

CS 450: Numerical Analysis – Lecture 8

Multiple Systems, Sherman–Morrison, and Linear Least Squares

Edgar Solomonik

September 17, 2026

(Lecture 8 begins in the Chapter 2 notes, with solving multiple systems and Sherman–Morrison.)

Chapter 3: linear least squares

Assume a simpler scenario than rank deficiency: $A \in \mathbb{R}^{m \times n}$ with $m > n$ and $\text{rank}(A) = n$ (full column rank). Given $b \in \mathbb{R}^m$, solve

$$\min_x \|Ax - b\|_2.$$

Optimality. For the solution x^* , the residual is orthogonal to the column span:

$$b - Ax^* \perp \text{colspan}(A).$$

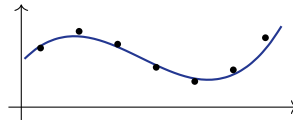
Otherwise, expand the residual $r = b - Ax^*$ as $r = r_1 + r_2$ with $0 \neq r_1 \in \text{colspan}(A)$ and $r_2 \perp \text{colspan}(A)$, and pick z with $Az = r_1$. Since $r_1 \perp r_2$,

$$\|r\|_2^2 = \|r_1\|_2^2 + \|r_2\|_2^2, \quad \text{hence} \quad \|r_2\|_2 < \|r\|_2,$$

and $b - A(x^* + z) = r - r_1 = r_2$, so $x^* + z$ is a better solution—a contradiction.

Numerous applications. For example, polynomial fitting: given data $(z_1, f_1), \dots, (z_m, f_m)$, find the best cubic polynomial fit

$$p_x(z) = x_1 + x_2 z + x_3 z^2 + x_4 z^3.$$



We want

$$\min_x \sum_{i=1}^m |p_x(z_i) - f_i|^2.$$

The model values $p_x(z_i)$ are collected by the Vandermonde matrix,

$$\begin{bmatrix} p_x(z_1) \\ \vdots \\ p_x(z_m) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & z_1 & z_1^2 & z_1^3 \\ 1 & z_2 & z_2^2 & z_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & z_m & z_m^2 & z_m^3 \end{bmatrix}}_V \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix},$$

so with $f = (f_1, \dots, f_m)^T$ the problem is $\min_x \|Vx - f\|_2$ —a linear least-squares (LLSQ) problem.

Solution: the normal equations. We want x such that $b - Ax \perp \text{colspan}(A)$:

$$A^T(b - Ax) = 0 \iff A^T A x = A^T b \quad (\text{the normal equations}). \quad (8.2)$$

In-class activity: properties of $A^T A$. For $A \in \mathbb{R}^{m \times n}$ with $\text{rank}(A) = n$, what properties does the $n \times n$ matrix $A^T A$ have? *Resolution.* It is symmetric, $(A^T A)^T = A^T A$, and positive definite: with $y = Ax$,

$$x^T A^T A x = y^T y = \|y\|_2^2 > 0 \quad \text{for } x \neq 0$$

($y \neq 0 \iff x \neq 0$, by full column rank). In particular, $A^T A$ is SPD, hence invertible, and Cholesky applies with no pivoting.

A naive algorithm: Cholesky on the normal equations. Factorize $A^T A = LL^T$ by Cholesky (no pivoting needed, by the activity), then solve $LL^T x = A^T b$. The cost is $O(mn^2)$ to form $A^T A$, plus $O(n^3)$ for the factorization.

Numerical stability of the normal equations. Is solving the normal equations numerically stable? Two parts are needed:

- What is the conditioning of solving $Ax \cong b$?
- What is the worst-case rounding error of solving $A^T Ax = A^T b$?

Conditioning of linear least squares. The problem: input A, b ; output x , such that $Ax \cong b$. Perturb the input, $b \rightarrow b + \delta b$, and ask for the resulting output perturbation, $x \rightarrow x + \delta x$. From the normal equations,

$$x = (A^T A)^{-1} A^T b, \quad x + \delta x = (A^T A)^{-1} A^T (b + \delta b) \implies \delta x = (A^T A)^{-1} A^T \delta b.$$

We want the minimal c such that $\|\delta x\|_2 / \|x\|_2 \leq c \|\delta b\|_2 / \|b\|_2$. First,

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \frac{\|(A^T A)^{-1} A^T\|_2 \|\delta b\|_2}{\|x\|_2},$$

and since the residual $r = b - Ax$ gives $Ax = b - r$, we have $\|A\|_2 \|x\|_2 \geq \|Ax\|_2 = \|b - r\|_2$, i.e., $\|x\|_2 \geq \|b - r\|_2 / \|A\|_2$. Hence

