}_n &= Q_1\, R_1 \\ H_1 &\leftarrow R_1\, Q_1 + \lambda\, \mathbb{1}_n \\ H_1 - \overline{\lambda}\, \mathbb{1}_n &= Q_2\, R_2 \\ H_2 &\leftarrow R_2\, Q_2 + \overline{\lambda}\, \mathbb{1}_n \end{aligned}$$

Set

$$Q = Q_1\, Q_2 \quad \text{and} \quad R = R_2\, R_1$$

It can be shown that these $n \times n$ matrices are *real* and that

$$H_2 = Q^T\, H\, Q \in \mathbb{R}^{n \times n}$$

The double shift strategy (cont.)

To avoid complex arithmetics altogether, we would like to pass from H to H_2 directly and using only real numbers

Set $M = Q\, R$, which verifies that

$$M = H^2 - 2\, \mathrm{Re}(\lambda)\, H + |\lambda|^2\, \mathbb{1}_n$$

Then the strategy is:

- compute $M = H^2 - 2\, \mathrm{Re}(\lambda)\, H + |\lambda|^2\, \mathbb{1}_n$
- compute the QR factorization $M = Q\, R$
- set $H_2 = Q^T H\, Q$

The first step requires $O(n^3)$ flops and so it is still not practical...

The implicit double shift

The *implicit double shift* (or *Francis QR step*) computes this double shift in $O(n^2)$ flops:

- compute $M\,e_1 =$ multiple of e_1,
- determine Householder matrices $P_0, P_1, \ldots, P_{n-2}$ such that if $Z = P_0\,P_1 \cdots P_{n-2}$ then $Z^T\,H\,Z$ is Hessenberg and the first columns of Q and of Z are equal.

Why does it work?

The implicit QR theorem

If both

$$Q^T A\,Q = H \quad \text{and} \quad Z^T A\,Z = Q$$

are Hessenberg, H is unreduced, and Q and Z have the same first column, then

$$G = D^{-1} H\,D$$

with $D = \text{diag}(\pm 1, \ldots, \pm 1)$

- If both $Q^T H\,Q$ and $Z^T H\,Z$ are unreduced, then they are essentially equal
- Else the problem decouples into smaller unreduced subproblems

Implementing the implicit double shift

Set $s = \text{Re}(\lambda)$ and $t = |\lambda|^2$, write

$$M\,e_1 = \begin{bmatrix} x \\ y \\ z \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

with

$$\begin{aligned} x &= h_{1,1}^2 + h_{1,2}\,h_{2,1} - s\,h_{1,1} + t \\ y &= h_{2,1}\,(h_{1,1} + h_{2,2} - s) \\ z &= h_{2,1}\,h_{3,2} \end{aligned}$$

Implementing the implicit double shift

The computation of $M\,e_1$ and P_0 takes $O(1)$ flops

A similarity with P_0 only changes rows and columns 1, 2 and 3:

$$P_0^T H\,P_0 = \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Implementing the implicit double shift (cont.)

The mission of $P_1, \ldots, P_{n-2}$ is to restore this matrix to Hessenberg form:

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \end{bmatrix} \rightarrow \cdots$$

Hence

$$P_0 \quad \text{and} \quad Z = P_0 P_1 \cdots P_{n-2}$$

have the same first column

$\rightsquigarrow Z e_1 = Q e_1$ and so Z and Q coincide, up to signs

Implementing the implicit double shift (cont.)

To implement this strategy, we check when $h_{n-1,n-2}$ is small, in which case the eigenvalues of

$$\begin{bmatrix} h_{n-1,n-1} & h_{n-1,n} \\ h_{n,n-1} & h_{n,n} \end{bmatrix}$$

approximate the eigenvalues of A

In this case

$$s = h_{n-1,n-1} + h_{n,n} \quad \text{(trace)}$$
$$t = h_{n-1,n-1}\, h_{n,n} - h_{n-1,n}\, h_{n,n-1} \quad \text{(determinant)}$$

gives quadratic convergence

On the average, two QR iterations are needed per eigenvalue, and the overall process costs

$$O(n^3) \text{ flops}$$

Basic iterative methods

Let A be a nonsingular $n \times n$ matrix and b an n-vector

Given an *initial* n*-vector* x_0, these methods generate a sequence of other n-vectors

$$(x_l)_{l \geqslant 0}$$

hopefully converging to the solution $x = A^{-1} b$ and where for each $l \geqslant 0$, the vector x_{l+1} is easy to compute from x_l

Splittings

A *splitting* of A is a decomposition

$$A = M - K$$

with M nonsingular

$\rightsquigarrow$ iterative method:

$$A x = b \iff M x = K x + b \iff x = M^{-1} K x + M^{-1} b$$

We then set

$$x_{l+1} = R x_l + c \quad \text{for } l \geqslant 0 \qquad (10)$$

A convergence criterion

The *spectral radius*

$$\rho(R)$$

is the maximum absolute value of the eigenvalues of R

The iteration in (10) converges for every choice of initial vector x_0 if and only if

$$\rho(R) < 1$$

A convergence criterion (cont.)

