Matrix Computations
Chapter 3: Least-squares Problems and QR Decomposition

Section 3.3 QR Decomposition

Jie Lu
ShanghaiTech University

Thin QR Decomposition for Full Column-Rank Matrices

Theorem

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$ *has full column rank. Then,* $\mathbf{A}$ *admits a decomposition*

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1 \qquad \textit{(Thin QR Decomposition)}$$

where $\mathbf{Q}_1 \in \mathbb{R}^{m \times n}$ *is semi-orthogonal and* $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$ *is nonsingular and upper triangular.*
In addition, if we restrict $[\mathbf{R}_1]_{ii} > 0$ *for all* $i = 1, \ldots, n$*, then* $(\mathbf{Q}_1, \mathbf{R}_1)$ *is unique.*

Proof:
Since $\mathbf{A}$ has full column rank, $\mathbf{C} := \mathbf{A}^T \mathbf{A}$ is positive definite.
Hence, there exists a unique Cholesky decomposition $\mathbf{C} = \mathbf{R}_1^T \mathbf{R}_1$ where $\mathbf{R}_1$ is upper triangular with positive diagonal entries.
Let $\mathbf{Q}_1 = \mathbf{A}\mathbf{R}_1^{-1}$. It can be verified that $\mathbf{Q}_1^T \mathbf{Q}_1 = \mathbf{I}$ and $\mathbf{Q}_1 \mathbf{R}_1 = \mathbf{A}$.

Note: We don't find QR decomposition via Cholesky decomposition in practice

Gram-Schmidt Procedure

Aim: Given a basis $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n\}$ of a subspace $\mathcal{S} \subset \mathbb{R}^m$, find an orthonormal basis $\{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n\}$ of $\mathcal{S}$, i.e.,

1. $\mathrm{span}\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n\} = \mathrm{span}\{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n\}$
2. $\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix}$ is a semi-orthogonal matrix (an orthogonal matrix if $m = n$)

Idea: Let $\mathbf{q}_1$ be normalized $\mathbf{a}_1$
Each $\mathbf{q}_{i+1}$ is obtained by removing $\mathbf{q}_1-, \ldots, \mathbf{q}_i-$component from $\mathbf{a}_{i+1}$, $i = 1, \ldots, n-1$ and then normalizing it

Note: Orthogonal projection of vector $\mathbf{a}$ onto vector $\mathbf{b}$ is given by

$$\frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\langle \mathbf{b}, \mathbf{b} \rangle}\mathbf{b},$$

Gram-Schmidt Procedure (cont'd)

$$
\begin{aligned}
\tilde{\mathbf{q}}_1 &= \mathbf{a}_1 \\
\mathbf{q}_1 &= \frac{\tilde{\mathbf{q}}_1}{\|\tilde{\mathbf{q}}_1\|_2} \\
\tilde{\mathbf{q}}_2 &= \mathbf{a}_2 - (\mathbf{q}_1^T \mathbf{a}_2)\mathbf{q}_1 \\
\mathbf{q}_2 &= \frac{\tilde{\mathbf{q}}_2}{\|\tilde{\mathbf{q}}_2\|_2} \\
&\dots \\
\tilde{\mathbf{q}}_i &= \mathbf{a}_i - (\mathbf{q}_1^T \mathbf{a}_i)\mathbf{q}_1 - (\mathbf{q}_2^T \mathbf{a}_i)\mathbf{q}_2 - \cdots - (\mathbf{q}_{i-1}^T \mathbf{a}_i)\mathbf{q}_{i-1} \\
\mathbf{q}_i &= \frac{\tilde{\mathbf{q}}_i}{\|\tilde{\mathbf{q}}_i\|_2} \\
&\dots \\
\tilde{\mathbf{q}}_n &= \mathbf{a}_n - \sum_{i=1}^{n-1} (\mathbf{q}_i^T \mathbf{a}_n)\mathbf{q}_i \\
\mathbf{q}_n &= \frac{\tilde{\mathbf{q}}_n}{\|\tilde{\mathbf{q}}_n\|_2}
\end{aligned}
$$

Gram-Schmidt Procedure (cont'd)

