9. Equality constraints and tradeoffs

- Least squares recap
- Minimum-norm least squares
- Equality-constrained least squares
- Optimal tradeoffs
- Example: hovercraft

Laurent Lessard (www.laurentlessard.com)

Least squares recap

Solving the least squares optimization problem:

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2$$

Is equivalent to solving the normal equations:

$$A^\mathsf{T} A \hat{x} = A^\mathsf{T} b$$

- If $A^\mathsf{T} A$ is invertible (A has linearly independent columns)
  The least squares problem has a unique solution:

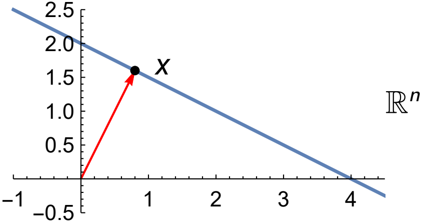
$$\hat{x} = (A^\mathsf{T} A)^{-1} A^\mathsf{T} b$$

- $A^\dagger := (A^\mathsf{T} A)^{-1} A^\mathsf{T}$ is called the **pseudoinverse** of A.

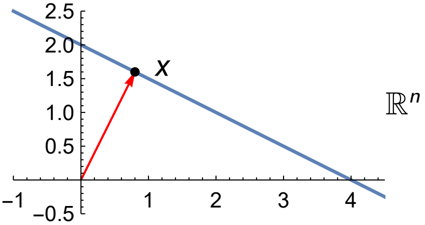
Minimum-norm least squares

Underdetermined case: $A \in \mathbb{R}^{m \times n}$ is a wide matrix ($m \leq n$), so $Ax = b$ typically has infinitely many solutions.

- The set of solutions of $Ax = b$ forms an affine subspace. Recall: if $Ay = b$ and $Az = b$ then $A(\alpha y + (1-\alpha)z) = b$.
- One possible choice: pick the x with smallest norm.



Minimum-norm least squares



- **Insight:** The optimal $\hat{x}$ satisfies $A\hat{x} = b$ and $\hat{x}^\mathsf{T}(\hat{x} - w) = 0$ for all w satisfying $Aw = b$.
- Solutions to $Aw = b$ take the form $\hat{x} + z$, where $\hat{x}$ is any *particular* solution ($A\hat{x} = b$) and z is any *homogeneous* solution ($Az = 0$).
- $\hat{x}^\mathsf{T} z = 0$ for all z satisfying $Az = 0$.

Minimum-norm least squares

$\hat{x}^{\mathsf{T}} z = 0$ for all z satisfying $Az = 0$.

- z is orthogonal to $\{\tilde{a}_1, \dots, \tilde{a}_m\}$ (rows of A)
- $\hat{x}$ is orthogonal to z.
- Therefore: $\hat{x}$ is a linear combination of $\{\tilde{a}_1, \dots, \tilde{a}_m\}$. In other words, $\hat{x} = A^{\mathsf{T}} q$ for some q.

Recap: we must find q such that

$$A\hat{x} = b \text{ and } A^{\mathsf{T}} q = \hat{x}$$

Alternatively, solve $AA^{\mathsf{T}} q = b$ then let $\hat{x} = A^{\mathsf{T}} q$.

Optimality conditions

Theorem: The following statements are equivalent:

1. $\hat{x}$ is a solution of the minimum-norm least squares

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x\|^2 \\ \text{such that:} \quad & Ax = b \end{aligned}$$

2. $\hat{x} = A^\mathsf{T} q$, where q is any solution to $AA^\mathsf{T} q = b$.

- If AA^T is invertible (A has linearly independent rows), we can solve for q and substitute into the expression for $\hat{x}$:

$$\hat{x} = A^\mathsf{T}(AA^\mathsf{T})^{-1} b$$

- $A^\dagger := A^\mathsf{T}(AA^\mathsf{T})^{-1}$ is also called the **pseudoinverse** of A.

Optimality conditions

Theorem: The following statements are equivalent:

1. $\hat{x}$ is a solution of the minimum-norm least squares

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x\|^2 \\ \text{such that:} \quad & Ax = b \end{aligned}$$

2. $\hat{x} = A^\mathsf{T} q$, where q is any solution to $AA^\mathsf{T} q = b$.

- There may not be a any solution to $AA^\mathsf{T} q = b$. This corresponds precisely to the case where the optimization problem is infeasible (no solution to $Ax = b$).

