QR Decomposition: Computation and Applications

Shilin Lai

Princeton University

06 February 2020

Motivation

Problem

Given a system $A\vec{x} = \vec{b}$ which is inconsistent, how to find the optimal choice of $\vec{x}$ to minimize some objectives.

Motivation

Problem

Given a system $A\vec{x} = \vec{b}$ which is inconsistent, how to find the optimal choice of $\vec{x}$ to minimize some objectives.

One way of formalizing this is the following notion.

Definition

A vector $\vec{x}^*$ is a *least square solution* if for all $\vec{x}$, we have

$$\|A\vec{x}^* - \vec{b}\| \leq \|A\vec{x} - \vec{b}\|$$

The value of $A\vec{x}^*$ is the *least square approximation* to $\vec{b}$.

The projection method

The least squared approximation is the projection of $\vec{b}$ to $\mathrm{Im}(A)$, so we can also solve the problem in three steps:

The projection method

The least squared approximation is the projection of $\vec{b}$ to $\mathrm{Im}(A)$, so we can also solve the problem in three steps:

(i) Compute the QR factorization of A to find an orthonormal basis for $\mathrm{Im}(A)$ (columns of Q).

The projection method

The least squared approximation is the projection of $\vec{b}$ to $\mathrm{Im}(A)$, so we can also solve the problem in three steps:

(i) Compute the QR factorization of A to find an orthonormal basis for $\mathrm{Im}(A)$ (columns of Q).

(ii) The least squared approximation is $QQ^T\vec{b}$ by the projection formula.

The projection method

The least squared approximation is the projection of $\vec{b}$ to $\mathrm{Im}(A)$, so we can also solve the problem in three steps:

(i) Compute the QR factorization of A to find an orthonormal basis for $\mathrm{Im}(A)$ (columns of Q).

(ii) The least squared approximation is $QQ^T\vec{b}$ by the projection formula.

(iii) The least squared solution satisfies $A\vec{x}^* = QQ^T\vec{b}$, which implies that

$$R\vec{x}^* = Q^T\vec{b}$$

Solve this by back-substitution.

The normal equation

There is also a direct approach.

Theorem

(1) The least squared solutions to $A\vec{x} = \vec{b}$ are exactly the solutions to the *normal equation*

$$A^T A\vec{x} = A^T \vec{b}$$

(2) $\ker(A^T A) = \ker(A)$.

(3) If $\ker(A) = \{\vec{0}\}$, then the least squared solution is unique, and the least squared approximation is

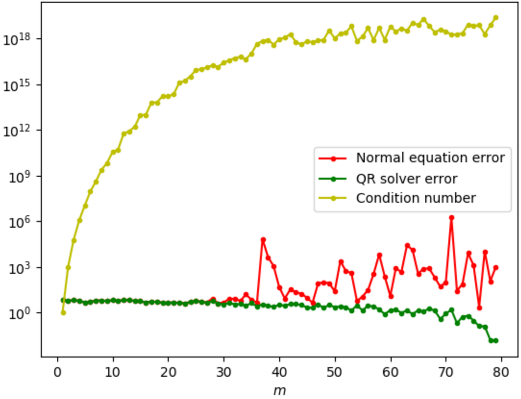
$$A(A^T A)^{-1} A^T \vec{b}$$

Why use QR?

The normal equation typically behaves much worse numerically compared to the original equation.

Why use QR?

A is the first m columns of a ill-conditioned 80×80 matrix.
$\vec{b}$ is a randomly chosen vector in $\mathbb{R}^{80}$.



Why use QR?

The computation of $A^T A$ may already lead to round-off error.

Why use QR?

The computation of $A^T A$ may already lead to round-off error.

$$A = \begin{bmatrix} 10^8 & -10^8 \\ 1 & 1 \end{bmatrix}, \quad A^T A = 10^{16} \left(\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + 10^{-16} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right)$$

The 10^{-16} term is below standard machine precision, so $A^T A$ looks singular to the computer.

Gram–Schmidt algorithm

Idea

Project $\vec{v}_i$ to the orthogonal complement of the subspace spanned by the vectors which come before it.

Problem: only have projection formula if we have an orthonormal basis.

Solution: compute the orthonormal basis inductively!

Extended Gram–Schmidt

Input: $n \times m$ matrix A with linearly independent columns $\vec{v}_1, \cdots, \vec{v}_m$.
Output: The QR decomposition of A.

Pseudocode

$Q := [\,];$
For $(i = 1;\ i \leq m;\ i{+}{+})\{$
  For $(j = 1;\ j < i;\ j{+}{+})$: $r_{ji} = \vec{u}_j^T \vec{v}_i$; // Projection coefficients
  $\vec{v}_i^{\perp} := \vec{v}_i - (r_{1i}\vec{u}_1 + \cdots + r_{i-1,i}\vec{u}_{i-1})$; // Compute projection
  $r_{ii} := \|\vec{v}_i^{\perp}\|$; // Normalizing constant
  $\vec{u} := \frac{1}{r_{ii}}\vec{v}_i^{\perp}$; // Normalize
  Append $\vec{u}$ to Q;
$\}$

Why not Gram–Schmidt

Key issue: in high dimension space, two randomly chosen vectors are nearly orthogonal.

Why not Gram–Schmidt

Key issue: in high dimension space, two randomly chosen vectors are nearly orthogonal.

