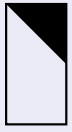
QR factorization

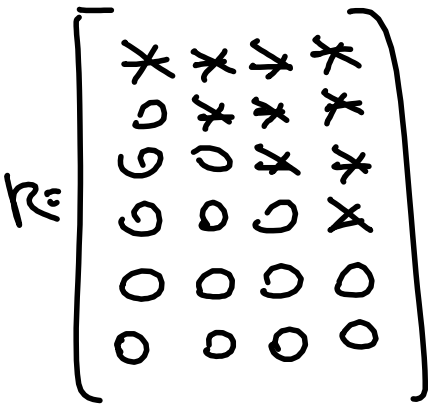
Next goal: proving existence of another kind of factorization, QR factorization.

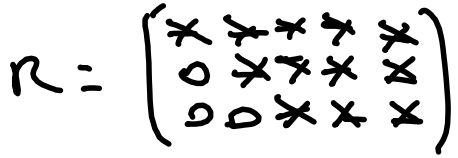
Theorem

For every $A \in \mathbb{R}^{m \times n}$, there exist $Q \in \mathbb{R}^{m \times m}$ orthogonal, R upper triangular (i.e. ) such that $A = QR$.

Not as powerful / revealing as SVD, but easier to compute.

Most interesting case: $m \geq n$ (square or tall-thin).





A invertible $\Leftrightarrow$ Q is invertible AND R is invertible

QR

Q invertible $\rightarrow$ always

R invertible $\Leftrightarrow$ no zeros on diagonal

Factorizations and linear systems

What do we use all these factorizations for?

- They reveal properties: singularity, rank, ...
- As an intermediate step in algorithms, e.g., solving linear systems.

To solve a linear system $Ax = b$ ($A \in \mathbb{R}^{m \times m}$ invertible):

1. Compute $A = QR$.

   Now it holds $x = A^{-1}b = R^{-1}Q^{-1}b = R^{-1}(Q^T b)$.
2. Compute $c = Q^T b$ (i.e., solve $Qc = b$).
3. Compute $x = R^{-1}c$ (i.e., solve $Rx = c$).

$(QR)^{-1} = R^{-1}Q^{-1}$

Factorizations and linear systems

What do we use all these factorizations for?

- They reveal properties: singularity, rank, ...
- As an intermediate step in algorithms, e.g., solving linear systems.

To solve a linear system $Ax = b$ ($A \in \mathbb{R}^{m \times m}$ invertible):

1. Compute $A = QR$. $O(m^3)$
   Now it holds $x = A^{-1}b = R^{-1}Q^{-1}b = R^{-1}Q^T b$.
2. Compute $c = Q^T b$ (i.e., solve $Qc = b$). $O(m^2)$
3. Compute $x = R^{-1}c$ (i.e., solve $Rx = c$). $O(m^2)$ (back-subst)

Step 1 does not require b. If we have to solve several linear systems with the same A, we can precompute the expensive factorization.

(Remark: one could do the exact same trick by precomputing A^{-1} — not as effective in terms of computational cost and stability.)

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \leftarrow \textcircled{*}$$

$\leftarrow$ $a_{33}x_3 + a_{34}x_4 = b_3 \rightsquigarrow x_3 = \dfrac{b_3 - a_{34}x_4}{a_{33}}$

$\leftarrow$ $a_{44}x_4 = b_4 \rightsquigarrow x_4 = \dfrac{b_4}{a_{44}}$

$\textcircled{*}$ $a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \rightsquigarrow x_2 = \dfrac{b_2 - \dots}{a_{22}}$

$O(n^2)$ back-substitution

The case of a vector

First simpler case: compute QR when A has only one column.