Algorithm: Gram-Schmidt
input: a collection of linearly independent vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$
$\tilde{\mathbf{q}}_1 = \mathbf{a}_1$, $\mathbf{q}_1 = \tilde{\mathbf{q}}_1 / \|\tilde{\mathbf{q}}_1\|_2$
for $i = 2, \ldots, n$
  $\tilde{\mathbf{q}}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} (\mathbf{q}_j^T \mathbf{a}_i)\mathbf{q}_j$
  $\mathbf{q}_i = \tilde{\mathbf{q}}_i / \|\tilde{\mathbf{q}}_i\|_2$
end
output: $\mathbf{q}_1, \ldots, \mathbf{q}_n$

- Complexity: $\mathcal{O}(mn^2)$
- $\mathrm{span}\{\mathbf{a}_1, \cdots, \mathbf{a}_i\} = \mathrm{span}\{\mathbf{q}_1, \cdots, \mathbf{q}_i\} = \mathrm{span}\{\tilde{\mathbf{q}}_1, \cdots, \tilde{\mathbf{q}}_i\}$ for all $i = 1, \ldots, n$

Thin QR Decomposition via Gram-Schmidt

From Gram-Schmidt,

$$\mathbf{a}_i = \sum_{j=1}^{i} r_{ji}\mathbf{q}_j, \quad i = 1, \ldots, n$$

where

$$r_{ii} = \|\tilde{\mathbf{q}}_i\|_2, \quad r_{ji} = \mathbf{q}_j^T \mathbf{a}_i, \; j = 1, \ldots, i-1$$

Equivalently,

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1$$

where

$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}$ full column rank

$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix}$ semi-orthogonal

$\mathbf{R}_1$ is upper triangular with $[\mathbf{R}_1]_{ij} = r_{ij}$ for $i \leq j$

- $\mathbf{R}_1$ is nonsingular because $\det(\mathbf{R}) = \Pi_{i=1}^{n} r_{ii} \neq 0$

General Gram-Schmidt Procedure

Extension to the case where $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \dots & \mathbf{a}_n \end{bmatrix}$ may not have full column rank

Observation from Gram-Schmidt:

- If $\mathbf{a}_j$ is linearly dependent of $\mathbf{a}_1, \dots, \mathbf{a}_{j-1}$, then $\tilde{\mathbf{q}}_j = 0$
- The number of nonzero $\tilde{\mathbf{q}}_i$'s is rank(A)

Idea: If $\tilde{\mathbf{q}}_j = 0$, skip to $j+1$ without computing $\mathbf{q}_j$

All the $\mathbf{q}_i$'s form an orthonormal basis for $\mathcal{R}(\mathbf{A})$

Algorithm: General Gram-Schmidt

input: a collection of possibly linearly dependent vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$

k=0

for $i = 1, \dots, n$

  $\tilde{\mathbf{q}}_i = \mathbf{a}_i - \sum_{j=1}^{k} (\mathbf{q}_j^T \mathbf{a}_i) \mathbf{q}_j$

  if $\tilde{\mathbf{q}}_i \neq 0$

    $k \leftarrow k+1$

    $\mathbf{q}_k = \tilde{\mathbf{q}}_i / \|\tilde{\mathbf{q}}_i\|_2$

  end % $\mathrm{span}\{\mathbf{a}_1, \dots, \mathbf{a}_i\} = \mathrm{span}\{\mathbf{q}_1 \dots, \mathbf{q}_k\}$

end

output: $\mathbf{q}_1, \dots, \mathbf{q}_k$ % $k = \mathrm{rank}(A)$

General Gram-Schmidt Procedure (cont'd)

Example: Let $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 \end{bmatrix} \in \mathbb{R}^{6\times 5}$
Suppose $\mathbf{a}_1 \neq \mathbf{0}$; $\mathbf{a}_2$ is linearly independent from $\mathbf{a}_1$; $\mathbf{a}_3$ is linearly dependent of $\mathbf{a}_1$ and $\mathbf{a}_2$; $\mathbf{a}_4$ is linearly independent from $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$; $\mathbf{a}_5$ is linearly dependent of $\mathbf{a}_2$ only

General Gram-Schmidt Procedure (cont'd)

