Outline

1 Householder Reflectors

2 Linear Least Squares Problems

Householder Reflectors

Gram-Schmidt as Triangular Orthogonalization

- Every step of Gram-Schmidt can be viewed as multiplication with triangular matrix. For example, at first step:

$$[\mathbf{v}_1|\mathbf{v}_2|\ldots\mathbf{v}_n]\underbrace{\begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{bmatrix}}_{\mathbf{R}_1} = [q_1|\mathbf{v}_2^{(2)}|\ldots|\mathbf{v}_n^{(2)}],$$

- Gram-Schmidt therefore multiplies triangular matrices to orthogonalize column vectors, and in turns can be viewed as triangular orthogonalization

$$\mathbf{A}\underbrace{\mathbf{R}_1\mathbf{R}_2\ldots\mathbf{R}_n}_{\hat{\mathbf{R}}^{-1}} = \hat{\mathbf{Q}}$$

where $\mathbf{R}_i$ is a triangular matrix.
- A 'dual' approach would be orthogonal triangularization, i.e., multiply $\mathbf{A}$ by unitary matrices to make it triangular matrix

Householder Triangularization

- Method introduced by Alston Scott Householder in 1958
- It multiplies unitary matrices to make column triangular, e.g.

$$\underbrace{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}}_{\mathbf{A}} \xrightarrow{\mathbf{Q}_1} \underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}}_{\mathbf{Q}_1\mathbf{A}} \xrightarrow{\mathbf{Q}_2} \underbrace{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & 0 & \times \\ & 0 & \times \\ & 0 & \times \end{bmatrix}}_{\mathbf{Q}_2\mathbf{Q}_1\mathbf{A}} \xrightarrow{\mathbf{Q}_3} \underbrace{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \\ & & 0 \\ & & 0 \end{bmatrix}}_{\mathbf{Q}_3\mathbf{Q}_2\mathbf{Q}_1\mathbf{A}}$$

- After n steps, we get a product of unitary matrices

$$\underbrace{\mathbf{Q}_n \ldots \mathbf{Q}_2 \mathbf{Q}_1}_{\mathbf{Q}^*} \mathbf{A} = \mathbf{R}$$

  and in turn we get full QR factorization $\mathbf{A} = \mathbf{QR}$
- $\mathbf{Q}_k$ introduces zeros below diagonal of kth column while preserving zeros below diagonal in preceding columns
- The key question is how to find $\mathbf{Q}_k$

Householder Reflectors

- First, consider $\mathbf{Q}_1 : \mathbf{Q}_1\mathbf{a}_1 = \|\mathbf{a}_1\| \mathbf{e}_1$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$. Why the length is $\|\mathbf{a}_1\|$?
- $\mathbf{Q}_1$ reflects a_1 across hyperplane H orthogonal to $\mathbf{v} = \|a_1\| e_1 - a_1$, and therefore

$$\mathbf{Q}_1 = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^*}{\mathbf{v}^*\mathbf{v}}$$

- More generally,

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{F} \end{bmatrix}$$

  where $\mathbf{I}$ is $(k-1) \times (k-1)$ and $\mathbf{F}$ is $(m-k+1) \times (m-k+1)$ such that $\mathbf{F}\mathbf{x} = \|x\|_2 e_1$, where $\mathbf{x}$ is $(a_{k,k}, a_{k,k+1}, \ldots, a_{k,m})^T$
- What is $\mathbf{F}$? It has similar form as $\mathbf{Q}_1$ with $\mathbf{v} = \|x\| e_1 - \mathbf{x}$.

Choice of Reflectors

- We could choose $\mathbf{F}$ such that $\mathbf{Fx} = -\|x\| e_1$ instead of $\mathbf{Fx} = \|x\| e_1$, or more generally, $\mathbf{Fx} = z\|\mathbf{x}\| e_1$ with $|z| = 1$ for $z \in \mathbb{C}$
- This leads to an infinite number of possible QR factorizations of $\mathbf{A}$
- If we require $z \in \mathbb{R}$, we still have two choices
- Numerically, it is undesirable for $\mathbf{v}^*\mathbf{v}$ to be close to zero for $\mathbf{v} = z\|\mathbf{x}\| e_1 - \mathbf{x}$, and $\|v\|$ is larger if $z = -sign(x_1)$
- Therefore, $\mathbf{v} = -sign(x_1)\|\mathbf{x}\| e_1 - \mathbf{x}$. Since $\mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^*}{\mathbf{v}^*\mathbf{v}}$ is independent of sign, clear out the factor -1 and obtain $\mathbf{v} = sign(x_1)\|\mathbf{x}\| e_1 + \mathbf{x}$
- For completeness, if $x_1 = 0$, set z to 1 (instead of 0)