Problem

Given $x \in \mathbb{R}^n$, find an orthogonal matrix Q such that $Q^T x$ is of the form $\begin{bmatrix} s \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = se_1$.

$$x = Q \cdot \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix} \iff Q^T x = \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

(We call e_j the jth column of I.) $e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Remark Since orthogonal matrices preserve norm, s can only be $\pm\|x\|$.

$$\|x\| = \left\| Q \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\| = \left\| \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\| = \sqrt{s^2} = |s|$$

Householder reflectors

Lemma

For every $u \in \mathbb{R}^m$, the matrix $H = I - \frac{2}{u^T u} uu^T$ is orthogonal.

Written also $I - \frac{2}{\|u\|^2} uu^T$, or $I - 2vv^T$ where $v = \frac{1}{\|u\|} u$ has norm 1.

Proof: verify directly $HH^T = I$.

Geometric idea: in a coordinate system such that $u = e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, it is a reflection of the first coordinate.

Note that for each $x \in \mathbb{R}^{m \times m}$ we can compute $Hx = (I - 2vv^T)x = x - 2v(v^T x)$ in $O(m)$, and HA for any $A \in \mathbb{R}^{m \times m}$ in $O(m^2)$.

scalar product, gives a number

$v_1 x_1 + \dots + v_m x_m$

$$I - \frac{2}{\|u\|^2} uu^T = I - 2\left(\frac{1}{\|u\|}\cdot u\right)\left(\frac{1}{\|u\|}\cdot u\right)^T = I - 2vv^T$$

$\|v\| = 1$

$$\left(I - 2vv^T\right)\left(I - 2vv^T\right)^T = \left(I - 2vv^T\right)\left(I^T - \left(2vv^T\right)^T\right) =$$

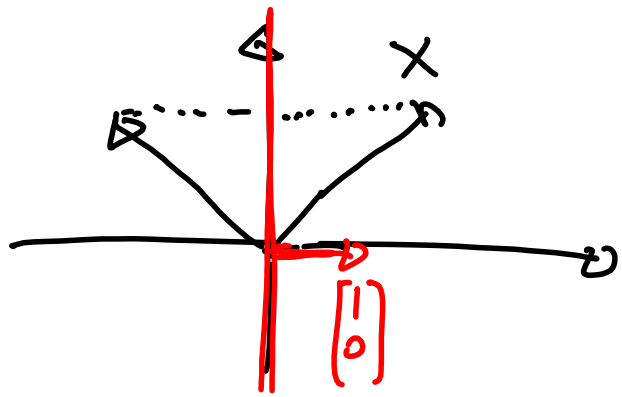
$$= \left(I - 2vv^T\right)\left(I - 2vv^T\right) = I\cdot I - 2vv^T\cdot I - I\cdot 2vv^T + 4v\left(v^T v\right)v^T$$

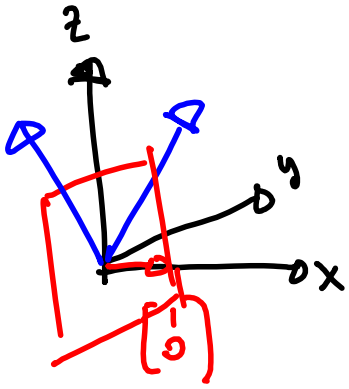
$$= I - 2vv^T - 2vv^T + 4vv^T = I$$

Choose coord. system such that $u = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

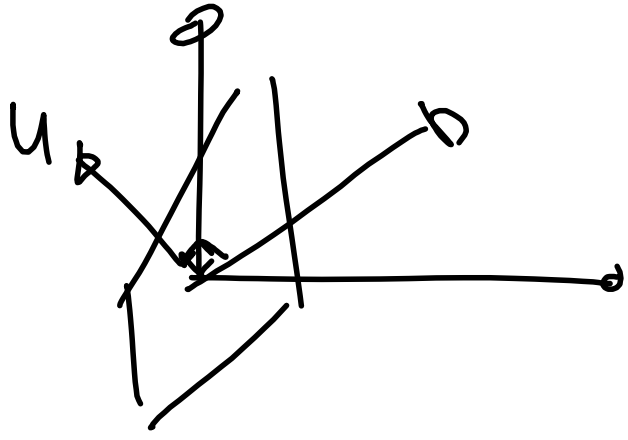
$$I - \frac{2}{\|u\|^2}\cdot\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} = I - 2\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & & \\ \vdots & & \ddots & \\ 0 & & & 0 \end{bmatrix} = \begin{bmatrix} -1 & & & \\ & 1 & & 0 \\ & & 1 & \\ & 0 & & \ddots \\ & & & & 1 \end{bmatrix}$$

