Lecture Notes to Accompany

Scientific Computing

An Introductory Survey

Second Edition

by Michael T. Heath

Chapter 3

Linear Least Squares

Method of Least Squares

Measurement errors inevitable in observational and experimental sciences

Errors smoothed out by averaging over many cases, i.e., taking more measurements than strictly necessary to determine parameters of system

Resulting system *overdetermined*, so usually no exact solution

Project higher dimensional data into lower dimensional space to suppress irrelevant detail

Projection most conveniently accomplished by method of *least squares*

Linear Least Squares

For linear problems, obtain *overdetermined* linear system $\boldsymbol{Ax} = \boldsymbol{b}$, with $m \times n$ matrix $\boldsymbol{A}$, $m > n$

Better written $\boldsymbol{Ax} \cong \boldsymbol{b}$, since equality usually not exactly satisfiable when $m > n$

Least squares solution $\boldsymbol{x}$ minimizes squared Euclidean norm of residual vector $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{Ax}$,

$$\min_{\boldsymbol{x}} \|\boldsymbol{r}\|_2^2 = \min_{\boldsymbol{x}} \|\boldsymbol{b} - \boldsymbol{Ax}\|_2^2$$

Example: Data Fitting

Given m data points (t_i, y_i), find n-vector $\boldsymbol{x}$ of parameters that gives "best fit" to model function $f(t, \boldsymbol{x})$:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{m} (y_i - f(t_i, \boldsymbol{x}))^2$$

Problem *linear* if function f linear in components of $\boldsymbol{x}$:

$$f(t, \boldsymbol{x}) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_n\phi_n(t),$$

where functions ϕ_j depend only on t

Written in matrix form as $\boldsymbol{A}\boldsymbol{x} \cong \boldsymbol{b}$, with $a_{ij} = \phi_j(t_i)$ and $b_i = y_i$

Example: Data Fitting

Polynomial fitting,

$$f(t, \boldsymbol{x}) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1},$$

is linear, since polynomial linear in coefficients, though nonlinear in independent variable t

Fitting sum of exponentials, with

$$f(t, \boldsymbol{x}) = x_1 e^{x_2 t} + \cdots + x_{n-1} e^{x_n t},$$

is nonlinear problem

For now, consider only linear least squares problems

Example: Data Fitting

Fitting quadratic polynomial to five data points gives linear least squares problem

$$\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ 1 & t_4 & t_4^2 \\ 1 & t_5 & t_5^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cong \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \boldsymbol{b}$$

Matrix with columns (or rows) successive powers of independent variable called *Vandermonde* matrix

Example: Data Fitting

For data

t	-1.0	-0.5	0.0	0.5	1.0
y	1.0	0.5	0.0	0.5	2.0

overdetermined 5×3 linear system is

$$\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cong \begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ 2.0 \end{bmatrix} = \boldsymbol{b}$$

Solution, which we will see later how to compute, is

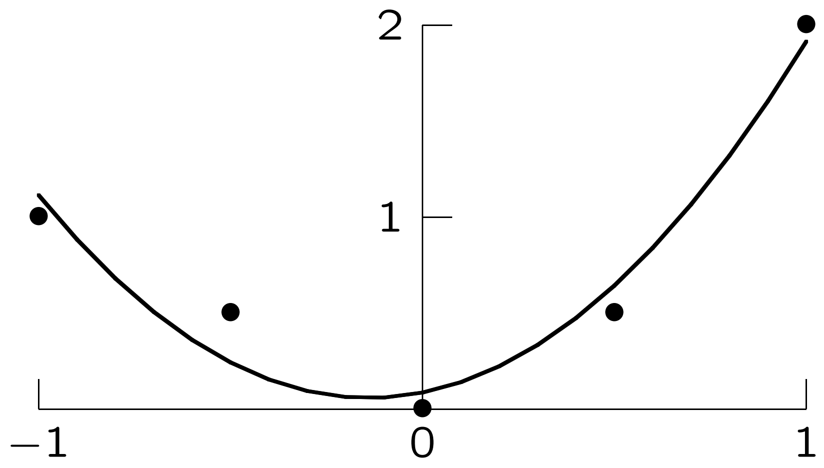
$$\boldsymbol{x} = [\,0.086 \quad 0.40 \quad 1.4\,]^T,$$

so approximating polynomial is

$$p(t) = 0.086 + 0.4t + 1.4t^2$$

Example: Data Fitting

Resulting curve and original data points shown in graph:



Existence and Uniqueness

Linear least squares problem $\boldsymbol{Ax} \cong \boldsymbol{b}$ *always* has solution

Solution unique if, and only if, columns of $\boldsymbol{A}$ linearly independent, i.e., rank$(\boldsymbol{A}) = n$, where $\boldsymbol{A}$ is $m \times n$

If rank$(\boldsymbol{A}) < n$, then $\boldsymbol{A}$ is *rank-deficient*, and solution of linear least squares problem is not unique

