Numerical Linear Algebra for Computational Science and Information Engineering

Least-Squares Problems

Nicolas Venkovic
nicolas.venkovic@tum.de

Chair of Computational Mathematics
School of Computation, Information and Technology
Technical University of Munich



Outline I

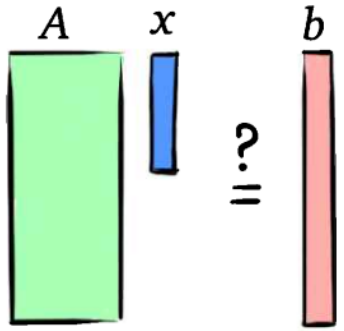
1. Geometry of least-squares problems
   Section 4.4 in Darve & Wootters (2021) 1
2. Method of normal equations 2
3. QR factorization and SVD for least-squares problems 4
4. LSQR 9

Geometry of least-squares problems

Section 4.4 in Darve & Wootters (2021)

Geometry of least-squares

- One of the applications of the QR decomposition is the solving of least-squares problems $\arg\min_x \|Ax - b\|_2$:

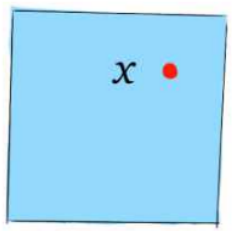


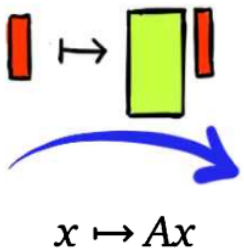


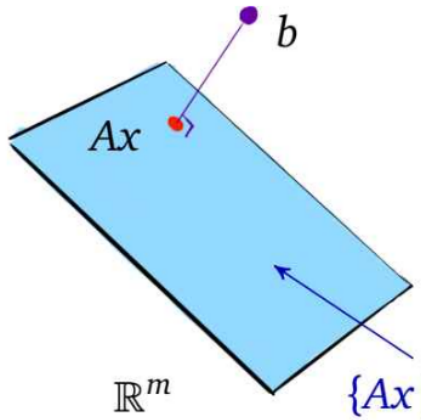
Since A is tall and skinny, there are more equations than unknowns! So there may not be some x so that $Ax = b$. The least-squares problem is to find the x so that Ax is closest to b, in 2-norm.

with a tall-and-skinny matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^n$.

- We seek the element in $\{Ax,\ x \in \mathbb{R}^n\}$ closest to b in the 2-norm. This is achieved by orthogonal projection:







The point Ax which is closest to b in 2-norm is the orthogonal projection of b onto the subspace $\{Ax : x \in \mathbb{R}^n\}$.

Darve, E., & Wootters, M. (2021). Numerical linear algebra with Julia. Society for Industrial and Applied Mathematics.

Method of normal equations

Method of normal equations

- As per property of orthogonal projections, the x that minimizes $\|Ax - b\|_2$ has an error $e := Ax - b$ which is orthogonal to the range of A.
  This can be written as

$$A^T(Ax - b) = 0. \tag{1}$$

Assuming A is full-rank, this equation can be used to solve for x by a method called **normal equations**.

Eq. (1) can also be derived from calculus, namely, seeking the optimal x which minimizes the cost function

$$f(x) = \|Ax - b\|_2^2 = (Ax - b)^T(Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b$$

is obtained for $\nabla f(x) = 0$ where

$$\nabla f(x) = 2A^T Ax - 2A^T b,$$

which equivalently yields Eq. (1).

Method of normal equations, cont'd

- Assuming A is full-rank, A^TA is SPD so that we may compute its Cholesky factorization and solve for x in $A^TAx = A^Tb$.

Normal equations

Finding the solution x to the least-suqares problem $\arg\min \|Ax - b\|_2$ by solving the system $A^TAx = A^Tb$ is called the method of **normal equations**.

- Since the condition number of A^TA is the square of that of A, the method of normal equations can run into issues when A is poorly conditioned.
- For cases where A is poorly conditioned, the QR factorization can be used to yield a more accurate computation of the solution x to the least-squares problem.

QR factorization and SVD for least-squares problems

QR factorization for least-squares problems

- The origin of the method of normal equations stems from saying that the error $Ax - b$ is orthogonal to the range of A.