Householder Algorithm

Householder QR Factorization

for $k = 1$ to n
$\mathbf{x} = \mathbf{A}_{k:m,k}$
$\mathbf{v}_k = sign(x_1)\,\|x\|\,e_1 + \mathbf{x}$
$\mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|$
$\mathbf{A}_{k:m,k:n} = \mathbf{A}_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{A}_{k:m,k:n})$

- Note that $sign(x) = 1$ if $\mathbf{x} \geq 0$ and $= -1$ if $\mathbf{x} < 0$
- Leave **R** in place of **A**
- Matrix **Q** is not formed explicitly but reflection vector $\mathbf{v}_k$ is stored

Householder Algorithm

Householder QR Factorization

for $k = 1$ to n
$\mathbf{x} = \mathbf{A}_{k:m,k}$
$\mathbf{v}_k = sign(x_1)\,\|x\|\,e_1 + \mathbf{x}$
$\mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|$
$\mathbf{A}_{k:m,k:n} = \mathbf{A}_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^* \mathbf{A}_{k:m,k:n})$

- Note that $sign(x) = 1$ if $\mathbf{x} \geq 0$ and $= -1$ if $\mathbf{x} < 0$
- Leave $\mathbf{R}$ in place of $\mathbf{A}$
- Matrix $\mathbf{Q}$ is not formed explicitly but reflection vector $\mathbf{v}_k$ is stored
- Question: Can $\mathbf{A}$ be reused to store both $\mathbf{R}$ and $\mathbf{v}_k$ completely?

Householder Algorithm

Householder QR Factorization

for $k = 1$ to n
$\mathbf{x} = \mathbf{A}_{k:m,k}$
$\mathbf{v}_k = sign(x_1)\,\|x\|\,e_1 + \mathbf{x}$
$\mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|$
$\mathbf{A}_{k:m,k:n} = \mathbf{A}_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^* \mathbf{A}_{k:m,k:n})$

- Note that $sign(x) = 1$ if $\mathbf{x} \geq 0$ and $= -1$ if $\mathbf{x} < 0$
- Leave **R** in place of **A**
- Matrix **Q** is not formed explicitly but reflection vector $\mathbf{v}_k$ is stored
- Question: Can **A** be reused to store both **R** and $\mathbf{v}_k$ completely?
- Answer: We can use lower diagonal portion of **A** to store all but one entry in each $\mathbf{v}_k$. So an additional array of size n is needed.

Householder Algorithm

Householder QR Factorization

for $k = 1$ to n
$\mathbf{x} = \mathbf{A}_{k:m,k}$
$\mathbf{v}_k = sign(x_1)\,\|x\|\,e_1 + \mathbf{x}$
$\mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|$
$\mathbf{A}_{k:m,k:n} = \mathbf{A}_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{A}_{k:m,k:n})$

- Note that $sign(x) = 1$ if $\mathbf{x} \geq 0$ and $= -1$ if $\mathbf{x} < 0$
- Leave **R** in place of **A**
- Matrix **Q** is not formed explicitly but reflection vector $\mathbf{v}_k$ is stored
- Question: Can **A** be reused to store both **R** and $\mathbf{v}_k$ completely?
- Answer: We can use lower diagonal portion of **A** to store all but one entry in each $\mathbf{v}_k$. So an additional array of size n is needed.
- Question: What happens if $\mathbf{v}_k$ is 0 in line 3 of the loop?

