THE GRAM-SCHMIDT ALGORITHM AND QR [6.4]

Orthogonality – The Gram-Schmidt algorithm

1. Two vectors u and v are orthogonal if $u.v = 0$.
2. They are orthonormal if in addition $\|u\| = \|v\| = 1$
3. In this case the matrix $Q = [u, v]$ is such

$$\boxed{Q^T Q = I}$$

- We say that the system $\{u, v\}$ is orthonormal ..
- .. and that the matrix Q has orthonormal columns
- .. or that it is orthogonal [Text reserves this term to $n \times n$ case]

Example: An orthonormal system $\{u, v\}$

$$u = \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} \quad v = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

Generalization:

➤ A system of vectors $\{v_1, \ldots, v_n\}$ is orthogonal if $v_i . v_j = 0$ for $i \neq j$; and orthonormal if in addition $\|v_i\| = 1$ for $i = 1, \cdots, n$

- A matrix is orthogonal if its columns are orthonormal
- Then: $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$ has orthonormal columns

[Note: The term 'orthonormal matrix' is not used. 'orthogonal' is often used for square matrices only (textbook)]

The Gram-Schmidt algorithm

Problem: Given a set $\{u_1, u_2\}$ how can we generate another set $\{q_1, q_2\}$ from linear combinations of u_1, u_2 so that $\{q_1, q_2\}$ is orthonormal?

Step 1 Define first vector: $q_1 = u_1/\|u_1\|$ ('Normalization')

Step 2: Orthogonalize u_2 against q_1: $\hat{q} = u_2 - (u_2.q_1)\ q_1$

Step 3 Normalize to get second vector: $q_2 = \hat{q}/\|\hat{q}\|$

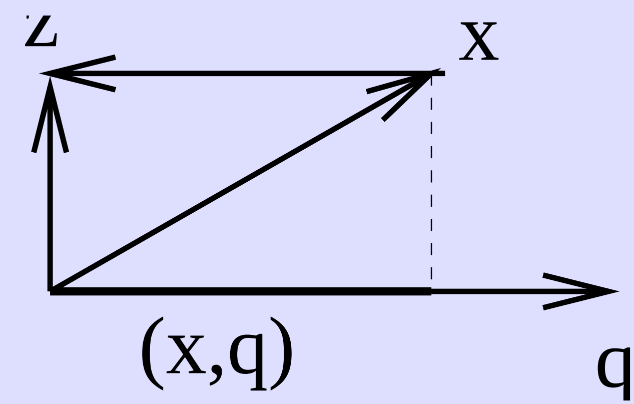
➤ Result: $\{q_1, q_2\}$ is an orthonormal set of vectors which spans the same space as $\{u_1, u_2\}$.

The operations in step 2 can be written as

$$\hat{q} := \boldsymbol{ORTH}(\boldsymbol{u_2}, \boldsymbol{q_1})$$

ORTH $(\boldsymbol{u_2}, \boldsymbol{q_1})$: "orthogonalize $\boldsymbol{u_2}$ against $\boldsymbol{q_1}$"

➤ $\boldsymbol{ORTH}(\boldsymbol{x}, \boldsymbol{q})$ denotes the operation of orthogonalizing a vector $\boldsymbol{x}$ against a unit vector $\boldsymbol{q}$.



Result of $\boldsymbol{z} = \boldsymbol{ORTH}(\boldsymbol{x}, \boldsymbol{q})$

Example: $u_1 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$ $u_2 = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 2 \end{pmatrix}$

Step 1: $q_1 = \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$ *Step 2:* First compute $u_2.q_1 = ... = 2$. Then:

$$\hat{q} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 2 \end{pmatrix} - 2 \times \frac{1}{2} \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

Step 3: Normalize $q_2 = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$

Generalization: 3 vectors

- How to generalize to 3 or more vectors?
- For 3 vectors : $[\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]$
  - First 2 steps are the same $\rightarrow \boldsymbol{q}_1, \boldsymbol{q}_2$
  - Then orthogonalize $\boldsymbol{u}_3$ against $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$:

$$\hat{q} = \boldsymbol{u}_3 - (\boldsymbol{u}_3.\boldsymbol{q}_1)\boldsymbol{q}_1 - (\boldsymbol{u}_3.\boldsymbol{q}_2)\boldsymbol{q}_2$$

  - Finally, normalize:

$$\boldsymbol{q}_3 = \hat{q}/\|\hat{q}\|$$

General problem: Given $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$, compute $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$ which is orthonormal and s.t. $\text{Col}(\boldsymbol{Q}) = \text{Col}(\boldsymbol{U})$.

ALGORITHM : 1. *Classical Gram-Schmidt*

1. For $j = 1 : n$ Do:
2. $\quad \hat{q} = u_j$
3. $\quad$ For $i = 1 : j - 1$
4. $\quad\quad \hat{q} := \hat{q} - (u_j.q_i)q_i$ $\qquad$ / set $r_{ij} = (u_j.q_i)$
5. $\quad$ End
6. $\quad q_j := \hat{q}/\|\hat{q}\|$ $\qquad$ / set $r_{jj} = \|\hat{q}\|$
7. End

➤ All n steps can be completed iff $u_1, u_2, \ldots, u_n$ are linearly independent.