Minimum-norm problems

Minimum-norm LS is **easy** to solve!

- Same as solving $AA^\mathsf{T}q = b$ (similar to normal equations).
- Can be solved in a variety of standard ways: LU (Cholesky) factorization, for example.
- More specialized methods are available if A is very large, sparse, or has a particular structure that can be exploited.
- Comparable to LPs in terms of solution difficulty.

Minimum-norm in Julia

1. Using JuMP, as before:

```
using JuMP, HiGHS
model = Model(HiGHS.Optimizer)
@variable(model, x[1:n])
@constraint(model, A*x == b)
@objective(model, Min, sum(x).^2)
optimize!(model)
```

2. Solving the optimality condition directly:
   `x = A'*inv(A*A')*b`
   Note: Requires A to have full row rank (AA^T invertible)

3. Using the backslash operator (*different* from Matlab!):
   `x = A\b`
   Note: Fastest and most reliable option!

Warning!

Behavior of `A\b` in Matlab and Julia is not the same!

Case	Matlab	Julia
A square	$A^{-1}b$	$A^{-1}b$
A tall	Least squares solution	Least squares solution
A wide	Sparse solution	Min-norm solution

To find min-norm solution in Matlab, use `lsqminnorm(A,b)`.

Equality-constrained least squares

A more general optimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|Ax - b\|^2 \\ \text{subject to:} \quad & Cx = d \end{aligned}$$

(Equality-constrained least squares)

- If $C = 0$, $d = 0$, we recover ordinary least squares
- If $A = I$, $b = 0$, we recover minimum-norm least squares

Equality-constrained least squares

Solving the equality-constrained least squares problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|Ax - b\|^2 \\ \text{subject to:} \quad & Cx = d \end{aligned}$$

Is equivalent to solving the linear equations:

$$A^\mathsf{T} A\hat{x} + C^\mathsf{T} q = A^\mathsf{T} b \quad \text{and} \quad C\hat{x} = d$$

Or, equivalently:

$$\begin{bmatrix} A^\mathsf{T} A & C^\mathsf{T} \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ q \end{bmatrix} = \begin{bmatrix} A^\mathsf{T} b \\ d \end{bmatrix}$$

Mathematical proof

We will prove that the optimality conditions are

$$A^{\mathsf{T}}A\hat{x} + C^{\mathsf{T}}q = A^{\mathsf{T}}b \quad \text{and} \quad C\hat{x} = d$$

Proof is in two parts:

1. **Sufficiency:** If $\hat{x}$ satisfies the optimality conditions for some choice of q, then $\hat{x}$ is an optimal solution to the equality constrained least-squares problem.

2. **Necessity:** If $\hat{x}$ is an optimal solution to the equality constrained least-squares problem, then there exists some q that make the optimality conditions hold.

Mathematical proof

Sufficiency: Suppose $\hat{x}$ and q satisfy $A^\mathsf{T} A\hat{x} + C^\mathsf{T} q = A^\mathsf{T} b$ and $C\hat{x} = d$. Let x be any other point satisfying $Cx = d$. Then,

$$\begin{aligned}
\|Ax - b\|^2 &= \|A(x - \hat{x}) + (A\hat{x} - b)\|^2 \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 + 2(x - \hat{x})^\mathsf{T} A^\mathsf{T}(A\hat{x} - b) \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 - 2(x - \hat{x})^\mathsf{T} C^\mathsf{T} q \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 - 2(Cx - C\hat{x})^\mathsf{T} q \\
&= \|A(x - \hat{x})\|^2 + \|A\hat{x} - b\|^2 \\
&\geq \|A\hat{x} - b\|^2
\end{aligned}$$

Therefore, $\hat{x}$ achieves a lower objective cost than any other feasible x. In other words, $\hat{x}$ solves the optimization problem.

Necessity: Suppose $\hat{x}$ is optimal. So, $C\hat{x} = d$ and for all other x satisfying $Cx = d$, we have $\|Ax - b\|^2 \geq \|A\hat{x} - b\|^2$. Write $x = \hat{x} + z$ where $Cz = 0$ (particular + homogeneous):

$$\|A\hat{x} - b + Az\|^2 \geq \|A\hat{x} - b\|^2 \quad \text{for all } z \text{ with } Cz = 0$$

$$\implies \|Az\|^2 + 2\underbrace{\left(A^\mathsf{T} A\hat{x} - A^\mathsf{T} b\right)^\mathsf{T}}_{w} z \geq 0 \quad \text{for all } z \text{ with } Cz = 0$$

As $z \to 0$, the $w^\mathsf{T} z$ term dominates. We can always replace z by $-z$, so it must be that $w^\mathsf{T} z = 0$ for all z satisfying $Cz = 0$.