For now, assume $\boldsymbol{A}$ has full column rank n

Normal Equations

Least squares minimizes squared Euclidean norm

$$\|\boldsymbol{r}\|_2^2 = \boldsymbol{r}^T\boldsymbol{r}$$

of residual vector

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$$

To minimize

$$\begin{aligned} \|\boldsymbol{r}\|_2^2 &= \boldsymbol{r}^T\boldsymbol{r} = (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \\ &= \boldsymbol{b}^T\boldsymbol{b} - 2\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b} + \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}, \end{aligned}$$

take derivative with respect to $\boldsymbol{x}$ and set to $\boldsymbol{o}$,

$$2\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{o},$$

which reduces to $n \times n$ linear system

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$$

known as system of *normal equations*

Orthogonality

Vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are *orthogonal* if their inner product is zero, $\boldsymbol{v}_1^T \boldsymbol{v}_2 = 0$

Space spanned by columns of $m \times n$ matrix $\boldsymbol{A}$, $\text{span}(\boldsymbol{A}) = \{\boldsymbol{A}\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\}$, is of dimension at most n

If $m > n$, $\boldsymbol{b}$ generally does not lie in $\text{span}(\boldsymbol{A})$, so no exact solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$

Vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ in $\text{span}(\boldsymbol{A})$ closest to $\boldsymbol{b}$ in 2-norm occurs when residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ orthogonal to $\text{span}(\boldsymbol{A})$

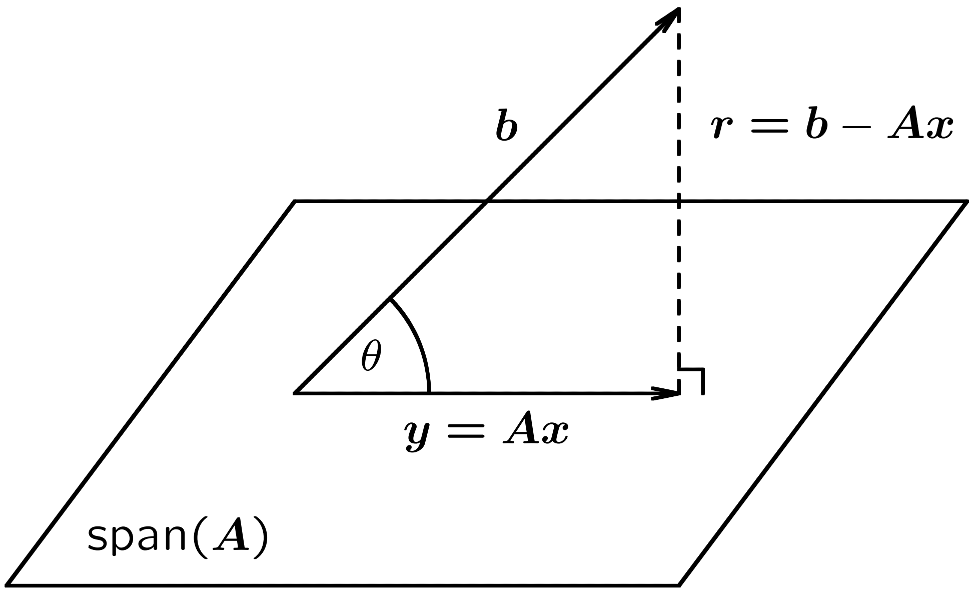
Thus,

$$\boldsymbol{o} = \boldsymbol{A}^T \boldsymbol{r} = \boldsymbol{A}^T (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}),$$

or

$$\boldsymbol{A}^T \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T \boldsymbol{b}$$

Orthogonality, continued



Orthogonal Projectors

Matrix $\boldsymbol{P}$ is *orthogonal projector* if *idempotent* ($\boldsymbol{P}^2 = \boldsymbol{P}$) and *symmetric* ($\boldsymbol{P}^T = \boldsymbol{P}$)

Orthogonal projector onto orthogonal complement span$(\boldsymbol{P})^\perp$ given by $\boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{P}$

For any vector $\boldsymbol{v}$,

$$\boldsymbol{v} = (\boldsymbol{P} + (\boldsymbol{I} - \boldsymbol{P}))\,\boldsymbol{v} = \boldsymbol{P}\boldsymbol{v} + \boldsymbol{P}_\perp \boldsymbol{v}$$

For least squares problem $\boldsymbol{A}\boldsymbol{x} \cong \boldsymbol{b}$, if rank$(\boldsymbol{A}) = n$, then

$$\boldsymbol{P} = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$$

is orthogonal projector onto span$(\boldsymbol{A})$, and

$$\boldsymbol{b} = \boldsymbol{P}\boldsymbol{b} + \boldsymbol{P}_\perp \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} + (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) = \boldsymbol{y} + \boldsymbol{r}$$

Pseudoinverse and Condition Number

Nonsquare $m \times n$ matrix $\boldsymbol{A}$ has no inverse in usual sense

If rank$(\boldsymbol{A}) = n$, *pseudoinverse* defined by

$$\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$$

and

$$\text{cond}(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \cdot \|\boldsymbol{A}^+\|_2$$