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \underbrace{\|(A^T A)^{-1} A^T\|_2 \|A\|_2}_{\kappa(A) \text{ for rectangular } A} \cdot \frac{\|\delta b\|_2}{\|Ax\|_2}. \quad (8.3)$$

Conditioning of rectangular matrices. To evaluate the constant, use the SVD: for any $A \in \mathbb{R}^{m \times n}$ there exist orthogonal $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ and a nonnegative diagonal $S \in \mathbb{R}^{m \times n}$ such that

$$A = USV^T.$$

For $m > n$, the trailing $m - n$ columns of U multiply zero rows of S and are redundant, giving the *reduced SVD*

$$A = U_1 S_1 V^T, \quad U_1 \in \mathbb{R}^{m \times n} \text{ with orthonormal columns}, \quad S_1 = \text{diag}(\sigma_1, \dots, \sigma_n),$$

with $\sigma_1 = \sigma_{\max} \geq \dots \geq \sigma_n = \sigma_{\min} \geq 0$:

$$\boxed{A} = \boxed{U_1} \boxed{S_1} \boxed{V^T}$$

orthonormal columns

The pseudoinverse. Define the *pseudoinverse*

$$A^+ = (A^T A)^{-1} A^T \in \mathbb{R}^{n \times m}.$$

(To be continued.)

CS 450: Numerical Analysis – Lecture 9

The Pseudoinverse, Stability of the Normal Equations, and QR

Edgar Solomonik

September 22, 2026

Today

- The pseudoinverse
- Stability of the normal equations

Review: linear least squares

Given $A \in \mathbb{R}^{m \times n}$ with full column rank ($\text{rank}(A) = n$, as in Lecture 8) and $b \in \mathbb{R}^m$, solve

$$\min_x \|Ax - b\|_2.$$

The *residual* is $r = b - Ax$; at the solution x , it is orthogonal to the column span,

$$r \perp \text{colspan}(A).$$

Conditioning analysis. With $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$ —the condition number of any $A \in \mathbb{R}^{m \times n}$ —(8.3) becomes, for LLSQ,

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \underbrace{\kappa(A) \cdot \frac{\|b\|_2}{\|Ax\|_2}}_{\text{bound on relative conditioning for perturbations in } b} \cdot \frac{\|\delta b\|_2}{\|b\|_2}. \quad (9.1)$$

The second factor is $\|b\|_2 / \|Ax\|_2 = 1/\cos \theta$, with θ the angle between b and $\text{colspan}(A)$; the worst case is b (nearly) orthogonal to the column span.

The pseudoinverse

Moore–Penrose conditions. For $A \in \mathbb{R}^{m \times n}$, the pseudoinverse $A^+ \in \mathbb{R}^{n \times m}$ must satisfy

1. $AA^+A = A$;
2. $A^+AA^+ = A^+$;
3. $(AA^+)^T = AA^+$;
4. $(A^+A)^T = A^+A$.

Conditions (3)–(4) are the symmetry conditions (Hermitian transpose for complex A). These four conditions in fact determine A^+ uniquely (a short exercise), so “the” pseudoinverse is well defined; if A^{-1} exists, it satisfies (1)–(4), hence $A^+ = A^{-1}$.

Scalar case. For a scalar $\alpha \in \mathbb{R}$, let

$$\alpha^+ = \begin{cases} 1/\alpha & \alpha \neq 0, \\ 0 & \alpha = 0. \end{cases}$$

(This satisfies (1)–(4).)

Diagonal case. For a diagonal matrix $D \in \mathbb{R}^{n \times n}$, let

$$D^+ = \begin{bmatrix} d_{11}^+ & & \\ & \ddots & \\ & & d_{nn}^+ \end{bmatrix}$$

(again, (1)–(4) are satisfied).

General case. For a general $A \in \mathbb{R}^{m \times n}$, take the full SVD of Lecture 8, $A = USV^T$, and let

$$A^+ = VS^+U^T,$$

where $S^+ \in \mathbb{R}^{n \times m}$ carries the reciprocals of the nonzero singular values ($s_{ii}^+ = 1/\sigma_i$ if $\sigma_i \neq 0$, all other entries zero). Check conditions (3)–(4):

$$A^+A = (VS^+U^T)(USV^T) = V(S^+S)V^T, \quad AA^+ = (USV^T)(VS^+U^T) = U(SS^+)U^T,$$

where

$$S^+S = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (\text{identity block of size } \text{rank}(A); SS^+ \text{ has the same block form in } \mathbb{R}^{m \times m}).$$

Thus $P = A^+A$ is an orthogonal projector, $PP = P$ and $P^T = P$ (and likewise AA^+), so conditions (3)–(4) are satisfied. Moreover,

$$AA^+A = (USV^T)(V S^+ S V^T) = U \underbrace{SS^+S}_S V^T = USV^T = A,$$

and similarly $A^+AA^+ = V(S^+SS^+)U^T = VS^+U^T = A^+$, so conditions (1)–(2) hold as well.

