Lec 19: Overdetermined Linear Systems

– QR Algorithm

Revisiting Least Squares

Moore-Penrose Pseudoinverse

Let $A \in \mathbb{R}^{m\times n}$ with $m \geqslant n$ and suppose that columns of A are linearly independent.

- The least square problem $A\mathbf{x}$ "=" $\mathbf{b}$ is equivalent to the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$, which is a square matrix equation.
- The solution can be written as

$$\mathbf{x} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\mathbf{b}.$$

- The matrix

$$A^{+} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}} \in \mathbb{R}^{n\times m},$$

  is called the **(Moore-Penrose) pseudoinverse**.
- MATLAB's backslash is mathematically equivalent to left-multiplication by the inverse or pseudoinverse of a matrix.
- MATLAB's `pinv` calculates the pseudoinverse, but it is rarely used in practice, just as `inv`.

Moore-Penrose Pseudoinverse (cont')

- A^+ can be calculated by using the thin QR factorization[1] $A = \widehat{Q}\widehat{R}$.

$$A^+ = \widehat{R}^{-1}\widehat{Q}^{\mathrm{T}}.$$

[1] It can be done using the thick QR factorization as seen on p.1624 of the text.

Least Squares and QR Factorization

Substitute the thin factorization $A = \widehat{Q}\widehat{R}$ into the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ and simplify.

Least Squares and QR Factorization (cont')

Summary: Algorithm for LLS Approximation

If A has rank n, the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ is consistent and is equivalent to $\widehat{R}\mathbf{x} = \widehat{Q}^{\mathrm{T}}\mathbf{b}$.

1. Factor $A = \widehat{Q}\widehat{R}$.
2. Let $\mathbf{z} = \widehat{Q}^{\mathrm{T}}\mathbf{b}$.
3. Solve $\widehat{R}\mathbf{x} = \mathbf{z}$ for $\mathbf{x}$ using backward substitution.

Least Squares and QR Factorization (cont')

```
function x = lsqrfact(A,b)
% LSQRFACT x = lsqrfact(A,b)
% Sove linear least squares by QR factorization
% Input:
%   A    coefficient matrix (m-by-n, m>n)
%   b    right-hand side (m-by-1)
% Output:
%   x    minimizer of || b - Ax || (2-norm)
    [Q,R] = qr(A,0);              % thin QR fact.
    z = Q'*b;
    x = backsub(R,c);
end
```

Householder Transformation and QR Algorithm

Motivation

Problem

Given $\mathbf{z} \in \mathbb{R}^m$, find an orthogonal matrix $H \in \mathbb{R}^{m \times m}$ such that $H\mathbf{z}$ is nonzero only in the first element.

- Since orthogonal matrices preserve the 2-norm, H must satisfy

$$H\mathbf{z} = \begin{bmatrix} \pm \|\mathbf{z}\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \pm \|\mathbf{z}\|_2 \, \mathbf{e}_1 .$$

- The **Householder transformation matrix** H defined by

$$H = I - 2\frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}, \quad \text{where } \mathbf{v} = \pm \|\mathbf{z}\|_2 \, \mathbf{e}_1 - \mathbf{z},$$

solves the problem. See Theorem 1.

Properties of Householder Transformation

Theorem 1

Let $\mathbf{v} = \|\mathbf{z}\|_2 \mathbf{e}_1 - \mathbf{z}$ *and let* H *be the Householder transformation defined by*

$$H = I - 2\frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}.$$

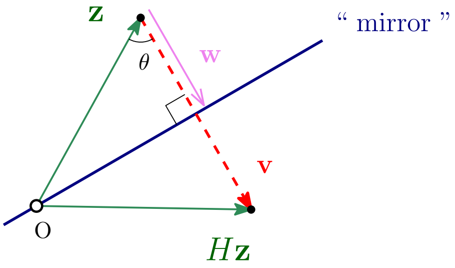
Then

1. H *is symmetric;*
2. H *is orthogonal;*
3. $H\mathbf{z} = \|\mathbf{z}\|_2 \mathbf{e}_1$.

- H is invariant under scaling of $\mathbf{v}$.
- If $\|\mathbf{v}\|_2 = 1$, then $H = I - \mathbf{v}\mathbf{v}^{\mathrm{T}}$.

Geometry Behind Householder Transformation (cont')

The Householder transformation matrix H can be thought of as a *reflector*[2].



[2] See Supplementary 1 on for review on projection and reflection operators

Factorization Algorithm

- The Gram-Schmidt orthogonalization (thin QR factorization) is unstable in floating-point calculations.
- **Stable alternative:** Find orthogonal matrices $H_1, H_2, \ldots, H_n$ so that

$$\underbrace{H_n H_{n-1} \cdots H_2 H_1}_{=: Q^{\mathrm{T}}} A = R\,.$$

  introducing zeros one column at a time below diagonal terms.
- As a product of orthogonal matrices, Q^{T} is also orthogonal and so $(Q^{\mathrm{T}})^{-1} = Q$. Therefore,

$$A = QR\,.$$

MATLAB Demonstration Code MYQR

```
function [Q, R] = myqr(A)
  [m, n] = size(A);
  A0 = A;
  Q = eye(m);
  for j = 1:min(m,n)
      Aj = A(j:m, j:n);
      z = Aj(:, 1);
      v = z + sign0(z(1))*norm(z)*eye(length(z), 1);
      Hj = eye(length(v)) - 2/(v'*v) * v*v';
      Aj = Hj*Aj;
      H = eye(m);
      H(j:m, j:m) = Hj;
      Q = Q*H;
      A(j:m, j:n) = Aj;
  end
  R = A;
end
```

MATLAB Demonstration Code MYQR (cont')

(continued from the previous page)

```
% local function
function sign0(x)
  y = ones(size(x));
  y(x < 0) = -1;
end
```

- The MATLAB command `qr` works similar to, but more efficiently than, this.
- The function finds the factorization in $\sim (2mn^2 - n^3/3)$ flops asymptotically.

