Lecture 10 - *Householder Triangularization*

OBJECTIVE:
The Householder algorithm is a process of *orthogonal triangularization*
i.e., it reduces a matrix to triangular form by a sequence of orthogonal matrix operations.
It is more stable than Gram-Schmidt, but it cannot be used in an iterative fashion.

◇ HOUSEHOLDER AND GRAM-SCHMIDT

Recalling Lecture 8, the Gram-Schmidt iteration applies a succession of elementary triangular matrices $\mathbf{R}_k$ to the right of $\mathbf{A}$ so that the resulting matrix has orthonormal columns:

$$\mathbf{A}\mathbf{R}_1\mathbf{R}_2 \ldots \mathbf{R}_n = \hat{\mathbf{Q}}$$

where

$$\hat{\mathbf{R}}^{-1} = \mathbf{R}_1\mathbf{R}_2 \ldots \mathbf{R}_n.$$

Because each $\mathbf{R}_k$ is upper-triangular, so is

$$\hat{\mathbf{R}} = \mathbf{R}_n^{-1}\mathbf{R}_{n-1}^{-1}\dots\mathbf{R}_1^{-1}.$$

Thus, $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ is a reduced QR factorization of $\mathbf{A}$.

In contrast, the Householder method applies a succession of elementary orthogonal matrices $\mathbf{Q}_k$ on the left of $\mathbf{A}$, so that the resulting matrix is upper-triangular:

$$\mathbf{Q}_n\mathbf{Q}_{n-1}\dots\mathbf{Q}_1\mathbf{A} = \mathbf{R}$$

where

$$\mathbf{Q}^T = \mathbf{Q}_n\mathbf{Q}_{n-1}\dots\mathbf{Q}_1.$$

Because each $\mathbf{Q}_k$ is orthogonal, so is

$$\mathbf{Q} = \mathbf{Q}_1^T\mathbf{Q}_2^T\dots\mathbf{Q}_n^T.$$

Thus, $\mathbf{A} = \mathbf{QR}$ is a full QR factorization of $\mathbf{A}$.

So, Gram-Schmidt $=$ triangular orthogonalization
Householder $=$ orthogonal triangularization

$\diamond$ TRIANGULARIZATION BY INTRODUCING ZEROS

The Householder method was first proposed by Alston Householder in 1958.

It is an ingenious way to systematically design orthogonal matrices $\mathbf{Q}_k$ such that

$$\mathbf{Q}_n\mathbf{Q}_{n-1}\dots\mathbf{Q}_1\mathbf{A}$$

is upper-triangular.

IDEA: Each matrix $\mathbf{Q}_k$ is designed to introduce zeros below the diagonal in column k while preserving all zeros previously introduced.

Here is a 5×3 example:

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{\mathbf{Q}_1} \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \\ 0 & * & * \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & \times & \times \\ & \times & \times \\ & \times & \times \end{bmatrix} \xrightarrow{\mathbf{Q}_2} \begin{bmatrix} \times & \times & \times \\ & * & * \\ & 0 & * \\ & 0 & * \\ & 0 & * \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \\ & & \times \\ & & \times \end{bmatrix} \xrightarrow{\mathbf{Q}_3} \begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & * \\ & & 0 \\ & & 0 \end{bmatrix}$$

In general, $\mathbf{Q}_k$ operates on rows k to m.

At the beginning of step k, there is a block of zeros in the first $k-1$ columns of these rows.

$\mathbf{Q}_k$ forms linear combinations of these rows, and thus zeros are undisturbed (a linear combination of zeros is still zero!).

After all n steps, all entries below the diagonal are zero.

Thus,

$$\mathbf{Q}_n \mathbf{Q}_{n-1} \dots \mathbf{Q}_1 \mathbf{A} = \mathbf{R}$$

is upper-triangular.

◇ HOUSEHOLDER REFLECTORS

We now investigate the construction of the $\mathbf{Q}_k$.