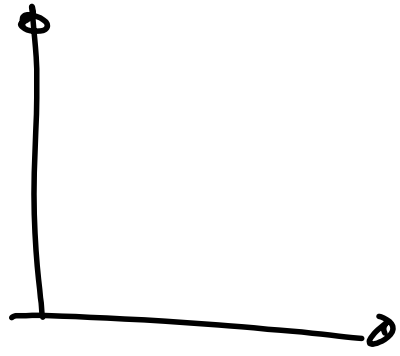
$$\begin{bmatrix} -1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} -x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix}$$





H reflects along the plane perpendicular to u





Where can we get by reflecting

Lemma

Let x, y be two vectors such that $\|x\| = \|y\|$. If one chooses $u = x - y$, then $H = I - \frac{2}{u^T u} uu^T$ is such that $Hx = y$.

Proof: boring algebra (clear denominators, expand...).

Geometric idea: reflecting through the plane perpendicular to $x - y$ sends x into y.

In particular, we can take $y = \|x\| e_1 = \begin{bmatrix} \|x\| \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

This is already a QR factorization of $A = x \in \mathbb{R}^{m \times 1}$.

Matlab: `function u = householder_vector(x)`

Matlab implementation

```
function [v, s] = householder_vector(x)
s = norm(x);
v = x;
v(1) = v(1) - s;
v = v / norm(v);
```

Testing it:

```
>> x = randn(4,1);
>> [u, s] = householder_vector(x);
>> x - 2*u*(u'*x)
ans =
   2.2541e+00
            0
  -1.1102e-16
            0
>> s
s =
   2.2541e+00
```

An extreme example

```
>> x = [1e10; 1e-6; 1e-6; 1e-6];
>> x
x =
   1.0000e+10
   1.0000e-06
   1.0000e-06
   1.0000e-06
>> [u, s] = householder_vector(x);
>> x = [1e10; 1e-6; 1e-6; 1e-6];
>> x - 2*u*(u'*x)
ans =
   1.0000e+10
  -1.0000e-06
  -1.0000e-06
  -1.0000e-06
```

The transformation did not make this vector any closer to being a multiple of e_1...

Reason for instability

Subtraction between two almost-equal values $\rightarrow$ cancellation.

```
>> x(1) - norm(x)
ans =
     0
```

This should be negative – the norm is always larger than each entry of the vector...

Solution

Instead of choosing $y = \|x\|e_1$, we choose $y = -\|x\|e_1$.

In general, without cancellation:

```
function [v, s] = householder_vector(x)
s = -sgn(x(1)) * norm(x);
v = x;
v(1) = v(1) - s;
v = v / norm(v);
```

Now that example works better:

```
>> x = [1e10; 1e-6; 1e-6; 1e-6];
>> [u, s] = householder_vector(x);
>> x - 2*u*(u'*x)
ans =
  -1.0000e+10
            0
            0
            0
```

QR factorization via Householder matrices

Next case: A is square.

Step 1: take $u_1 = \text{householder\_vector}(\text{A}(:, 1))$ to get

$$H_1 A = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix} =: \begin{bmatrix} B_2 & C_2 \\ 0 & A_2 \end{bmatrix}.$$

Step 2 (wrong): we can't simply multiply by another reflector $H_2 A_1$ — it would 'spoil' the zeros in the first column.