Since z is orthogonal to all rows of C and w is orthogonal to z, we conclude that z is a linear combination of the rows of C. i.e., $z = C^\mathsf{T} q$ for some q. Therefore:

$$A^\mathsf{T} A\hat{x} + C^\mathsf{T} q = A^\mathsf{T} b \quad \text{and} \quad C\hat{x} = d.$$

Recap so far

Several different variants of least squares problems are **easy** to solve in the sense that they are equivalent to solving systems of linear equations.

Least squares	Minimum-norm	Equality constrained
$\min_x \ \|Ax - b\|^2$	$\min_x \ \|x\|^2$ s.t. $Ax = b$	$\min_x \ \|Ax - b\|^2$ s.t. $Cx = d$

Optimal tradeoffs

We often want to optimize several different objectives simultaneously, but these objectives are **conflicting**.

- risk vs expected return (finance)
- power vs fuel economy (automobiles)
- quality vs memory (audio compression)
- space vs time (computer programs)
- mittens vs gloves (winter)

Optimal tradeoffs

- Suppose $J_1 = \|Ax - b\|^2$ and $J_2 = \|Cx - d\|^2$.
- We would like to make **both** J_1 and J_2 small.
- A sensible approach: solve the optimization problem:

$$\underset{x}{\text{minimize}} \quad J_1 + \lambda J_2$$

  where $\lambda > 0$ is a (fixed) **tradeoff parameter**.
- Then tune λ to explore possible results.
  - When $\lambda \to 0$, we place more weight on J_1
  - When $\lambda \to \infty$, we place more weight on J_2

Optimal tradeoffs

This problem is also equivalent to solving linear equations!

$$
\begin{aligned}
J_1 + \lambda J_2 &= \|Ax - b\|^2 + \lambda \|Cx - d\|^2 \\
&= \left\| \begin{bmatrix} Ax - b \\ \sqrt{\lambda}(Cx - d) \end{bmatrix} \right\|^2 \\
&= \left\| \begin{bmatrix} A \\ \sqrt{\lambda} C \end{bmatrix} x - \begin{bmatrix} b \\ \sqrt{\lambda} d \end{bmatrix} \right\|^2
\end{aligned}
$$

- An ordinary least squares problem!
- Equivalent to solving normal equations:

$$
(A^\mathsf{T} A + \lambda C^\mathsf{T} C)\,\hat{x} = (A^\mathsf{T} b + \lambda C^\mathsf{T} d)
$$

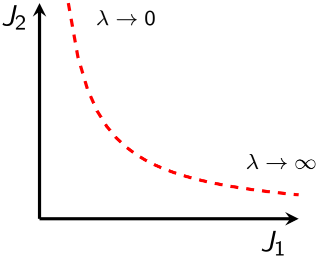
Tradeoff analysis

1. Choose values for λ (usually log-spaced). A useful command: `lambda = 10.^LinRange(p,q,n)` produces n points logarithmically spaced between 10^p and 10^q.

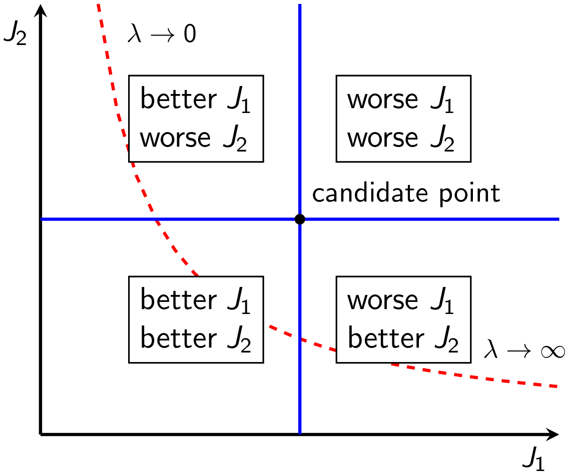
2. For each λ value, find $\hat{x}_\lambda$ that minimizes $J_1 + \lambda J_2$.