Supplementary 1: Projection and Reflection

Projection and Reflection Operators

Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ be nonzero vectors.

- Projection of $\mathbf{u}$ onto $\langle \mathbf{v} \rangle = \mathrm{span}(\mathbf{v})$:

$$\frac{\mathbf{v}^{\mathrm{T}}\mathbf{u}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v} = \underbrace{\left(\frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\right)}_{=:P}\mathbf{u} =: P\mathbf{u}.$$

- Projection of $\mathbf{u}$ onto $\langle \mathbf{v} \rangle^{\perp}$, the orthogonal complement of $\langle \mathbf{v} \rangle$:

$$\mathbf{u} - \frac{\mathbf{v}^{\mathrm{T}}\mathbf{u}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v} = \left(I - \frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\right)\mathbf{u} =: (I - P)\mathbf{u}.$$

- Reflection of $\mathbf{u}$ across $\langle \mathbf{v} \rangle^{\perp}$:

$$\mathbf{u} - 2\frac{\mathbf{v}^{\mathrm{T}}\mathbf{u}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\mathbf{v} = \left(I - 2\frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\right)\mathbf{u} =: (I - 2P)\mathbf{u}.$$

Projection and Reflection Operators (cont')

Summary: for given $\mathbf{v} \in \mathbb{R}^m$, a nonzero vector, let

$$P = \frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}} \in \mathbb{R}^{m \times m}.$$

Then the following matrices carry out geometric transformations

- Projection onto $\langle \mathbf{v} \rangle$: P
- Projection onto $\langle \mathbf{v} \rangle$: $I - P$
- Reflection across $\langle \mathbf{v} \rangle^{\perp}$: $I - 2P$

Note. If $\mathbf{v}$ were a unit vector, the definition of P simplifies to $P = \mathbf{v}\mathbf{v}^{\mathrm{T}}$.

Supplementary 2: Conditioning and Stability

Analytical Properties of Pseudoinverse

The matrix $A^{\mathrm{T}}A$ appearing in the definition of A^{+} satisfies the following properties.

Theorem 2

For any $A \in \mathbb{R}^{m\times n}$ with $m \geqslant n$, the following are true:

1. *$A^{\mathrm{T}}A$ is symmetric.*
2. *$A^{\mathrm{T}}A$ is singular if and only if rank$(A) < n$.*
3. *If $A^{\mathrm{T}}A$ is nonsingular, then it is positive definite.*

A symmetric positive definite (SPD) matrix S such as $A^{\mathrm{T}}A$ permits so-called the **Cholesky factorization**

$$S = R^{\mathrm{T}}R$$

where R is an upper triangular matrix.

Least Squares Using Normal Equation

One can solve the LLS problem $A\mathbf{x}$ "=" $\mathbf{b}$ by solving the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ directly as below.

1. Compute $N = A^{\mathrm{T}}A$.
2. Compute $\mathbf{z} = A^{\mathrm{T}}\mathbf{b}$.
3. Solve the square linear system $N\mathbf{x} = \mathbf{z}$ for $\mathbf{x}$.

Step 3 is done using `chol` which implements the Cholesky factorization.

MATLAB Implementarion.

```
N = A'*A;
z = A'*b;
R = chol(N);
w = forelim(R',z);     % solve R'w = z
x = backsub(R,w);      % solve R x = w
```

Conditioning of Normal Equations

- Recall that the condition number of solving a square linear system $A\mathbf{x} = \mathbf{b}$ is $\kappa(A) = \|A\| \left\|A^{-1}\right\|$.
- Provided that the residual norm at the least square solution is relatively small, the conditioning of LLS problem is similar:

$$\kappa(A) = \|A\| \left\|A^{+}\right\| .$$

- If A is rank-deficient (columns are linearly dependent), then $\kappa(A) = \infty$.
- If an LLS problem is solved solving the normal equation, it can be shown that the condition number is

$$\kappa(A^{\mathrm{T}}A) = \kappa(A)^2.$$

Which Reflector Is Better?

- Recall:

$$H = I - 2\frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}, \quad \text{where } \mathbf{v} = \pm \|\mathbf{z}\|_2 \, \mathbf{e}_1 - \mathbf{z},$$

- In `myqr.m`, the statement

```
v = z + sign0(z(1))*norm(z)*eye(length(z), 1);
```

defines $\mathbf{v}$ slightly differently[3], namely,

$$\mathbf{v} = \mathbf{z} \pm \|\mathbf{z}\|_2 \, \mathbf{e}_1.$$

[3]This does not cause any difference since H is invariant under scaling of $\mathbf{v}$; see p.10

Which Reflector Is Better? (cont')

The sign of $\pm \|\mathbf{z}\|_2$ is chosen so as to avoid possible catastrophic cancellation in forming $\mathbf{v}$:

$$\mathbf{v} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} + \begin{bmatrix} \pm \|\mathbf{z}\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} z_1 \pm \|\mathbf{z}\|_2 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}$$

Subtractive cancellation may arise when $z_1 \approx \pm \|\mathbf{z}\|_2$.

- if $z_1 > 0$, use $z_1 + \|\mathbf{z}\|_2$;
- if $z_1 < 0$, use $z_1 - \|\mathbf{z}\|_2$;
- if $z_1 = 0$, either works.

For numerical stability, it is desirable to reflect $\mathbf{z}$ to the vector $s \|\mathbf{z}\|_2 \mathbf{e}_1$ that is not too close to $\mathbf{z}$ itself. (Trefethen & Bau)