Definition (pseudoinverse via the SVD). For any $A \in \mathbb{R}^{m \times n}$ with SVD $A = USV^T$,

$$A^+ = VS^+U^T \in \mathbb{R}^{n \times m}. \quad (9.2)$$

This formula also defines A^+ when A is rank deficient. (Recall from Lecture 8: $A^+ = (A^T A)^{-1} A^T$ when $\text{rank}(A) = n$.) In particular, $\|A^+\|_2 = 1/\sigma_{\min}(A)$ for full column rank, while $\|A\|_2 = \sigma_{\max}(A)$ —which completes the evaluation of the conditioning constant as $\kappa(A)$ in (9.1).

Solving LLSQ with the pseudoinverse. To solve $Ax \cong b$, compute $x = A^+b$. How to actually compute A^+ ? It suffices to *apply* it—for full-column-rank A , the normal equations with Cholesky do exactly that:

$$A^T A x = A^T b, \quad A^T A = LL^T.$$

The normal-equations algorithm.

1. compute $B = A^T A$: $O(mn^2)$;
2. factorize $B = LL^T$ by Cholesky: $O(n^3)$;
3. compute $y = A^T b$: $O(mn)$;
4. solve $LL^T x = y$: $O(n^2)$.

Steps 3–4 are the b -dependent part: for a new right-hand side, only they need to be repeated.

Stability of the normal-equations algorithm

Is forming B and y safe with respect to round-off? Even if $y = A^T b$ is formed with relative error $O(\varepsilon_{\text{mach}})$, i.e., with a rounding error δy satisfying

$$\frac{\|\delta y\|_2}{\|y\|_2} = O(\varepsilon_{\text{mach}}),$$

the conditioning analysis of linear systems (Chapter 2) shows that δy causes a perturbation δx of the solution of $Bx = y$ where, in the worst case,

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \kappa(B) \frac{\|\delta y\|_2}{\|y\|_2} = O(\kappa(A)^2 \varepsilon_{\text{mach}}), \quad (9.3)$$

since

$$B = A^T A = (VS^T U^T)(USV^T) = V \underbrace{S^T S}_{\text{“}S^2\text{”}} V^T, \quad \text{so} \quad \sigma_{\max}(B) = \sigma_{\max}(A)^2, \quad \sigma_{\min}(B) = \sigma_{\min}(A)^2,$$

and hence $\kappa(B) = \kappa(A)^2$. *Conclusion:* solving the normal equations is not numerically stable—it squares the conditioning of the problem.

Suboptimal. When $b \in \text{colspan}(A)$ —the case $\cos \theta = 1$ in (9.1)—we want

$$\frac{\|\delta x\|_2}{\|x\|_2} = O(\kappa(A) \varepsilon_{\text{mach}}).$$

High-level problem. The product with A^T may lead to large intermediate results.

Instead: orthogonalize. We want Q with $Q^T Q = I$ and $\text{colspan}(Q) = \text{colspan}(A)$. Then $r \perp \text{colspan}(Q)$, and so

$$Q^T r = 0 \implies Q^T (b - Ax) = 0 \implies Q^T Ax = Q^T b, \quad (9.4)$$

an $n \times n$ system $Zx = \tilde{b}$ with $Z := Q^T A$ and $\tilde{b} := Q^T b$. Orthogonal transformations avoid forming $A^T A$ and hence avoid squaring the condition number (and multiplication by orthonormal columns does not amplify absolute errors: $\|Q^T v\|_2 \leq \|v\|_2$).

QR factorization

Given $A \in \mathbb{R}^{m \times n}$ with $m \geq n$, there exist an orthogonal $Q \in \mathbb{R}^{m \times m}$ and an upper-triangular $R \in \mathbb{R}^{n \times n}$ such that

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (\text{full}), \quad A = Q_1 R \quad (\text{reduced}), \quad (9.5)$$

where $Q_1 \in \mathbb{R}^{m \times n}$ collects the first n columns of Q :

$$\begin{array}{ccc} \boxed{A} & = & \boxed{Q} \begin{array}{c} \boxed{R} \\ \boxed{0} \end{array} \\ \text{full} & & \text{full} \end{array} \quad \begin{array}{ccc} \boxed{A} & = & \boxed{Q_1} \boxed{R} \\ \text{reduced} & & \text{reduced} \end{array}$$

Proof. (Assume $\text{rank}(A) = n$ for simplicity.) There exists L such that $A^T A = LL^T$, hence

$$L^{-1} A^T \underbrace{A L^{-T}}_Q = I.$$

This Q has orthonormal columns and $QL^T = A$; set $R = L^T$. (The construction yields the reduced factorization $A = Q_1 R$; completing Q_1 to an orthogonal basis of $\mathbb{R}^m$ gives the full form.)