3. For each $\hat{x}_\lambda$, also compute the corresponding J_1^λ and J_2^λ.

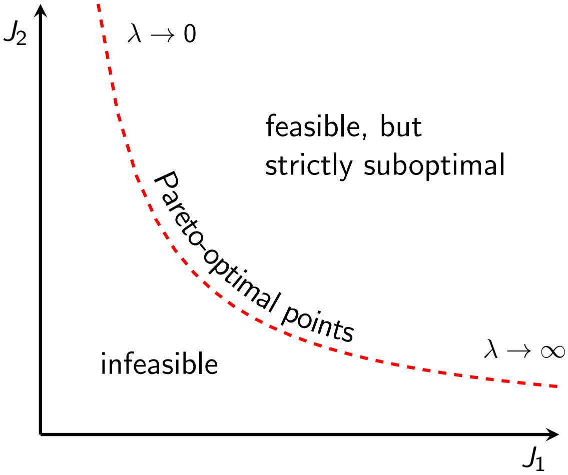
4. Plot $(J_1^\lambda, J_2^\lambda)$ for each λ and connect the dots.



Pareto curve

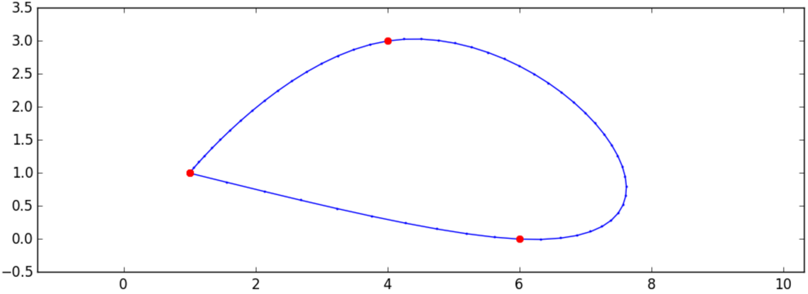


Pareto curve



Example: hovercraft

We are in command of a hovercraft. We are given a set of k waypoint locations and times. The objective is to hit the waypoints at the prescribed times while minimizing fuel use.



Goal is to choose appropriate thruster inputs at each instant.

Example: hovercraft

We are in command of a hovercraft. We are given a set of k waypoint locations and times. The objective is to hit the waypoints at the prescribed times while minimizing fuel use.

- Discretize time: $t = 0, 1, 2, \dots, T$.
- Important variables: position x_t, velocity v_t, thrust u_t.
- Simplified model of the dynamics:

$$\begin{aligned} x_{t+1} &= x_t + v_t \\ v_{t+1} &= v_t + u_t \end{aligned} \qquad \text{for } t = 0, 1, \dots, T-1$$

- We must choose $u_0, u_1, \dots, u_T$.
- Initial position and velocity: $x_0 = 0$ and $v_0 = 0$.
- Waypoint constraints: $x_{t_i} = w_i$ for $i = 1, \dots, k$.
- Minimize fuel use: $\|u_0\|^2 + \|u_1\|^2 + \cdots + \|u_T\|^2$

Example: hovercraft

First model: hit the waypoints exactly

$$
\begin{aligned}
\underset{x_t, v_t, u_t}{\text{minimize}} \quad & \sum_{t=0}^{T} \|u_t\|^2 \\
\text{subject to:} \quad & x_{t+1} = x_t + v_t \quad \text{for } t = 0, 1, \ldots, T-1 \\
& v_{t+1} = v_t + u_t \quad \text{for } t = 0, 1, \ldots, T-1 \\
& x_0 = v_0 = 0 \\
& x_{t_i} = w_i \qquad\qquad \text{for } i = 1, \ldots, k
\end{aligned}
$$

Julia model: Hovercraft.ipynb

Example: hovercraft

Second model: allow waypoint misses

$$\begin{aligned} \underset{x_t, v_t, u_t}{\text{minimize}} \quad & \sum_{t=0}^{T} \|u_t\|^2 + \lambda \sum_{i=1}^{k} \|x_{t_i} - w_i\|^2 \\ \text{subject to:} \quad & x_{t+1} = x_t + v_t \quad \text{for } t = 0, 1, \dots, T-1 \\ & v_{t+1} = v_t + u_t \quad \text{for } t = 0, 1, \dots, T-1 \\ & x_0 = v_0 = 0 \end{aligned}$$

- λ controls the tradeoff between making u small and hitting all the waypoints.