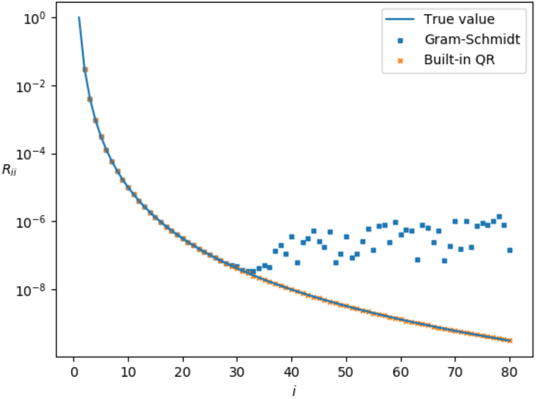
$\Longrightarrow \vec{v}_i^{\perp}$ is near $\vec{0}$, and its computation involves subtracting two numbers which are close.

$\Longrightarrow$ Deterioration of accuracy.

Why not Gram–Schmidt

$A = QR$ is 80×80 with rapidly decaying diagonal entries in R.

This graph shows the diagonal entries computed using the Gram–Schmidt.



Better algorithms

Idea

Do a fancy row reduction using *orthogonal matrices*.

Better algorithms

Idea

Do a fancy row reduction using *orthogonal matrices*.

More precisely, repeatedly left-multiplying by orthogonal matrices to create zeroes in A:

$$Q_1 A = \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & * & * \end{pmatrix}$$

Better algorithms

Idea

Do a fancy row reduction using *orthogonal matrices*.

More precisely, repeatedly left-multiplying by orthogonal matrices to create zeroes in A:

$$Q_2 Q_1 A = \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \end{pmatrix}$$

$$A = Q_1^T Q_2^T R$$

Householder reflection

Problem

Given $\vec{x}$ and $\vec{y}$ in $\mathbb{R}^n$ of the same length, find an orthogonal matrix Q such that $Q\vec{x} = \vec{y}$.

Householder reflection

Problem

Given $\vec{x}$ and $\vec{y}$ in $\mathbb{R}^n$ of the same length, find an orthogonal matrix Q such that $Q\vec{x} = \vec{y}$.

Observation

Let $\vec{u}$ be a non-zero vector, then the reflection across the hyperplane $\vec{u}^\perp$ is

$$H_{\vec{u}} = I_n - 2\frac{\vec{u}\vec{u}^T}{\|\vec{u}\|^2}$$

Householder reflection

Given $\vec{x}, \vec{y} \in \mathbb{R}^n$ of equal length, let $\vec{u} = \vec{x} - \vec{y}$,

Householder reflection

Given $\vec{x}, \vec{y} \in \mathbb{R}^n$ of equal length, let $\vec{u} = \vec{x} - \vec{y}$, then

$$\begin{aligned} H_{\vec{u}}(\vec{x}) &= \vec{x} - 2\frac{\vec{u}^T\vec{x}}{\vec{u}^T\vec{u}}\vec{u} \\ &= \vec{x} - 2\frac{\|\vec{x}\|^2 - \vec{y}^T\vec{x}}{2(\|\vec{x}\|^2 - \vec{y}^T\vec{x})}\vec{u} \\ &= \vec{x} - \vec{u} \\ &= \vec{y} \end{aligned}$$

We have shown that orthogonal matrices act *transitively* on vectors of the same length.

Householder reflection

Input: $n \times n$ matrix A

Output: The QR decomposition of A

Pseudocode

For $(i = 1;\ i \leq m;\ i{+}{+})\{$

$\quad \vec{u} := A[i:n, i] - \vec{e}_1;$

$\quad Q_i := \begin{pmatrix} I_i & 0 \\ 0 & H_{\vec{u}} \end{pmatrix};$

$\quad A := Q_i A;$

$\}$

$R := A;$

$Q := Q_1^T Q_2^T \cdots Q_n^T;$

Complexity

Reflection of a vector in $\mathbb{R}^n$:

- n for dot product.
- n for scalar multiplication.
- n for subtraction.

Computing vector norm takes n operations.

Total complexity:

$$\sum_{i=0}^{n-1} 4(n-i)^2 \simeq \frac{4}{3}n^3$$

Complexity

Reflection of a vector in $\mathbb{R}^n$:

- n for dot product.
- n for scalar multiplication.
- n for subtraction.

Computing vector norm takes n operations.
Total complexity:

$$\sum_{i=0}^{n-1} 4(n-i)^2 \simeq \frac{4}{3}n^3$$

This is comparable to Gram–Schmidt, but the algorithm is stable (with some tweaks).

Givens rotation

Can also kill off entries one by one using plane rotations

$$\begin{bmatrix} I_i & & & & \\ & c & \cdots & s & \\ & \vdots & I_{j-i-1} & \vdots & \\ & -s & \cdots & c & \\ & & & & I_{n-j-1} \end{bmatrix}$$

Givens rotation

Can also kill off entries one by one using plane rotations

$$\begin{bmatrix} I_i & & & & \\ & c & \cdots & s & \\ & \vdots & I_{j-i-1} & \vdots & \\ & -s & \cdots & c & \\ & & & & I_{n-j-1} \end{bmatrix}$$

Each rotation takes $O(n)$ operations to perform, so if A is sparse, this can be much faster (with good application order).