Indeed, for every $\varepsilon > 0$ there is a vector norm $\|\cdot\|$ such that the associated operator norm verifies that

$$\|R\| \leqslant \rho(R) + \varepsilon$$

Hence if $\rho(R) < 1$ then we can choose ε sufficiently small so that the associated vector norm $\|\cdot\|$ verifies that $\|R\| < 1$

Then

$$\begin{array}{rcl} x & = & R\,x + c \\ - \quad x_{l+1} & = & R\,x_l + c \\ \hline x - x_{l+1} = R\,(x - x_l) & & \end{array}$$

and so in this case

$$\|x - x_{l+1}\| = \|R\,(x - x_l)\| \leqslant \|R\|\,\|x - x_l\| \leqslant \|R\|^{l+1}\,\|x - x_0\| \longrightarrow 0$$

when $l \to +\infty$

A convergence criterion (cont.)

Conversely, if $\rho(R) \geqslant 1$ then there is $x_0 \neq x$ such that $x - x_0$ is an eigenvector for an eigenvalue λ of absolute value $\geqslant 1$

Hence

$$x - x_{l+1} = R^{l+1}\,(x - x_0) = \lambda^{l+1}\,(x - x_{l+1})$$

which does not approach 0 when $k \to +\infty$

The speed of convergence

When $\rho(R) < 1$ the approximation of the l-th iteration is

$$-\log_b \|x - x_l\| \geqslant -l\,\log_b \rho(R) - \log_b \|x - x_0\|$$

$\rightsquigarrow$ the increase in precision at each step is *linear* with rate

$$-\log_b \rho(R)$$

Our goal is to fing a splitting $A = M - K$ verifying the conditions

- $R = M^{-1}K$ and $c = M^{-1}b$ are easy to compute
- $\rho(R)$ is small

The classical iterative methods

Suppose that no diagonal entry of A is zero and set

$$A = D - \widetilde{L} - \widetilde{U} = D\,(\mathbb{1}_n - L - U)$$

with

- D is diagonal
- $\widetilde{L}$ and L strictly lower triangular
- $\widetilde{U}$ and U are strictly upper triangular

Jacobi's method

Jabobi's method can be interpreted as going successively through the equations, changing at the j-th step of the $(l+1)$th iteration the variable x_j so that the j-th equation is satisfied:

for $j = 1, \ldots, n$

$$x_{l+1,j} \leftarrow \tfrac{1}{a_{j,j}} \left(b_j - \textstyle\sum_{k \neq j} a_{j,k}\, x_{l,k}\right)$$

Hence for all j we have that

$$a_{j,1}\, x_{l,1} + \cdots + a_{j,j-1}\, x_{l,j-1} + a_{j,j}\, x_{l+1,j} + a_{l,j+1}\, x_{l,j+1} + \cdots + a_{l,n}\, x_{l,n} = b_j$$

In matrix notation, $x_{l+1} = R_{\mathrm{J}}\, x_l + c_{\mathrm{J}}$ with

$$R_{\mathrm{J}} = D^{-1}(\widetilde{L} + \widetilde{U}) = L + U \quad \text{and} \quad c_{\mathrm{J}} = D^{-1} b$$

Gauss-Seidel method

The *Gauss-Seidel method* takes advantage of the previously computed coordinates $x_{l+1,k}$, $k = 1, \ldots, j-1$:

for $j = 1, \ldots, n$

$$x_{l+1,j} \leftarrow \tfrac{1}{a_{j,j}} \Big(b_j - \underbrace{\sum_{k<j} a_{j,k}\, x_{l+1,k}}_{\text{updated } x\text{'s}} - \underbrace{\sum_{k>j} a_{j,k}\, x_{l,k}}_{\text{older } x\text{'s}}\Big)$$

Hence for all j

$$a_{j,1}\, x_{l+1,1} + \cdots + a_{j,j-1}\, x_{l+1,j-1} + a_{j,j}\, x_{l+1,j} + a_{l+1,j+1}\, x_{l,j+1} + \cdots + a_{l,n}\, x_{l,n} = b_j$$