The standard approach is as follows:

Each $\mathbf{Q}_k$ is chosen to be an orthogonal matrix of the form

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix}$$

where $\mathbf{I}$ is the $(k-1) \times (k-1)$ identity matrix and $\mathbf{F}$ is an $(m-k+1) \times (m-k+1)$ orthogonal matrix.

Multiplication by $\mathbf{F}$ must introduce zeros in column k.

The Householder algorithm chooses $\mathbf{F}$ to be a *Householder reflector*.

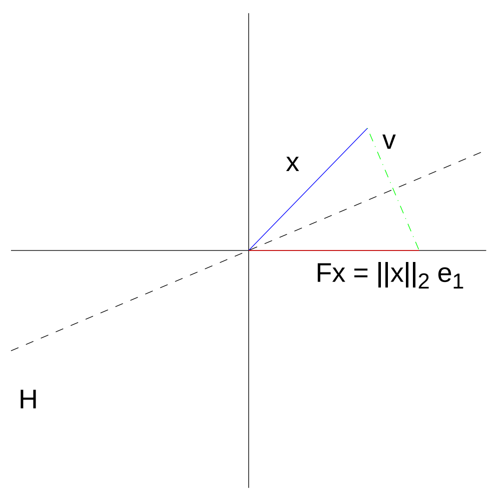
Suppose at the beginning of step k, the (potentially) nonzero entries to be zeroed are called $\mathbf{x} \in \mathbb{R}^{m-k+1}$.

The Householder reflector $\mathbf{F}$ should then essentially perform the following operation

$$\mathbf{x} = \begin{bmatrix} \times \\ \times \\ \vdots \\ \times \end{bmatrix} \qquad \xrightarrow{\mathbf{F}} \qquad \mathbf{Fx} = \begin{bmatrix} \|\mathbf{x}\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|\mathbf{x}\|\mathbf{e}_1$$

(Recall, $\mathbf{F}$ is orthogonal, so it cannot change the norm of $\mathbf{x}$!)

Geometrically the idea is given in the following diagram:



$\mathbf{F}$ reflects $\mathbf{x}$ across the hyperplane $\mathbf{H}$ orthogonal to

$$\mathbf{v} = \|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$$

(A *hyperplane* is just an ordinary "plane" in a higher dimension.)

e.g., A plane is a two-dimensional subspace in three dimensions.

So, a hyperplane could be a 3D subspace in 4D, etc.

It is a subspace of one less dimension than the space it is in.

It can be said to have *codimension* 1.

In general, a hyperplane can be characterized as the set of points orthogonal to some nonzero vector $\mathbf{v}$ (see figure).

H maps every point on one side to its mirror image on the other side.

In particular, $\mathbf{x}$ is mapped to $\|\mathbf{x}\|\mathbf{e}_1$.

The formula for $\mathbf{H}$ can be derived as follows:

We have seen that for any $\mathbf{y} \in \mathbb{R}^m$,

$$\begin{aligned} \mathbf{P}\mathbf{y} &= \left(\mathbf{I} - \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{y} \\ &= \mathbf{y} - \mathbf{v}\left(\frac{\mathbf{v}^T\mathbf{y}}{\mathbf{v}^T\mathbf{v}}\right) \end{aligned}$$

is the orthogonal projection of $\mathbf{y}$ onto $\mathbf{H}$.

To reflect $\mathbf{y}$ across $\mathbf{H}$, we need to go twice as far!

$$\begin{aligned} \therefore \qquad \mathbf{F}\mathbf{y} &= \left(\mathbf{I} - \frac{2\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right)\mathbf{y} \\ &= \mathbf{y} - 2\mathbf{v}\left(\frac{\mathbf{v}^T\mathbf{y}}{\mathbf{v}^T\mathbf{v}}\right) \end{aligned}$$