The proof suggests the algorithm “Cholesky QR”, but it has the same stability issues as the normal equations. (These can be remedied by iteration.)

QR algorithms overview.

1. *Gram–Schmidt*: form Q one column at a time by orthogonalizing each column of A ;
2. *Householder and Givens*: apply orthogonal $Q_1, \dots, Q_k$ that zero out entries, $Q_k \cdots Q_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$, so that $Q = Q_1^T \cdots Q_k^T$.

Approach (2) is the most stable, since each Q_i is orthogonal.

Gram–Schmidt orthogonalization

Write $A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$ and $Q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$; assume $\text{rank}(A) = n$, so every normalization below divides by a nonzero norm.

Classical Gram–Schmidt (CGS). Let $q_1 = a_1 / \|a_1\|_2$ and $r_{11} = \|a_1\|_2$. For $j = 2$ to n :

$$\hat{q} = a_j - \overbrace{q_1 (q_1^T a_j)}^{r_{1j}} - \cdots - \overbrace{q_{j-1} (q_{j-1}^T a_j)}^{r_{j-1,j}}, \quad q_j = \hat{q} / \|\hat{q}\|_2, \quad r_{jj} = \|\hat{q}\|_2.$$

Modified Gram–Schmidt (MGS). Let $q_1 = a_1 / \|a_1\|_2$ and $r_{11} = \|a_1\|_2$. For $j = 2$ to n : set $\hat{q}^{(0)} = a_j$; for $k = 1$ to $j - 1$,

$$\hat{q}^{(k)} = \hat{q}^{(k-1)} - \overbrace{q_k (q_k^T \hat{q}^{(k-1)})}^{r_{kj}}; \quad \text{then} \quad r_{jj} = \|\hat{q}^{(j-1)}\|_2, \quad q_j = \hat{q}^{(j-1)} / r_{jj}.$$

MGS is more stable than CGS because it computes all the same projections, but using a vector whose norm is generally smaller ($\|\hat{q}^{(k)}\|_2 \leq \|a_j\|_2$), hence with less round-off error; in exact arithmetic, the two variants are equivalent.

The cost of both variants is $O(mn^2)$.

CS 450: Numerical Analysis – Lecture 10

QR Algorithms: Gram–Schmidt, Householder, and Givens

Edgar Solomonik

September 24, 2026

Today

- Review: QR factorization
- QR algorithms: Gram–Schmidt; Householder and Givens

Review: QR factorization

For any $A \in \mathbb{R}^{m \times n}$ with $m \geq n$, there exist $Q \in \mathbb{R}^{m \times n}$ with $Q^T Q = I$ and an upper-triangular $R \in \mathbb{R}^{n \times n}$ such that

$$A = QR \quad (\text{reduced form}), \quad (10.1)$$

unique for full-column-rank A if we normalize $r_{ii} > 0$. Appending $\bar{Q} \in \mathbb{R}^{m \times (m-n)}$ whose columns complete an orthonormal basis gives the full form,

$$A = \begin{bmatrix} Q & \bar{Q} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad \begin{bmatrix} Q & \bar{Q} \end{bmatrix} \text{ orthogonal:}$$

$$\begin{array}{ccc} \boxed{A} & = & \boxed{Q} \begin{array}{|c|} \hline \boxed{R} \\ \hline \end{array} \\ \text{reduced} & & \end{array} \quad \begin{array}{ccc} \boxed{A} & = & \boxed{Q} \quad \boxed{\bar{Q}} \begin{array}{|c|} \hline \boxed{R} \\ \hline 0 \end{array} \\ \text{full} & & \end{array}$$

Linear least squares via QR. To solve $Ax \cong b$ given $A = QR$, solve the triangular system

$$Rx = Q^T b :$$

$O(mn)$ to form $Q^T b$, and $O(n^2)$ to solve given $Q^T b$.

Gram–Schmidt orthogonalization

Orthogonalize the columns, $\begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \rightarrow \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$.