By convention, cond$(\boldsymbol{A}) = \infty$ if rank$(\boldsymbol{A}) < n$

Just as condition number of square matrix measures closeness to singularity, condition number of rectangular matrix measures closeness to rank deficiency

Least squares solution of $\boldsymbol{A}\boldsymbol{x} \cong \boldsymbol{b}$ is given by $\boldsymbol{x} = \boldsymbol{A}^+\boldsymbol{b}$

Sensitivity and Conditioning

Sensitivity of least squares solution to $\boldsymbol{Ax} \cong \boldsymbol{b}$ depends on $\boldsymbol{b}$ as well as $\boldsymbol{A}$

Define angle θ between $\boldsymbol{b}$ and $\boldsymbol{y} = \boldsymbol{Ax}$ by

$$\cos(\theta) = \frac{\|\boldsymbol{y}\|_2}{\|\boldsymbol{b}\|_2} = \frac{\|\boldsymbol{Ax}\|_2}{\|\boldsymbol{b}\|_2}$$

(see previous drawing)

Bound on perturbation $\Delta\boldsymbol{x}$ in solution $\boldsymbol{x}$ due to perturbation $\Delta\boldsymbol{b}$ in $\boldsymbol{b}$ given by

$$\frac{\|\Delta\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \leq \text{cond}(\boldsymbol{A}) \, \frac{1}{\cos(\theta)} \, \frac{\|\Delta\boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2}$$

Sensitivity and Conditioning, cont.

Similarly, for perturbation $\boldsymbol{E}$ in matrix $\boldsymbol{A}$,

$$\frac{\|\Delta \boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \lessapprox \Big([\mathrm{cond}(\boldsymbol{A})]^2 \tan(\theta) + \mathrm{cond}(\boldsymbol{A})\Big) \frac{\|\boldsymbol{E}\|_2}{\|\boldsymbol{A}\|_2}$$

Condition number of least squares solution about cond($\boldsymbol{A}$) if residual small, but can be squared or arbitrarily worse for large residual

Normal Equations Method

If $m \times n$ matrix $\boldsymbol{A}$ has rank n, then symmetric $n \times n$ matrix $\boldsymbol{A}^T\boldsymbol{A}$ is positive definite, so its Cholesky factorization

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T,$$

can be used to obtain solution $\boldsymbol{x}$ to system of normal equations

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b},$$

which has same solution as linear least squares problem $\boldsymbol{A}\boldsymbol{x} \cong \boldsymbol{b}$

Normal equations method involves transformations

rectangular $\longrightarrow$ square $\longrightarrow$ triangular

Example: Normal Equations Method

For polynomial data-fitting example given previously, normal equations method gives

$$A^T A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1.0 & -0.5 & 0.0 & 0.5 & 1.0 \\ 1.0 & 0.25 & 0.0 & 0.25 & 1.0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \end{bmatrix} = \begin{bmatrix} 5.0 & 0.0 & 2.5 \\ 0.0 & 2.5 & 0.0 \\ 2.5 & 0.0 & 2.125 \end{bmatrix},$$

$$A^T b =$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1.0 & -0.5 & 0.0 & 0.5 & 1.0 \\ 1.0 & 0.25 & 0.0 & 0.25 & 1.0 \end{bmatrix} \begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ 2.0 \end{bmatrix} = \begin{bmatrix} 4.0 \\ 1.0 \\ 3.25 \end{bmatrix}$$

Example Continued

Cholesky factorization of symmetric positive definite matrix gives

$$A^T A = \begin{bmatrix} 5.0 & 0.0 & 2.5 \\ 0.0 & 2.5 & 0.0 \\ 2.5 & 0.0 & 2.125 \end{bmatrix} =$$

$$\begin{bmatrix} 2.236 & 0 & 0 \\ 0 & 1.581 & 0 \\ 1.118 & 0 & 0.935 \end{bmatrix} \begin{bmatrix} 2.236 & 0 & 1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & 0.935 \end{bmatrix}$$

$$= LL^T$$

Example Continued

Solving lower triangular system $\boldsymbol{Lz} = \boldsymbol{A}^T\boldsymbol{b}$ by forward-substitution gives

$$\boldsymbol{z} = \begin{bmatrix} 1.789 \\ 0.632 \\ 1.336 \end{bmatrix}$$

Solving upper triangular system $\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{z}$ by back-substitution gives least squares solution

$$\boldsymbol{x} = \begin{bmatrix} 0.086 \\ 0.400 \\ 1.429 \end{bmatrix}$$

Shortcomings of Normal Equations

Information can be lost in forming $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{A}^T\boldsymbol{b}$

For example, take

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

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

Then in floating-point arithmetic

$$\boldsymbol{A}^T\boldsymbol{A} = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

which is singular

Sensitivity of solution also worsened, since

$$\text{cond}(\boldsymbol{A}^T\boldsymbol{A}) = [\text{cond}(\boldsymbol{A})]^2$$

Augmented System Method

Definition of residual and orthogonality requirement give $(m+n) \times (m+n)$ augmented system