Example: Let $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 \end{bmatrix} \in \mathbb{R}^{6\times 5}$
Suppose $\mathbf{a}_1 \neq \mathbf{0}$; $\mathbf{a}_2$ is linearly independent from $\mathbf{a}_1$; $\mathbf{a}_3$ is linearly dependent of $\mathbf{a}_1$ and $\mathbf{a}_2$; $\mathbf{a}_4$ is linearly independent from $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$; $\mathbf{a}_5$ is linearly dependent of $\mathbf{a}_2$ only

General Gram-Schmidt Procedure (cont'd)

Using General Gram-Schmidt, $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $\text{rank}(\mathbf{A}) = k \le n$ can be decomposed as

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1$$

where

$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_k \end{bmatrix} \in \mathbb{R}^{m \times k}$ is semi-orthogonal

$\mathbf{R}_1 \in \mathbb{R}^{k \times n}$ is in an upper staircase form, where each staircase corresponds to a column of $\mathbf{A}$ that is independent from previous columns

$\mathbf{R}_1 \in \mathbb{R}^{k \times n}$ is upper triangular[1]

Applications:

- Obtain an orthonormal basis for $\mathcal{R}(\mathbf{A})$
- Check whether $\mathbf{b} \in \text{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ by applying general Gram-Schmidt to $\{\mathbf{a}_1, \ldots, \mathbf{a}_n, \mathbf{b}\}$
- The staircase pattern of $\mathbf{R}_1$ indicates the dependence of each column of $\mathbf{A}$ on previous columns

[1] From now on, we say a rectangular matrix is upper triangular if its (i, j)-entry is zero for all $i > j$

QR Decomposition

Theorem

Any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ admits a decomposition

$$\mathbf{A} = \mathbf{QR} \qquad \textit{(QR Decomposition)}$$

where $\mathbf{Q} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix and $\mathbf{R} \in \mathbb{R}^{m \times n}$ is an upper triangular matrix.
In addition, when $m = n$ and $\mathbf{A}$ has full rank, $(\mathbf{Q}, \mathbf{R})$ is unique if we restrict $r_{ii} > 0$ for all i.

Finding QR Decomposition via General Gram-Schmidt

1. Find any matrix $\tilde{\mathbf{A}}$ s.t. the matrix $\begin{bmatrix} \mathbf{A} & \tilde{\mathbf{A}} \end{bmatrix}$ has full row rank
   - We may simply let $\tilde{\mathbf{A}} = \mathbf{I}_m$
2. Applying General Gram-Schmidt gives

$$\begin{bmatrix} \mathbf{A} & \tilde{\mathbf{A}} \end{bmatrix} = \mathbf{Q}\bar{\mathbf{R}}, \quad \mathbf{Q} \in \mathbb{R}^{m \times m} \text{ orthogonal}$$

3. Write $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix}$ where
   - $\mathbf{Q}_1 \in \mathbb{R}^{m \times k}$ with $k = \text{rank}(\mathbf{A})$ provides an orthonormal basis for $\mathcal{R}(A)$
   - $\mathbf{Q}_2 \in \mathbb{R}^{m \times (m-k)}$ provides an orthonormal basis for $\mathcal{R}(\tilde{\mathbf{A}})$
4. Note that

$$\mathbf{A} = \underbrace{\mathbf{Q}_1}_{m \times k} \underbrace{\mathbf{R}_1}_{k \times n} = \underbrace{\begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0}_{(m-k) \times n} \end{bmatrix}}_{\mathbf{R}}$$

Discussions

Thin QR Decomposition for general $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \geq n$:

$$\mathbf{A} = \tilde{\mathbf{Q}}_1 \underbrace{\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0}_{(n-k)\times n} \end{bmatrix}}_{\tilde{\mathbf{R}}_1}$$

where

$\tilde{\mathbf{Q}}_1 \in \mathbb{R}^{m \times n}$ is semi-orthogonal

$\tilde{\mathbf{R}}_1 \in \mathbb{R}^{n \times n}$ is upper triangular

When $\mathbf{A}$ has full column rank, then $\tilde{\mathbf{Q}}_1 = \mathbf{Q}_1$ and $\tilde{\mathbf{R}}_1 = \mathbf{R}_1$