Classical and modified Gram–Schmidt. CLASSICAL-GS(A): for $i = 1$ to n , set $u = a_i$; for $j = 1$ to $i - 1$,

$$u \leftarrow u - q_j \overbrace{\langle q_j, a_i \rangle}^{r_{ji}};$$

then

$$q_i = u / \|u\|_2, \quad r_{ii} = \|u\|_2.$$

Modified Gram–Schmidt (MGS) instead uses the current, partially orthogonalized vector in each inner product: $u \leftarrow u - q_j \langle q_j, u \rangle$. The cost of either variant is $O(mn^2)$: $O(n^2)$ inner-loop iterations at $O(m)$ inner-product cost.

CGS vs. MGS in finite arithmetic. At the i th iteration, CGS and MGS both compute (in exact arithmetic)

$$a_i - \underbrace{\begin{bmatrix} q_1 & \cdots & q_{i-1} \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_{i-1} \end{bmatrix}^T}_P a_i,$$

where P is the orthogonal projector onto the span of $\{a_1, \dots, a_{i-1}\}$. In finite arithmetic, we have $\hat{q}_1 \approx q_1, \dots, \hat{q}_{i-1} \approx q_{i-1}$, and $\{\hat{q}_1, \dots, \hat{q}_{i-1}\}$ (1) are not exactly orthogonal and (2) span a slightly different space. CGS applies

$$\hat{P} = \begin{bmatrix} \hat{q}_1 & \cdots & \hat{q}_{i-1} \end{bmatrix} \begin{bmatrix} \hat{q}_1 & \cdots & \hat{q}_{i-1} \end{bmatrix}^T$$

directly, but $\hat{P}$ is not an orthogonal projector. MGS instead applies

$$(I - \hat{q}_{i-1}\hat{q}_{i-1}^T) \cdots (I - \hat{q}_1\hat{q}_1^T).$$

Each factor $I - \hat{q}_j\hat{q}_j^T$ is itself an exact orthogonal projector (onto the complement of $\hat{q}_j$) whenever $\hat{q}_j$ is accurately normalized; the product is not an orthogonal projector in general, but it is a better approximation of one than $I - \hat{P}$. Hence MGS alleviates the error due to (1).

Stability of Gram–Schmidt. Consider orthonormal q_1, q_2 with $q_1 \perp q_2$, and let

$$A = \begin{bmatrix} \underbrace{q_1}_{a_1} & \underbrace{q_1 + \alpha q_2}_{a_2} \end{bmatrix}, \quad 0 < \alpha \ll 1.$$

Gram–Schmidt computes

$$u = a_2 - q_1 \langle q_1, a_2 \rangle = (q_1 + \alpha q_2) - q_1 = \alpha q_2,$$

but the floating-point subtraction $\hat{u} = \text{fl}(a_2 - q_1)$ suffers cancellation when α is small: in the worst case,

$$\frac{\|\hat{u} - u\|_2}{\|u\|_2} \approx \varepsilon_{\text{mach}} \cdot \frac{1}{\alpha},$$

since $\|\hat{u} - u\|_2 \approx \varepsilon_{\text{mach}}$ while $\|u\|_2 = \alpha$. In fact $\kappa(A) \approx 1/\alpha$, and in general MGS incurs an error—loss of orthogonality $\|I - \hat{Q}^T \hat{Q}\|_2$ —of $O(\kappa(A) \varepsilon_{\text{mach}})$; CGS is even worse (possible HW 2 addendum).

Householder and Givens QR

How do we improve stability? Avoid projecting; apply orthogonal transformations instead. Write the full QR as $A = \tilde{Q} \hat{R}$ with $\tilde{Q} = \begin{bmatrix} Q & \bar{Q} \end{bmatrix}$ orthogonal and $\hat{R} = \begin{bmatrix} R \\ 0 \end{bmatrix}$; the goal is to transform

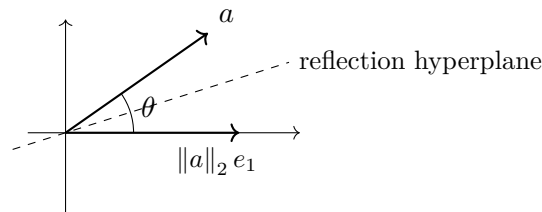
$$A \xrightarrow{\tilde{Q}^T} \hat{R}, \quad \text{i.e.} \quad \hat{R} = \tilde{Q}^T A.$$

Define orthogonal $\tilde{Q}_1, \dots, \tilde{Q}_k$ such that $\tilde{Q} = \tilde{Q}_1 \cdots \tilde{Q}_k$ and each $\tilde{Q}_i$ “zeroes out” part of A . How do we define each $\tilde{Q}_i$?