  But if we know a QR factorization $A = QR$ where $Q \in \mathbb{R}^{m \times n}$, then the range of A is the same as the range of Q.

  The orthogonality condition can then be re-stated as

$$Q^T(Ax - b) = 0. \tag{2}$$

  Since Q is orthogonal, it is necessarily well-conditioned, and the conditioning problem of the method of normal equations can be avoided.

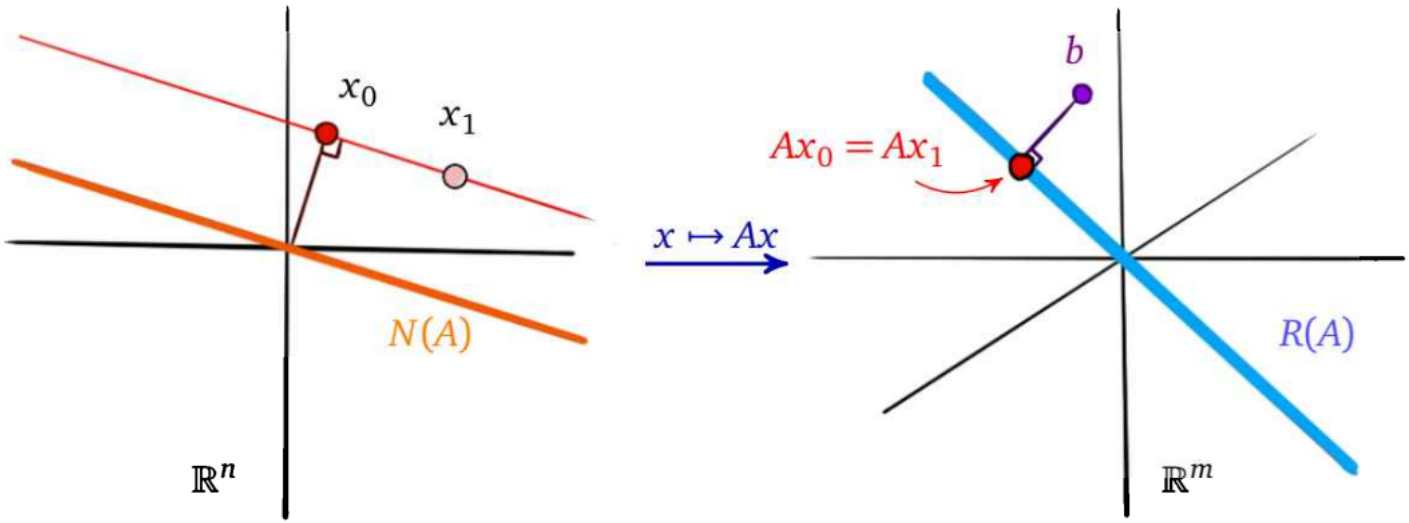
  Since $A = QR$, due to the orthogonality of Q, we have $Q^T A = R$ so that Eq. (2) becomes

$$Rx = Q^T b$$

  where R is non-singular as long as A is non-singular, so that there exists a unique solution x.

Case of rank-deficient A

- If A is rank-deficient, the null space of A is non-trivial. Then, for some x that minimizes $\|Ax - b\|_2$, there are infinitely many $\delta x \in \text{null}(A)$ such that $A(x + \delta x) = Ax$. Hence, the solution to the least-squares problem is not unique.
- In case of non-uniqueness of solution, one can search for the unique x_0 which minimizes both $\|Ax - b\|_2$ and $\|x\|_2$:



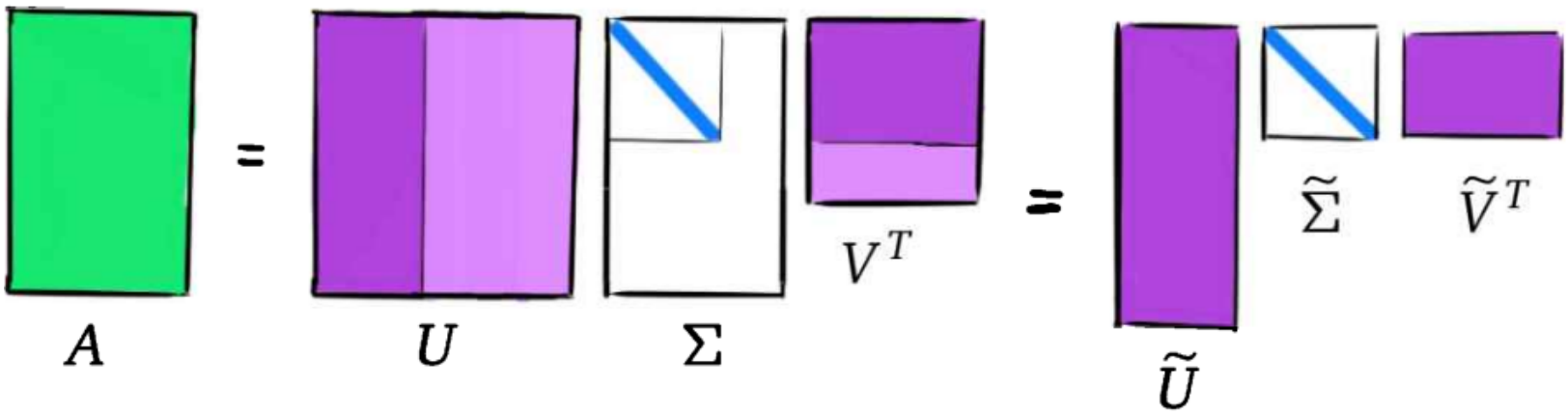
Darve, E., & Wootters, M. (2021). Numerical linear algebra with Julia. Society for Industrial and Applied Mathematics.

We can see the x_0 we are after is orthogonal to the null space of A, while any other solution x_1 is of the form $x_0 + \delta x$.

SVD method for solving least-squares with rank-deficient A

- Let $A \in \mathbb{R}^{m \times n}$ be of rank $r < n < m$ have an SVD given by $A = U\Sigma V^T$ with $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$ and $\Sigma \in \mathbb{R}^{m \times n}$ where Σ has zeros from row $r+1$ to m.

  Then we can ignore the columns of U and V that correspond to zeros in Σ to create the thin SVD $A = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$:



Darve, E., & Wootters, M. (2021). Numerical linear algebra with Julia. Society for Industrial and Applied Mathematics.

- Now, $Ax = 0$ if and only if $\widetilde{V}^T x = 0$, which means that the null space of A is the same as that of $\widetilde{V}^T$, i.e., $\text{null}(A) = \text{null}(\widetilde{V}^T)$.

SVD method for solving least-squares with rank-deficient A

- We know that any solution x to the least-squares problem satisfies

$$\begin{aligned} A^T A x &= A^T b \\ (\widetilde{U}\widetilde{\Sigma}\widetilde{V}^T)^T \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T x &= (\widetilde{U}\widetilde{\Sigma}\widetilde{V}^T)^T b \\ \widetilde{V}\widetilde{\Sigma}^T\widetilde{\Sigma}\widetilde{V}^T x &= \widetilde{V}\widetilde{\Sigma}^T\widetilde{U}^T b \\ \widetilde{\Sigma}\widetilde{V}^T x &= \widetilde{U}^T b \end{aligned}$$

  where $r < n$ so that $\widetilde{\Sigma}\widetilde{V}^T$ is not full-column-rank and this equation admits infinitely many solutions.

- However, we can find one solution as follows.

  First, let's solve the system $\widetilde{\Sigma}\omega = \widetilde{U}^T b$ for $\omega \in \mathbb{R}^r$. This gives

$$\omega_i = \frac{\widetilde{u}_i^T b}{\widetilde{\sigma}_{ii}}$$

  where $\widetilde{U} = [\widetilde{u}_1, \ldots, \widetilde{u}_r]$ and $\widetilde{\Sigma} = \mathrm{diag}(\widetilde{\sigma}_{11}, \ldots, \widetilde{\sigma}_{rr})$.