$\mathbf{A}$ has full column rank if and only if $[\mathbf{R}_1]_{ii} \neq 0$ for all i

Discussions (cont'd)

Since $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \in \mathbb{R}^{m \times m}$ is orthogonal,

- $\mathcal{R}(\mathbf{Q}_1)$ and $\mathcal{R}(\mathbf{Q}_2)$ are orthogonal
- $\mathcal{R}(\mathbf{Q}_1)$ and $\mathcal{R}(\mathbf{Q}_2)$ spans $\mathbb{R}^m$

Therefore, $\mathcal{R}(\mathbf{Q}_1)$ and $\mathcal{R}(\mathbf{Q}_2)$ are orthogonal complements of each other, i.e.,

$$\mathcal{R}(\mathbf{Q}_1)^\perp = \mathcal{R}(\mathbf{Q}_2)$$

It follows that

$$\mathcal{R}(\mathbf{A}) = \mathcal{R}(\mathbf{Q}_1), \quad \mathcal{R}(\mathbf{A})^\perp = \mathcal{R}(\mathbf{Q}_2)$$

- The columns of $\mathbf{Q}_1$ form an orthonormal basis for $\mathcal{R}(\mathbf{A})$
- The columns of $\mathbf{Q}_2$ form an orthonormal basis for $\mathcal{R}(\mathbf{A})^\perp = \mathcal{N}(\mathbf{A}^T)$

LS via QR

Let $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\text{rank}(\mathbf{A}) = k$

$$\mathbf{A} = \begin{bmatrix} \underbrace{\mathbf{Q}_1}_{m\times k} & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \underbrace{\mathbf{R}_1}_{k\times n} \\ \mathbf{0}_{(m-k)\times n} \end{bmatrix}$$

Using the QR decomposition,

$$\begin{aligned}
\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 &= \|\mathbf{Q}^T\mathbf{A}\mathbf{x} - \mathbf{Q}^T\mathbf{y}\|_2^2 \qquad \text{because orthogonal } Q \text{ preserves norm} \\
&= \left\| \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \mathbf{y} \right\|_2^2 \\
&= \left\| \begin{bmatrix} \underbrace{\mathbf{Q}_1^T\mathbf{Q}_1}_{\mathbf{I}} & \underbrace{\mathbf{Q}_1^T\mathbf{Q}_2}_{\mathbf{0}} \\ \underbrace{\mathbf{Q}_2^T\mathbf{Q}_1}_{\mathbf{0}} & \underbrace{\mathbf{Q}_2^T\mathbf{Q}_2}_{\mathbf{I}} \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{Q}_1^T\mathbf{y} \\ \mathbf{Q}_2^T\mathbf{y} \end{bmatrix} \right\|_2^2 \\
&= \left\| \begin{bmatrix} \mathbf{R}_1\mathbf{x} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{Q}_1^T\mathbf{y} \\ \mathbf{Q}_2^T\mathbf{y} \end{bmatrix} \right\|_2^2 = \|\mathbf{R}_1\mathbf{x} - \mathbf{Q}_1^T\mathbf{y}\|_2^2 + \|\mathbf{Q}_2^T\mathbf{y}\|_2^2
\end{aligned}$$

LS via QR (cont'd)

$$\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 = \|\mathbf{R}_1\mathbf{x} - \mathbf{Q}_1^T\mathbf{y}\|_2^2 + \|\mathbf{Q}_2^T\mathbf{y}\|_2^2$$

Conclusion: $\mathbf{x}_{\text{LS}}$ is a least-squares solution to $\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ if and only if it is a least-squares solution to $\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{b}_1 - \mathbf{R}_1\mathbf{x}\|_2^2$ where $\mathbf{b}_1 = \mathbf{Q}_1^T\mathbf{y}$

Suppose $\mathbf{A}$ has full column rank, i.e., $k = n$

Then, $\mathbf{R}_1$ is nonsingular and the unique least-squares solution is

$$\mathbf{x}_{\text{LS}} = \mathbf{R}_1^{-1}\mathbf{b}_1$$

We may solve the triangular system $\mathbf{R}_1\mathbf{x} = \mathbf{b}_1$ by backward substitution