Consider $n = 1$: we want $\tilde{Q}$ such that

$$\tilde{Q}^T \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \pm \|a\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \hat{R}.$$

For $m = 2$, we can draw, assuming target $\begin{bmatrix} \|a\|_2 & 0 \end{bmatrix}^T$:



Two choices of $\tilde{Q}$ are possible:

1. *rotate* (define $\tilde{Q}$ to rotate any vector by $-\theta$) $\rightarrow$ *Givens rotations*, which work on two components of a vector at a time;
2. *reflect* (define $\tilde{Q}$ to reflect across the hyperplane) $\rightarrow$ *Householder reflections*, which work for general m by reflecting across an $(m - 1)$ -dimensional hyperplane.

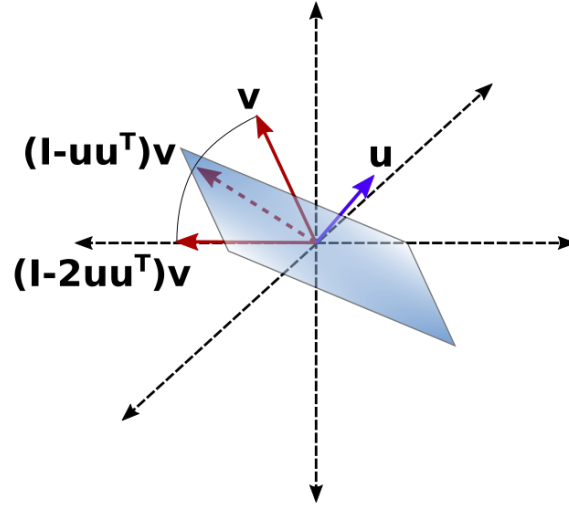
Householder reflections. Let

$$\tilde{Q}_i = H = I - 2uu^T, \quad \|u\|_2 = 1. \quad (10.2)$$

Then

$$Hv = (I - 2uu^T)v = v - 2u\langle u, v \rangle,$$

so Hv negates the component of v in the direction of u , and u specifies the hyperplane: $(I - uu^T)v$ is the projection of v onto the hyperplane, $Hv = (I - 2uu^T)v$ is its mirror image, and vectors in the hyperplane are unchanged.



To transform $a \mapsto \|a\|_2 e_1$, pick $H = I - 2uu^T$ with

$$u = \frac{a - \|a\|_2 e_1}{\|a - \|a\|_2 e_1\|_2}, \quad \text{so} \quad Ha = a - 2u\langle u, a \rangle = \begin{bmatrix} \|a\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|a\|_2 e_1.$$

Forming u is a potential source of cancellation when $a \approx \|a\|_2 e_1$ (equivalently $a_1 \approx \|a\|_2$); hence choose the target sign, $\pm \|a\|_2 e_1$, based on the sign of a_1 .

Householder QR algorithm. HHQR(A), with a the first column of A :

1. $v = a \pm \|a\|_2 e_1$ (sign of a_1), $u = v / \|v\|_2$, $H = I - 2uu^T$;
2. $\tilde{A} = HA = A - 2u(u^T A) = \begin{bmatrix} r_{11} & r_{12} \\ 0 & \hat{A}_{22} \end{bmatrix}$;
3. $[\hat{Q}, R_{22}] = \text{HHQR}(\hat{A}_{22})$ (recurring until no columns remain);
4. $Q = H \begin{bmatrix} 1 & \\ & \hat{Q} \end{bmatrix}$, $R = \begin{bmatrix} r_{11} & r_{12} \\ & R_{22} \end{bmatrix}$.

The cost is $O(mn^2)$, dominated by forming $\tilde{A}$: $O(mn)$ for each H .

Givens rotations. The rotation

$$G = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (10.3)$$

rotates any $v \in \mathbb{R}^2$ by $-\theta$. One rotation zeroes out one entry and affects two rows:

$$\begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & \cos \theta & \sin \theta \\ & & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} -a_1^T \\ \vdots \\ -a_{m-1}^T \\ -a_m^T \end{bmatrix} = \begin{bmatrix} -a_1^T \\ \vdots \\ -\tilde{a}_{m-1}^T \\ -\tilde{a}_m^T \end{bmatrix},$$

with θ chosen so that the leading entry of $\tilde{a}_m^T$ is zero, at $O(n)$ cost per rotation. Zeroing the subdiagonal entries of each column from the bottom up reduces A to $\hat{R}$:

$$\begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ 0 & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ 0 & \times \\ 0 & \times \end{bmatrix} \rightarrow \dots$$

Givens rotations are useful for sparse A , e.g. upper Hessenberg (rectangular, with one nonzero subdiagonal):

$$A = \begin{bmatrix} \diagup \\ \diagdown \\ 0 \end{bmatrix}$$

requires $O(n)$ rotations (one per subdiagonal entry), $O(n^2)$ cost total, to reduce to R . Both Householder and Givens can form Q efficiently by applying each transformation to I .

