E7441: Scientific computing in biology and biomedicine

Non-square linear systems

Vlad Popovici, Ph.D.

RECETOX

Outline

The systems of linear equations

General form:

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ & \ddots & \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

- if $m < n$: underdetermined case; find a minimum-norm solution
- if $m > n$: overdetermined case; minimize the squared error
- if $m = n$: determined case; already discussed

Reminder

- two vectors $\mathbf{y}, \mathbf{z}$ are orthogonal if $\mathbf{y}^T\mathbf{z} = 0$
- the span of a set of n independent vectors is $\mathtt{span}(\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}) = \left\{\sum_{i=1}^{n} \alpha_i \mathbf{v}_i \mid \alpha_i \in \mathbb{R}\right\}$
- the *row (column) space* of a matrix $\mathbf{A}$ is the linear subspace generated (or spanned) by the rows (colums) of $\mathbf{A}$. Its dimension is equal to $\mathtt{rank}(\mathbf{A}) \leq \mathtt{min}(m, n)$.
- by definition, $\mathtt{span}(\mathbf{A})$ is the column space of $\mathbf{A}$ and can be written as

$$C(\mathbf{A}) = \{\mathbf{v} \in \mathbb{R}^m : \mathbf{v} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \mathbb{R}^n\},$$

  so it is the space of transformed vectors by the action of multiplication by the matrix.

Underdetermined case

- $m < n$ there are more variables than equations, hence the solution is not unique
- consider the rows to be *linearly independent*
- then, any n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$ can be decomposed into

$$\mathbf{x} = \mathbf{x}^+ + \mathbf{x}^-$$

where $\mathbf{x}^+$ is in the row space of $\mathbf{A}$ and $\mathbf{x}^-$ is in the null space of $\mathbf{A}$ (orthogonal to the previous space):

$$\mathbf{x}^+ = \mathbf{A}^T\alpha \qquad \mathbf{A}\mathbf{x}^- = 0$$

- this leads to

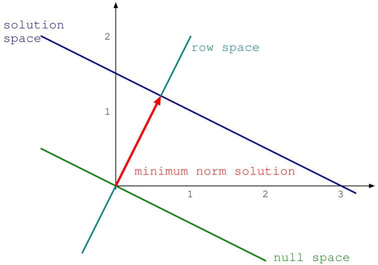
$$\mathbf{A}(\mathbf{x}^+ + \mathbf{x}^-) = \mathbf{A}\mathbf{A}^T\alpha + \mathbf{A}\mathbf{x}^- = \mathbf{A}\mathbf{A}^T\alpha = \mathbf{b}$$

- $\mathbf{A}\mathbf{A}^T$ is a $m \times m$ nonsingular matrix, so $\mathbf{A}\mathbf{A}^T\alpha = \mathbf{b}$ has a unique solution $\alpha_0 = (\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$
- the corresponding minimal norm solution to original system is

$$\mathbf{x}_0^+ = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$$

- note, however, that the orthogonal component $\mathbf{x}^-$ remains unspecified
- the matrix $\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$ is called the right pseudo-inverse of $\mathbf{A}$ (right: $\mathbf{A} \cdot \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{I}$)
- PYTHON: `scipy.linalg.pinv()` or `numpy.linalg.pinv()`

Example: let $\mathbf{A} = [1\ 2]$ and $\mathbf{b} = [3]$ (hence $m = 1$).



- solution space:

$$x_2 = -\frac{1}{2}x_1 + \frac{3}{2}$$

  is a solution, for any $x_1 \in \mathbb{R}$.

- $\mathbf{x}^+ = \mathbf{A}^T\alpha = \begin{bmatrix}1\\2\end{bmatrix}\alpha$ (row space)
- $\mathbf{Ax}^- = 0 \Rightarrow [1\ 2][x_1^-\ x_2^-]^T = 0.$
  $\Rightarrow x_2^- = -\frac{1}{2}x_1^-$ (null space)

The **minimal norm solution** is *the intersection of solution space with the row space* and is the closest vector to the origin, among all vectors in the solution space:

$$\mathbf{x}_0^+ = [0.6\ 1.2]^T$$

Overdetermined case

- if the rows of $\mathbf{A}$ are independent, there is no *perfect* solution to the system ($\mathbf{b} \notin \mathrm{span}(\mathbf{A})$)
- one needs some other criterion to call a solution *acceptable*
- least squares solution $\mathbf{x}_0$ minimizes the square Euclidean norm of the residual vector:

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} \|\mathbf{r}\|_2^2 = \arg\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$$

Solution to the LS problem

From a linear system problem, we arrived at solving an optimization problem with objective function

$$J = \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \frac{1}{2}(\mathbf{b} - \mathbf{A}\mathbf{x})^T(\mathbf{b} - \mathbf{A}\mathbf{x})$$

Set the derivative wrt $\mathbf{x}$ to zero:

$$\frac{\partial}{\partial \mathbf{x}} J = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x} = 0$$

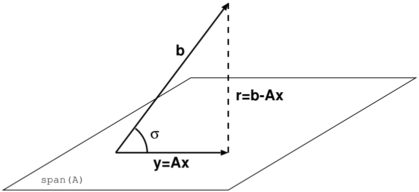
which leads to normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$, with the solution

$$\mathbf{x}_0 = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the *left pseudo-inverse* of $\mathbf{A}$.

Solution to the LS problem - geometric interpretation

- let $\mathbf{y} = \mathbf{Ax}$, where $\mathbf{x}$ is the LS solution
- the residual $\mathbf{r} = \mathbf{b} - \mathbf{y}$ is orthogonal to $\mathrm{span}(\mathbf{A})$,



LS data approximation

Model: $y = c_3x^2 + c_2x + c_1$. Problem: $c_i =?$ when (x_i, y_i) are given.
See *Example 1* in Jupyter notebook.

Condition number

- if $\text{rank}(\mathbf{A}) = n$ (columns are independent), the condition number is

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^\dagger\|_2$$

- by convention, if $\text{rank}(\mathbf{A}) < n$, $\text{cond}(\mathbf{A}) = \infty$
- for non-square matrices, the condition number measures the closeness to rank deficiency

Numerical methods for LS problem

- the LS solution can be obtained using the pseudo-inverse $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ or by solving the normal equations

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

  which is a system of n equations
- $\mathbf{A}^T\mathbf{A}$ is symmetric positive definite, so it admits a Cholesky decomposition,

$$\mathbf{A}^T\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

Issues with normal equations method

- floating-point computations in $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}^T\mathbf{b}$ may lead to information loss
- sensitivity of the solution is worsen, since $\mathrm{cond}(\mathbf{A}^T\mathbf{A}) = [\mathrm{cond}(\mathbf{A})]^2$

Example:

Let $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix}$ with $\epsilon \in \mathbb{R}_+$ and $\epsilon < \sqrt{\epsilon_{\text{mach}}}$. Then, in floating-point arithmetic, $\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 1 & 1+\epsilon^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ which is singular!

Augmented systems

- idea: find the solution and the residual as a solution of an extended system, under the orthogonality requirement
- the new system is

$$\begin{bmatrix} \mathbf{I} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

- despite requiring more storage and not being positive definite, it allows more freedom in choosing pivots for LU decomposition
- in some cases it is useful, but not much used in practice

Orthogonal transformations

- a matrix $\mathbf{Q}$ is orthogonal if $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$
- multiplication of a vector by an orthogonal matrix does not change its *Euclidean* norm:

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

- so, multiplying the two sides of the system by $\mathbf{Q}$ does not change the solution
- again: try to transform the system so it's easy to solve e.g. triangular system

- an upper triangular overdetermined ($m > n$) LS problem has the form

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \approx \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

  where $\mathbf{R}$ is an $n \times n$ upper triangular matrix and $\mathbf{b}$ is partitioned accordingly
- the residual becomes

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

- to minimize the residual, one has to minimize $||\mathbf{b}_1 - \mathbf{R}\mathbf{x}||_2^2$ (since $||\mathbf{b}_2||_2^2$ is fixed) and this leads to the system

$$\mathbf{R}\mathbf{x} = \mathbf{b}_1$$

  which can be solved by back-substitution
- the residual becomes $||\mathbf{r}||_2^2 = ||\mathbf{b}_2||_2^2$ and $\mathbf{x}$ is the LS solution

QR factorization

- problem: find an $m \times m$ orthogonal matrix $\mathbf{Q}$ such that an $m \times n$ matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

  where $\mathbf{R}$ is $n \times n$ upper triangular
- the new problem to solve is

$$\mathbf{Q}^T \mathbf{A} \mathbf{x} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{x} \approx \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \mathbf{Q}^T \mathbf{b}$$

