The QR factorization for Least-Square problems
Numerical Linear Algebra 091119

The QR-factorization.

Let $A \in \mathbb{R}^{m\times n}$, $m \geq n$. The (full) QR-factorization of A is $A = QR$, where $Q \in \mathbb{R}^{m\times m}$ is orthogonal (i.e. $Q^tQ = Id$) and $R \in \mathbb{R}^{m\times n}$ is upper triangular.

One has

$$Q = (Q_1|Q_2),\ R = \left(\begin{array}{c} R_1 \\ 0 \end{array} \right)$$

where $Q_1 \in \mathbb{R}^{n\times m}$ and $R_1 \in \mathbb{R}^{n\times n}$. The thin QR-factorization of A is $A = Q_1R_1$.

Properties: If $rank(A) = n$ (i.e. A is full-rank) then

1. $Im(A) = Im(Q_1)$ and $Im(A^{\perp}) = Im(Q_2)$.
2. The thin factorization $A = Q_1R_1$ is unique.
3. $R_1 = G^t$ where G is given by the Cholesky factorization of $A^tA = GG^t$.

Methods: To compute the thin factorization one can use Gram-Schmidt. More numerically stable methods (specially if the vectors of A are not fairly independent) are based on orthogonal transformations: one can use either Householder reflexions or Givens rotations. On the other hand, Gram-Schmidt requires $2mn^2$ flops, Givens $3n^2(m - n/3)$ flops and Householder $4(m^2n - mn^2 + n^3/3)$ flops. Givens rotations allows to easily compute the factorization in cases with structure preserving it (e.g. Hessenberg matrices, sparse matrices, etc).

The Least-Squares (LS) problem.

Given $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$ with $m \geq n$, we want to compute $x_{LS} \in \mathbb{R}^n$ such that the minimum of $\|Ax - b\|_2$ is achieved for $x = x_{LS}$.

Ex.1 Consider $x, z \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$ and $b \in \mathbb{R}^m$. Prove that one has

$$\|A(x + \alpha z) - b\|^2 = \|Ax - b\|^2 + 2\alpha z^tA^t(Ax - b) + \alpha^2\|Az\|.$$

The full-rank LS problem.

It follows from the previous equality in Ex.1 that if x solves the LS problem then $A^t(Ax-b) = 0$ (i.e. x is the solution of the so-called normal equations). Moreover, if x and $x+\alpha z$ solve the LS problem then $z \in Ker(A)$ and one concludes that, if A is full-rank, the least-squares problem has unique solution.

Three basic methods to solve the LS problem:

1. *Normal equations:*

   compute A^tA
   compute $d = A^tb$
   compute Cholesky $A^tA = GG^t$
   solve $Gy = d$ and $G^tx = y$ to obtain x_{LS}

   Drawback: $\|x_{LS}\|_2 \approx k_2(A)^2\epsilon_{\text{machine}}$ because we solve a linear system with matrix A^tA.

2. Use QR-factorization (see below)

3. Use SVD-factorization (it will be explained in the theoretical lessons by M. Sombra)

We focus on the use of QR for solving the LS problem. One has

$$Ax = b \Leftrightarrow QRx = b \Leftrightarrow Rx = Q^tb$$

If one writes $Q^tb = (y_1|y_2)^t \in \mathbb{R}^m$, with $y_1 \in \mathbb{R}^n$, then

$$Rx = Q^tb \Leftrightarrow \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

One solves $R_1x = y_1$ and obtain x_{LS} with a residual error $\|r\|_2^2 = \|y_1 - R_1x\|^2 + \|y_2\|^2 = \|y_2\|^2$. Note that we solve a system with matrix R_1, since Q orthogonal $k_2(R_1) = k_2(A)$. This justifies the fact that QR is better than directly solve normal equations.

Ex. 2 Code the previous algorithm to solve full-rank LS problem. Use it to polynomial fitting a data set.

The rank deficient LS problem using QR.

If $r = \text{rank}(A) < n$ there is, in general, no unique solution of the LS problem and one is lead to require an extra condition to obtain x_{LS} (e.g. to be a minimum $\|\cdot\|_2$-norm solution).

Idea: One applies QR with pivoting

$$AP = QR,\ Q \in \mathbb{R}^{m\times m} \text{ orthogonal },R \in \mathbb{R}^{m\times n} \text{ upper triangular},$$

being P a pivoting matrix so that

$$R = \begin{pmatrix} R_1 & S \\ 0 & 0 \end{pmatrix}$$

with R_1 non-singular upper triangular matrix.

More concretely, since $A = QRP^t$, one has,

$$\|b - Ax\|_2^2 = \left\| b - Q\begin{pmatrix} R_1 & S \\ 0 & 0 \end{pmatrix} P^tx \right\|_2^2 = \left\| Q^tb - \begin{pmatrix} R_1 & S \\ 0 & 0 \end{pmatrix} P^tx \right\|_2^2$$

since Q orthogonal implies that $\|QA\|_2 = \|A\|_2$. Writing $P^t x = (u, v)^t$ and $Q^t b = (c, d)^t$, with $u, c \in \mathbb{R}^r$ and $v, d \in \mathbb{R}^{n-r}$, one has

$$\|b - Ax\|_2^2 = \left\| \begin{pmatrix} c \\ d \end{pmatrix} - \begin{pmatrix} R_1 & S \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} c - R_1 u + Sv \\ d \end{pmatrix} \right\|_2^2 = \|c - R_1 u + Sv\|_2^2 + \|d\|_2^2$$

The left summand maybe vanishes for a suitable x (because depends on $P^t x = (u, v)^t$). The right one does not depend on x, then

$$\min \|b - Ax\|_2^2 = \|d\|_2^2 \tag{1}$$

and is attained whenever

$$R_1 u + Sv = c.$$

There are different ways to choose (u, v) solving the last equation. One can look for the so-called *minimal solution* of the LS problem meaning that, among the possible solutions (u, v) of (1) one looks for the one which minimizes $\|(u, v)\|_2$ (this requires an optimization problem). Another option is to look for the so called *basic solution* of the LS problem. Then one takes $v = 0$ and solves $R_1 u = c$ to obtain u.

Ex. 3 Code the previous algorithm to compute a basic solution of the rank-deficient LS problem.