QR factorization

(Gram-Schmidt orthogonalization)

1. Least squares
   Regression problem

2. Eigenvalues and Eigenvectors of
   Large scale Symmetric / Hermitian matrices

3. SVD uses bi-diagonalization techniques
   and have an underlying EVP to solve.

4. Solve linear system of equations!

Reduced QR factorization:

$$\underset{\sim}{A} = \begin{bmatrix} | & | & & | \\ \underset{\sim}{a_1} & \underset{\sim}{a_2} & \cdots & \underset{\sim}{a_n} \\ | & | & & | \end{bmatrix}_{m \times n}$$ and a full rank matrix

Spaces spanned by the columns of $\underset{\sim}{A}$ in succession

$$\langle \underbrace{a_1} \rangle \subseteq \langle \underbrace{\underset{\sim}{a_1}, \underset{\sim}{a_2}} \rangle \subseteq \langle \underbrace{\underset{\sim}{a_1}, \underset{\sim}{a_2}, a_3} \rangle \subseteq \cdots$$

are called successive column spaces of $\underset{\sim}{A}$

i.e $\langle a_1 \rangle$ is 1D space spanned by a_1 and $\langle a_1, a_2 \rangle$ is 2D space spanned by a_1, a_2 and so on.

The idea behind QR factorization is to successively construct orthonormal vectors $q_1, q_2, \dots q_n$ that span these successive column spaces of A as described before.

" Let $A \in \mathbb{R}^{m \times n}$ $(m \geq n)$ have full rank n, we want to find orthonormal vectors $\langle q_1, q_2, \dots q_n \rangle$ such that

$$\langle q_1, q_2, \dots q_j \rangle = \langle a_1, a_2, \dots a_j \rangle$$

for $j = 1, 2, \dots n$

$$\left.\begin{aligned} a_1 &= r_{11} q_1 \\ a_2 &= r_{12} q_1 + r_{22} q_2 \\ a_3 &= r_{13} q_1 + r_{23} q_2 + r_{33} q_3 \end{aligned}\right\}$$

$\vdots$

$$\underset{\sim}{a_n} = r_{1n}\underset{\sim}{q_1} + r_{2n}\underset{\sim}{q_2} + \dots r_{nn}\underset{\sim}{q_n}$$

$$\begin{bmatrix} | & | & | & & | \\ \underset{\sim}{a_1} & \underset{\sim}{a_2} & a_3 & \cdots & a_n \\ | & | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \underset{\sim}{q_1} & \underset{\sim}{q_2} & \cdots & \underset{\sim}{q_n} \\ | & | & & | \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}$$

$\underset{\sim}{A}$ $m \times n$ $\qquad \hat{Q}$ $m \times n$ $\qquad \hat{R}$ $n \times n$

where $r_{kk} \neq 0$ for $k = 1, \dots n$

$\underset{\sim}{A} = \underset{\sim}{\hat{Q}}\,\underset{\sim}{\hat{R}}$ is called

