Lecture 10: Householder triangularization

Householder triangularization is a method to compute QR factorization.

This method is more stable than Gram-Schmidt (both classical and modified).

We saw that the Gram-Schmidt algorithm is a method of triangular orthogonalization, that is

$$A \underbrace{R_1 R_2 \cdots R_n}_{\hat{R}^{-1}} = \hat{Q}.$$

On the other hand, the Householder method applies a succession of unitary matrices Q_k on the left of A

$$\underbrace{Q_n \cdots Q_2 Q_1}_{Q^*} A = R$$

so that the result is upper triangular. The product $Q = Q_1^* \cdots Q_n^*$ is also unitary. Hence, $A = QR$ is the required QR factorization. So, we call this orthogonal triangularization.

Finding the unitary matrices Q_k

We let A be a 5×3 matrix. We will find Q_1, Q_2, Q_3 so that $Q_3 Q_2 Q_1 A$ is upper triangular.

$$
\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}}
\xrightarrow{Q_1}
\underset{Q_1 A}{\begin{bmatrix} \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}}
\xrightarrow{Q_2}
\underset{Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \end{bmatrix}}
\xrightarrow{Q_3}
\underset{Q_3 Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \mathbf{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}}
$$

- the matrix Q_1 is defined to make the entries below the first diagonal element zero.
- the matrix Q_2 is defined to make the entries below the second diagonal element zero, and it does not change the first column.
- the matrix Q_3 is defined to make the entries below the thrid diagonal element zero, and it does not change the first and the second columns.

In general, the matrix Q_k is defined to make the entries below the k-th diagonal element zero, and it does not change the first $k - 1$ columns.

We present the construction of Q_k.

Recall that, the matrix Q_k is defined to make the entries below the k-th diagonal element zero, and it does not change the first $k-1$ columns.

So, Q_k is chosen to have the form

$$\begin{pmatrix} I & 0 \\ 0 & F \end{pmatrix}$$

where I is the $(k-1) \times (k-1)$ identity matrix, and F is an $(m-k+1) \times (m-k+1)$ unitary matrix. A requirement for F is that it will create zero below the k-th diagonal element.

The matrix F will be chosen as the Householder reflector, which is defined next.

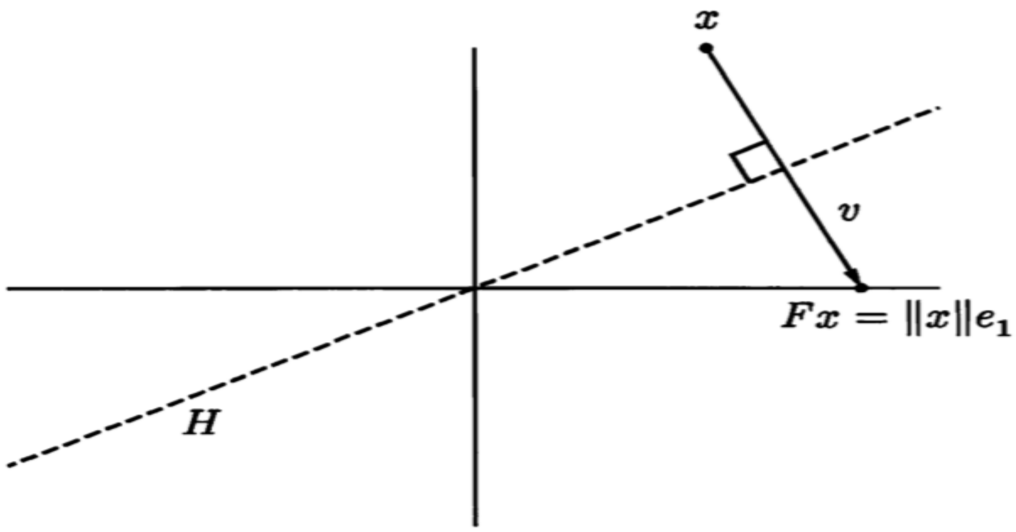
Householder reflector

We let $x \in \mathbb{C}^{m-k+1}$ be a vector consisting of the entries $k, \cdots, m$ of the k-th column.