Summary. QR factorization via orthogonal transformations costs $O(mn^2)$, with error $O(\varepsilon_{\text{mach}})$ independent of $\kappa(A)$: the computed factors satisfy $\hat{Q}\hat{R} = A + \delta A$ with $\|\delta A\|_2 / \|A\|_2 = O(\varepsilon_{\text{mach}})$, and $\|\hat{Q}^T \hat{Q} - I\|_2 = O(\varepsilon_{\text{mach}})$.

CS 450: Numerical Analysis – Lecture 11

Projectors, Rank-Deficient Least Squares, and Low-Rank Approximation

Edgar Solomonik

September 29, 2026

Today

- Review: QR algorithms
- Rank-deficient least squares
- Low-rank approximation

Review: QR algorithms

Project, rotate, or reflect?

Orthogonal projectors. $P \in \mathbb{R}^{n \times n}$ is an *orthogonal projector* if

1. $P^2 = P$;
2. P is symmetric ($P^T = P$).

Lemma. The orthogonal projector onto $\text{span}\{q_1, \dots, q_k\}$, where $\{q_i\}$ are orthonormal, has the form

$$P = \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}^T = \sum_{i=1}^k q_i q_i^T. \quad (11.1)$$

Completing $q_1, \dots, q_k$ to an orthonormal basis $q_1, \dots, q_n$ of $\mathbb{R}^n$, so that $\text{span}\{q_1, \dots, q_k\}^\perp = \text{span}\{q_{k+1}, \dots, q_n\}$, we then have

$$P = I - \begin{bmatrix} q_{k+1} & \cdots & q_n \end{bmatrix} \begin{bmatrix} q_{k+1} & \cdots & q_n \end{bmatrix}^T.$$

To see the action of P , consider $x = \sum_{i=1}^n \alpha_i q_i$:

$$Px = \sum_{i=1}^k \alpha_i q_i, \quad \text{since} \quad Px = \sum_{i=1}^n \alpha_i Pq_i$$

and $Pq_i = q_i$ for $i \leq k$ while $Pq_i = 0$ for all $i > k$.

Lemma. Given the (economy) SVD $A = USV^T$, AA^+ is the projector onto the column span of A , and A^+A is the projector onto the row span of A .

Proof. $\text{colspan}(A) = \text{colspan}(U)$, since $Ax = U(SV^T x)$; and $\text{rowspan}(A) = \text{colspan}(V)$, since $A^T y = V(S^T U^T y)$. Now

$$AA^+ = USV^T V S^+ U^T = U(SS^+)U^T :$$

if A has full rank, $S^+ = S^{-1}$; otherwise, the zero singular values correspond to the null space of A (not in the span). A^+A is similar.

Gram-Schmidt as projection. With P the orthogonal projector onto $\text{span}\{q_1, \dots, q_k\}$,

$$\begin{aligned} \text{classical Gram-Schmidt: } q_{k+1} &= (I - P) a_{k+1} / \|(I - P) a_{k+1}\|_2, \\ \text{modified Gram-Schmidt: } q_{k+1} &= \underbrace{(I - q_k q_k^T) \cdots (I - q_1 q_1^T)}_{\substack{\text{more accurate approximation of } I - P \\ \text{if } q_1, \dots, q_k \text{ perturbed by round-off}}} a_{k+1} \quad (\text{then normalized}). \end{aligned}$$

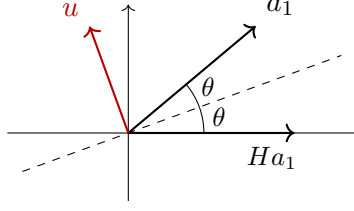
Applying $I - P$ amplifies round-off in a_{k+1} (in relative terms). Instead, we want to apply orthogonal transformations.

Householder reflectors. For any $u \in \mathbb{R}^n$ with $\|u\|_2 = 1$,

$$H = I - 2uu^T \quad \text{satisfies} \quad H^{-1} = H^T = H.$$

Given $x = \sum_{i=1}^n \alpha_i q_i$ in an orthonormal basis with $q_1 = u$, the reflector $H = I - 2q_1 q_1^T$ gives

$$Hx = -\alpha_1 q_1 + \sum_{i=2}^n \alpha_i q_i, \quad \|x\|_2^2 = \sum_{i=1}^n \alpha_i^2 = \|Hx\|_2^2.$$



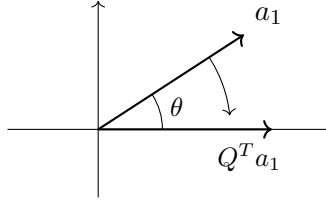
To send a_1 to the axis,

$$u = \frac{a_1 \mp \|a_1\|_2 e_1}{\|a_1 \mp \|a_1\|_2 e_1\|_2}; \quad \text{pick the sign to avoid cancellation.}$$

Givens rotations.

$$Q = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad s^2 + c^2 = 1, \quad s = \sin(\theta), \quad c = \cos(\theta),$$

so that $Q^T a_1 = \|a_1\|_2 e_1$:



Rotations make sense in 2D only (encoded by an angle $\theta \in \mathbb{R}$); reflections work in n D for any n , encoded by the normal vector of the reflection hyperplane (n D).