reduced QR factorization of $\underset{\sim}{A}$

where $\hat{Q} \in \mathbb{R}^{m \times n}$ matrix having

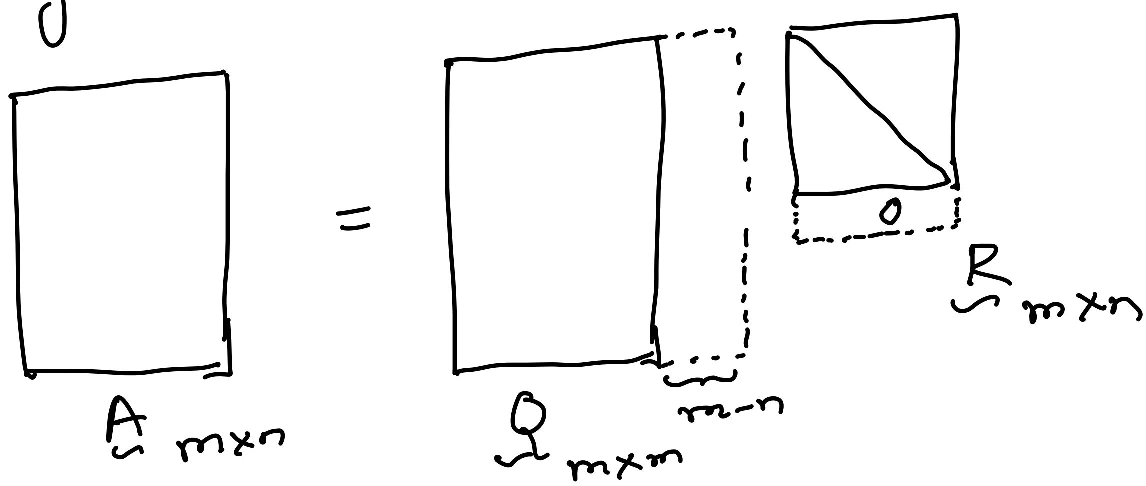
n orthonormal columns and

$\hat{R} \in \mathbb{R}^{n \times n}$ is upper triangular matrix.

Full $\underset{\sim}{Q}\,\underset{\sim}{R}$ factorization

A full QR factorization of $\underset{\sim}{A} \in \mathbb{R}^{m \times n}$

$(m \geq n)$ appends an additional $(m-n)$ orthonormal columns to $\hat{Q}$ to make it an $m \times m$ orthogonal matrix. $(m-n)$ rows of zeros are appended to $\hat{R}$ making it $m \times n$ matrix $\underset{\sim}{R}$



In full QR, the additional columns $\underset{\sim}{q}_j$ $(j > n)$ are orthogonal to range$(\underset{\sim}{A})$

If $\underset{\sim}{A}$ had full rank, these additional are orthonormal basis for range$(A)^{\perp}$ (i.e the space orthogonal to range(A) or equivalently to null(A^T))

Gram - Schmidt orthogonalization :-

Given $\underset{\sim}{a_1}, \underset{\sim}{a_2}, \ldots \underset{\sim}{a_n}$ (linearly independent vectors)

and we want to build

$\underset{\sim}{q_1}, \underset{\sim}{q_2}, \ldots \underset{\sim}{q_n}$ and entries

r_{ij} by successive orthogonalization

At step j, we want a unit-vector

$\underset{\sim}{q_j} \in \langle \underset{\sim}{a_1}, \underset{\sim}{a_2}, \ldots a_j \rangle \quad j = 1 \ldots n$

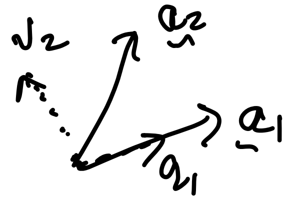
that is orthogonal to

$\langle \underset{\sim}{q_1}, \underset{\sim}{q_2}, \ldots \underset{\sim}{q_{j-1}} \rangle$

Take a vector $\underset{\sim}{a_j}$

$$\underset{\sim}{V_j} = \underset{\sim}{a_j} - (q_1^T a_j)\, \underset{\sim}{q_1} - (q_2^T a_j)\, \underset{\sim}{q_2} \cdots\cdots$$

We divide $\underset{\sim}{V_j}$ by $\|V_j\|_2$ and we get our $\underset{\sim}{q_j}$



$$\underset{\sim}{q_1} = \frac{\underset{\sim}{a_1}}{\|\underset{\sim}{a_1}\|}$$

$$\underset{\sim}{a_2} = (\underset{\sim}{q_1}^T \underset{\sim}{a_1})\, \underset{\sim}{q_1} + \underset{\sim}{V_2}$$

$$\underset{\sim}{V_2} = \underset{\sim}{a_2} - (\underset{\sim}{q_1}^T \underset{\sim}{a_2})\, \underset{\sim}{q_1}$$

$$\underset{\sim}{q_2} = \frac{\underset{\sim}{V_2}}{\|\underset{\sim}{V_2}\|}$$

$$\underset{\sim}{q_1} = \frac{\underset{\sim}{a_1}}{r_{11}} \; ; \; \underset{\sim}{q_2} = \frac{\underset{\sim}{a_2} - r_{12}\underset{\sim}{q_1}}{r_{22}} \; ;$$

$$\underset{\sim}{q_3} = \frac{\underset{\sim}{a_3} - r_{13}\underset{\sim}{q_1} - r_{23}\underset{\sim}{q_2}}{r_{33}}$$

$$\vdots$$

$$\underset{\sim}{q_n} = \frac{\underset{\sim}{a_n} - \sum_{i=1}^{n-1} r_{in}\underset{\sim}{q_i}}{r_{nn}}$$

where $r_{ij} = \underset{\sim}{q_i}^T \underset{\sim}{a_j} \quad (i \neq j)$

$$|r_{jj}| = \left\| \underset{\sim}{a_j} - \sum_{\underbrace{i=1}}^{j-1} r_{ij}\underset{\sim}{q_i} \right\|$$