Hence,

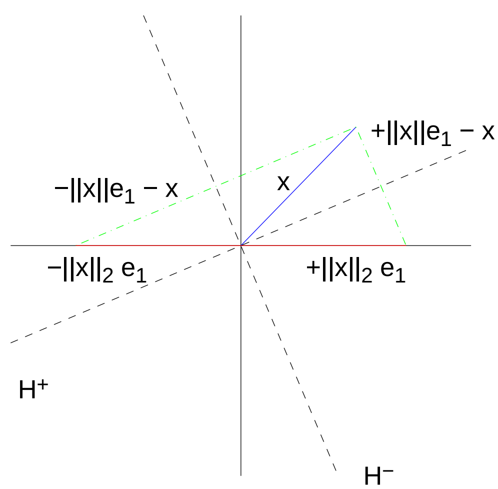
$$\mathbf{F} = \mathbf{I} - \frac{2\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}$$

so that the only difference between the full rank and orthogonal $\mathbf{F}$ and the rank $(m-1)$ and orthogonal $\mathbf{P}$ is the factor of 2.

◇ THE BETTER OF TWO REFLECTORS

There is another possibility for a Householder reflector that we have ignored so far. It corresponds to mapping $\mathbf{x}$ to $-\|\mathbf{x}\|\mathbf{e}_1$.

Mathematically, this is also an acceptable choice according to our criteria for defining $\mathbf{F}$.



Let us call these two mappings $\mathbf{H}^+, \mathbf{H}^-$ according to whether $\mathbf{x}$ is mapped to $+\|\mathbf{x}\|\mathbf{e}_1$ or $-\|\mathbf{x}\|\mathbf{e}_1$.

Although both mappings are satisfactory mathematically, for a given $\mathbf{x}$, one will have better numerical stability properties then the other.

For numerical stability, we want that $\mathbf{Hx}$ not be too close to $\mathbf{x}$ itself.

To achieve this, we can choose

$$\mathbf{v} = -\mathrm{sgn}(x_1)\|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$$

where x_1 is the first component of $\mathbf{x}$ and

$$\mathrm{sgn}(x_1) = \begin{cases} +1 \text{ if } x_1 \geq 0 \\ -1 \text{ otherwise} \end{cases}$$

To see why the choice of sign affects stability, consider what happens if $\mathbf{H}^+$ makes a small angles with respect to $\mathbf{e}_1$. Then, $\mathbf{v} = \|\mathbf{x}\|\mathbf{e}_1 - \mathbf{x}$ is much smaller than either $\|\mathbf{x}\|\mathbf{e}_1$ or $\mathbf{x}$.

In other words, there is a significant loss of precision when computing $\mathbf{v}$ thanks to cancellation.

Our choice of $\mathbf{v}$ however ensures that $\|\mathbf{v}\| \geq \|\mathbf{x}\|$.

◇ THE HOUSEHOLDER ALGORITHM

Notation:
Let $\mathbf{A}(i_1 : i_2, j_1 : j_2)$ be the $(i_2 - i_1 + 1) \times (j_2 - j_1 + 1)$ submatrix of matrix $\mathbf{A}$ with upper left corner $\mathbf{a}_{i_1,j_1}$ and lower right corner $\mathbf{a}_{i_2,j_2}$.

If the submatrix reduces to a single row or column, we write $\mathbf{A}(i, j_1 : j_2)$ or $\mathbf{A}(i_1 : i_2, j)$ respectively.

The following algorithm computes the factor $\mathbf{R}$ of a QR factorization of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$, overwriting $\mathbf{A}$ with the result.

We also store n reflection vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ for future use.

ALGORITHM 10.1: HOUSEHOLDER QR FACTORIZATION

for $k = 1$ to n **do**
  $\mathbf{x} = \mathbf{A}(k:m, k)$
  $\mathbf{v}_k = \mathrm{sgn}(x_1)\|\mathbf{x}\|_2\mathbf{e}_1 + \mathbf{x}$ % This is $-\mathbf{v}$, but
  % that doesn't matter
  $\mathbf{v}_k = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|_2}$
  $\mathbf{A}(k:m, k:n) = \mathbf{A}(k:m, k:n) - 2(\mathbf{v}_k^T\mathbf{A}(k:m, k:n))\mathbf{v}_k$
end for

$\diamond$ APPLYING OR FORMING $\mathbf{Q}$

This algorithm reduces $\mathbf{A}$ to upper triangular form (the "$\mathbf{R}$" in the QR factorization).