The Householder reflector F will perform the following

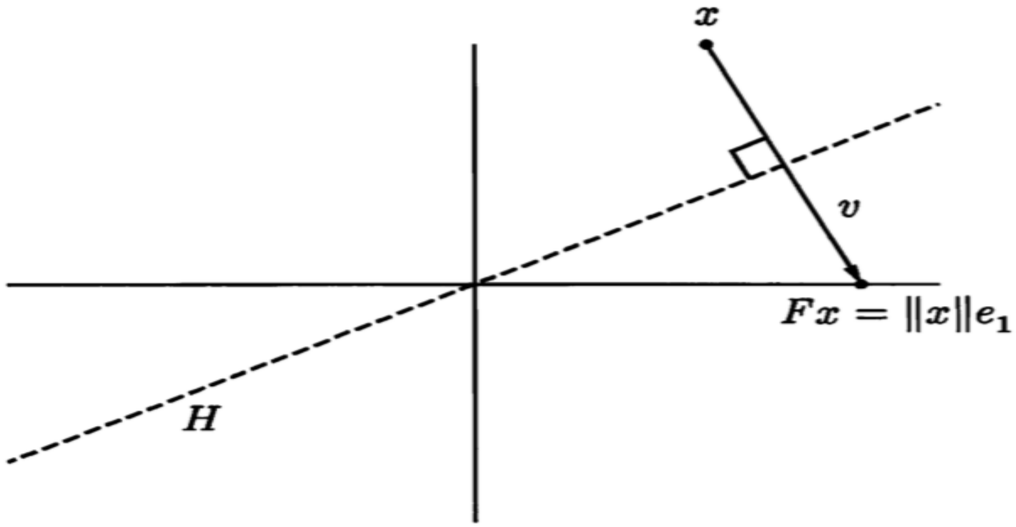
$$x = \begin{bmatrix} \times \\ \times \\ \times \\ \vdots \\ \times \end{bmatrix} \quad \xrightarrow{F} \quad Fx = \begin{bmatrix} \|x\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1$$

The action of F can be described as follows:



It is a reflection across a plane H with normal vector v.

It is a reflection across a plane H with normal vector v.



Note that, the vector v is given by

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

Recall that the orthogonal projector P defined by

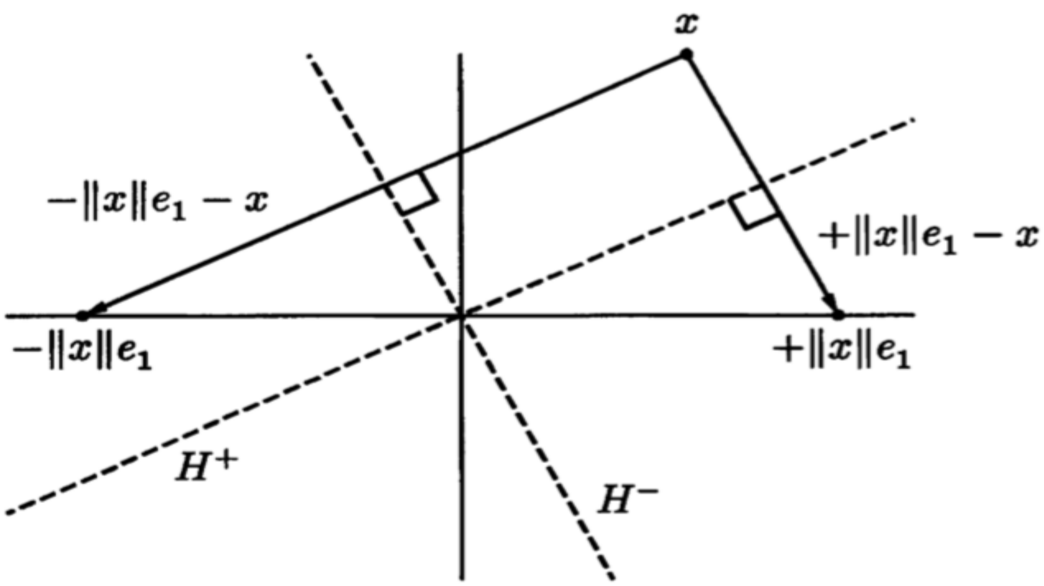
$$P = I - \frac{vv^*}{v^* v}$$

projects vectors to the plane orthogonal to v. We need to go twice in the direction in order to get our desired result. So, we need

$$F = I - 2\frac{vv^*}{v^* v}$$

Two possibilities

We defined F^+ so that $F^+ x = \|x\| e_1$. We can also define F^- so that $F^- x = -\|x\| e_1$.