In matrix notation, $x_{l+1} = R_{\mathrm{GS}}\, x_l + c_{\mathrm{GS}}$ with

$$\begin{aligned} R_{\mathrm{GS}} &= (D - \widetilde{L})^{-1}\widetilde{U} = (\mathbb{1}_n - L)^{-1} U \\ c_{\mathrm{GS}} &= (D - \widetilde{L})^{-1} b = (\mathbb{1}_n - L)^{-1} D^{-1} b \end{aligned}$$

SOR

Successive overrelaxation for a parameter $\omega \in \mathbb{R}$ is a weighted average of the vectors x_{l+1} and x_l from the Gauss-Seidel method:

$$x_{l+1}^{\mathrm{SOR}(\omega)} = (1-\omega)\, x_l^{\mathrm{GS}} + \omega x_{l+1}^{\mathrm{GS}}$$

for $j = 1, \ldots, n$

$$x_{l+1,j} \leftarrow (1-\omega) x_{l,j} + \tfrac{\omega}{a_{j,j}} \left(b_j - \textstyle\sum_{k<j} a_{j,k}\, x_{l+1,k} - \sum_{k>j} a_{j,k}\, x_{l,k}\right)$$

In matrix notation, $x_{l+1} = R_{\mathrm{SOR}(\omega)}\, x_l + c_{\mathrm{SOR}(\omega)}\, b$ with

$$\begin{aligned} R_{\mathrm{SOR}(\omega)} &= (D - \omega\,\widetilde{L})^{-1}((1-\omega)\, D + \omega\,\widetilde{U}) \\ &= (\mathbb{1}_n - \omega\, L)^{-1}((1-\omega)\,\mathbb{1}_n + \omega\, U) \\ c_{\mathrm{SOR}(\omega)} &= \omega\,(D - \omega\,\widetilde{L}) = \omega\,(\mathbb{1}_n - \omega\, L)\, D^{-1} \end{aligned}$$

- $\omega = 1$: Gauss-Seidel method
- $\omega > 1$: *overrelaxation*
- $\omega < 1$: *underrelaxation*

Example

Consider the 2×2 matrix and the 2-vector

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The solution of $Ax = b$ is the 2-vector

$$x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

The decompositions $A = D - \widetilde{L} - \widetilde{U} = D\,(\mathbb{1}_2 - L - U)$ are

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ \frac{-1}{2} & 0 \end{bmatrix} - \begin{bmatrix} 0 & \frac{-1}{2} \\ 0 & 0 \end{bmatrix} \right).$$

Example (cont.)

Hence

$$R_{\mathrm{J}} = L + U = \begin{bmatrix} 0 & \frac{-1}{2} \\ \frac{-1}{2} & 0 \end{bmatrix} \quad \text{and} \quad c_{\mathrm{J}} = D^{-1} b = \begin{bmatrix} \frac{1}{2} \\ \frac{-1}{2} \end{bmatrix}$$

and so the corresponding Jacobi iteration writes as

$$\begin{bmatrix} x_{l+1,1} \\ x_{l+1,2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{2} \\ \frac{-1}{2} & 0 \end{bmatrix} \begin{bmatrix} x_{l,1} \\ x_{l,2} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ \frac{-1}{2} \end{bmatrix} = \begin{bmatrix} \frac{-x_{l,2}}{2} + \frac{1}{2} \\ \frac{x_{l,1}}{2} - \frac{1}{2} \end{bmatrix}$$

The characteristic polynomial of the Jacobi matrix is

$$\chi_{R_{\mathrm{J}}} = \det(R_{\mathrm{J}} - t\,\mathbb{1}_2) = \det \begin{bmatrix} -t & \frac{-1}{2} \\ \frac{-1}{2} & -t \end{bmatrix} = t^2 - \frac{1}{4}$$

Hence $\lambda(R_{\mathrm{J}}) = \{t \mid \chi_{R_{\mathrm{J}}} = 0\} = \left\{ \pm \frac{1}{2} \right\}$ and so the spectral radius is

$$\rho(R_{\mathrm{J}}) = \frac{1}{2}$$

$\rightsquigarrow$ the method converges to x for every choice of initial vector x_0

Example (cont.)