Algo :-

for $j = 1$ to n

$\quad \underset{\sim}{V_j} = \underset{\sim}{a_j}$

$\quad$ for $i = 1$ to $j-1$

$\quad\quad r_{ij} = \underset{\sim}{q_i}^T \underset{\sim}{a_j}$

$\quad\quad \underset{\sim}{V_j} = \underset{\sim}{V_j} - r_{ij}\underset{\sim}{q_i}$

$r_{jj} = \|\underset{\sim}{V_j}\|_2 \; ; \; q_j = V_j / r_{jj}$

Existence and uniqueness:-

Thm:- Every matrix $\underset{\sim}{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ has a full QR factorization and hence also reduced QR factorization!

Thm:- Every $\underset{\sim}{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$ of full rank has a unique reduced QR factorization $\underset{\sim}{A} = \underset{\sim}{\hat{Q}}\underset{\sim}{\hat{R}}$ with $r_{jj} > 0$

Solution of $\underset{\sim}{A}\underset{\sim}{x} = \underset{\sim}{b}$:-

$\underset{\sim}{A} \in \mathbb{R}^{m \times m}$ is non singular

$$\underset{\sim}{A}\underset{\sim}{x} = \underset{\sim}{b}$$

Construct QR factorization of $A = QR$

$$\Rightarrow \underset{\sim}{Q}\underset{\sim}{R}\underset{\sim}{x} = \underset{\sim}{b}$$

$$\Rightarrow \underset{\sim}{R}\underset{\sim}{x} = \underbrace{\underset{\sim}{Q}^T\underset{\sim}{b}}$$

Solve $\underset{\sim}{R}\underset{\sim}{x} = y$ for $\underset{\sim}{x}$

$$\begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

GS algo is one of the algorithms for QR factorization!

$\underset{\sim}{A} \in \mathbb{R}^{m \times n} \; (m \geq n)$ has full rank!

$$\underset{\sim}{q}_1 = \frac{\underset{\sim}{a}_1}{r_{11}} \; ; \; \underset{\sim}{q}_2 = \frac{\underset{\sim}{a}_2 - r_{12}\underset{\sim}{q}_1}{r_{22}}$$

$$\underset{\sim}{q}_3 = \frac{\underset{\sim}{a}_3 - r_{13}\underset{\sim}{q}_1 - r_{23}\underset{\sim}{q}_2}{r_{33}}$$

$$\vdots$$

$$\underset{\sim}{q}_n = \frac{\underset{\sim}{a}_n - \sum_{i=1}^{n-1} r_{in}\underset{\sim}{q}_i}{r_{nn}}$$

① $\underset{\sim}{q}_1 = \dfrac{\underset{\sim}{a}_1}{r_{11}} = \dfrac{\underset{\sim}{P}_1 \underset{\sim}{a}_1}{\|\underset{\sim}{P}_1 \underset{\sim}{a}_1\|}$ where $\underset{\sim}{P}_1 = \underset{\sim}{I}$

② $$\underset{\sim}{q}_2 = \frac{\underset{\sim}{a}_2 - r_{12}\underset{\sim}{q}_1}{r_{22}} = \frac{\underset{\sim}{a}_2 - (\underset{\sim}{q}_1^T \underset{\sim}{a}_2)\underset{\sim}{q}_1}{r_{22}}$$

$$= \frac{\underset{\sim}{a}_2 - (\underset{\sim}{q}_1 \underset{\sim}{q}_1^T)\underset{\sim}{a}_2}{r_{22}}$$

$$= \frac{(\underset{\sim}{I} - \underset{\sim}{q}_1 \underset{\sim}{q}_1^T)\underset{\sim}{a}_2}{r_{22}} = \frac{\underset{\sim}{P}_2 \underset{\sim}{a}_2}{\|\underset{\sim}{P}_2 \underset{\sim}{a}_2\|}$$

where $\underset{\sim}{P}_2 = \underset{\sim}{I} - \underset{\sim}{q}_1 \underset{\sim}{q}_1^T$