In this case, the optimal residual $\|\mathbf{A}\mathbf{x}_{\text{LS}} - \mathbf{y}\|_2$ is

$$\|\mathbf{Q}_2^T\mathbf{y}\|_2 = \|\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{y}\|_2$$

Note that $\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{y} \in \mathcal{R}(\mathbf{Q}_2) = \mathcal{R}(\mathbf{Q}_1)^\perp = \mathcal{R}(\mathbf{A})^\perp$

$\mathbf{Q}_2\mathbf{Q}_2^T\mathbf{y}$ is the component of $\mathbf{y}$ orthogonal to $\mathcal{R}(\mathbf{A})$

Numerical Error Issue of Gram-Schmidt

Gram-Schmidt is numerically unstable due to propagation of numerical errors

Example: Given
$\mathbf{a}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T, \mathbf{a}_2 = \begin{bmatrix} 1 & 0 & \epsilon & 0 \end{bmatrix}^T, \mathbf{a}_3 = \begin{bmatrix} 1 & 0 & 0 & \epsilon \end{bmatrix}^T$ with tiny ϵ so that the approximation $1 + \epsilon^2 \approx 1$ can be made

Applying Gram-Schmidt with the above approximation yields

- $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2} = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$
- $\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - \mathbf{q}_1^T \mathbf{a}_2 \mathbf{q}_1 = \begin{bmatrix} 0 & -\epsilon & \epsilon & 0 \end{bmatrix}^T$
  $\mathbf{q}_2 = \frac{\tilde{\mathbf{q}}_2}{\|\tilde{\mathbf{q}}_2\|_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}^T$
- $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - \mathbf{q}_1^T \mathbf{a}_3 \mathbf{q}_1 - \mathbf{q}_2^T \mathbf{a}_3 \mathbf{q}_2 = \begin{bmatrix} 0 & -\epsilon & 0 & \epsilon \end{bmatrix}^T$
  $\mathbf{q}_3 = \frac{\tilde{\mathbf{q}}_3}{\|\tilde{\mathbf{q}}_3\|_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix}^T$

  Orthogonality is lost!

Modified Gram-Schmidt

Instead of computing $\tilde{\mathbf{q}}_i = \mathbf{a}_i - (\mathbf{q}_1^T \mathbf{a}_i)\mathbf{q}_1 - (\mathbf{q}_2^T \mathbf{a}_i)\mathbf{q}_2 - \cdots - (\mathbf{q}_{i-1}^T \mathbf{a}_i)\mathbf{q}_{i-1}$ in Gram-Schmidt (full column rank case), compute

$$
\begin{aligned}
\tilde{\mathbf{q}}_i^{(1)} =& \mathbf{a}_i - (\mathbf{q}_1^T \mathbf{a}_i)\mathbf{q}_1 \\
\tilde{\mathbf{q}}_i^{(2)} =& \tilde{\mathbf{q}}_i^{(1)} - (\mathbf{q}_2^T \tilde{\mathbf{q}}_i^{(1)})\mathbf{q}_2 \\
&\vdots \\
\tilde{\mathbf{q}}_i^{(j)} =& \tilde{\mathbf{q}}_i^{(j-1)} - (\mathbf{q}_j^T \tilde{\mathbf{q}}_i^{(j-1)})\mathbf{q}_j \\
&\vdots \\
\tilde{\mathbf{q}}_i = \tilde{\mathbf{q}}_i^{(i-1)} =& \tilde{\mathbf{q}}_i^{(i-2)} - (\mathbf{q}_{i-1}^T \tilde{\mathbf{q}}_i^{(i-2)})\mathbf{q}_{i-1}
\end{aligned}
$$

Complexity: $\mathcal{O}(mn^2)$

Modified Gram-Schmidt (cont'd)

Example (revisit): Given

$\mathbf{a}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T, \mathbf{a}_2 = \begin{bmatrix} 1 & 0 & \epsilon & 0 \end{bmatrix}^T, \mathbf{a}_3 = \begin{bmatrix} 1 & 0 & 0 & \epsilon \end{bmatrix}^T$ with tiny ϵ so that the approximation $1 + \epsilon^2 \approx 1$ can be made