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{o} \end{bmatrix}$$

System not positive definite, larger than original, and requires storing two copies of $\boldsymbol{A}$

But allows greater freedom in choosing pivots in computing $\boldsymbol{LDL}^T$ or $\boldsymbol{LU}$ factorization

Augmented System Method, continued

Introducing scaling parameter α gives system

$$\begin{bmatrix} \alpha \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^T & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}/\alpha \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{o} \end{bmatrix},$$

which allows control over relative weights of two subsystems in choosing pivots

Reasonable rule of thumb

$$\alpha = \max_{i,j} |a_{ij}|/1000$$

Augmented system sometimes useful, but far from ideal in work and storage required

Orthogonal Transformations

Seek alternative method that avoids numerical difficulties of normal equations

Need numerically robust transformation that produces easier problem

What kind of transformation leaves least squares solution unchanged?

Square matrix $\boldsymbol{Q}$ is *orthogonal* if $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$

Preserves Euclidean norm, since

$$\|\boldsymbol{Qv}\|_2^2 = (\boldsymbol{Qv})^T\boldsymbol{Qv} = \boldsymbol{v}^T\boldsymbol{Q}^T\boldsymbol{Qv} = \boldsymbol{v}^T\boldsymbol{v} = \|\boldsymbol{v}\|_2^2$$

Multiplying both sides of least squares problem by orthogonal matrix does not change solution

Triangular Least Squares Problems

As with square linear systems, suitable target in simplifying least squares problems is triangular form

Upper triangular overdetermined ($m > n$) least squares problem has form

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x} \cong \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix},$$

with $\boldsymbol{R}$ $n \times n$ upper triangular and $\boldsymbol{b}$ partitioned similarly

Residual is

$$\|\boldsymbol{r}\|_2^2 = \|\boldsymbol{b}_1 - \boldsymbol{R}\boldsymbol{x}\|_2^2 + \|\boldsymbol{b}_2\|_2^2$$

Triangular Least Squares Problems, cont.

Have no control over second term, $\|\boldsymbol{b}_2\|_2^2$, but first term becomes zero if $\boldsymbol{x}$ satisfies triangular system

$$\boldsymbol{R}\boldsymbol{x} = \boldsymbol{b}_1,$$

which can be solved by back-substitution

Resulting $\boldsymbol{x}$ is least squares solution, and minimum sum of squares is

$$\|\boldsymbol{r}\|_2^2 = \|\boldsymbol{b}_2\|_2^2$$

So strategy is to transform general least squares problem to triangular form using orthogonal transformation

QR Factorization

Given $m \times n$ matrix $\boldsymbol{A}$, with $m > n$, we seek $m \times m$ orthogonal matrix $\boldsymbol{Q}$ such that

$$\boldsymbol{A} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix},$$

with $\boldsymbol{R}$ $n \times n$ and upper triangular

Linear least squares problem $\boldsymbol{Ax} \cong \boldsymbol{b}$ transformed into triangular least squares problem

$$\boldsymbol{Q}^T \boldsymbol{Ax} = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x} \cong \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{bmatrix} = \boldsymbol{Q}^T \boldsymbol{b},$$

which has same solution, since $\|\boldsymbol{r}\|_2^2 =$

$$\begin{aligned} \|\boldsymbol{b} - \boldsymbol{Ax}\|_2^2 &= \|\boldsymbol{b} - \boldsymbol{Q} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x}\|_2^2 = \|\boldsymbol{Q}^T \boldsymbol{b} - \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x}\|_2^2 \\ &= \|\boldsymbol{c}_1 - \boldsymbol{Rx}\|_2^2 + \|\boldsymbol{c}_2\|_2^2 \end{aligned}$$

because orthogonal transformation preserves Euclidean norm

Orthogonal Bases

Partition $m \times m$ orthogonal matrix $\boldsymbol{Q} = [\boldsymbol{Q}_1\ \boldsymbol{Q}_2]$, with $\boldsymbol{Q}_1$ $m \times n$

Then

$$\boldsymbol{A} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} = [\boldsymbol{Q}_1\ \boldsymbol{Q}_2] \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} = \boldsymbol{Q}_1 \boldsymbol{R}$$

is *reduced* QR factorization of $\boldsymbol{A}$

Columns of $\boldsymbol{Q}_1$ are *orthonormal basis* for span($\boldsymbol{A}$), and columns of $\boldsymbol{Q}_2$ are orthonormal basis for span$(\boldsymbol{A})^{\perp}$

$\boldsymbol{Q}_1 \boldsymbol{Q}_1^T$ is orthogonal projector onto span($\boldsymbol{A}$)

Solution to least squares problem $\boldsymbol{A}\boldsymbol{x} \cong \boldsymbol{b}$ given by solution to square system

$$\boldsymbol{Q}_1^T \boldsymbol{A}\boldsymbol{x} = \boldsymbol{R}\boldsymbol{x} = \boldsymbol{c}_1 = \boldsymbol{Q}_1^T \boldsymbol{b}$$