(3) $$\underset{\sim}{q}_3 = \frac{\underset{\sim}{a}_3 - r_{13}\underset{\sim}{q}_1 - r_{23}\underset{\sim}{q}_2}{r_{33}}$$

$$= \frac{\underset{\sim}{a}_3 - (\underset{\sim}{q}_1\underset{\sim}{q}_1^T)\underset{\sim}{a}_3 - (\underset{\sim}{q}_2\underset{\sim}{q}_2^T)\underset{\sim}{a}_3}{r_{33}}$$

$$= \frac{(\underset{\sim}{I} - \underset{\sim}{q}_1\underset{\sim}{q}_1^T - \underset{\sim}{q}_2\underset{\sim}{q}_2^T)\underset{\sim}{a}_3}{r_{33}} = \frac{\underset{\sim}{P}_3\underset{\sim}{a}_3}{r_{33}}$$

$\vdots$

(4) $$\underset{\sim}{q}_n = \frac{\underset{\sim}{P}_n \underset{\sim}{a}_n}{\|\underset{\sim}{P}_n \underset{\sim}{a}_n\|}$$

$\underset{\sim}{P}_j = \underset{\sim}{I} - \hat{Q}_{j-1}\hat{Q}_{j-1}^T$ where

$$\hat{Q}_{j-1} = \left[\underset{\sim}{q}_1 \,\middle|\, \underset{\sim}{q}_2 \,\middle|\, \cdots \,\middle|\, \underset{\sim}{q}_{j-1}\right]$$

<u>Remark</u> :-

Each $\underset{\sim}{P}_j \in \mathbb{R}^{m\times m}$ is the matrix of rank $m-(j-1)$ that projects a vector in $\mathbb{R}^m$ onto a space orthogonal to $\langle \underset{\sim}{q_1}, \underset{\sim}{q_2}, \dots \underset{\sim}{q_{j-1}} \rangle$

Classical Gram-Schmidt is unstable numerically!

i.e loss of orthogonality occurs because of round off error accumulation!

<u>Algo for</u> CGS:-

Compute single orthogonal projection

$$\underset{\sim}{v_j} = \underset{\sim}{P_j}\, \underset{\sim}{a_j}$$

<u>Modified GS</u>:-

We get the same result by a sequence of $(j-1)$ projectors of rank $m-1$

$\underset{\sim}{P_{\perp q}}$ denotes $(m-1)$ rank orthogonal projector onto space orthogonal to non-zero vector $\underset{\sim}{q} \in \mathbb{R}^m$

$$\boxed{\underset{\sim}{v}_j = \underset{\sim}{P}_j \underset{\sim}{a}_j} \quad \text{— (1) (CGS)}$$

$$\underset{\sim}{P}_j = P_{\perp q_{j-1}} \cdots P_{\perp q_3} P_{\perp q_2} P_{\perp q_1}$$

$$\boxed{\underset{\sim}{v}_j = P_{\perp q_{j-1}} \cdots P_{\perp q_3} P_{\perp q_2} P_{\perp q_1} \underset{\sim}{a}_j} \quad \text{— (2) (MGS)}$$

(1) and (2) are equivalent mathematically but sequence of arithmetic operations are different!

$$P_3 = I - q_1 q_1^T - q_2 q_2^T$$

$$= P_{\perp q_2} P_{\perp q_1}$$

$$(I - q_2 q_2^T)(I - q_1 q_1^T)$$

MGS Algo

for $j = 1$ to n

$$\underset{\sim}{v}_j^{(1)} = \underset{\sim}{a}_j$$

$$\underset{\sim}{v}_j^{(2)} = P_{\perp q_1} \underset{\sim}{a}_j = \underset{\sim}{v}_j - q_1 q_1^T v_j^{(1)}$$

$$v_j^{(3)} = P_{\perp q_2} v_j^{(2)} = v_j^{(2)} - q_2 q_2^T v_j^{(2)}$$

$$\vdots$$

$$v_j = v_j^{(j)} = P_{\perp q_{j-1}} v_j^{(j-1)}$$

$$v_j = v_j^{(j-1)} - q_{j-1} q_{j-1}^T v_j^{(j-1)}$$

In finite precision arithmetic, you can show MGS algo introduces smaller errors than CGS algo!