Applying modified Gram-Schmidt with the above approximation yields

- $\tilde{\mathbf{q}}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$
  $\mathbf{q}_1 = \begin{bmatrix} 1 & \epsilon & 0 & 0 \end{bmatrix}^T$
- $\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - \mathbf{q}_1^T \mathbf{a}_2 \mathbf{q}_1 = \begin{bmatrix} 0 & -\epsilon & \epsilon & 0 \end{bmatrix}^T$
  $\mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}^T$
- $\tilde{\mathbf{q}}_3^{(1)} = \mathbf{a}_3 - \mathbf{q}_1^T \mathbf{a}_3 \mathbf{q}_1 = \begin{bmatrix} 0 & -\epsilon & 0 & \epsilon \end{bmatrix}^T$
  $\tilde{\mathbf{q}}_3 = \tilde{\mathbf{q}}_3^{(2)} = \tilde{\mathbf{q}}_3^{(1)} - \mathbf{q}_2^T \tilde{\mathbf{q}}_3^{(1)} \mathbf{q}_2 = \begin{bmatrix} 0 & -\frac{\epsilon}{2} & -\frac{\epsilon}{2} & \epsilon \end{bmatrix}^T$
  $\mathbf{q}_3 = \frac{1}{\sqrt{6}} \begin{bmatrix} 0 & -1 & -1 & 2 \end{bmatrix}^T$

  Orthogonality is preserved approximately

We may also compute QR using reflection and rotation approaches

Reflection Matrices

A matrix $\mathbf{H} \in \mathbb{R}^{m \times m}$ is called a reflection matrix if

$$\mathbf{H} = \mathbf{I} - 2\mathbf{P},$$

where $\mathbf{P}$ is an orthogonal projector (symmetric and idempotent)

Interpretation: Let $\mathbf{P}^{\perp} = \mathbf{I} - \mathbf{P}$ be the orthogonal complement projector

$$\mathbf{x} = \mathbf{P}\mathbf{x} + \mathbf{P}^{\perp}\mathbf{x}, \qquad \mathbf{H}\mathbf{x} = -\mathbf{P}\mathbf{x} + \mathbf{P}^{\perp}\mathbf{x}$$

The vector $\mathbf{H}\mathbf{x}$ is a reflected version of $\mathbf{x}$, with $\mathcal{R}(\mathbf{P}^{\perp})$ being the "mirror"

A reflection matrix is orthogonal:

$$\mathbf{H}^T\mathbf{H} = (\mathbf{I} - 2\mathbf{P})(\mathbf{I} - 2\mathbf{P}) = \mathbf{I} - 4\mathbf{P} + 4\mathbf{P}^2 = \mathbf{I} - 4\mathbf{P} + 4\mathbf{P} = \mathbf{I}$$

Householder Reflections

Problem: Given $\mathbf{x} \in \mathbb{R}^m$, find an orthogonal $\mathbf{H} \in \mathbb{R}^{m \times m}$ such that

$$\mathbf{H}\mathbf{x} = \begin{bmatrix} \beta \\ \mathbf{0} \end{bmatrix} = \beta \mathbf{e}_1, \qquad \text{for some } \beta \in \mathbb{R}$$

Householder reflection: Let $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{v} \neq \mathbf{0}$, and let

$$\mathbf{H} = \mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2} \mathbf{v}\mathbf{v}^T,$$

which is a reflection matrix with $\mathbf{P} = \mathbf{v}\mathbf{v}^T / \|\mathbf{v}\|_2^2$

Householder Reflections (cont'd)

$$\mathbf{Hx} = \left(\mathbf{I} - \frac{2}{\|\mathbf{v}\|_2^2}\mathbf{v}\mathbf{v}^T\right)\mathbf{x} = \mathbf{x} - \frac{2\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{v}$$

We want $\mathbf{Hx}$ to be a multiple of $\mathbf{e}_1$. Hence, we require $\mathbf{v} \in \mathrm{span}\{\mathbf{x}, \mathbf{e}_1\}$
Let $\mathbf{v} = \mathbf{x} + \alpha\mathbf{e}_1$. Then,

$$\mathbf{v}^T\mathbf{x} = \mathbf{x}^T\mathbf{x} + \alpha x_1, \quad \mathbf{v}^T\mathbf{v} = \mathbf{x}^T\mathbf{x} + 2\alpha x_1 + \alpha^2$$