Householder QR, step by step. Pick u_1 from the first column of A , giving $H_1 = I - 2u_1 u_1^T$; then pick u_2 from the trailing part of the second column of $H_1 A$, giving $H_2 = I - 2u_2 u_2^T$; and so on:

$$H_1 A = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ \vdots & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}, \quad H_2 H_1 A = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ \vdots & 0 & \times & \times \\ 0 & \vdots & \times & \times \end{bmatrix} \rightarrow u_3, \text{ etc.}$$

Computing $H_k B$, where $B = H_{k-1} \cdots H_1 A$ is given, costs $O(mn)$ via

$$H_k B = B - 2u_k (u_k^T B).$$

Givens QR, step by step. Each rotation combines two rows: Q_1^T zeroes the lowest subdiagonal entry of the first column, affecting the last two rows; Q_2^T zeroes the next one up, affecting the two rows above it; and so on:

$$Q_1^T A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}, \quad Q_2^T Q_1^T A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}, \quad \text{etc.,}$$

at $O(mn^2)$ cost in total. Given QR is effective for some sparse matrices, e.g. tridiagonal: the first rotation zeroes the $(2, 1)$ entry and creates one fill-in entry (bold),

$$A = \begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}, \quad Q_1^T A = \begin{bmatrix} \times & \times & \times & & \\ 0 & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}, \quad Q_2^T Q_1^T A = \dots,$$

and after $n - 1$ rotations,

$$Q_{n-1}^T \dots Q_1^T A = R = \begin{bmatrix} \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

(upper triangular, with two superdiagonals).

Rank-deficient linear least squares

If $A \in \mathbb{R}^{m \times n}$ has $\text{rank}(A) < n$ (with $m \geq n$), then for every x there exists $x' \neq x$ such that

$$Ax = Ax'.$$

Instead, solve for the *minimum-norm* solution:

$$\min_{x: Ax \cong b} \|x\|_2, \quad \text{or equivalently} \quad \min_x \|Ax - b\|_2 \quad \text{s.t.} \quad x \in \text{rowspan}(A).$$

The solution is

$$x = A^+ b, \quad A^+ = V S^+ U^T, \quad S^+ = \text{diag}(\sigma_1^{-1}, \dots, \sigma_r^{-1}, 0, \dots, 0).$$

QR with column pivoting. For rank-deficient A , column $k + 1$ will be in the span of the first k columns, for some k . For $A \in \mathbb{R}^{m \times n}$ with $\text{rank}(A) = r$, there exist $Q \in \mathbb{R}^{m \times r}$ with orthonormal columns, an upper-trapezoidal $R \in \mathbb{R}^{r \times n}$, and a permutation matrix $P \in \mathbb{R}^{n \times n}$, so that

$$AP = QR : \tag{11.2}$$

$$\boxed{A} \boxed{P} = \boxed{Q} \boxed{R}$$

The *QRCP algorithm*:

1. pick the highest-norm column and swap it with the leading column;
2. apply one Householder reflector;
3. recurse on the remaining columns;
4. stop when all remaining columns are zero.

The cost is $O(mnr)$ for rank r .

What about approximation? We can stop QRCP when all remaining columns have small norm; the residual $\|AP - QR\|_F$ is then also small. In general, QRCP gives a greedy algorithm for finding the best low-rank approximation in the column space of k columns.

Low-rank approximation

Eckart–Young–Mirsky theorem. Given $A \in \mathbb{R}^{m \times n}$ with SVD $A = USV^T$ and $r < m, n$, the objective

$$\|A - XY\| \quad \text{over} \quad X \in \mathbb{R}^{m \times r}, Y \in \mathbb{R}^{r \times n},$$

is minimized—in both $\|\cdot\|_F$ and $\|\cdot\|_2$ —by the truncated SVD,

$$X = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}, \quad Y = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} -v_1^T & - \\ \vdots & \\ -v_r^T & - \end{bmatrix}. \quad (11.3)$$

$$\boxed{A} - \boxed{X} \boxed{Y}$$