QR Factorization

To compute QR factorization of $m \times n$ matrix $\boldsymbol{A}$, with $m > n$, annihilate subdiagonal entries of successive columns of $\boldsymbol{A}$, eventually reaching upper triangular form

Similar to LU factorization by Gaussian elimination, but uses orthogonal transformations instead of elementary elimination matrices

Possible methods include

- Householder transformations
- Givens rotations
- Gram-Schmidt orthogonalization

Householder Transformations

Householder transformation has form

$$\boldsymbol{H} = \boldsymbol{I} - 2\frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}$$

for nonzero vector $\boldsymbol{v}$

$\boldsymbol{H} = \boldsymbol{H}^T = \boldsymbol{H}^{-1}$, so $\boldsymbol{H}$ is orthogonal and symmetric

Given vector $\boldsymbol{a}$, choose $\boldsymbol{v}$ so that

$$\boldsymbol{H}\boldsymbol{a} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \boldsymbol{e}_1$$

Substituting into formula for $\boldsymbol{H}$, can take

$$\boldsymbol{v} = \boldsymbol{a} - \alpha \boldsymbol{e}_1$$

and $\alpha = \pm\|\boldsymbol{a}\|_2$, with sign chosen to avoid cancellation

Example: Householder Transformation

Let $\boldsymbol{a} = [\,2 \quad 1 \quad 2\,]^T$

By foregoing recipe,

$$\boldsymbol{v} = \boldsymbol{a} - \alpha \boldsymbol{e}_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix},$$

where $\alpha = \pm\|\boldsymbol{a}\|_2 = \pm 3$

Since a_1 positive, choosing negative sign for α avoids cancellation

Thus, $\boldsymbol{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$

To confirm that transformation works,

$$\boldsymbol{H}\boldsymbol{a} = \boldsymbol{a} - 2\frac{\boldsymbol{v}^T\boldsymbol{a}}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 2\,\frac{15}{30}\begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$$

Householder QR Factorization

To compute QR factorization of $\boldsymbol{A}$, use Householder transformations to annihilate subdiagonal entries of each successive column

Each Householder transformation applied to entire matrix, but does not affect prior columns, so zeros preserved

In applying Householder transformation $\boldsymbol{H}$ to arbitrary vector $\boldsymbol{u}$,

$$\boldsymbol{H}\boldsymbol{u} = \left(\boldsymbol{I} - 2\frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}\right)\boldsymbol{u} = \boldsymbol{u} - \left(2\frac{\boldsymbol{v}^T\boldsymbol{u}}{\boldsymbol{v}^T\boldsymbol{v}}\right)\boldsymbol{v},$$

which is much cheaper than general matrix-vector multiplication and requires only vector $\boldsymbol{v}$, not full matrix $\boldsymbol{H}$

Householder QR Factorization, cont.

Process just described produces factorization

$$\boldsymbol{H}_n \cdots \boldsymbol{H}_1 \boldsymbol{A} = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix}$$

with $\boldsymbol{R}$ $n \times n$ and upper triangular

If $\boldsymbol{Q} = \boldsymbol{H}_1 \cdots \boldsymbol{H}_n$, then

$$\boldsymbol{A} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix}$$

To preserve solution of linear least squares problem, right-hand-side $\boldsymbol{b}$ transformed by same sequence of Householder transformations

Then solve triangular least squares problem

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x} \cong \boldsymbol{Q}^T \boldsymbol{b}$$

for solution $\boldsymbol{x}$ of original least squares problem

Householder QR Factorization, cont.

For solving linear least squares problem, product $\boldsymbol{Q}$ of Householder transformations need not be formed explicitly

$\boldsymbol{R}$ can be stored in upper triangle of array initially containing $\boldsymbol{A}$

Householder vectors $\boldsymbol{v}$ can be stored in (now zero) lower triangular portion of $\boldsymbol{A}$ (almost)

Householder transformations most easily applied in this form anyway

Example: Householder QR Factorization

For polynomial data-fitting example given previously, with

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ 2.0 \end{bmatrix},$$

Householder vector $\boldsymbol{v}_1$ for annihilating subdiagonal entries of first column of $\boldsymbol{A}$ is

$$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -2.236 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3.236 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Example Continued

Applying resulting Householder transformation $\boldsymbol{H}_1$ yields transformed matrix and right-hand side

$$\boldsymbol{H}_1\boldsymbol{A} = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & -0.191 & -0.405 \\ 0 & 0.309 & -0.655 \\ 0 & 0.809 & -0.405 \\ 0 & 1.309 & 0.345 \end{bmatrix}$$

$$\boldsymbol{H}_1\boldsymbol{b} = \begin{bmatrix} -1.789 \\ -0.362 \\ -0.862 \\ -0.362 \\ 1.138 \end{bmatrix}$$