These two choices are based on the choice of v:

$$v^{\pm} = \pm\|x\| e_1 - x$$

A practical choice is

$$v = -\text{sign}(x_1)\|x\| e_1 - x \qquad (\text{or } v = \text{sign}(x_1)\|x\| e_1 + x)$$

so that the reflected point is not too close to the original point.

The algorithm

We use MATLAB notations: Let A be a matrix.

- $A_{i:i',j:j'}$ is the $(i'-i+1)\times(j'-j+1)$ sub-matrix of A with upper-left corner a_{ij} and lower-right corner $a_{i',j'}$
- $A_{i,j:j'}$ is the row vector starting from a_{ij} to $a_{i,j'}$
- $A_{i:i',j}$ is the column vector starting from a_{ij} to $a_{i',j}$

Let $A \in \mathbb{C}^{m\times n}$ with $m \geq n$.

Algorithm 10.1. Householder QR Factorization

for $k = 1$ **to** n

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

This algorithms give the full QR factorization of A, with the matrix R overwrites A. It also generates the reflection vectors $v_1, \cdots, v_n$.

Applying or forming Q

The above algorithms gives the vectors $v_1, \cdots, v_n$, but not the matrices $Q_1, \cdots, Q_n$.

From the algorithm, we obtain R where $A = QR$, and we know that

$$Q^* = Q_n \cdots Q_2 Q_1 \quad \text{or} \quad Q = Q_1 Q_2 \cdots Q_n$$

where we recall that Q_k are orthogonal projections.

Most cases, there is no need to form Q. Only need to compute $Q^* b$ or Qx.

Algorithm 10.2. Implicit Calculation of a Product Q^*b

for $k = 1$ **to** n

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

Algorithm 10.3. Implicit Calculation of a Product Qx

for $k = n$ **downto** 1

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

If we need the matrix Q, then we compute $Qe_1, \cdots Qe_m$.

Operation count

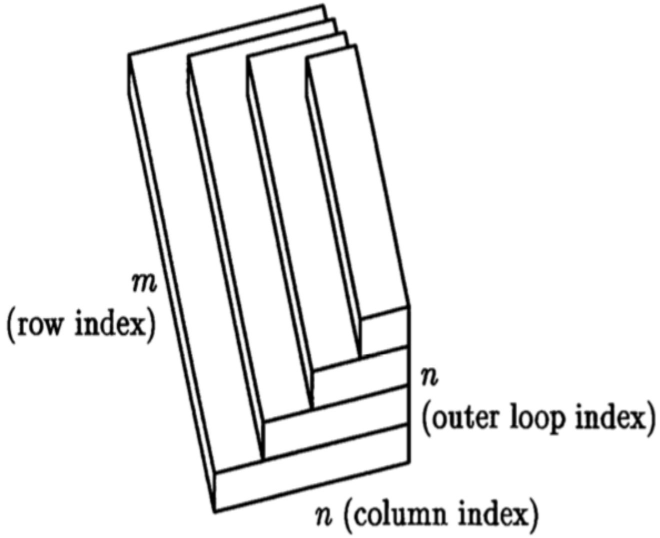
The work in the Householder QR factorization (Algorithm 10.1) is dominated by