➤ Define a matrix R as follows

$$r_{ij} = \begin{cases} u_j.q_i & \text{if} \quad i < j \ \text{(see line 4)} \\ \|\hat{q}\| & \text{if} \quad i = j \ \text{(see line 6)} \\ 0 & \text{if} \quad i > j \ \text{(lower part)} \end{cases}$$

- We have from the algorithm: (For $j = 1, 2, \cdots, n$)

$$u_j = r_{1j} q_1 + r_{2j} q_2 + \ldots + r_{jj} q_j$$

- If $U = [u_1, u_2, \ldots, u_n]$, $Q = [q_1, q_2, \ldots, q_n]$, and if R is the $n \times n$ upper triangular matrix defined above:

$$R = \{r_{ij}\}_{i,j=1,\ldots,n}$$

then the above relation can be written as

$$U = QR$$

- This is called the QR factorization of U.

- Q has orthonormal columns. It satisfies:

$$Q^T Q = I$$

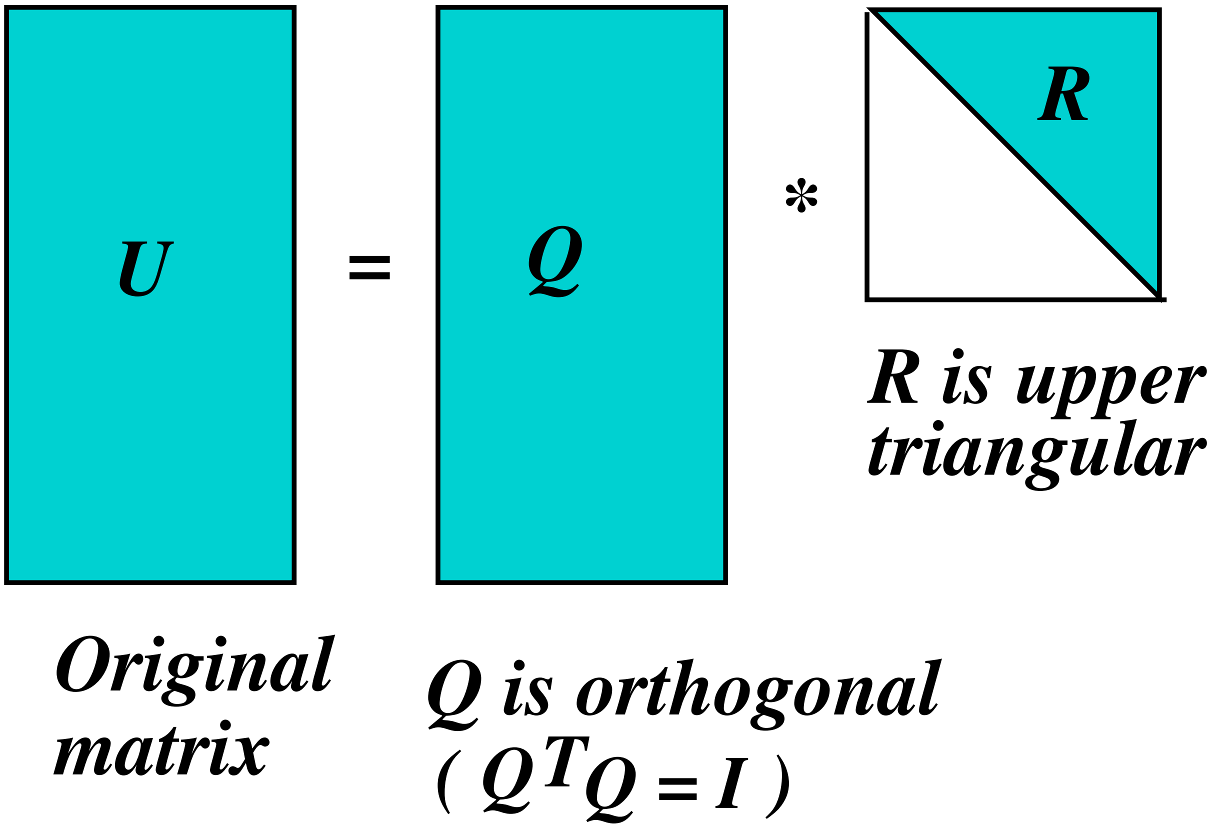
- It is said to be orthogonal
- R is upper triangular

✍ What is the inverse of an orthogonal $n \times n$ matrix?

✍ What is the cost of the factorization when $U \in \mathbb{R}^{m \times n}$?

Another decomposition:

A matrix U, with linearly independent columns, is the product of an orthogonal matrix Q and a upper triangular matrix R.



✍ Orthonormalize the system of vectors:

$$U = [u_1, u_2, u_3] = \begin{pmatrix} 1 & -4 & 3 \\ -1 & 2 & -1 \\ 1 & 0 & 1 \\ 1 & -2 & -1 \end{pmatrix}$$

For this example:

✍ 1) what is Q? what is R?

✍ 2) Verify (matlab) that $U = QR$

✍ 3) Compute $Q^T Q$. [Result should be the identity matrix]

Solving LS systems via QR factorization

- In practice: not a good idea to solve the system $A^T A x = A^T b$. Use the QR factorization instead. How?

- Answer in the form of an exercise

Problem: $Ax \approx b$ in least-squares sense

A is an $m \times n$ (full-rank) matrix. Consider the QR factorization of A

$$A = QR$$

✍ Approach 1: Write the normal equations – then 'simplify'

✍ Approach 2: Write the condition $b - Ax \perp \mathrm{Col}(A)$ and recall that A and Q have the same column space.

✍ Total cost?