- if $\mathbf{Q}$ is partitioned as $\mathbf{Q} = [\mathbf{Q}_1 \mathbf{Q}_2]$ with $\mathbf{Q}_1$ having n columns, then

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_1 \mathbf{R}$$

  is called reduced QR factorization of $\mathbf{A}$ (Python: `scipy.linalg.qr()`)
- columns of $\mathbf{Q}_1$ form an orthonormal basis of $\mathrm{span}(\mathbf{A})$, and the columns of $\mathbf{Q}_2$ form an orthonormal basis of $\mathrm{span}(\mathbf{A})^\perp$
- $\mathbf{Q}_1 \mathbf{Q}_1^T$ is orthogonal projector onto $\mathrm{span}(\mathbf{A})$
- the solution to the initial problem is given by the solution to the square system

$$\mathbf{Q}_1^T \mathbf{A} \mathbf{x} = \mathbf{Q}_1^T \mathbf{b}$$

QR factorization

In general, for an $m \times n$ matrix A, with $m > n$, the factorization is

$$\mathbf{A} = \mathbf{QR}$$

and

- $\mathbf{Q}$ is an *orthogonal* matrix: $\mathbf{Q}^T\mathbf{Q} = \mathbf{I} \Leftrightarrow \mathbf{Q}^{-1} = \mathbf{Q}^T$
- $\mathbf{R}$ is an *upper triangular* matrix
- solving the normal equations (for LS solution) $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$ comes to solving
$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathbf{b}$$

Example

See Example 2 in Jupyter notebook.

A statistical perspective

Changing a bit the notation, the linear model is

$$E[\mathbf{y}] = \mathbf{X}\beta, \qquad \mathrm{Cov}(y) = \sigma^2 I$$

It can be shown that the *best linear unbiased estimator* is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y}$$

for a decomposition $\mathbf{X} = \mathbf{QR}$. Then $\hat{\mathbf{y}} = \mathbf{QQ}^T\mathbf{y}$. (Gauss-Markov thm.: LS estimator has the lowest variance among all unbiased linear estimators.) Also,

$$\mathrm{Var}(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 = (\mathbf{R}^T\mathbf{R})^{-1}\sigma^2$$

where $\sigma^2 = \|\mathbf{y} - \hat{\mathbf{y}}\|^2/(m - n - 1)$.

Computing the QR factorization

- similarly to LU factorization, we nullify entries under the diagonal, column by column
- now, use orthogonal transformations:
  - Householder transformations
  - Givens rotations
  - Gram-Schmidt orthogonalization
- PYTHON:`scipy.linalg.qr()`

Householder transformations

$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}, \qquad \mathbf{v} \neq 0$$

- $\mathbf{H}$ is orthogonal and symmetric: $\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$
- $\mathbf{v}$ are chosen such that for a vector $\mathbf{a}$:

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

- this leads to $\mathbf{v} = \mathbf{a} - \alpha\mathbf{e}_1$ with $\alpha = \pm\|\mathbf{a}\|_2$, where the sign is chosen to avoid cancellation

Householder QR factorization

- apply, the Householder transformation to nuliffy the entries below diagonal
- the process is applied to each column (of the n) and produces a transformation of the form

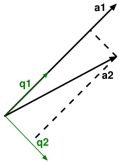
$$\mathbf{H}_n \dots \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

  where $\mathbf{R}$ is $n \times n$ upper triangular
- then take $\mathbf{Q} = \mathbf{H}_1 \dots \mathbf{H}_n$
- note that the multiplication of $\mathbf{H}$ with a vector $\mathbf{u}$ is much cheaper than a general matrix-vector multiplication:

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

Gram-Schmidt orthogonalization

- idea: given two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$, we seek orthonormal vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ having the same span



- method: subtract from $\mathbf{a}_2$ its projection on $\mathbf{a}_1$ and normalize the resulting vectors
- apply this method to each column of $\mathbf{A}$ to obtain the classical Gram-Schmidt procedure

Algorithm: Classical Gram-Schmidt

for $k = 1$ **to** n **do**

$\quad \mathbf{q}_k \leftarrow \mathbf{a}_k$;

$\quad$ **for** $j = 1$ **to** $k - 1$ **do**

$\quad\quad r_{jk} \leftarrow \mathbf{q}_j^T \mathbf{a}_k$;

$\quad\quad \mathbf{q}_k \leftarrow \mathbf{q}_k - r_{jk} \mathbf{q}_j$;

$\quad r_{kk} \leftarrow \|\mathbf{q}_k\|_2$;

$\quad \mathbf{q}_k \leftarrow \mathbf{q}_k / r_{kk}$;

The resulting matrices $\mathbf{Q}$ (with $\mathbf{q}_k$ as columns) and $\mathbf{R}$ (with elements r_{jk}) form the reduced QR factorization of $\mathbf{A}$.