$$A_{k:m,j} - 2v_k(v_k^* A_{k:m,j})$$

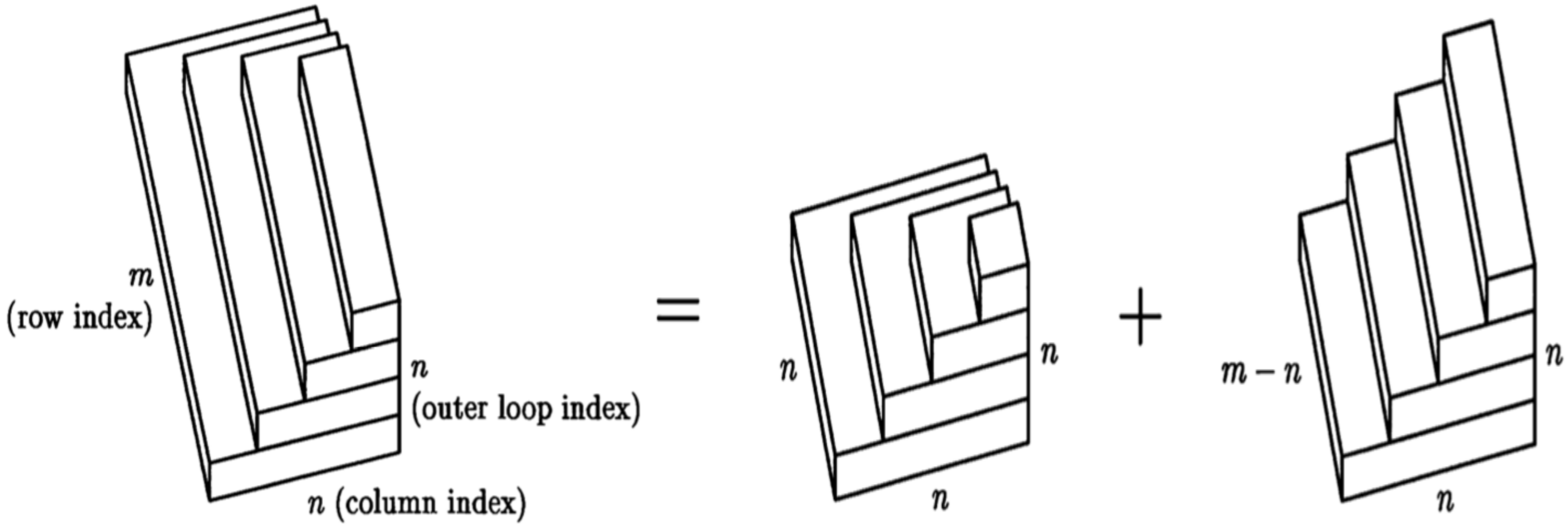
Let $\ell = m - k + 1$ be the length of the vector. This step needs $4\ell - 1 \approx 4\ell$ scalar operations:

- ℓ for the subtraction
- ℓ for the scalar multiplication
- $2\ell - 1$ for the dot product

Thus, we need ≈ 4 operations for each entry.
The following figure shows the number of entries we need to handle:



We can separate this as follows:



- for the first term, the number is $1^2 + 2^2 + \cdots + n^2 = n(n+1)(2n+1)/6 \approx n^3/3$ when n is large
- for the second term, the number is $(m-n)(n + \cdots + 2 + 1) \approx (m-n)n^2/2$ when n is large

Hence the total operations is

$$4\left(\frac{1}{2}mn^2 - \frac{1}{6}n^3\right) = 2mn^2 - \frac{2}{3}n^3$$

Lecture 11: Least squares problems

Consider the system $Ax = b$, with $A \in \mathbb{C}^{m \times n}$ and $m > n$.

In general, it has no solution.

Least squares problem: find $x \in \mathbb{C}^n$ such that

$$\|b - Ax\|_2$$

is minimized.

Data fitting

Given the data points $(x_1, y_1), \cdots, (x_m, y_m)$, we find a polynomial of degree $n-1$

