Notes on Assignment 5

Question 1

Write an R program to implement,
QR factorization using the following.

1. Classic Gram Schmidt,
2. Modified Gram Schmidt (MGS) by modifying CGS as discussed in class,
3. Modified Gram Schmidt as outlined in the text book,
4. Householder method.

Directions

Program should take a matrix A as input and return both Q and R as output.
Use the following matrix :

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ -1 & 4 \end{bmatrix}$$

Compare results and verify that $A = QR$

Solution

Let

$$\begin{aligned} A_{CGS} &= \text{Classic Gram Schmidt} \\ A_{MGS} &= \text{Modified Gram Schmidt} \\ A_{TMGS} &= \text{Text Book Modified Gram Schmidt} \\ A_{HH} &= \text{Householder Method} \end{aligned}$$

Pseudocode

Classic Gram Schmidt A_{CGS}

Algorithm
For j $=1$ **to** n **do:**
$v_j = a_j$
For i $=1$ **to** j - 1 **do:**
$r_{ij} = q_i^* a_j$
$v_j = v_j - r_{ij} q_i$
$r_{jj} = \|v_j\|_2$
$$q_j = \frac{v_j}{r_{jj}}$$

Notes

Gram-Schmidt Orthogonalization using this algorithm produce a reduced QR fractorization denoted as $A = \hat{Q}\hat{R}$

Theorem 7.1

- Every $A \in \mathbb{C}^{m \times n}$ $(m \geq n)$ has a full QR factorization, hence also a reduced QR factorization.

Proof :
Suppose first that A has full rank and that we want just a reduced QR factorization.
In this case, a proof of existence is provided by the Gram-Schmidt algorithm.
By construction, this process generates orthonormal columns of $\hat{Q}$ and entries of $\hat{R}$ such that (7.4) holds.
Failure can occur only if at some step, v_j is zero and thus cannot be normalized to produce q_j .
However, this would imply $a_j \in \langle q_1, \cdots, q_{j-1} \rangle = \langle a_1, \cdots, a_{j-1} \rangle$, contradicting the assumption that A has full rank.

> Suppose that A does not have full rank. Then at one or more steps j, we shall find that (7.5) gives $v_j = 0$.
> Pick q_j arbitrarily to be any normalized vector orthogonal to $\langle q_1, \cdots, q_{j-1} \rangle$ and then continue the Gram-Schmidt process.
>
> Finally, the full, QR Factorization of an $m \times n$ matrix with $m > n$ can be constructed using arbitrary orthonormal vectors in the same way.
> Follow G-S process through step n,
>
> - then continue additional $m - n$ steps,
> - introducing q_j at each step.
>
> Now we turn to $\textbf{Uniqueness}$.
>
> Suppose$A = \hat{Q}\hat{R}$ is a reduced QR Factorization.
> If the i^{th} column of $\hat{Q}$ is multiplied by z
> and the i^{th} row of $\hat{R}$ is multiplied by $z^{-1} for some scalar \$ z$ with $|z| = 1$,
> we obtain another QR factorization of A.
> The next theorem asserts that if A has *full rank*
>
> - this is the only way to obtain distinct reduced QR factorizations.

Summary

How to do Classic by hand

Modified Gram Schmidt A_{MGS}

For $j - 1$ **to** n **do:**
$v_j^{(1)} = a_j$
end for
For $i = 1$ **to** n **do:**
$r_{ij} = \|v_j^{(1)}$

$q_i = \frac{v_i^{(1)}}{r_{ij}}$

For $j = i + 1$ **to** n **do:**

$r_{ij} = q_i^{(1)} v_j^{(1)}$

$v_j^{i+1} = v_j^{(1)} - r_{ij} q_i$

end for

end for

Text Book Modified Gram Schmidt A_{TMGS}

Pseudocode

For $i = 1$ **to** n **do:**

$v_i = a_i$

For $i = 1$ **to** n **do:**

$r_{ii} = \|v_i\|$

$q_i = \frac{v_i}{r_{ii}}$

For $j = i + 1$ **to** n **do:**

$r_{ij} = q_i^* v_j$

$v_j = v_j - r_{ij} q_i$

Note :

common to let v_i overwrite a_i

& q_i overwrite v_i

Householder method A_{HH}

Pseudocode

The Algorithm

For $k = 1$ to n do:

- $x = A_{k:m,\, k}$
- $v_k = \operatorname{sign}(x_1)\, \|x\|\, e_1 + x$
- $v_k = \dfrac{v_k}{\|v_k\|_2}$
- $A_{k:m,\, k:n} = A_{k:m,\, k:n} - 2v_k(v_k^* A_{k:m,\, k:n})$

Implicit calculation of a product Q^*B

---

sequence of n operations applied to b the same operations that were applied to A to make it triangular

---

For $k = 1$ to n do:

- $b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$

Implicit calculation of a Product Qx

Computation of a product Qx can be achieved by the same process in reverse order

---

For $k = n$ down to 1 do:

- $x_{k:m} = x_{k:m} - 2v_k(v_k^* x_{k:m})$

Upon completion of this algo,
A has been reduced to upper triangullar form

- **(We have calculated R)**

But

- we still need to calculate Q and,
- $n-$column submatrix $\hat{Q}$
    - corresponding to a reduced $\mathbf{QR}$ factorization

> ## The Algorithm
>
> ---
>
> **For $k = 1$ to n do:**
>
> - $x = A_{k:m,\, k}$
> - $v_k = \text{ sign } (x_1) \; \|x\| \; e_1 + x$
> - $v_k = \dfrac{v_k}{\|v_k\|_2}$
> - $A_{k:m,\, k:n} = A_{k:m,\, k:n} - 2v_k(v_k^* A_{k:m,\, k:n})$
>
> ## Implicit calculation of a product Q^*B
>
> ---
>
> sequence of n operations applied to b the same operations that were applied to A to make it triangular
>
> ---
>
> **For $k = 1$ to n do:**
>
> - $b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$
>
> ## Implicit calculation of a Product Qx
>
> ---
>
> Computation of a product Qx can be achieved by the same process in reverse order

For $k = n$ down to 1 do:

- $x_{k:m} = x_{k:m} - 2v_k(v_k^* x_{k:m})$