SVD method for solving least-squares with rank-deficient A

Then, since $\omega = \widetilde{\Sigma}^{-1}\widetilde{U}^T b$, we have

$$\widetilde{\Sigma}\widetilde{V}^T(\widetilde{V}\omega) = \widetilde{\Sigma}\widetilde{V}^T\widetilde{V}\tilde{\Sigma}^{-1}\widetilde{U}^T b = \widetilde{U}^T b$$

so that $x_0 := \widetilde{V}\omega$ is solution of $\widetilde{\Sigma}\widetilde{V}^T x_0 = \widetilde{U}^T b$ and thus, as explained before, it is also solution of the least-squares problem.

Note that $x_0 := \widetilde{V}\omega$ is the solution with smallest norm.

To see this, we need to show $x_0 \perp \mathrm{null}(A)$. First, let

$$\mathrm{null}(\widetilde{V}^T) = \{y \in \mathbb{R}^n,\ \widetilde{V}^T y = 0\}$$

and consider that for each $y \in \mathrm{null}(\widetilde{V}^T)$, we have

$$x_0^T y = (\widetilde{V}\omega)^T y = \omega\widetilde{V}^T y = 0$$

so that $x_0 \perp \mathrm{null}(\widetilde{V}^T)$.

But since $\mathrm{null}(\widetilde{V}^T) = \mathrm{null}(A)$, we have that $x_0 \perp \mathrm{null}(A)$.

LSQR

Iterative solve of normal equations

- We saw that the least-squares solution x_* of $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$ for $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ with $m > n$ is such that $A^T A x_* = A^T b$, i.e., x_* is solution of the normal equation.
- For very large matrices, the cost of computing a QR factorization by Householder QR, or even by CholeskyQR, can be prohibitive. When the matrix is sparse, computing a QR factorization is generally an overkill.
- In future lectures, we will look into iterative methods to solve linear systems of the form $Bx = b$ with a square matrix $B \in \mathbb{R}^{n \times n}$.

  In particular, if A is full-rank, one can use the conjugate gradient algorithm to solve $A^T A x_* = b$.

  However, in practice, for cases where A is ill-conditioned, this approach can suffer from significantly delayed convergence.
- **LSQR** is an algorithm proposed by Paige and Sanders (1982) which, in case of exact arithmetic, reproduces the iterates of the conjugate gradient algorithm applied to the normal equation but, in practice, is **more reliable**.

Paige, C. C., & Saunders, M. A. (1982). LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Transactions on Mathematical Software (TOMS), 8(1), 43-71.

Bidiagonalization

- Bidiagonalization is a procedure proposed by Golub and Kahan (1965) which reduces any general matrix $A \in \mathbb{R}^{m \times n}$ into lower bidiagonal form. Let $x_0 \in \mathbb{R}^n$ be an initial approximation of x_* with residual $r_0 := b - Ax_0$. Starting the bidiagonalization procedure with r_0 goes as follows:

$$\beta_1 u_1 = r_0, \ \alpha_1 v_1 = A^T u_1$$

$$\left.\begin{aligned} \beta_{i+1} u_{i+1} &= A v_i - \alpha_i u_i \\ \alpha_{i+1} v_{i+1} &= A^T u_{i+1} - \beta_{i+1} v_i \end{aligned}\right\} \text{for } i = 1, 2 \ldots$$

where the scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|u_i\|_2 = \|v_i\|_2 = 1$. Let $U_k := [u_1, \ldots, u_k]$, $V_k := [v_1, \ldots, v_k]$,

$$B_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \end{bmatrix} \text{ and } \underline{B}_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

Golub, G., & Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2(2), 205-224.

Paige, C. C. (1974). Bidiagonalization of matrices and solution of linear equations. SIAM Journal on Numerical Analysis, 11(1), 197-209.

Bidiagonalization, cont'd

Then, we have

$$U_{k+1}(\beta_1 e_1) = r_0 \qquad (e_1 := I_{k+1}[:,1])$$

$$AV_k = U_{k+1}\underline{B}_k = U_k B_k + \beta_{k+1} u_{k+1} e_k^T \qquad (e_k := I_k[:,k])$$

$$A^T U_{k+1} = V_k \underline{B}_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T. \qquad (e_{k+1} := I_{k+1}[:,k+1])$$