We also have that

$$R_{\mathrm{GS}} = (\mathbb{1}_2 - L)^{-1} U = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & \frac{-1}{2} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{2} \\ 0 & \frac{1}{4} \end{bmatrix}$$

$$c_{\mathrm{GS}} = (D - \widetilde{L})^{-1} b = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{-1}{4} & \frac{1}{2} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{-3}{4} \end{bmatrix}$$

and so the Gauss-Seidel iteration writes as

$$\begin{bmatrix} x_{l+1,1} \\ x_{l+1,2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{2} \\ 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} x_{l,1} \\ x_{l,2} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ \frac{-3}{4} \end{bmatrix} = \begin{bmatrix} \frac{-x_{l,2}}{2} + \frac{1}{2} \\ \frac{x_{l,2}}{4} - \frac{3}{4} \end{bmatrix}$$

Example (cont.)

We have that

$$\chi_{R_{\mathrm{GS}}} = \det(R_{\mathrm{GS}} - t\,\mathbb{1}_2) = \det \begin{bmatrix} -t & \frac{1}{2} \\ 0 & -t + \frac{1}{4} \end{bmatrix} = t^2 - \frac{1}{4}\,t$$

Hence $\lambda(R_{\mathrm{GS}}) = \left\{0, \frac{1}{4}\right\}$ and so $\rho(R_{\mathrm{GS}}) = \frac{1}{4}$

In this example

$$\rho(R_{\mathrm{GS}}) = \rho(R_{\mathrm{J}})^2$$

and so the Gauss-Seidel method converges with the *double of the speed* of the Jacobi method

Example (cont.)

For $\omega \in \mathbb{R}$

$$R_{\mathrm{SOR}(\omega)} = (\mathbb{1}_2 - \omega L)^{-1}((1-\omega)\,\mathbb{1}_2 + \omega U)$$
$$= \begin{bmatrix} 1 & 0 \\ \omega/2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1-\omega & -\omega/2 \\ 0 & 1-\omega \end{bmatrix} = \begin{bmatrix} 1-\omega & -\frac{\omega}{2} \\ \frac{\omega^2}{2} - \frac{\omega}{2} & \frac{\omega^2}{4} - \omega + 1 \end{bmatrix}$$

and

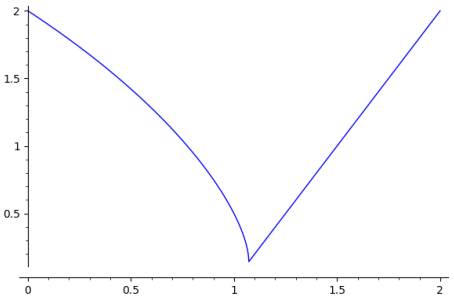
$$c_{\mathrm{SOR}(\omega)} = \omega\,(D - \omega\,\widetilde{L})^{-1} b = \omega \begin{bmatrix} 2 & 0 \\ \frac{\omega}{2} & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \frac{\omega}{2} \\ \frac{-\omega^2}{4} - \frac{\omega}{2} \end{bmatrix}$$

Hence

$$\chi_{R_{\mathrm{SOR}(\omega)}} = \det(R_{\mathrm{SOR}(\omega)} - t\,\mathbb{1}_2) = \det \begin{bmatrix} 1-\omega-t & -\frac{\omega}{2} \\ \frac{\omega^2}{2} - \frac{\omega}{2} & \frac{\omega^2}{4} - \omega + 1 - t \end{bmatrix}$$
$$= t^2 + \left(\frac{-\omega^2}{4} + 2\,\omega - 2\right) t + (\omega^2 - 2\,\omega + 1).$$

Example (cont.)

The spectral radius $R_{\mathrm{SOR}(\omega)}$ for $\omega \in [0, 2]$:



The optimal value is $\omega_{\mathrm{opt}} = 1.0717$ and we have that

$$\rho(R_{\mathrm{SOR}(1.0717)}) = 0.1535$$

Convergence of the basic iterative schemes

Consider a splitting $A = M - K$ with M nonsingular and the associated iterative scheme

$$x_{l+1} \leftarrow R\,x_l \quad \text{with } R = M^{-1}K \text{ and } c = M^{-1}b$$

If it converges, its limit $x_\infty = \lim_{l\to\infty} x_l$ satisfies

$$x_\infty = R\,x_\infty + c = M^{-1}K\,x_\infty + M^{-1}b$$

which implies that x_∞ is the (unique) solution of the equation $A\,x = b$

Convergence of the basic iterative schemes (cont.)

The convergence of the iteration for an arbitrary initial vector x_0 is equivalent to the condition that

$$\rho(R) < 1$$

In this case, the rate of convergence of the method is linear with coefficient

$$-\log_b \rho(R)$$

In practice, the spectral radius is difficult to compute: need sufficient conditions that are easy to apply!