Example Continued

Householder vector $\boldsymbol{v}_2$ for annihilating subdiagonal entries of second column of $\boldsymbol{H}_1\boldsymbol{A}$ is

$$\boldsymbol{v}_2 = \begin{bmatrix} 0 \\ -0.191 \\ 0.309 \\ 0.809 \\ 1.309 \end{bmatrix} - \begin{bmatrix} 0 \\ 1.581 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.772 \\ 0.309 \\ 0.809 \\ 1.309 \end{bmatrix}$$

Applying resulting Householder transformation $\boldsymbol{H}_2$ yields

$$\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A} = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & -0.725 \\ 0 & 0 & -0.589 \\ 0 & 0 & 0.047 \end{bmatrix}$$

Example Continued

$$\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{b} = \begin{bmatrix} -1.789 \\ 0.632 \\ -1.035 \\ -0.816 \\ 0.404 \end{bmatrix}$$

Householder vector $\boldsymbol{v}_3$ for annihilating subdiagonal entries of third column of $\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}$ is

$$\boldsymbol{v}_3 = \begin{bmatrix} 0 \\ 0 \\ -0.725 \\ -0.589 \\ 0.047 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0.935 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1.660 \\ -0.589 \\ 0.047 \end{bmatrix}$$

Example Continued

Applying resulting Householder transformation $\boldsymbol{H}_3$ yields

$$\boldsymbol{H}_3\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A} = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & 0.935 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{H}_3\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{b} = \begin{bmatrix} -1.789 \\ 0.632 \\ 1.336 \\ 0.026 \\ 0.337 \end{bmatrix}$$

Now solve upper triangular system $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{c}_1$ by back-substitution to obtain

$$\boldsymbol{x} = [\,0.086 \quad 0.400 \quad 1.429\,]^T$$

Givens Rotations

Givens rotations introduce zeros one at a time

Given vector $[a_1 \quad a_2]^T$, choose scalars c and s so that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix},$$

with $c^2 + s^2 = 1$, or equivalently, $\alpha = \sqrt{a_1^2 + a_2^2}$

Previous equation can be rewritten

$$\begin{bmatrix} a_1 & a_2 \\ a_2 & -a_1 \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

Gaussian elimination yields triangular system

$$\begin{bmatrix} a_1 & a_2 \\ 0 & -a_1 - a_2^2/a_1 \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \alpha \\ -\alpha a_2/a_1 \end{bmatrix}$$

Givens Rotations, continued

Back-substitution then gives

$$s = \frac{\alpha a_2}{a_1^2 + a_2^2}, \qquad c = \frac{\alpha a_1}{a_1^2 + a_2^2}$$

Finally, $c^2 + s^2 = 1$, or $\alpha = \sqrt{a_1^2 + a_2^2}$, implies

$$c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}, \qquad s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}$$

Example: Givens Rotation

Let $\boldsymbol{a} = [\,4 \quad 3\,]^T$

Computing cosine and sine,

$$c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}} = \frac{4}{5} = 0.8$$

$$s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}} = \frac{3}{5} = 0.6$$

Rotation given by

$$\boldsymbol{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{bmatrix}$$

To confirm that rotation works,

$$\boldsymbol{G}\boldsymbol{a} = \begin{bmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

Givens QR Factorization

To annihilate selected component of vector in n dimensions, rotate target component with another component

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} a_1 \\ \alpha \\ a_3 \\ 0 \\ a_5 \end{bmatrix}$$

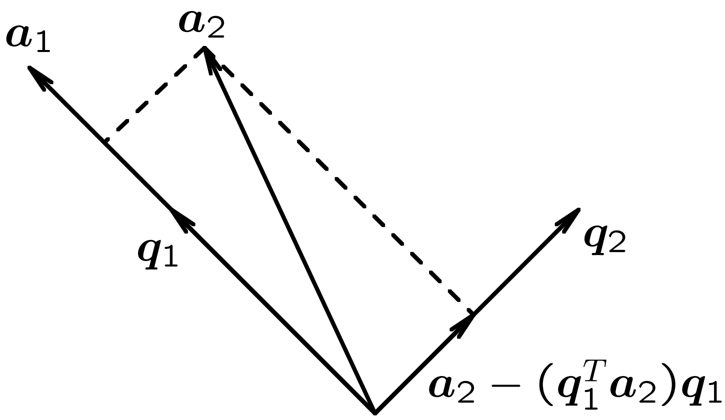
Reduce matrix to upper triangular form using sequence of Givens rotations

Each rotation orthogonal, so their product is orthogonal, producing QR factorization

Straightforward implementation of Givens method requires about 50% more work than Householder method, and also requires more storage, since each rotation requires two numbers, c and s, to define it

Gram-Schmidt Orthogonalization

Given vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, can determine orthonormal vectors $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ with same span by orthogonalizing one vector against other:



for $k = 1$ **to** n
    $\boldsymbol{q}_k = \boldsymbol{a}_k$
    for $j = 1$ **to** $k - 1$
        $r_{jk} = \boldsymbol{q}_j^T \boldsymbol{a}_k$
        $\boldsymbol{q}_k = \boldsymbol{q}_k - r_{jk}\boldsymbol{q}_j$
    end
    $r_{kk} = \|\boldsymbol{q}_k\|_2$
    $\boldsymbol{q}_k = \boldsymbol{q}_k / r_{kk}$
end