In exact arithmetic, we have $U_k^T U_k = I_k$ and $V_k^T V_k = I_k$.
Clearly, $U_{k+1}^T A V_k = \underline{B}_k$, $U_k^T A V_k = B_k$, $\alpha_k = u_k^T A v_k$ and $\beta_k = u_k^T A v_{k-1}$.
Moreover, the columns of U_k and V_k are orthonormal bases of Krylov subspaces of AA^T and $A^T A$, respectively,i.e.,

$$\text{range}(U_k) = \mathcal{K}_k(AA^T, u_1) := \text{span}\left\{u_1, AA^T u_1, \ldots, (AA^T)^{k-1} u_1\right\},$$

$$\text{range}(V_k) = \mathcal{K}_k(A^T A, v_1) := \text{span}\left\{v_1, A^T A v_1, \ldots, (A^T A)^{k-1} v_1\right\}.$$

Bidiagonalization is central to the definition of LSQR. It also plays a key role in the iterative solve of singular value decompositions.

Golub, G., & Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2(2), 205-224.

Paige, C. C. (1974). Bidiagonalization of matrices and solution of linear equations. SIAM Journal on Numerical Analysis, 11(1), 197-209.

LSQR

- LSQR defines a sequence of iterates $x_1, x_2, \ldots, x_k$ which approximate the solution x_* of $\min_{x \in \mathbb{R}^n} \|b - Ax\|_2$ by

$$x_k \in x_0 + \mathcal{K}_k(A^T A, v_1).$$

That is, we search for $x_k = x_0 + V_k y_k$ such that

$$\begin{aligned} x_k = \underset{x \in x_0 + \mathsf{range}(V_k)}{\arg\min} \|b - Ax\|_2 \implies y_k &= \underset{y \in \mathbb{R}^k}{\arg\min} \|b - A(x_0 + V_k y)\|_2 \\ &= \underset{y \in \mathbb{R}^k}{\arg\min} \|r_0 - U_{k+1} \underline{B}_k y\|_2 \\ &= \underset{y \in \mathbb{R}^k}{\arg\min} \|U_{k+1}(\beta_1 e_1) - U_{k+1} \underline{B}_k y\|_2 \\ &= \underset{y \in \mathbb{R}^k}{\arg\min} \|\beta_1 e_1 - \underline{B}_k y\|_2. \end{aligned}$$

In exact arithmetic, the LSQR iterates exhibit monotonic decrease of residual norm, i.e., $\|r_{k+1}\|_2 \leq \|r_k\|_2$.

LSQR, cont'd

- A basic implementation of the LSQR algorithm goes as follows:

1. $r_0 := b - Ax_0$
2. $u_1 := r_0$, $\beta_1 := \|u_1\|_2$, $u_1 := u_1/\beta_1$
3. $v_1 := A^T u_1$, $\alpha_1 := \|v_1\|_2$, $v_1 := v_1/\alpha_1$
4. for $i = 1, 2, \ldots$
5. $\quad u_{i+1} := Av_i - \alpha_i u_i$, $\beta_{i+1} := \|u_{i+1}\|_2$, $u_{i+1} := u_{i+1}/\beta_{i+1}$
6. $\quad v_{i+1} := A^T u_{i+1} - \beta_{i+1} v_i$, $\alpha_{i+1} := \|v_{i+1}\|_2$, $v_{i+1} := v_{i+1}/\alpha_{i+1}$
7. $\quad y_i := \underset{y \in \mathbb{R}^i}{\arg\min} \|\beta_1 e_1 - \underline{B}_i y\|_2$

When convergence is achieved, the iterate $x_i := x_0 + V_i y_i$ is formed.

Convergence monitoring is reliant on the evaluation of $\|r_i\|_2$ and $\|A^T r_i\|_2$. Note that we have

$$\|r_i\|_2 = \|b - A(x_0 + V_i y_i)\|_2 = \|\underline{t}_i\|_2 \quad \text{where} \quad \underline{t}_i := \beta_1 e_1 - \underline{B}_i y_i$$

$$\|A^T r_i\|_2 = \|A^T U_{i+1} \underline{t}_i\|_2 = \|(V_i \underline{B}_i^T + \alpha_{i+1} v_{i+1} e_{i+1}^T)\underline{t}_i\|_2 = \|B_{i+1}^T \underline{t}_i\|_2$$

so that convergence can be monitored without forming neither x_i nor r_i.