Applying or Forming $\mathbf{Q}$

- Compute $\mathbf{Q}^*\mathbf{b} = \mathbf{Q}_n \dots \mathbf{Q}_1\mathbf{b}$

> Implicit calculation of $\mathbf{Q}^*\mathbf{b}$
> for $k = 1$ to n
> $b_{k:m} = b_{k:m} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{b}_{k:m})$

- Compute $\mathbf{Q}\mathbf{x} = \mathbf{Q}_1\mathbf{Q}_2 \dots \mathbf{Q}_n\mathbf{x}$

> Implicit calculation of $\mathbf{Q}\mathbf{x}$
> for $k = n$ down to 1
> $\mathbf{x}_{k:m} = \mathbf{x}_{k:m} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{x}_{k:m})$

- Question: How to form $\mathbf{Q}$ and $\hat{\mathbf{Q}}$, respectively?
- Answer: Apply $\mathbf{x} = \mathbf{I}_{m\times m}$ or first n columns of $\mathbf{I}$, respectively

Operation Count

- Most work done at step $\mathbf{A}_{k:m,k:n} = \mathbf{A}_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{A}_{k:m,k:n})$
- Flops per iteration:
  - $\sim 2(m-k)(n-k)$ for dot products $\mathbf{v}_k^*\mathbf{A}_{k:m,k:n}$
  - $\sim (m-k)(n-k)$ for outer product $2v_k(\dots)$
  - $\sim (m-k)(n-k)$ for subtraction
  - $\sim 4(m-k)(n-k)$ total
- Including outer loop, total flops is

$$\begin{aligned}\sum_{k=1}^{n} 4(m-k)(n-k) &= 4\sum_{k=1}^{n}(mn-km-kn+k^2)\\ &\sim 4mn^2 - 4(m+n)n^2/2 + 4n^3/3\\ &= 2mn^2 - \frac{2}{3}n^3\end{aligned}$$

If $m \approx n$, it is more efficient than Gram-Schmidt method, but if $m \gg n$, similar to Gram-Schmidt

Givens Rotations

- Instead of using reflection, we can rotate **x** to obtain $\|x\| e_1$
- A Given rotation $\mathbf{R} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$ rotates $\mathbf{x} \in \mathbb{R}^2$ counterclockwise by θ
- Choose θ to be angle between $(x_i, x_j)^T$ and $(1,0)^T$, and we have

$$\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{bmatrix}$$

where

$$cos\theta = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad sin\theta = \frac{-x_j}{\sqrt{x_i^2 + x_j^2}}$$

Givens QR

- Introduce zeros in column bottom-up, one zero at a time

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{(4,5)} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \\ & \times & \times \end{bmatrix} \cdots$$

- To zero a_{ij} , left-multiply matrix $\mathbf{F}$ with $\mathbf{F}_{i:i+1,i:i+1}$ being rotation matrix and $\mathbf{F}_{kk} = 1$ for $k \neq i, i+1$
- Flop count of Givens **QR** is $3mn^2 - n^3$, which is about 50% more expensive than Householder QR

Linear Least Squares Problems

Linear Least Squares Problems

- Overdetermined system of equations $\mathbf{Ax} \approx \mathbf{b}$, where $\mathbf{A}$ has more rows than columns and has full rank, in general has no solutions
- Example application: Polynomial least squares fitting
- In general, minimize the residual $\mathbf{r} = \mathbf{b} - \mathbf{Ax}$
- In terms of 2-norm, we obtain linear least squares problem: Given $\mathbf{A} \in \mathbb{C}^{m \times n}$, $m \geq n$, and $\mathbf{b} \in \mathbb{C}^m$, find $\mathbf{x} \in \mathbb{C}^n$ such that $\|\mathbf{b} - \mathbf{Ax}\|_2$ is minimized
- If $\mathbf{A}$ has full rank, the minimizer $\mathbf{x}$ is the solution to the normal equation

$$\mathbf{A}^*\mathbf{Ax} = \mathbf{A}^*\mathbf{b}$$

or in terms of the pseudoinverse $\mathbf{A}^+$,

$$\mathbf{x} = \mathbf{A}^+\mathbf{b}, \text{ where } \mathbf{A}^+ = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^* \in \mathbb{C}^{n \times m}$$

Geometric Interpretation

- $\mathbf{Ax}$ is in range($\mathbf{A}$), and the point in range($\mathbf{A}$) closest to $\mathbf{b}$ is its orthogonal projection onto range($\mathbf{A}$)
- Residual $\mathbf{r}$ is then orthogonal to range($\mathbf{A}$), and hence $\mathbf{A}^*\mathbf{r} = \mathbf{A}^*(\mathbf{b} - \mathbf{Ax}) = 0$
- $\mathbf{Ax}$ is orthogonal projection of $\mathbf{b}$, where $\mathbf{x} = \mathbf{A}^+\mathbf{b}$, so $\mathbf{P} = \mathbf{AA}^+ = \mathbf{A}(\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*$ is orthogonal projection (recall lecture 5)