$$q_1 = \frac{a_1}{r_{11}} ; \quad q_2 = \frac{a_2 - (q_1^T a_2) q_1}{r_{22}} ; \quad q_3 = \left(a_3 - (q_1^T a_3) q_1 - (q_2^T a_3) q_2\right) / r_{33}$$

$$q_1^T q_2 = \delta$$

$$q_1^T q_3 = \frac{1}{r_{33}} \left[\cancel{q_1^T a_3} - \cancel{q_1^T a_3} - (q_2^T a_3)\underbrace{(q_1^T q_2)}_{\delta} \right]$$

$$= \frac{-1}{r_{33}} q_2^T a_3 \delta$$

$$q_2^T q_3 = -\frac{(q_1^T a_3)\delta}{r_{33}}$$

$$q_4 = \left[a_4 - (q_1^T a_4) q_1 - (q_2^T a_4) q_2 - (q_3^T a_4) q_3 \right] \frac{1}{r_{44}}$$

$$q_1^T q_4 = \left[-(q_2^T a_4)\delta + \frac{(q_3^T a_4)(q_2^T a_3)\delta}{r_{33}} \right] \frac{1}{r_{44}} .$$

$$q_2^T q_4 = \left[-(q_1^T a_4)\delta + \frac{(q_3^T a_4)(q_1^T a_3)\delta}{r_{33}} \right] \frac{1}{r_{44}} ,$$

$$q_3^T q_4 = \left[\frac{(q_1^T a_4)(q_2^T a_3)\delta}{r_{33}} + \frac{(q_1^T a_3)(q_2^T a_4)\delta}{r_{33}} \right] \frac{1}{r_{44}}$$

<u>In MGS :-</u>

$$v_3^{(3)} = v_3$$

$$v_3^{(3)} = v_3^{(2)} - q_2 q_2^T v_3^{(2)}$$

$$= v_3^{(2)} - \left(q_2^T v_3^{(2)}\right) q_2$$

$$q_2^T v_3 = q_2^T v_3^{(2)} - q_2^T v_3^{(2)} = 0 \quad \checkmark$$

$$\underbrace{q_1^T v_3} = \underbrace{q_1^T}\left[v_3^{(2)} - \left(q_2^T v_3^{(2)}\right) q_2 \right]$$

$$= \underbrace{q_1^T v_3^{(2)}} - \underbrace{q_2^T v_3^{(2)}}\delta$$

$$q_1^T v_3^{(2)} = q_1^T \left[v_3^{(1)} - q_1 q_1^T v_3^{(1)} \right] = 0$$

$$q_2^T v_3^{(2)} = q_2^T \left[v_3^{(1)} - q_1 q_1^T v_3^{(1)} \right]$$

$$= \underbrace{q_2^T v_3^{(1)} - q_1^T v_3^{(1)}\delta}$$

$$q_1^T v_3 = -\left[q_2^T v_3^{(1)} - q_1 v_3^{(1)}\delta \right]\delta$$

$$= -q_2^T v_3^{(1)} \delta + O(\delta^2)$$

$q_1^T v_3$ is no worse than CGS but no error in $q_2^T v_3$

$$v_4 = v_4^{(4)} = v_4^{(3)} - q_3 q_3^T v_4^{(3)}$$

$$q_3^T v_4 = 0 \; ; \; q_2^T v_4 = 0 \; ; \; q_1^T v_4 = \underset{O(\delta)}{(\quad)}!$$

Exercise

<u>Operation count of G.S:-</u>

Any of addition, subtraction, multiplication are counted as 1 flop

<u>Thm</u>:- G.S orthogonalization requires $\sim 2mn^2$ flops to compute QR factorization!

<u>Remark</u>:- Symbol $\sim$ has the meaning representing of asymptotic complexity

i.e $$\lim_{m,n\to\infty} \frac{\text{number of flops}}{2mn^2} = 1$$

for $j = 1$ to n

$\quad \underset{\sim}{v}_j = \underset{\sim}{a}_j$

$\quad$ for $i = 1$ to $j-1$

$\quad\quad r_{ij} = \underset{\sim}{q}_i^T \underset{\sim}{a}_j$ ✓

$\quad\quad \underset{\sim}{v}_j = \underset{\sim}{v}_j - r_{ij} \underset{\sim}{q}_i$ ✓