Modified Gram-Schmidt

Classical Gram-Schmidt procedure often suffers loss of orthogonality in finite-precision

Also, separate storage is required for $\boldsymbol{A}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$, since original $\boldsymbol{a}_k$ needed in inner loop, so $\boldsymbol{q}_k$ cannot overwrite columns of $\boldsymbol{A}$

Both deficiencies improved by *modified* Gram-Schmidt procedure, with each vector orthogonalized in turn against all *subsequent* vectors so $\boldsymbol{q}_k$ can overwrite $\boldsymbol{a}_k$:

for $k = 1$ **to** n
  $r_{kk} = \|\boldsymbol{a}_k\|_2$
  $\boldsymbol{q}_k = \boldsymbol{a}_k / r_{kk}$
  for $j = k+1$ **to** n
    $r_{kj} = \boldsymbol{q}_k^T \boldsymbol{a}_j$
    $\boldsymbol{a}_j = \boldsymbol{a}_j - r_{kj}\boldsymbol{q}_k$
  end
end

Rank Deficiency

If rank($\boldsymbol{A}$) $< n$, then QR factorization still exists, but yields singular upper triangular factor $\boldsymbol{R}$, and multiple vectors $\boldsymbol{x}$ give minimum residual norm

Common practice selects minimum residual solution $\boldsymbol{x}$ having smallest norm

Can be computed by QR factorization with column pivoting or by singular value decomposition (SVD)

Rank of matrix often not clear cut in practice, so relative tolerance used to determine rank

Example: Near Rank Deficiency

Consider 3×2 matrix

$$\boldsymbol{A} = \begin{bmatrix} 0.641 & 0.242 \\ 0.321 & 0.121 \\ 0.962 & 0.363 \end{bmatrix}$$

Computing QR factorization,

$$\boldsymbol{R} = \begin{bmatrix} 1.1997 & 0.4527 \\ 0 & 0.0002 \end{bmatrix}$$

$\boldsymbol{R}$ extremely close to singular (exactly singular to 3-digit accuracy of problem statement)

If $\boldsymbol{R}$ used to solve linear least squares problem, result is highly sensitive to perturbations in right-hand side

For practical purposes, rank($\boldsymbol{A}$) $= 1$ rather than 2, because columns nearly linearly dependent

QR with Column Pivoting

Instead of processing columns in natural order, select for reduction at each stage column of remaining unreduced submatrix having maximum Euclidean norm

If rank($\boldsymbol{A}$) $= k < n$, then after k steps, norms of remaining unreduced columns will be zero (or "negligible" in finite-precision arithmetic) below row k

Yields orthogonal factorization of form

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{S} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix},$$

with $\boldsymbol{R}$ $k \times k$, upper triangular, and nonsingular, and permutation matrix $\boldsymbol{P}$ performing column interchanges

QR with Column Pivoting, cont.

Basic solution to least squares problem $\boldsymbol{Ax} \cong \boldsymbol{b}$ can now be computed by solving triangular system $\boldsymbol{Rz} = \boldsymbol{c}_1$, where $\boldsymbol{c}_1$ contains first k components of $\boldsymbol{Q}^T\boldsymbol{b}$, and then taking

$$\boldsymbol{x} = \boldsymbol{P} \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{o} \end{bmatrix}$$

Minimum-norm solution can be computed, if desired, at expense of additional processing to annihilate $\boldsymbol{S}$

rank($\boldsymbol{A}$) usually unknown, so rank determined by monitoring norms of remaining unreduced columns and terminating factorization when maximum value falls below tolerance

Singular Value Decomposition

Singular value decomposition (SVD) of $m \times n$ matrix $\boldsymbol{A}$ has form

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

where $\boldsymbol{U}$ is $m \times m$ orthogonal matrix, $\boldsymbol{V}$ is $n \times n$ orthogonal matrix, and $\boldsymbol{\Sigma}$ is $m \times n$ diagonal matrix, with

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

Diagonal entries σ_i, called *singular values* of $\boldsymbol{A}$, usually ordered so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

Columns $\boldsymbol{u}_i$ of $\boldsymbol{U}$ and $\boldsymbol{v}_i$ of $\boldsymbol{V}$ called left and right *singular vectors*

Example: SVD

SVD of

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$$

given by $U\Sigma V^T =$

$$\begin{bmatrix} .141 & .825 & -.420 & -.351 \\ .344 & .426 & .298 & .782 \\ .547 & .0278 & .664 & -.509 \\ .750 & -.371 & -.542 & .0790 \end{bmatrix}$$

$$\begin{bmatrix} 25.5 & 0 & 0 \\ 0 & 1.29 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} .504 & .574 & .644 \\ -.761 & -.057 & .646 \\ .408 & -.816 & .408 \end{bmatrix}$$

Applications of SVD

Minimum norm solution to $\boldsymbol{Ax} \cong \boldsymbol{b}$:

$$\boldsymbol{x} = \sum_{\sigma_i \neq 0} \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i$$