It follows that

$$\mathbf{Hx} = \frac{\alpha^2 - \mathbf{x}^T\mathbf{x}}{\mathbf{x}^T\mathbf{x} + 2\alpha x_1 + \alpha^2}\mathbf{x} - 2\alpha\frac{\mathbf{v}^T\mathbf{x}}{\mathbf{v}^T\mathbf{v}}\mathbf{e}_1$$

The coefficient of $\mathbf{x}$ has to be zero, so that $\alpha^2 = \|\mathbf{x}\|_2^2$. Therefore,

$$\mathbf{v} = \mathbf{x} \mp \|\mathbf{x}\|_2\mathbf{e}_1 \quad \Longrightarrow \quad \mathbf{Hx} = \pm\|\mathbf{x}\|_2\mathbf{e}_1$$

The sign in the expression of $\mathbf{v}$ may be determined to be the one that maximizes $\|\mathbf{v}\|_2$ for the sake of numerical stability

Householder QR

1. Let $\mathbf{H}_1 \in \mathbb{R}^{m \times m}$ be the Householder reflection w.r.t. $\mathbf{a}_1$. Transform $\mathbf{A}$ as

$$\mathbf{A}^{(1)} = \mathbf{H}_1 \mathbf{A} = \begin{bmatrix} \times & \times & \dots & \times \\ 0 & \color{red}{\times} & \dots & \times \\ \vdots & \vdots & & \vdots \\ 0 & \color{red}{\times} & \dots & \times \end{bmatrix}$$

2. Let $\tilde{\mathbf{H}}_2 \in \mathbb{R}^{(m-1) \times (m-1)}$ be the Householder reflection w.r.t. $\mathbf{A}^{(1)}(2:m, 2)$ (marked red above). Transform $\mathbf{A}^{(1)}$ as

$$\mathbf{A}^{(2)} = \underbrace{\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}}_2 \end{bmatrix}}_{=\mathbf{H}_2} \mathbf{A}^{(1)} = \begin{bmatrix} \times & \times \quad \dots \times \\ \mathbf{0} & \tilde{\mathbf{H}}_2 \mathbf{A}^{(1)}(2:m, 2:n) \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \dots & \times \\ 0 & \times & \times & \dots & \times \\ \vdots & 0 & \times & \dots & \times \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \times & \dots & \times \end{bmatrix}$$

3. By repeating this, $\mathbf{A}$ is transformed to $\mathbf{R}$

Householder QR (cont'd)

WLOG, assume $m \geq n$

$\mathbf{A}^{(0)} = \mathbf{A}$
for $k = 1, \ldots, n-1$
  $\mathbf{A}^{(k)} = \mathbf{H}_k \mathbf{A}^{(k-1)}$, where

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}}_k \end{bmatrix}$$

  $\tilde{\mathbf{H}}_k$ is the Householder reflection of $\mathbf{A}^{(k-1)}(k:m, k)$
end

- The above procedure results in

$$\mathbf{A}^{(n-1)} = \mathbf{H}_{n-1} \cdots \mathbf{H}_2 \mathbf{H}_1 \mathbf{A}, \quad \mathbf{A}^{(n-1)} \text{ is upper triangular}$$

- QR decomposition is obtained by letting $\mathbf{R} = \mathbf{A}^{(n-1)}$ and $\mathbf{Q} = (\mathbf{H}_{n-1} \cdots \mathbf{H}_2 \mathbf{H}_1)^T$
- A widely used method for QR decomposition

Householder QR (cont'd)

$\mathbf{A}^{(0)} = \mathbf{A}$
for $k = 1, \ldots, n-1$
  $\mathbf{A}^{(k)} = \mathbf{H}_k \mathbf{A}^{(k-1)}$, where

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}}_k \end{bmatrix}$$

  $\tilde{\mathbf{H}}_k$ is the Householder reflection of $\mathbf{A}^{(k-1)}(k:m,k)$
end