$\quad r_{jj} = \| \underset{\sim}{v}_j \|_2$

$\quad \underset{\sim}{q}_j = \dfrac{\underset{\sim}{v}_j}{r_{jj}}$

$r_{ij} = \underset{\sim}{q}_i^T \underset{\sim}{a}_j$

m multiplications + $(m-1)$ additions

$\underset{\sim}{v}_j = \underset{\sim}{v}_j - r_{ij} \underset{\sim}{q}_i$

m multiplications + m subtractions

Total work $\sim 4m$ flops

Total flops $\sim \displaystyle\sum_{j=1}^{n} \sum_{i=1}^{j-1} 4m$

$$\sim \sum_{j=1}^{n} (j-1)\, 4m$$

$$\sim \left(\frac{n(n+1)}{2} - 1 \right) 4m$$

$$\sim 2mn^2$$

Householder triangularization :-

When it comes to produce orthogonality

how close $Q^T Q$ is close to identity

$O(K(A)^2 \varepsilon_M)$ $\underbrace{\| \underset{\sim}{I} - \tilde{Q}^T \tilde{Q} \|}$

ⓐ CGS → usually poor orthogonality

$O(K(A)\varepsilon_M)$

ⓑ MGS → Dependeng on condition number of $\underset{\sim}{A}$ $K(\underset{\sim}{A})$

$O(\varepsilon_M)$

ⓒ Householder → good orthogonality better than MGS ~~but dependent on $K(\underset{\sim}{A})$~~

→ Householder orthogonalization are popular for sequential dense matrices

→ Householder transformation has its application in algorithms for solving eigenvalue problem!

Key idea in Householder method is to apply succession of elementary orthogonal matrices Q_K to the left of $A \in \mathbb{R}^{m\times n}$ so that resulting matrix is upper triangular.

$$Q_n \cdots Q_3 Q_2 Q_1 A = R \quad \text{upper triangular}$$

The product $Q = Q_1^T Q_2^T \cdots Q_n^T$ is orthogonal

$A = QR$ is full QR factorization of A

$$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}_{5\times 3} \xrightarrow{Q_1} \underbrace{\begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix}}_{Q_1 A}$$

Orthogonal triangularization!

$$Q_1 A \xrightarrow{Q_2} \underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}}_{Q_2 Q_1 A} \xrightarrow{Q_3} \underbrace{\begin{bmatrix} \times & \times & * \\ & \times & * \\ & & * \\ & & 0 \\ & & 0 \end{bmatrix}}_{Q_3 Q_2 Q_1 A}$$

Beginning of step 1, there are no zeros

" of step 2, block of zeros in first column

" of step 3, block of zeros in first 2 columns

⋮

Step k, block of zeros in first (k-1) columns

After n steps, all entries below diagonal are zero!

How to construct $\underset{\sim}{Q}_k$?

Each $\underset{\sim}{Q}_K$ is chosen to be orthogonal matrix of the form

$$\underset{\sim}{Q}_K = \begin{bmatrix} \underset{\sim}{I}_{(k-1\times k-1)} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{F}_{(m-k+1)\times(m-k+1)} \end{bmatrix}$$

$\underset{\sim}{I}_{(k-1)\times(k-1)} \rightarrow$ Identity matrix

$\underset{\sim}{F} \rightarrow (m-k+1)\times(m-k+1)$ orthogonal matrix

$\underset{\sim}{F}$ is chosen to be Householder reflector!

In our example before

$$\underset{\sim}{Q}_1 = \begin{bmatrix} \underset{\sim}{F} \end{bmatrix}_{5\times5}$$

$$\underset{\sim}{Q}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \underset{\sim}{F}_{4\times4} \end{bmatrix}_{5\times5}$$