Further topics on QR factorization

- if $\mathrm{rank}(\mathbf{A}) < n$ then $\mathbf{R}$ is singular and there are multiple solutions $\mathbf{x}$; choose the $\mathbf{x}$ with the smallest norm
- in limited precision, the rank can be lower than the theoretical one, leading to highly sensitive solutions → an alternative could be the SVD method (next)
- there exists a version, QR with pivoting, that chooses everytime the column with largest Euclidean norm for reduction → improves stability in rank deficient scenarios
- another method of factorization: Givens rotations - makes one 0 at a time

Singular Value Decomposition - SVD

- SVD of an $m \times n$ matrix $\mathbf{A}$ has the form

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

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

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

- σ_i are usually ordered such that $\sigma_1 \geq \cdots \geq \sigma_n$ and are called singular values of $\mathbf{A}$
- the columns $\mathbf{u}_i$ and $\mathbf{v}_i$ are called left and right singular vectors of $\mathbf{A}$, respectively

- minimum norm solution to $\mathbf{Ax} \approx \mathbf{b}$ is

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

- for ill-conditioned or rank-deficient problems, the sum should be taken over "large enough" σ's: $\sum_{\sigma_i \geq \epsilon} \ldots$
- Euclidean norm: $\|\mathbf{A}\|_2 = \max_i\{\sigma_i\}$
- Euclidean condition number: $\mathrm{cond}(\mathbf{A}) = \frac{\max_i\{\sigma_i\}}{\min_i\{\sigma_i\}}$
- Rank of $\mathbf{A}$: $\mathrm{rank}(\mathbf{A}) = \#\{\sigma_i > 0\}$

Pseudoinverse (again)

- the pseudoinverse of an $m \times n$ matrix $\mathbf{A}$ with SVD decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ is

$$\mathbf{A}^+ = \mathbf{V}\Sigma^{-1}\mathbf{U}^T$$

  where

$$[\Sigma^{-1}]_{ii} = \begin{cases} 1/\sigma_i & \text{for } \sigma_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

- pseudoinverse always exists and minimum norm solution to $\mathbf{Ax} \approx \mathbf{b}$ is $\mathbf{x} = \mathbf{A}^+\mathbf{b}$
- if $\mathbf{A}$ is square and nonsingular, $\mathbf{A}^{-1} = \mathbf{A}^+$

SVD and subspaces relevant to $\mathbf{A}$

- $\mathbf{u}_i$ for which $\sigma_i > 0$ form the orthonormal basis of $\mathsf{span}(\mathbf{A})$
- $\mathbf{u}_i$ for which $\sigma_i = 0$ form the orthonormal basis of the orthogonal complement of $\mathsf{span}(\mathbf{A})$
- $\mathbf{v}_i$ for which $\sigma_i = 0$ form the orthonormal basis of the null space of $\mathbf{A}$
- $\mathbf{v}_i$ for which $\sigma_i > 0$ form the orthonormal basis of the orthogonal complement of the null space of $\mathbf{A}$

SVD and matrix approximation

- $\mathbf{A}$ can be re-written as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \sigma_1\mathbf{u}_1\mathbf{v}_1^T + \cdots + \sigma_n\mathbf{u}_n\mathbf{v}_n^T$$

- let $\mathbf{E}_i = \mathbf{u}_i\mathbf{v}_i^T$; $\mathbf{E}_i$ has rank 1 and requires only $m+n$ storage locations
- $\mathbf{E}_i\mathbf{x}$ multiplication requires only $m+n$ multiplications
- assuming $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n$ then by using the largest k singular values, one obtains the closes approximation of $\mathbf{A}$ of rank k:

$$\mathbf{A} \approx \sum_{i=1}^{k} \sigma_i \mathbf{E}_i$$

- many applications to image processing, data compression, cryptography, etc.