For ill-conditioned or rank deficient problems, "small" singular values can be dropped from summation to stabilize solution

Euclidean matrix norm:

$$\|\boldsymbol{A}\|_2 = \max_{\boldsymbol{x} \neq \boldsymbol{o}} \frac{\|\boldsymbol{Ax}\|_2}{\|\boldsymbol{x}\|_2} = \sigma_{\max}$$

Euclidean condition number of matrix:

$$\text{cond}(\boldsymbol{A}) = \sigma_{\max}/\sigma_{\min}$$

Rank of matrix: number of nonzero, or non-negligible, singular values

Pseudoinverse

Define pseudoinverse of scalar σ to be $1/\sigma$ if $\sigma \neq 0$, zero otherwise

Define pseudoinverse of (possibly rectangular) diagonal matrix by transposing and taking scalar pseudoinverse of each entry

Then pseudoinverse of general real $m \times n$ matrix $\boldsymbol{A}$ given by

$$\boldsymbol{A}^{+} = \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^{T}$$

Pseudoinverse always exists whether or not matrix is square or has full rank

If $\boldsymbol{A}$ is square and nonsingular, then $\boldsymbol{A}^{+} = \boldsymbol{A}^{-1}$

In all cases, minimum-norm solution to $\boldsymbol{A}\boldsymbol{x} \cong \boldsymbol{b}$ is given by $\boldsymbol{A}^{+}\boldsymbol{b}$

Orthogonal Bases

Columns of $\boldsymbol{U}$ corresponding to nonzero singular values form orthonormal basis for span($\boldsymbol{A}$)

Remaining columns of $\boldsymbol{U}$ form orthonormal basis for orthogonal complement span$(\boldsymbol{A})^{\perp}$

Columns of $\boldsymbol{V}$ corresponding to zero singular values form orthonormal basis for null space of $\boldsymbol{A}$

Remaining columns of $\boldsymbol{V}$ form orthonormal basis for orthogonal complement of null space

Lower-Rank Matrix Approximation

Another way to write SVD:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sigma_1 \boldsymbol{E}_1 + \sigma_2 \boldsymbol{E}_2 + \cdots + \sigma_n \boldsymbol{E}_n,$$

with $\boldsymbol{E}_i = \boldsymbol{u}_i \boldsymbol{v}_i^T$

$\boldsymbol{E}_i$ has rank 1 and can be stored using only $m + n$ storage locations

Product $\boldsymbol{E}_i \boldsymbol{x}$ can be formed using only $m + n$ multiplications

Condensed approximation to $\boldsymbol{A}$ obtained by omitting from summation terms corresponding to small singular values

Approximation using k largest singular values is closest matrix of rank k to $\boldsymbol{A}$

Approximation is useful in image processing, data compression, information retrieval, cryptography, etc.

Total Least Squares

Ordinary least squares applicable when right hand side $\boldsymbol{b}$ subject to random error but matrix $\boldsymbol{A}$ known accurately

When all data, including $\boldsymbol{A}$, subject to error, then total least squares more appropriate

Total least squares minimizes orthogonal distances, rather than vertical distances, between model and data

Total least squares solution can be computed from SVD of $[\boldsymbol{A}, \boldsymbol{b}]$

Comparison of Methods

Forming normal equations matrix $\boldsymbol{A}^T\boldsymbol{A}$ requires about $n^2m/2$ multiplications, and solving resulting symmetric linear system requires about $n^3/6$ multiplications

Solving least squares problem using Householder QR factorization requires about $mn^2 - n^3/3$ multiplications

If $m \approx n$, two methods require about same amount of work

If $m \gg n$, Householder QR requires about twice as much work as normal equations

Cost of SVD proportional to $mn^2 + n^3$, with proportionality constant ranging from 4 to 10, depending on algorithm used

Comparison of Methods, continued

Normal equations method produces solution whose relative error is proportional to $[\mathrm{cond}(\boldsymbol{A})]^2$

Required Cholesky factorization can be expected to break down if $\mathrm{cond}(\boldsymbol{A}) \approx 1/\sqrt{\epsilon_{\mathrm{mach}}}$ or worse

Householder method produces solution whose relative error is proportional to

$$\mathrm{cond}(\boldsymbol{A}) + \|\boldsymbol{r}\|_2\,[\mathrm{cond}(\boldsymbol{A})]^2,$$

which is best possible, since this is inherent sensitivity of solution to least squares problem

Householder method can be expected to break down (in back-substitution phase) only if

$$\mathrm{cond}(\boldsymbol{A}) \approx 1/\epsilon_{\mathrm{mach}}$$

or worse

Comparison of Methods, continued

Householder is more accurate and more broadly applicable than normal equations

These advantages may not be worth additional cost, however, when problem is sufficiently well conditioned that normal equations provide adequate accuracy

For rank-deficient or nearly rank-deficient problem, Householder with column pivoting can produce useful solution when normal equations method fails outright

SVD is even more robust and reliable than Householder, but substantially more expensive