Example

The *Richardson iteration* for a parameter $\omega \in \mathbb{R}$ is

$$x_{l+1} \leftarrow R_\omega\, x_l + \omega\, b$$

with $R_\omega = \mathbb{1}_n - \omega\, A$. It corresponds to the splitting

$$A = \omega^{-1}\, \mathbb{1}_n - (\omega^{-1}\, \mathbb{1}_n - A)$$

Suppose that the eigenvalues of A are real and ordered as

$$\lambda_1 \geqslant \cdots \geqslant \lambda_n$$

Then the eigenvalues of R_ω are also real and satisfy that

$$\mu_1 = 1 - \omega\, \lambda_1 \leqslant \cdots \leqslant \mu_n = 1 - \omega\, \lambda_n$$

If $\lambda_n \leqslant 0$ then $\mu_n \geqslant 1$ and so $\rho(R_\omega) \geqslant 1$ and the iteration does not converge

Example (cont.)

Else suppose that $\lambda_n \geqslant 0$. Then the iteration converges if and only if

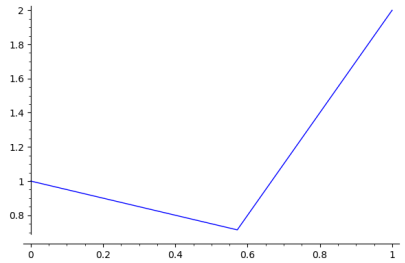
$$-1 \leqslant 1 - \omega\, \lambda_1 \quad \text{and} \quad 1 - \omega\, \lambda_n \leqslant 1$$

that is, if and only if

$$0 < \omega < \frac{2}{\lambda_1}$$

The spectral radius is the piecewise affine function

$$\rho(R_\omega) = \max(|1 - \omega\, \lambda_n|, |1 - \omega\, \lambda_1|)$$



Example (cont.)

The optimal value is given by $-1 + \lambda_1\, \omega_{\text{opt}} = 1 - \lambda_n \omega_{\text{opt}}$ and so

$$\omega_{\text{opt}} = \frac{2}{\lambda_1 + \lambda_n}$$

The corresponding spectral radius is

$$\rho(\omega_{\text{opt}}) = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}$$

- If A has both very large and very small eigenvalues, then the iteration will be slow
- The determination of the optimal parameter ω_{opt} requires knowledge of these eigenvalues

Diagonally dominant matrices

The matrix A is *diagonally dominant* if for all j

$$|a_{j,j}| > \sum_{i \neq j} |a_{i,j}|$$

If A is diagonally dominant then both the Jacobi and the Gauss-Seidel iterations will converge for every choice of initial vector x_0

Diagonally dominant matrices (cont.)

For instance, let

$$R_{\mathrm{J}} = D^{-1}(\widetilde{L} + \widetilde{U})$$

be the iteration matrix of Jacobi's method

For an eigenvalue $\lambda \in \lambda(R_{\mathrm{J}})$ let x be a corresponding eigenvector and m the index of its largest component, normalized so that

$$x_m = 1 \quad \text{and} \quad |x_j| \leqslant 1 \text{ for all } j$$

Then from $R_{\mathrm{J}}\, x = \lambda\, x$ we deduce that

$$\lambda\, x_m = -\sum_{j \neq m} \frac{a_{m,j}}{a_{m,m}}\, x_j$$

and so

$$|\lambda| \leqslant \sum_{j \neq m} \frac{|a_{m,j}|}{|a_{m,m}|}\, |x_j| < 1$$

which proves the result for the Jacobi iteration

Convergence of SOR

When A is symmetric and positive definite,

$$\mathrm{SOR}(\omega)$$

converges for every $0 < \omega < 2$

In particular, in this situation the Gauss-Seidel method ($\omega = 1$) also converges

Convergence of SOR

For successive overrelaxation, the condition $0 < \omega < 2$ is necessary for the convergence of the method

Indeed

$$R_{\mathrm{SOR}(\omega)} = (\mathbb{1}_n - \omega\, L)^{-1}((1-\omega)\, \mathbb{1}_n + \omega\, U)$$

and so $\chi_{R_{\mathrm{SOR}(\omega)}}(0) = \det(R_{\mathrm{SOR}(\omega)}) = (1-\omega)^n$

On the other hand

$$|\chi_{R_{\mathrm{SOR}(\omega)}}(0)| = \prod_{\lambda} |\lambda| \geqslant \rho(R_{\mathrm{SOR}(\omega)})^n$$

the product being over the eigenvalues of $R_{\mathrm{SOR}(\omega)}$, and so

$$\rho(R_{\mathrm{SOR}(\omega)}) \geqslant |1-\omega|$$