$$p(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

that fits the data, where we assume $m > n$. That is, we find $p(x)$ so that

$$p(x_i) \approx y_i, \qquad i = 1, 2, \cdots, m$$

Thus, we are solving the following system

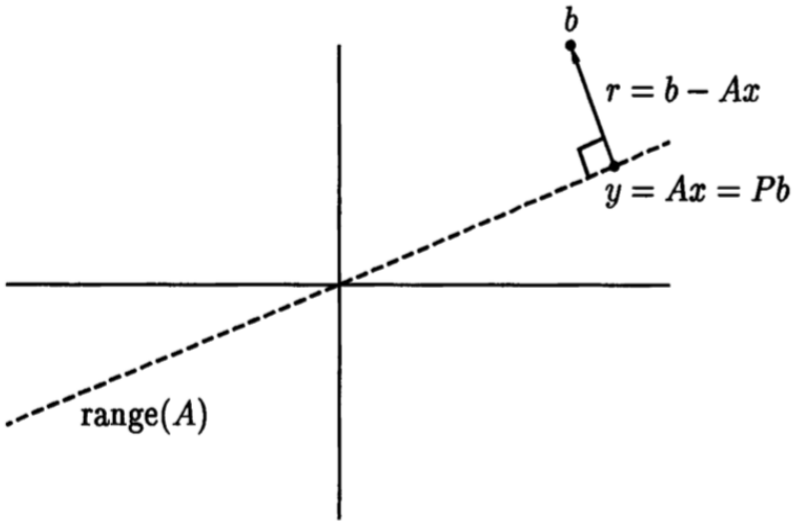
$$\begin{bmatrix} 1 & x_1 & & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ 1 & x_3 & & x_3^{n-1} \\ & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

Least squares problem: We find $p(x)$ by minimizing the squares of the derivations

$$\sum_{i=1}^{m} |p(x_i) - y_i|^2$$

Normal equation

We find x so that the residual $r = b - Ax$ is as small as possible.



Points to note:

- we need a point $y = Pb$, where Pb is the orthogonal projection of b
- then, we can find x such that $Ax = Pb$
- P is the orthogonal projection onto the range(A)
- the residual vector $r = b - Ax$ is orthogonal to range(A)

These points give the following theorem.

Theorem: Let $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$. A vector $x \in \mathbb{C}^n$ minimizes the residual $\|r\|_2 = \|b - Ax\|_2$ if and only if

$$r \perp \text{range}(A), \qquad \text{that is} \qquad A^* r = 0$$

If A is full rank if and only if the solution x is unique.

Points to note:

- The above condition is equivalent to

$$Pb = Ax$$

- The above condition is equivalent to

$$A^*Ax = A^*b \qquad \text{(called the normal equation)}$$

Next, we consider the minimum residual property. Let $y = Pb$ and it minimizes $\|b - y\|_2$. Let z be another point in range(A) and $z \neq y$. Then $(z - y) \perp (b - y)$.
So,

$$\|b - z\|_2^2 = \|(b - y) + (y - z)\|_2^2 = \|b - y\|_2^2 + \|y - z\|_2^2 > \|b - y\|_2^2$$

This proves the required residual minimization property.

Uniqueness:

- if not unique, there is $x \neq 0$ such that $A^*Ax = 0$. This implies $x^*A^*Ax = 0$ and $Ax = 0$
- if A not full rank, there is $x \neq 0$ such that $Ax = 0$, implying $A^*Ax = 0$

Pseudoinverse

If A has full rank, then the solution x to the least squares problem is

$$x = (A^*A)^{-1}A^*b$$

We call the matrix

$$A^+ = (A^*A)^{-1}A^*$$

the pseudoinverse of the matrix A.

Three methods

Least squares problem: find $x \in \mathbb{C}^n$ such that

$$\|b - Ax\|_2$$

is minimized.

- Normal equation. We solve

$$A^* Ax = A^* b$$

(not stable in the presence of rounding errors)

- QR factorization. Construct $A = \hat{Q}\hat{R}$. The orthogonal projector P is given by

$$P = \hat{Q}\hat{Q}^*$$

Then $y = Pb = \hat{Q}\hat{Q}^* b$. And

$$Ax = Pb \qquad \to \qquad \hat{Q}\hat{R}x = \hat{Q}\hat{Q}^* b$$

(good when A is full rank)

- SVD. Construct $A = \hat{U}\hat{\Sigma}V^*$. The orthogonal projector P is given by

$$P = \hat{U}\hat{U}^*$$

Then

$$Ax = Pb \qquad \to \qquad \hat{U}\hat{\Sigma}V^* x = \hat{U}\hat{U}^* b$$

(good when A is rank-deficient)