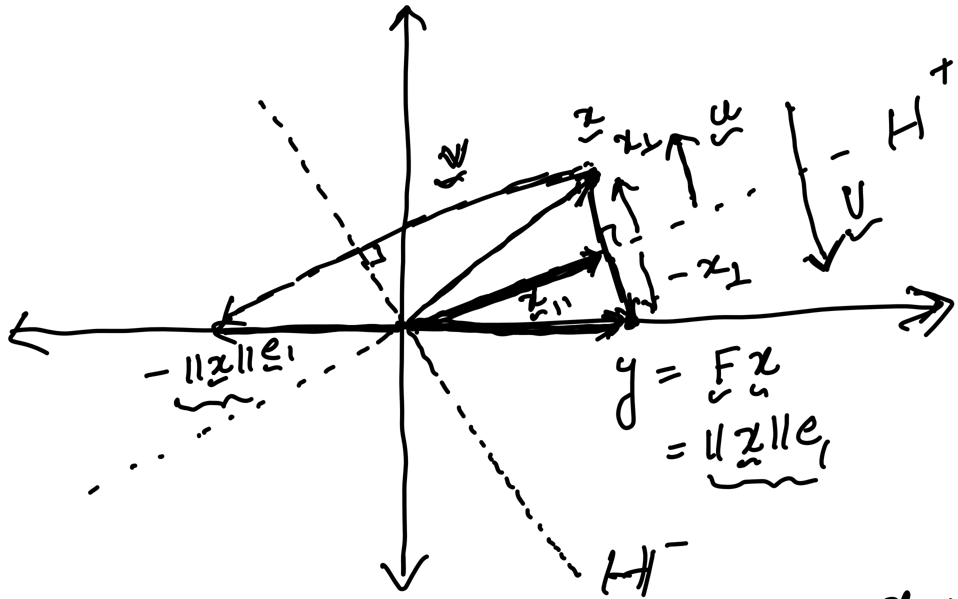
Suppose at beginning of step k the entries in the k^{th} column $(k+1, \dots m)$ has to be zeroed

Let the entries $k, \dots m$ for a vector $\underset{\sim}{x} \in \mathbb{R}^{m-k+1}$

$$\underset{\sim}{x} = \begin{bmatrix} x \\ x \\ \vdots \\ x \end{bmatrix}_{m-k+1} \xrightarrow{\underset{\sim}{F}} \underset{\sim}{F}\underset{\sim}{x} = \begin{bmatrix} * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \|\underset{\sim}{x}\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

(Recall $\underset{\sim}{F}$ is orthogonal matrix so it cannot change norm) $\|\underset{\sim}{x}\| \underset{\sim}{e}_1$



$$x = x_{\parallel} + x_{\perp}$$

$$x + v = \|x\| e_1$$

$$\boxed{v = \|x\| e_1 - x}$$

v = reflection vector

$$x_{\perp} = u u^T x$$

$$x_{\parallel} - x_{\perp} = y$$

$$y = (x_{\parallel} + x_{\perp}) - 2 x_{\perp}$$

$$= x - 2 u u^T x$$

$$u = -\frac{v}{\|v\|}$$

$$y = F x = (I - 2 u u^T) x$$

$$\boxed{F = I - 2 u u^T} \Rightarrow \boxed{F = I - 2 \frac{v v^T}{\|v\|^2}}$$

Infact we can also construct F such that

$$F x = -\|x\| e_1$$

$$x + v = -\|x\| e_1$$

$$v = -\|x\| e_1 - x$$

Which reflections do we pick?

We better pick the one that is not too close to $\underset{\sim}{x}$

Let us say $\underset{\sim}{x}$ is too close to $\|\underset{\sim}{x}\|\underset{\sim}{e}_1$ i.e angle b/w H^T and $\underset{\sim}{e}_1$ axis is very small, then we would incur lot of round off error in computing $\underset{\sim}{v} = \|\underset{\sim}{x}\|\underset{\sim}{e}_1 - \underset{\sim}{x}$

If θ is angle between $\underset{\sim}{x}$ and $\underset{\sim}{y}$

$$\cos\theta = \frac{\underset{\sim}{x} \cdot \|\underset{\sim}{x}\|\underset{\sim}{e}_1}{\|x\|\,\|\underset{\sim}{x}\|} = \frac{x_1}{\|x\|}$$

Let us say $x_1 > 0$ i.e $\operatorname{sgn}(x_1) = +1$

then $\underset{\sim}{v} = -\|\underset{\sim}{x}\|\underset{\sim}{e}_1 - \underset{\sim}{x}$