The "$\mathbf{Q}$" of this factorization (or even the $\hat{\mathbf{Q}}$) has not been constructed.

The reason is that this takes additional work!

Quite often we do not need $\mathbf{Q}$ anyway, only its effect on some vector
i.e., we only need the product $\mathbf{Q}\mathbf{x}$ or $\mathbf{Q}^T\mathbf{x}$.

For this we note

$$\mathbf{Q}^T = \mathbf{Q}_n \mathbf{Q}_{n-1} \ldots \mathbf{Q}_1$$

and

$$\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 \ldots \mathbf{Q}_n$$

(why?)

e.g., To solve $\mathbf{Ax} = \mathbf{b}$ by QR factorization, we write $\mathbf{QRx} = \mathbf{b}$, then $\mathbf{Rx} = \mathbf{Q}^T\mathbf{b}$
$\rightarrow$ so the only way we ever need $\mathbf{Q}$ is in $\mathbf{Q}^T\mathbf{b}$.

Recall $\mathbf{A} = \mathbf{QR} \implies \mathbf{R} = \mathbf{Q}^T\mathbf{A}$
i.e., the same process that reduced $\mathbf{A}$ to $\mathbf{R}$ is equivalent to multiplication by $\mathbf{Q}^T$.

ALGORITHM 10.2: CALCULATION OF $\mathbf{Q}^T\mathbf{b}$

for $k = 1$ to n **do**
  $\mathbf{b}(k:m) = \mathbf{b}(k:m) - 2(\mathbf{v}_k^T \mathbf{b}(k:m))\mathbf{v}_k$
end for

Similarly, calculation of $\mathbf{Qx}$ can be achieved by the same process reversed.

ALGORITHM 10.3: CALCULATION OF $\mathbf{Qx}$

for $k = n$ downto 1 **do**
  $\mathbf{x}(k:m) = \mathbf{x}(k:m) - 2(\mathbf{v}_k^T \mathbf{x}(k:m))\mathbf{v}_k$
end for

◇ OPERATION COUNT

The work involved in Algorithm 10.1 is dominated by the inner most loop

$$\mathbf{A}(k:m,j) - 2(\mathbf{v}_k^T \mathbf{A}(k:m,j))\mathbf{v}_k$$

If the vector has length $l = m - k + 1$, we need l subtractions, l scalar multiplications, and l multiplications and $l-1$ additions for the dot product.

$$= 4l - 1 \approx 4l \quad (\sim 4 \text{ flops per entry operated on})$$

Schematically, here is what is going on for a 5×4 matrix:

$$\underset{\text{original matrix}}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \longrightarrow \underset{\text{step 1}}{\begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{bmatrix}}$$

$$\longrightarrow \underset{\text{step 2}}{\begin{bmatrix} \times & \times & \times & \times \\ & * & * & * \\ & 0 & * & * \\ & 0 & * & * \\ & 0 & * & * \end{bmatrix}} \longrightarrow \underset{\text{step 3}}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & * & * \\ & & 0 & * \\ & & 0 & * \end{bmatrix}}$$

$$\longrightarrow \underset{\text{step 4}}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & * \\ & & & 0 \end{bmatrix}}$$

At step k, rows 1 to $k-1$ are unchanged and columns 1 to k are zero. We do not operate on any of these elements; we only operate on $(m-k+1)(n-k)+1$ elements.

$\therefore$ operation count

$$\begin{aligned}
&\sim \left(\sum_{k=1}^{n}(m-k+1)(n-k)+1\right) \\
&\qquad\qquad \text{elements * 4 operations/elements} \\
&\sim 4\sum_{k=1}^{n} mn-(m+n+1)k+k^2 \\
&= 4\left[mn^2-(m+n+1)\frac{n(n+1)}{2}+\right. \\
&\qquad\qquad \left.\frac{n(n+1)(2n+1)}{6}\right] \\
&\sim 4\left[\frac{mn^2}{2}-\frac{n^3}{6}\right]=2mn^2-\frac{2}{3}n^3
\end{aligned}$$