Complexity (for $m \geq n$):

- $\mathcal{O}(n^2(m - n/3))$ for $\mathbf{R}$ only
  - A direct implementation of the above pseudo-code does not lead to this complexity–Need to exploit the structures of $\mathbf{H}_k$ in the implementations
- $\mathcal{O}(m^2 n - mn^2 + n^3/3)$ if $\mathbf{Q}$ is also wanted
- See Section 5.2.2 of textbook

Givens Rotations

Example: Let

$$\mathbf{J} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

where $c = \cos(\theta), s = \sin(\theta)$ for some θ

$$\mathbf{y} = \mathbf{Jx} \iff \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} cx_1 + sx_2 \\ -sx_1 + cx_2 \end{bmatrix}$$

Observe that

- $\mathbf{J}$ is orthogonal
- $y_2 = 0$ if $\theta = \tan^{-1}(x_2/x_1)$, or if

$$c = \frac{x_1}{\sqrt{x_1^2 + x_2^2}}, \quad s = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}$$

Givens Rotations (cont'd)

Givens rotations:

$$\mathbf{J}(i,k,\theta) = \begin{array}{cc} \begin{array}{ccccc} \phantom{\mathbf{I}} & i & \phantom{\mathbf{I}} & k & \phantom{\mathbf{I}} \\ & \downarrow & & \downarrow & \end{array} & \\ \begin{bmatrix} \mathbf{I} & & & & \\ & c & & s & \\ & & \mathbf{I} & & \\ & -s & & c & \\ & & & & \mathbf{I} \end{bmatrix} & \begin{array}{c} \\ \leftarrow i \\ \\ \leftarrow k \\ \\ \end{array} \end{array}$$

where $c = \cos(\theta)$, $s = \sin(\theta)$

- $\mathbf{J}(i,k,\theta)$ is orthogonal
- Let $\mathbf{y} = \mathbf{J}(i,k,\theta)\mathbf{x}$. Then,

$$y_j = \begin{cases} cx_i + sx_k, & j = i \\ -sx_i + cx_k, & j = k \\ x_j, & j \neq i,k \end{cases}$$

- y_k is forced to zero if we choose $\theta = \tan^{-1}(x_k/x_i)$

Givens QR

Example: Consider a 4×3 matrix.

$$\mathbf{A} = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{J}_{1,2}} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{J}_{1,3}} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{J}_{1,4}} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$

$$\xrightarrow{\mathbf{J}_{2,3}} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{\mathbf{J}_{2,4}} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix} \xrightarrow{\mathbf{J}_{3,4}} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{R}$$

where $\mathbf{B} \xrightarrow{\mathbf{J}} \mathbf{C}$ means $\mathbf{C} = \mathbf{J}\mathbf{B}$; $\mathbf{J}_{i,k} = \mathbf{J}(i, k, \theta)$, with θ chosen to zero out the (k, i)th entry of the matrix transformed by $\mathbf{J}_{i,k}$

Givens QR (cont'd)

Givens QR ($m \geq n$): Perform a sequence of Givens rotations to annihilate the lower triangular parts of $\mathbf{A}$

$$\underbrace{(\mathbf{J}_{n,m} \dots \mathbf{J}_{n,n+2}\mathbf{J}_{n,n+1}) \dots (\mathbf{J}_{2m} \dots \mathbf{J}_{24}\mathbf{J}_{23})(\mathbf{J}_{1m} \dots \mathbf{J}_{13}\mathbf{J}_{12})}_{=\mathbf{Q}^T} \mathbf{A} = \mathbf{R}$$

where $\mathbf{R}$ is upper triangular and $\mathbf{Q}$ is orthogonal

- Complexity (for $m \geq n$): $\mathcal{O}(n^2(m - n/3))$ for $\mathbf{R}$ only (see Section 5.2.5 of textbook)
- Not as efficient as Householder QR for general (and dense) $\mathbf{A}$'s
  - The flop count for Householder QR is $2n^2(m - n/3)$
  - The flop count for Givens QR is $3n^2(m - n/3)$
- Givens QR can be faster than Householder QR if $\mathbf{A}$ has certain sparse structures and we exploit them