Step 2 (right): take $u_2 = \text{householder\_vector}(\text{A2}(:, 1))$, and compute

$$\begin{bmatrix} 1 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} B_2 & C_2 \\ 0 & A_2 \end{bmatrix} = \begin{bmatrix} B_2 & C_2 \\ 0 & H_2 A_2 \end{bmatrix} = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix} = \begin{bmatrix} B_3 & C_3 \\ 0 & A_3 \end{bmatrix}.$$

Continue. . .

$$\begin{bmatrix} I & 0 \\ 0 & H_3 \end{bmatrix} \begin{bmatrix} B_3 & C_3 \\ 0 & A_3 \end{bmatrix} = \begin{bmatrix} B_3 & C_3 \\ 0 & H_3 A_3 \end{bmatrix} = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix},$$

$$\begin{bmatrix} I & 0 \\ 0 & H_4 \end{bmatrix} \begin{bmatrix} B_4 & C_4 \\ 0 & A_4 \end{bmatrix} = \begin{bmatrix} B_4 & C_4 \\ 0 & H_4 A_4 \end{bmatrix} = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{bmatrix}.$$

We have computed a sequence of orthogonal matrices such that $Q_{m-1} \cdots Q_3 Q_2 Q_1 A = R$ is orthogonal.

$$A = \underbrace{(Q_1^* Q_2^* \cdots Q_{m-1}^*)}_{:=Q} R.$$

(Recall: product of orthogonal matrices is orthogonal.)

Matlab implementation

```
function [Q, R] = myqr(A)
[m, n] = size(A);
Q = eye(m);
for j = 1:n
    v = householder_vector(A(j:end, j));
    H = eye(length(v)) - 2*v*v';
    A(j:end,j:end) = H * A(j:end,j:end);
    Q(:, j:end) = Q(:, j:end) * H;
end
R = A;
```

Optimizations

Huge optimization: don't form H — use $HA_j = A_j - 2v(v^T A_j)$.

Minor optimization: write s and zeros manually in `A(j:end,j)`.

Detail: we can stop at $n-1$, the last step does (almost) nothing.

```
function [Q, A] = myqr(A)
[m, n] = size(A);
Q = eye(m);
for j = 1:n-1
    [v, s] = householder_vector(A(j:end, j));
    A(j,j) = s; A(j+1:end,j) = 0;
    A(j:end,j+1:end) = A(j:end,j+1:end) - ...
        2*v*(v'*A(j:end,j+1:end));
    Q(:, j:end) = Q(:, j:end) - Q(:,j:end)*v*2*v';
end
```

Rectangular QR

What if A is tall thin?

Same thing: just use fewer columns. $\begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Actually, QR is 'incremental': if I take only the first n columns of R, I get the factorization of the first n columns of A.

Thin QR (like thin SVD): we can use $Q_1 \in \mathbb{R}^{m \times n}$, $R_1 \in \mathbb{R}^{n \times n}$.

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1.$$

Good algorithms produce Q_1 directly.

Alternative: We can store and use the sequence of Householder vectors v to represent Q implicitly.

Remarks

Computational cost: $\frac{4}{3}n^3$ for square matrices; scales like $2mn^2$ when $m \gg n$ (thin QR).

Exercises

1. Is the QR factorization unique? (Hint: try to change a few signs).
2. Show that Householder reflectors are symmetric.
3. Suppose that a matrix $A \in \mathbb{R}^{m \times m}$ is 'upper triangular plus one more diagonal', e.g., $\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$ (these are called *Hessenberg matrices*). What is the cost of performing QR on a matrix with this structure?
4. Can you identify (without computation) a QR of a matrix with zero structure $\begin{bmatrix} 0 & 0 & 0 & 0 & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & * & * & * \\ 0 & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$? (Hint: swapping rows is an orthogonal operation).