Example - image compression

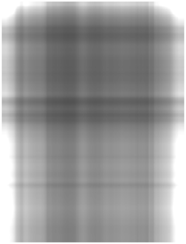
PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:



Example - image compression

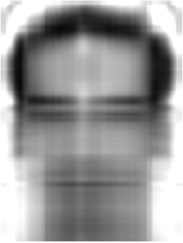
PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:



Example - image compression

PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Example - image compression

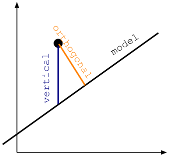
PYTHON: `scipy.linalg.svd()`

Original image and its approximations using 1,2,3,4,5 and 10 terms:

Total least squares

$$\mathbf{Ax} \approxeq \mathbf{b}$$

- ordinary least squares applies when the error affects only $\mathbf{b}$
- what if there is error (uncertainty) in $\mathbf{A}$ as well?
- total least squares minimizes the orthogonal distances, rather than vertical distances, between model and data



- can be computed using SVD of $[\mathbf{A}, \mathbf{b}]$

Comparison: work effort

- computing $\mathbf{A}^T\mathbf{A}$ requires about $n^2m/2$ multiplications and solving the resulting symmetric system, about $n^3/6$ multiplications
- LS problem solution by Householder QR requires about $mn^2 - n^3/3$ multiplications
- if $m \gg n$, Householder method requires about twice as much work normal eqs.
- cost of SVD is $\approx (4\ldots10) \times (mn^2 + n^3)$ depending on implementation

Comparison: precision

- relative error for normal eqs. is $\sim [\mathrm{cond}(\mathbf{A})]^2$; if $\mathrm{cond}(\mathbf{A}) \approx 1/\sqrt{\epsilon_{\mathrm{mach}}}$, Cholesky factorization will break
- Householder method has a relative error

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

  which is the best achievable for LS problems
- Householder method breaks (in back-substitution step) for $\mathrm{cond}(\mathbf{A}) \lessapprox 1/\epsilon_{\mathrm{mach}}$
- while Householder method is more general and more accurate than normal equations, it may not always be worth the additional cost

Comparison: precision, cont'd

- for (nearly) rank-deficient problems, the pivoting Householder method produces useful solution, while normal equations method fails
- SVD is more precise and more robust than Householder method, but much more expensive computationally

Eigenvalue problems

Standard eigenvalue problem

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

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

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

Characteristic polynomial

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

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

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

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

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

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

Example

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

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

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

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

so $v_{21} = 0$. We choose v_{11} such that $\|\mathbf{v}_1\| = 1$, so $v_{11} = 1$. Similarly, for $\lambda_2 = -1$ we get $\mathbf{v}_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$.

See Example 4.

Sensitivity of the characteristic polynomial

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

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

with the solution 1 (double root)

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

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

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

Eigen-decomposition

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

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

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

Eigenvectors

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

Some useful properties

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

Before solving an eigenvalue problem...

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

Conditioning of EV problems

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

Computation - general ideas

- a matrix $\mathbf{B}$ is similar to $\mathbf{A}$ if there exists a nonsingular matrix $\mathbf{T}$ such that $\mathbf{B} = \mathbf{T}^{-1}\mathbf{AT}$
- if $\mathbf{y}$ is an eigenvector of $\mathbf{B}$ then $\mathbf{x} = \mathbf{Ty}$ is an eigenvector of $\mathbf{A}$ and HOMEWORK: prove that $\mathbf{A}$ and $\mathbf{B}$ have the same eigenvalues
- transformations:
  - *shift:* $\mathbf{A} \leftarrow \mathbf{A} - \sigma\mathbf{I}$
  - *inversion:* $\mathbf{A} \leftarrow \mathbf{A}^{-1}$ (if $\mathbf{A}$ is nonsingular)
  - *power:* $\mathbf{A} \leftarrow \mathbf{A}^k$
  - *polynomial:* let p be a polynomial, then $\mathbf{A} \leftarrow p(\mathbf{A})$

Forms attainable by similarity

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

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

If **A** is diagonal...

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

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

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

If **A** is triangular (Schur form, in general)...

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

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

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

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

  is the corresponding eigenvector

Symmetric matrices - Jacobi method

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

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

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

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

from which we use the root with the smallest magnitude

- for more general matrices, there are other methods like *Power iterations*, with or without deflation, etc.
- a generalized eigenvalue problem,

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

  can be solved using the QZ algorithm

Singular Value Decomposition - again

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

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

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

The relation between SVD and the eigen-decomposition

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

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

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

Questions?