$x_1 < 0$ i.e $\operatorname{sgn}(x_1) = -1$

$\underset{\sim}{v} = \|\underset{\sim}{x}\|\underset{\sim}{e}_1 - \underset{\sim}{x}$

$$\boxed{\underset{\sim}{v} = -\operatorname{sgn}(x_1)\|\underset{\sim}{x}\|\underset{\sim}{e}_1 - \underset{\sim}{x}}$$

$$\boxed{\underset{\sim}{v} = \operatorname{sgn}(x_1)\|\underset{\sim}{x}\|\underset{\sim}{e}_1 + \underset{\sim}{x}}$$

for $k = 1 : n$

$$\underset{\sim}{x} = A(k:m, k)$$

$$\underset{\sim}{v}_k = \text{sgn}(x_1)\|\underset{\sim}{x}\|\underset{\sim}{e}_1 + \underset{\sim}{x}$$

$$\underset{\sim}{v}_k = \frac{\underset{\sim}{v}_k}{\|v_k\|_2}$$

$$A(k:m, k:n) = A(k:m, k:n) - 2\underset{\sim}{v}_k\underset{\sim}{v}_k^T A(k:m, k:n)$$

This algo reduces $\underset{\sim}{A}$ to upper triangular form, the $\underset{\sim}{R}$ in $\underset{\sim}{Q}\underset{\sim}{R}$ factorization

$$\underset{\sim}{Q}^T = \underset{\sim}{Q}_n \underset{\sim}{Q}_{n-1} \cdots \underset{\sim}{Q}_2 \underset{\sim}{Q}_1$$

$$\underset{\sim}{A}\underset{\sim}{x} = \underset{\sim}{b}$$

$$\underset{\sim}{Q}\underset{\sim}{R}\underset{\sim}{x} = b$$

$$\Rightarrow \underset{\sim}{R}\underset{\sim}{x} = \underset{\sim}{Q}^T\underset{\sim}{b} \quad \text{(Back substitution)}$$

<u>Algo for $\underset{\sim}{Q}^T \underset{\sim}{b}$</u>

for $k = 1 : n$

$$b(k:m) = b(k:m) - 2 v_k v_k^T b(k:m)$$

Computational flops $\sim (2mn^2 - \frac{2}{3}n^3)$ flops!

Stability of Householder triangularization :-

Householder triangularization is backward stable for all matrices A :

$\tilde{Q}\tilde{R} = A + \delta A$ where δA is small

i.e $\frac{\|\delta A\|}{\|A\|} = O(\epsilon_m)$

i.e $\tilde{Q}\tilde{R}$ is exact QR factorization of the matrix which is perturbed by small $\delta A \in \mathbb{R}^{m\times n}$

$\tilde{R}$ is upper triangular matrix constructed by Householder triangularization in floating point arithmetic. (FPA)

And define

$\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_n$ where $\tilde{Q}_k$ define exactly orthogonal matrix defined by $\tilde{v}_k$ which are obtained by FPA.

* Householder triangularization is backward stable but not always forward accurate, this means $\tilde{Q}$ and Q may have large error depending on conditioning of matrix $\underset{\sim}{A}$.

* Is the accuracy of product of $\underset{\sim}{Q}\underset{\sim}{R}$ enough for application or do we need accuracy of $\underset{\sim}{Q}$ and $\underset{\sim}{R}$ individually! It turns out accuracy of product of $\underset{\sim}{Q}\underset{\sim}{R}$ is enough as seen below!

Algo solve $\underset{\sim}{A}\underset{\sim}{x} = \underset{\sim}{b}$ by QR factorization

(i) $\tilde{Q}\tilde{R} = \underset{\sim}{A}$ with $\tilde{Q}$ as represented as product of reflectors

(ii) $\tilde{y} = \underset{\sim}{\tilde{Q}}^T \underset{\sim}{b}$ Construct $\tilde{Q}^T \underset{\sim}{b}$ without explicitly building $\tilde{Q}$!

(iii) $\underset{\sim}{R}\underset{\sim}{\tilde{x}} = \underset{\sim}{\tilde{y}}$ (Solve triangular s/m of eqns)!

Algo is backward stable i.e

$\tilde{x}$ is a solution $(A + \delta A)\tilde{x} = b$

for some $\frac{\|\delta A\|}{\|A\|} = O(\epsilon_M)$

because

(i) $\tilde{Q}\tilde{R} = A$ is backward stable

(ii) $\tilde{y} = \tilde{Q}^T b$ compute $\tilde{y}$ satisfies

$(\tilde{Q} + \delta Q)\tilde{y} = b$

for some $\|\delta Q\| = O(\epsilon_M)$

(iii) Similarly

backward substitution

is backward stable

compute $\tilde{x}$ satisfies $(\tilde{R} + \delta R)\tilde{x} = \tilde{y}$

for δR satisfying

$\frac{\|\delta R\|}{\|R\|} = O(\epsilon_M)$