Solution of Least Squares Problems

- One approach is to solve normal equation $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$ directly using Cholesky factorization
  - Is unstable, but is very efficient if $m \gg n$ $(mn^2 + \frac{1}{3}n^3)$
- More robust approach is to use QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$
  - $\mathbf{b}$ can be projected onto range($\mathbf{A}$) by $\mathbf{P} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^*$, and therefore $\hat{\mathbf{Q}}\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^*\mathbf{b}$
  - Left-multiply by $\hat{\mathbf{Q}}^*$ and we get $\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^*\mathbf{b}$(note $\mathbf{A}^+ = \hat{\mathbf{R}}^{-1}\hat{\mathbf{Q}}^*$)

> Least squares via QR Factorization:
> Compute reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$
> Compute vector $\mathbf{c} = \hat{\mathbf{Q}}^*\mathbf{b}$
> Solve upper-triangular system $\hat{\mathbf{R}}\mathbf{x} = \mathbf{c}$ for $\mathbf{x}$

Solution of Least Squares Problems contd.

Least squares via QR Factorization:
  Compute reduced QR factorization $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$
  Compute vector $\mathbf{c} = \hat{\mathbf{Q}}^*\mathbf{b}$
  Solve upper-triangular system $\hat{\mathbf{R}}\mathbf{x} = \mathbf{c}$ for $\mathbf{x}$

- Computation is dominated by QR factorization ($2mn^2 - \frac{2}{3}n^3$)
- Question: If Householder QR is used, how to compute $\hat{\mathbf{Q}}\mathbf{b}$?
- Answer: Compute $\mathbf{Q}^*\mathbf{b}$ (where $\mathbf{Q}$ is from full QR factorization) and then take first n entries of resulting $\mathbf{Q}^*\mathbf{b}$

Solution by SVD

- Using $\mathbf{A} = \hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*$, $\mathbf{b}$ can be projected onto range($\mathbf{A}$) by $\mathbf{P} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*$, and therefore $\hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*\mathbf{b}$
- Left-multiply by $\hat{\mathbf{U}}$ and we get $\hat{\Sigma}\hat{\mathbf{V}}^*\mathbf{x} = \hat{\mathbf{U}}^*\mathbf{b}$

> Least squares via SVD:
> Compute reduced SVD factorization $\mathbf{A} = \hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*$
> Compute vector $\mathbf{c} = \hat{\mathbf{U}}^*\mathbf{b}$
> Solve diagonal system $\hat{\Sigma}\mathbf{w} = \mathbf{c}$ for $\mathbf{w}$
> Set $\mathbf{x} = \mathbf{V}\mathbf{w}$

- Work is dominated by SVD, which is $\sim 2mn^2 + 11n^3$ flops, very expensive if $m \approx n$
- Best numerical stability
- Question: If $\mathbf{A}$ is rank deficient, how to solve $\mathbf{A}\mathbf{x} \approx \mathbf{b}$?

Solution by SVD

- Using $\mathbf{A} = \hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*$, $\mathbf{b}$ can be projected onto range($\mathbf{A}$) by $\mathbf{P} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*$, and therefore $\hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{U}}^*\mathbf{b}$
- Left-multiply by $\hat{\mathbf{U}}$ and we get $\hat{\Sigma}\hat{\mathbf{V}}^*\mathbf{x} = \hat{\mathbf{U}}^*\mathbf{b}$

> Least squares via SVD:
> Compute reduced SVD factorization $\mathbf{A} = \hat{\mathbf{U}}\hat{\Sigma}\mathbf{V}^*$
> Compute vector $\mathbf{c} = \hat{\mathbf{U}}^*\mathbf{b}$
> Solve diagonal system $\hat{\Sigma}\mathbf{w} = \mathbf{c}$ for $\mathbf{w}$
> Set $\mathbf{x} = \mathbf{V}\mathbf{w}$

- Work is dominated by SVD, which is $\sim 2mn^2 + 11n^3$ flops, very expensive if $m \approx n$
- Best numerical stability
- Question: If $\mathbf{A}$ is rank deficient, how to solve $\mathbf{A}\mathbf{x} \approx \mathbf{b}$?
- Answer: $\mathbf{x}$ is no longer unique. Constrain $\mathbf{x}$ to be orthogonal to null space of $\mathbf{A}$.